

40  
2ef.



**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO**

**ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES**

**CAMPUS ARAGON**

**Diseño de un Sistema de  
Procesamiento Digital de Señales  
Electroencefalográficas**

**T E S I S**

**QUE PARA OBTENER EL TITULO DE :  
INGENIERO EN COMPUTACION**

**P R E S E N T A :**

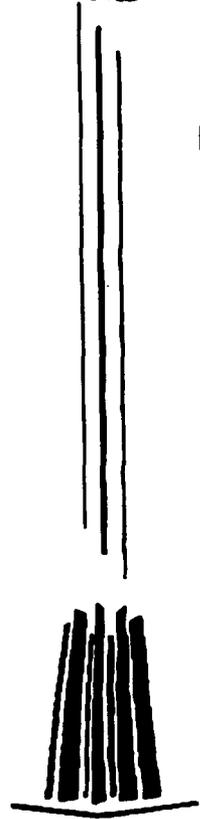
**GONZALO ANTONIO MÉNDEZ JIMÉNEZ**

*Director de Tesis:*  
**M.I. Lauro Santiago Cruz**

**MEXICO**

**1997**

**TESIS CON  
FALLA DE ORIGEN**





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.



UNIVERSIDAD NACIONAL  
AVENIDA 11  
MÉXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES  
ARAGÓN  
DIRECCION

GONZALO ANTONIO MÉNDEZ JIMÉNEZ  
P R E S E N T E .

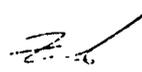
En contestación a su solicitud de fecha 17 de abril del año en curso, relativa a la autorización que se le debe conceder para que el señor profesor, M en I LAURO SANTIAGO CRUZ pueda dirigirle el trabajo de Tesis denominado, "DISEÑO DE UN SISTEMA DE PROCESAMIENTO DIGITAL DE SEÑALES ELECTROENCEFALOGRÁFICAS", con fundamento en el punto 6 y siguientes, del Reglamento para Exámenes Profesionales en esta Escuela, y toda vez que la documentación presentada por usted reúne los requisitos que establece el precitado Reglamento; me permito comunicarle que ha sido aprobada su solicitud.

Aprovecho la ocasión para reiterarle mi distinguida consideración.

ATENTAMENTE  
"POR MI RAZA HABLARA EL ESPÍRITU"  
San Juan de Aragón, México., 23 de abril de 1997  
EL DIRECTOR

  
M en I CLAUDIO C. MERRIFIELD CASTRO

  
c c p Jefe de la Unidad Académica.  
c c p Jefatura de Carrera de Ingeniería en Computación.  
c c p Asesor de Tesis.

  
CCMC/AIR/IIa.

**EN COLABORACION CON:**

**Verónica Juárez Escoriza**

**Rodolfo Landa González**

**Sergio Arturo García Castillo**

**Alumnos de la Facultad de Ingeniería  
de la UNAM**

**Agradecimientos:**

*A la Universidad, por la puerta abierta a la cultura que nos permitió entrar al camino del conocimiento,*

*a nuestra Escuela, por proveer todas las herramientas que hicieron posible que alcanzáramos la primera meta del camino,*

*a nuestros profesores, quienes influyeron en nuestra vida con algo más que conocimiento.*

*Al Dr. Angel Zapata Ferrer,  
Dr. Miguel A. Guevara Pérez,  
M.C. Yolanda Del Rio Portilla,  
Matthijs Hajer,  
PhD. Tom O'Haver,  
por su tiempo, experiencia y conocimientos que enriquecieron este trabajo.*

*En Dios solo descansa el alma mía,  
de El viene mi salud, mi esperanza,  
mi salvación, mi roca sólo es El,  
mi fortaleza, no he de vacilar...*

*A mis padres:*

*Ya que gracias a su apoyo, consejo y guía me ha sido  
posible alcanzar esta meta.  
¡Gracias mami chula por todos tus desvelos y cuidados!  
¡Gracias papá por tu buen ejemplo!  
Esto es lo que han cosechado.  
Con amor su hija*

*Vero*

***A MIS QUERIDOS PADRES Y HERMANOS:***

*Por esos momentos de apoyo, paz,  
armonía y unión familiar que siempre  
me han brindado.*

***A MIS ABUELOS:***

*Por esos momentos de sabiduría y dulzura  
con los que me han encaminado.*

***A MIS TIOS, PRIMOS Y SOBRINOS:***

*Por esos momentos de devoción,  
comuni3n y festejo que me han  
compartido.*

***A MIS PROFESORES:***

*Por esos momentos de motivaci3n e  
impulso con los que me han conducido  
por la senda del conocimiento.*

***A MIS COMPAÑEROS Y AMIGOS:***

*Por esos momentos de solemnidad,  
jubilo y desenfreno que me han  
acompañado a vivir.*

***A DIOS:***

*Por todos esos momentos.*

## *A Alma Delia:*

*Este trabajo está dedicado a ti, Alma. Por todo el cariño que me has brindado, porque fuiste una persona importantísima en el logro de esta meta. Por todas tus palabras de ánimo, exhortación y apoyo que sembraste en mi corazón, que ahora dan este fruto y que a pesar de las circunstancias siempre me hicieron sentir que tú estuviste cerca. Porque nunca quitaste el dedo del renglón para que yo diera este paso tan importante. Por todo aquello que me has dado que no terminaría de enumerar aquí. Porque simplemente sin tu apoyo y tu guía no hubiera comenzado jamás. Este trabajo es para ti.*

*Gracias a Dios infinitamente. Gracias Padre porque me has llenado de bendición al permitirme finalizar este proyecto. Gracias por disponer el camino y las circunstancias, gracias por levantarme en los momentos más difíciles. Todo lo que he logrado lo he hecho gracias a ti. Te ofrezco pues todo el esfuerzo puesto en este trabajo como tributo a tu bondad, a que hayas puesto tu mirada sobre mí.*

*Gracias a mis padres, por toda una vida de esfuerzo y sacrificio para ver culminada una larga trayectoria de estudio. Gracias por el apoyo y entrega incondicional que siempre me demostraron. Muy especial agradecimiento a Mary Tere, Myriam y Hugo Enrique, su ayuda y comprensión no se pueden medir; gracias por tratar siempre de mejorar nuestra relación como hermanos. Gracias por los sobrinitos, quienes han traído luz a esta familia.*

*Gracias a toda, absolutamente toda la familia. Sin omitir a nadie, todos han sido muy importantes para mí. Un reconocimiento especial a mi tía América, quien siempre ha estado al pendiente de mi carrera; muchísimas gracias de todo corazón por toda la ayuda brindada, ésta ha sido invaluable. Gracias también a Alex, Yuri, Fer, Araceli, mi tía Delia y toda la familia de Cuernavaca por sus incontables muestras de aprecio, por su auténtico interés en verme convertido en una mejor persona.*

*Gracias a Jorge y a Julio por toda su ayuda, tangible e intangible sin la cual hubiera resultado prácticamente imposible la culminación exitosa de este trabajo.*

*A todos mis compañeros de la Facultad, gracias por hacer de esta etapa de mi vida una experiencia inolvidable. En especial a Enrique, de quien aprendí tantas cosas, académicas y de la vida en general; gracias Chino por tu amistad.*

*Quiero agradecer de una manera muy especial a Mónica Lavín. Gracias Moni porque en los momentos más difíciles fuiste tú quien me tendió su mano; cuando no hubo nada en que pudiera sostenerme fuiste tú mi único apoyo. Gracias porque además de todo he encontrado en ti una amiga de verdad. Gracias por tu oración, tu preocupación y tu interés. Tú ocupas un lugar muy especial en mi corazón. Que Dios te bendiga siempre.*

*Rodolfo Landa*

*A Dios, por la vida que disfruto en su presencia, y por cada promesa que ha cumplido y que cumplirá.*

*A mis padres y hermanos, que con su amor, cuidado y paciencia hicieron posible que llegara este momento.*

*A Joaquín, por su consejo y por mucho más.*

*A quienes me enseñaron con su vida lo que significa mostrarse amigo.*

**S.A.G.**

*Cuando algo, por especial, no lo  
puedes definir, puedes querer  
compararlo. Si no puedes compararlo  
por no haber nada semejante, y no  
quieres desistir, puedes quizá  
imaginarlo. No obstante, de esto nada  
tú escogas, y entonces, y sólo  
entonces, podrás realmente  
apreciarlo.*

*S.A.G.*

*Cuando algo, por especial, no lo puedes definir, puedes querer compararlo. Si no puedes compararlo por no haber nada semejante, y no quieres desistir, puedes quizá imaginarlo. No obstante, de esto nada tú escogas, y entonces, y sólo entonces, podrás realmente apreciarlo.*

*S.A.G.*

*If it works, don't change it!*  
*If it works in Windows, don't trust it!*

*M.J.*

# Índice

<b>Introducción</b>	1
<b>1. Generalidades del Procesamiento Digital de Señales Electroencefalográficas</b>	5
1.1. Principios de la Electroencefalografía (EEG)	6
1.1.1. Fisiología Cerebral	6
1.1.2. Antecedentes de la Electroencefalografía	11
1.1.3. Técnicas de Registro del EEG	14
1.1.4. Características de las Señales de un EEG	18
1.1.5. Aplicaciones del EEG	26
1.2. Conceptos Básicos de la Electroencefalografía	31
1.2.1. Electroodos	31
1.2.2. Muestreo de Datos	37
1.2.3. Amplificación de Señales	43
1.2.4. Filtros Analógicos	46
1.2.5. Convertidores A/D - D/A	51
<b>2. El Procesamiento Digital de Señales y sus Aplicaciones</b>	63
2.1. Análisis de Sistemas y Señales	64
2.1.1. Señales Discretas en el Tiempo	65
2.1.2. Sistemas Discretos Lineales e Invariantes en el Tiempo	69
2.2. Técnicas de Procesamiento	75
2.2.1. Transformada de Fourier Discreta (DFT), FFT, FHT	75

2.2.2. La Función de Correlación	84
2.2.3. Estimación de la Densidad Espectral	84
<b>3. Diseño del Sistema</b>	91
3.1. Diseño del Hardware	92
3.1.1. Registro de la Señal	92
3.1.2. Acondicionamiento de la Señal	93
3.1.3. Adquisición de Datos	116
3.2. Diseño del Software	121
3.2.1. Diseño Conceptual	122
3.2.2. Diseño Técnico	125
3.2.3. Diseño de Programas	128
<b>4. Desarrollo del Sistema</b>	161
4.1. Plataforma de Desarrollo	161
4.2. Interfaz Gráfica de Usuario	163
4.3 La Aplicación	165
4.3.1 Registro de la Señal	166
4.3.2 Tratamiento de la Señal	175
<b>Conclusiones</b>	181
<b>Bibliografía</b>	185
<b>Apéndices</b>	
Apéndice 1 Diagrama General del Hardware	A-1
Apéndice 2 Hojas de Datos del Hardware	A-3
Apéndice 3 Términos EEG	A-29
Apéndice 4 Listado de Programas	A-31

## Introducción

El cerebro humano, como muchas otras partes del cuerpo y de el mundo, produce cierto tipo de señales eléctricas de características especiales, cuyo análisis detallado es de gran interés desde el punto de vista médico, esto debido a que se ha encontrado que existe una relación específica entre la variación en el comportamiento de estas señales y ciertas disfunciones fisiológicas y fenómenos de conducta en el humano. Resulta claro, que aun cuando la naturaleza analógica de estas señales es fácil de identificar, para manejarlas digitalmente con una computadora, tendrán que estar apropiadamente convertidas, de manera de posibilitar su almacenamiento y procesamiento en el equipo de cómputo.

El procesamiento digital de señales (PDS) es tan antiguo como los procedimientos numéricos inventados por Newton<sup>1</sup> y Gauss<sup>2</sup> en los siglos diecisiete y diecinueve respectivamente. Sin embargo las nuevas tecnologías en los últimos veinticinco años han reducido notablemente el costo del equipo (*hardware*) digital, y su velocidad se ha incrementado a tal grado que el PDS ha reemplazado gran parte del procesamiento analógico de señales.

---

<sup>1</sup> Newton Isaac, 1642-1727. Matemático, filósofo y astrónomo inglés. Sus descubrimientos en el terreno del cálculo diferencial y su ley de la gravitación universal ofrecen la primera gran sistematización de la ciencia matemática. En el campo de los métodos numéricos son reconocidos sus trabajos en los métodos de aproximaciones sucesivas para la solución numérica de ecuaciones.

<sup>2</sup> Gauss Karl Friedrich, 1777-1855. Matemático físico y astrónomo alemán. En 1801 sentó la base de la teoría de los números. Descubrió un método general para la resolución de las ecuaciones binómicas. Estudió la convergencia y divergencia de las funciones. Enunció la ley de la distribución de error (curva de Gauss).

En este trabajo se presenta el diseño de un sistema de procesamiento digital de señales, que en sí, es un campo de estudio de gran interés y con múltiples aplicaciones en la ingeniería, en donde entran en uso los algoritmos numéricos resultantes del cambio en la manera de procesar señales, que se ha posibilitado con las nuevas tecnologías.

El procesamiento digital de señales es un campo de estudio y aplicación concerniente con el procesamiento de información (las señales) representada en forma digital. Ciertas técnicas en este campo como ya se comentó tienen sus orígenes en algoritmos numéricos desarrollados hace siglos. Sin embargo, el advenimiento de dispositivos digitales de cómputo de alta velocidad ha causado una revolución en la utilización de la teoría en gran variedad de aplicaciones, como procesamiento de datos biomédicos (como es el caso, con señales electroencefalográficas), audio digital, procesamiento de radar y sonar, procesamiento de voz, comunicación de datos, almacenamiento de datos confiables de información computarizada, procesamiento de señales sísmicas y un sinnúmero de otras aplicaciones. El procesamiento digital de señales tiene tan amplia variedad de aplicaciones que resulta uno de sus aspectos más interesantes, pues esta variedad ha servido para crear una vitalidad en este campo que no se tiene en otros campos de estudio científico.

Actualmente los equipos utilizados en los institutos de salud para realizar registros y monitoreo encefalográfico, son demasiado caros y de tecnología extranjera, con las consecuencias desventajas que esto implica (no hay flexibilidad para cambiar las configuraciones compradas, el soporte se realiza desde el extranjero, etc.). Por lo anterior, se puede considerar a la electroencefalografía como una prueba diagnóstica de difícil acceso para el grueso de la población mexicana, siendo su disponibilidad restringida a las personas con la capacidad económica para pagar un hospital particular equipado, o bien, de viajar al extranjero. Aun cuando en las instituciones públicas de salud del país se cuenta con equipos de monitoreo electroencefalográfico, un paciente, normalmente tiene que esperar largo tiempo antes de realizarse un estudio, y ya ni hablar acerca de los casos en que los tiempos de muestreo deben ser lo suficientemente grandes como para detectar una anomalía de las que sólo se presentan esporádicamente como la epilepsia.

La necesidad estriba en contar con una sola herramienta que permita no sólo poner en práctica aquellas técnicas que el PDS ha puesto y seguirá poniendo al alcance de prácticas como la electroencefalografía, sino en también tener un sistema integral con el que se adquieran, conviertan y procesen digitalmente señales electroencefalográficas, y de éstas se obtenga la información que se necesite, así como sus representaciones gráficas.

De lo anterior se desprende la propuesta de este trabajo, la cual consiste en plantear el diseño de un sistema que realice todas las etapas que involucran el estudio de una señal EEG. Es decir, un sistema con el cual se lleve a cabo tanto el registro de la misma directamente de la cabeza de un paciente, como la adecuación y digitalización de la señal, para su procesamiento y despliegue final mediante el uso de una computadora personal.

El objetivo es realizar la adquisición de las ondas cerebrales en tiempo real, como se ilustra en la figura I.1, utilizando una configuración y número de canales estándar, y también acondicionar la señal para trabajar con ella. Esto implica amplificación y filtrado, someter la

señal a un proceso de conversión analógica-digital (A/D) a alguna resolución específica (número de bits) y además procesarla digitalmente para obtener de manera flexible resultados interpretables. Todo esto con el fin de satisfacer la necesidad clínica de analizar en forma práctica las señales EEG, con ayuda de herramientas electrónicas y digitales al alcance presupuestal del mayor número posible de centros de diagnóstico

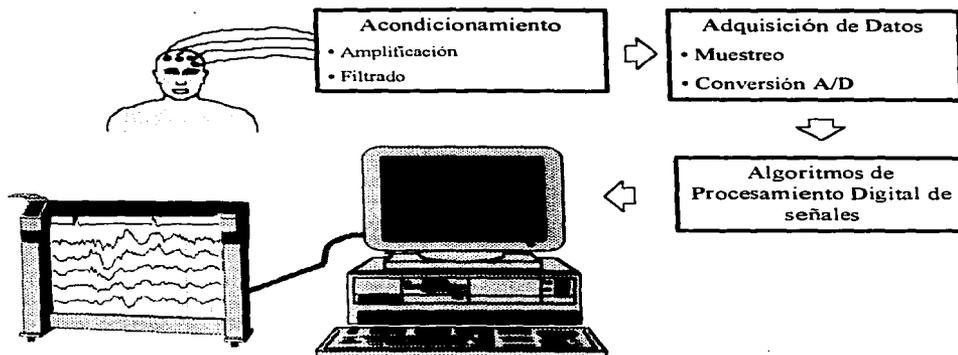


Figura 1.1. Forma general del sistema propuesto.

El procedimiento descrito se llevará al cabo utilizando una computadora, la cual contendrá un *software* que permitirá almacenar las señales y mostrar en pantalla las representaciones gráficas de las mismas, con un análisis visual de sus características de amplitud, frecuencia, forma, etc. Así también, el sistema permitirá hacer el análisis de la señal, para auxiliar al experto a detectar los puntos críticos de anormalidad, mediante comparaciones, correlaciones y demás herramientas numéricas.

La plataforma sobre la que operará el sistema es de computadoras personales con procesador de la familia 80x86, utilizando un ambiente operativo de interfaz gráfica (Windows o Windows 95) con el fin de hacer más amigable el uso de la aplicación.

En este sistema se busca utilizar tecnología a un presupuesto accesible y con una perspectiva de crecimiento y adecuación a diversas necesidades de este campo. Esto se fortalece con el hecho de usar tecnología propia, que libere la necesidad de depender de productos terminados y técnicos o expertos extranjeros que los ajusten a las necesidades particulares de las instituciones de salud nacionales. Así también, se busca sacar ventaja en cuanto a costo-beneficio, ventaja que se da con el hecho de automatizar con ayuda de una computadora, el procesamiento y análisis de señales de esta naturaleza.

Este trabajo está encausado a mostrar la utilización de los elementos necesarios para el diseño e implementación de un sistema de cómputo aplicado al procesamiento digital de señales electroencefalográficas, por tanto se presentan tópicos relacionados con la electroencefalografía, el procesamiento digital de señales y el diseño de hardware y de software.

El presente trabajo se estructuró de la siguiente manera: En el capítulo uno, se tratan las generalidades sobre la electroencefalografía, conceptos que permitirán conocer sobre los dispositivos necesarios para la adquisición y adecuación de las señales; así como la historia del desarrollo y aplicación de la electroencefalografía. El capítulo dos presenta una introducción al procesamiento digital de señales y algunas de sus aplicaciones, que se consideraron representativas de los campos de análisis científico en que se involucra el PDS (se recomienda al lector un conocimiento previo en teoría de sistemas lineales que incluya sistemas continuos y discretos, aunque éste puede ser prescindible). En el capítulo tres se presenta el diseño del *hardware* y del *software* que conformarán el sistema, y se presentan las diferentes etapas por las que pasará la señal, primero en su forma física, es decir como una corriente eléctrica que se registra y acondiciona, para ser posteriormente obtenida por la computadora en forma digital; y después en su forma lógica donde se realiza con la computadora el tratamiento numérico y la presentación de los resultados y las salidas obtenidas. En el capítulo cuatro se explican el desarrollo y la integración del sistema. Aquí se presta atención a algunos antecedentes de sistemas análogos varios de ellos ya realizados y otros solamente propuestos, y es aquí donde se desarrolla la interfaz de usuario, que resalta como una parte importante de cualquier sistema y por ende, de éste. En el capítulo cuatro es donde ya se muestra, como es la realización, aunque no siempre física, de todas las ideas y soluciones tratadas en los capítulos anteriores. Después de los cuatro capítulos descritos anteriormente vienen las conclusiones, donde se evalúan los resultados y en general se dictamina sobre el diseño, realización y funcionalidad de este sistema. Finalmente, se proporciona un panorama de las perspectivas que se alcanzan a ver para un sistema de esta naturaleza, mismas que abren un panorama de trabajos futuros realizables en electroencefalografía y en áreas íntimamente relacionadas, como serían el procesamiento digital de señales de audio, el procesamiento digital de imágenes etc.

Al final de este documento se encuentran la bibliografía<sup>3</sup> y los apéndices donde se han colocado un resumen de términos usados en EEG, la hojas técnicas de los circuitos integrados utilizados y las reproducciones impresas de los programas fuente del sistema.

Con este trabajo, se espera aportar al área de la Ingeniería en Computación, principalmente información teórica y ejemplos prácticos, acerca de estrategias de diseño y de implementación, que sean la base para el desarrollo de aplicaciones de diversa índole, que involucren el procesamiento de señales utilizando una computadora digital.

---

<sup>3</sup> Las referencias bibliográficas en este trabajo se presentan entre paréntesis cuadrados, mismos que encierran una clave de cuatro letras y dos números, que representan el apellido del autor (o autores) y el año de edición del ejemplar, por ejemplo [Ast90]. La bibliografía está ordenada alfabéticamente por esta clave.

## Capítulo 1

# Generalidades del procesamiento digital de señales electroencefalográficas

Con objeto de ahondar en materia es importante tener presentes ciertos conceptos que permitirán ir dando una idea clara de los elementos involucrados y la manera en que se interrelacionan en el procesamiento digital de señales. Por esta razón se empleará la primera parte de este capítulo para presentar de manera general el funcionamiento del cerebro, específicamente de las neuronas. Esto servirá para entender el por qué de la electroencefalografía, sus inicios, sus objetivos y sus aplicaciones en general. Además se hará una presentación de las señales electroencefalográficas y la manera en que son registradas.

En la segunda parte del capítulo se presenta, sin exceso de detalle, otro cúmulo de conceptos que se podría decir son la base técnica, eléctrica y electrónica, que permitirá trabajar la señal de interés. Es decir como es sensada, sometida a un proceso de amplificación y de filtrado y posteriormente muestreada y convertida a forma digital, que es en realidad la forma en la que se podrá empezar a aplicar el *Procesamiento Digital*.

## 1.1 Principios de la Electroencefalografía (EEG)

El conocimiento de la naturaleza eléctrica de la transmisión nerviosa tiene sus orígenes en los estudios realizados por Galvani<sup>1</sup> en la rana, en la Universidad de Bologna, en 1789. Estudios posteriores demostraron que el sistema nervioso presenta propiedades eléctricas, tanto de conducción como de generación de potenciales por la diferencia iónica entre el interior y el exterior de la membrana celular. Este conocimiento resultó ser determinante para posteriormente desarrollar equipos médicos tales como electrocardiógrafos, electromiografos, y por supuesto, electroencefalógrafos.

### 1.1.1 Fisiología Cerebral

El cerebro es la estructura de mayor complejidad en el universo y está constituido por  $10^{10}$  neuronas, donde cada una de ellas recibe y manda millones de impulsos eléctricos de y a otras células del sistema nervioso.

Ramón y Cajal<sup>2</sup>, por medio de técnicas histológicas, confirmó que el sistema nervioso en general y el cerebro en particular no es una masa uniforme de tejido neural, sino que está constituido por un conjunto de zonas heterogéneas. Cada una de estas zonas tiene una arquitectura distinta a la de las otras, comprendiendo un diverso número de neuronas de diferente forma, tamaño, respuestas fisiológicas, etc. Gracias a las técnicas anatómicas desarrolladas hasta el presente, se ha visto que estas zonas interactúan entre sí de múltiples formas, estableciendo circuitos y mallas de alta complejidad.

En forma general el cerebro ha sido dividido en cerebro anterior, cerebro medio (mesencéfalo) y cerebro posterior (rombencéfalo) [Lara87]. El cerebro anterior está constituido por los hemisferios cerebrales, cuya superficie es la corteza cerebral, y comprende al hipocampo, la neocorteza y los campos olfatorios. En los hemisferios se encuentran la amígdala y el cuerpo estriado, que incluye al núcleo caudado y al putamen. La parte inferior del cerebro anterior está constituida por el diencéfalo, formado por el tálamo con un gran número de subdivisiones y el hipotálamo. El cerebro medio está formado por el colículo superior e inferior y el cerebro posterior incluye el cerebelo (ver figura 1.1).

---

<sup>1</sup> Galvani, Luigi. 1737-1798. Anatomista italiano nacido en Bologna. Sus investigaciones sobre la fisiología de los batracios lo condujeron a la hipótesis de que el movimiento muscular era causado por el paso de corriente eléctrica en los tejidos. Descubrió el desarrollo de la electricidad por contacto entre dos metales diferentes (cobre y zinc) con líquido interpuesto. Sus descubrimientos sirvieron de base a Volta para sus estudios sobre la electricidad.

Galvanismo Fis. Electricidad desarrollada por el contacto de dos metales diferentes con un líquido interpuesto. || Fis. Propiedad de excitar, por medio de corrientes eléctricas, movimientos en los nervios y músculos de animales vivos o muertos.

<sup>2</sup> Ramón y Cajal, Santiago. 1852-1934. Histólogo, médico y escritor español nacido en Petilla de Aragón. Se dedicó, entre otras cosas, al estudio del sistema nervioso; aisló la neurona, cuya estructura, fisiología y patología también estableció; sentó las leyes de la conexión sináptica en los tejidos nerviosos de la corteza cerebral y de la médula espinal. Demostró la continuidad de dicho tejido, su complejidad funcional y su organización sistemática. Inventó técnicas de rastreo mediante tinción cromática para los estudios histológicos.

Desde un punto de vista morfológico, la neurona es una célula diferente a las otras células del organismo, ya que del cuerpo neuronal surgen prolongaciones finas que se desarrollan y ramifican a diferentes distancias; estas prolongaciones pueden ser de dos tipos, de acuerdo al papel fisiológico que desempeñan en el paso de las señales de una neurona a otra: las dendritas, que junto con el soma reciben señales de otras neuronas, y el axón, que termina en las dendritas o somas de otras células para establecer un contacto funcional con ellas. Las neuronas, sin embargo, conservan características comunes a otras células como son la presencia de un cuerpo celular o soma, donde se localizan el núcleo que contiene las características genéticas en los ácidos nucleicos, las mitocondrias que actúan como la maquinaria para la obtención de energía de las sustancias nutritivas, el retículo endoplasmático y ribosomas para la síntesis de proteínas y microtúbulos para las funciones de transporte y contracción. Las neuronas, no obstante, no se pueden reproducir como las otras células.

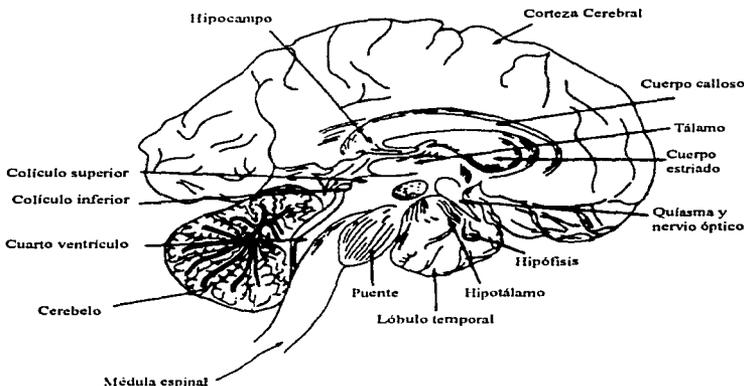


Figura 1.1. El cerebro y algunas regiones cerebrales.

### La Sinapsis

Al punto de contacto entre dos neuronas, Sherrington<sup>3</sup> lo denominó la *sinapsis* [DeMi83]. Se ha encontrado que existen básicamente dos tipos de sinapsis: *eléctrica* y *electroquímica*. En la primera, la transferencia de las señales se realiza simplemente por medios eléctricos, ya que las neuronas están conectadas entre sí, actuando como un cable con subunidades de regeneración de la señal. En la segunda, por otro lado, existe un espacio entre

<sup>3</sup> Sherrington, Charles Scott. 1861-1952. Fisiólogo inglés. Por sus descubrimientos sobre la función de la neurona, compartió el premio Nobel de Medicina de 1932, con E. D. Adrian.

las neuronas que interactúan llamado espacio intersináptico, el cual es de aproximadamente  $200 \text{ \AA}^4$ , pero que impide el paso directo del potencial eléctrico. La transmisión de la información en esta sinapsis se realiza por medio de una sustancia química a la que se ha denominado neurotransmisor, que se libera por medios eléctricos de la neurona emisora y que, a su vez, propicia cambios eléctricos en la neurona receptora.

La sinapsis electroquímica desde un punto de vista estructural está constituida por una porción presináptica que es el ensanchamiento del axón en su parte más distante y que se le ha denominado terminal nerviosa o botón sináptico y por una porción postsináptica constituida por la membrana del cuerpo neuronal o las dendritas (ver figura 1.2). Tanto el botón sináptico como la membrana postsináptica presentan especializaciones morfológicas relacionadas con su función.

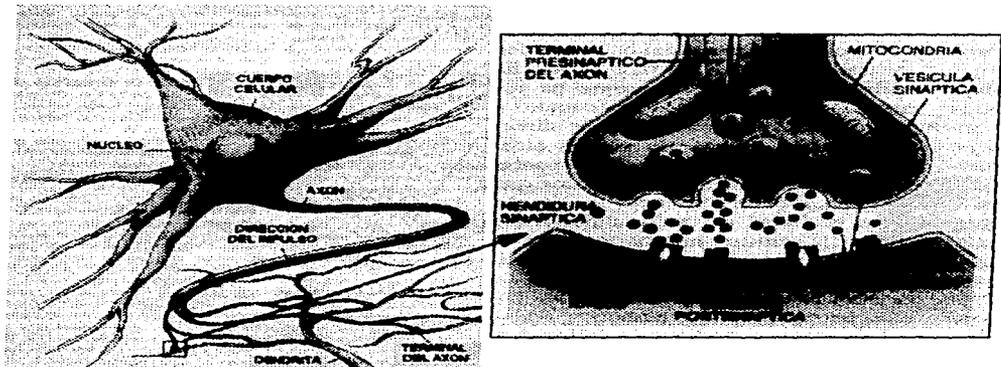


Figura 1.2. Transmisión sináptica.

En el botón sináptico se encuentra un engrosamiento de la membrana en el sitio de contacto funcional con otra neurona y presenta cuerpos vesiculados denominados vesículas sinápticas que almacenan el transmisor. La membrana postsináptica, por otro lado, presenta un engrosamiento en el sitio de contacto con la presinapsis, así como la presencia de un material denso contiguo a este engrosamiento de la membrana.

A la sinapsis de un axón con una dendrita se le denomina sinapsis axodendrítica; a la de un axón con el cuerpo o soma de la célula, axosomática; a la de un axón con otro axón,

---

$1 \text{ \AA} = \text{Angstrom} = 10^{-10} \text{ metros.}$

axoaxónica; y existen casos en que hay sinapsis entre dendritas que se denominan dendrodendríticas.

La membrana de las neuronas actúa como un aislante entre los medios interno y externo de la célula. La membrana celular está formada principalmente por fosfolípidos, que son sustancias con una zona hidrofóbica que propicia que se unan entre sí como un emparedado y es lo que les permite actuar como aislantes, y por proteínas, que pueden estar localizadas en distintas zonas de la membrana y que pueden actuar como mediadores entre las zonas externa e interna de la célula. La membrana neuronal presenta entre el interior y el exterior un potencial de reposo de aproximadamente -70 mv. Debido a que tanto el medio intracelular como el intracelular son ricos en iones de sodio (Na), potasio (K), cloro (Cl) y calcio (Ca), entre otros, y que éstos se encuentran en distinta concentración en el interior y el exterior de la célula, se ha postulado que los potenciales de reposo y de acción son generados por cambios en la concentración de estos iones: a esta teoría se le ha llamado la *Teoría Iónica de los Potenciales de Membrana*.

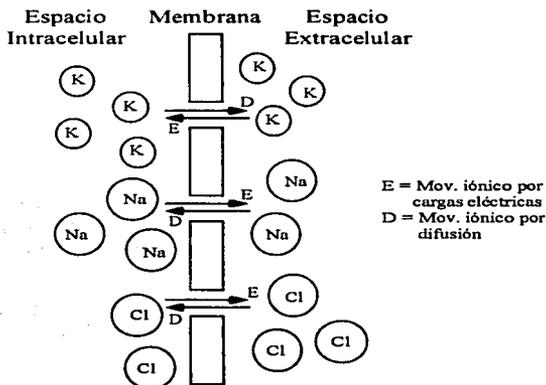


Figura 1.3. Mecanismos iónicos para la generación del potencial de membrana.

Nernst<sup>5</sup> y Planck<sup>6</sup> propusieron que el potencial de membrana surge como resultado de la diferente concentración iónica entre el interior y el exterior de la célula, debida al equilibrio entre la fuerza de difusión que tiende a equilibrar la concentración del ion en el interior y

<sup>5</sup> Nernst, Walter. 1864-1941. Físico y químico alemán. Emitió un teorema sobre el calor y una teoría acerca de la diferencia de potencial entre dos electrodos. Premio Nobel de Química en 1920.

<sup>6</sup> Planck, Max. 1858-1947. Físico alemán. Investigó la termodinámica. Formuló una ley sobre la conservación de la energía y otra sobre la radiación, llegando a la *Teoría de los Cuantos*. Premio Nobel en 1918.

exterior de la membrana, y las fuerzas eléctricas provocadas por las cargas eléctricas de los diferentes iones. El potencial de reposo a un ion específico es, por tanto, el potencial eléctrico que permite contrarrestar la fuerza de difusión provocada por la diferencia de concentración del ion entre el interior y exterior de la célula (ver figura 1.3). Los iones que influyen más fuertemente para la generación del potencial de membrana son: el potasio (K), el sodio (Na) y el cloro (CL).

Al modelo básico general que explica como se produce la transmisión sináptica con los mecanismos iónicos responsables de los potenciales de reposo y de acción de la neurona se le llama *modelo básico de la transmisión sináptica*.

### *Modelo Básico de la Transmisión Sináptica*

El transmisor químico se encuentra localizado en la terminal presináptica en las vesículas (ver figura 1.2). Al llegar el potencial de acción a la terminal presináptica, produce una despolarización que provoca que los iones de calcio localizados en el medio externo se introduzcan al interior de la terminal y, al hacer contacto con las vesículas y la membrana, provocan la salida del transmisor. El transmisor sale al espacio intersináptico llegando a la terminal postsináptica donde es recibido por un receptor específico para esa sustancia. La unión del transmisor y el receptor provoca cambios en la permeabilidad de la membrana a los distintos iones localizados tanto en el interior como en el exterior de la membrana, provocando con ello cambios en el potencial de reposo de la célula. Si los cambios en la permeabilidad de la célula son en los iones de sodio, ocurre una despolarización del potencial, es decir, una disminución del potencial de reposo, por lo cual se le considera un *transmisor excitador*; mientras que si la permeabilidad se modifica a los iones potasio y cloro, entonces ocurre una hiperpolarización del potencial de membrana, y al transmisor que la provoca se le denomina *transmisor inhibidor*. En caso de que la despolarización de la membrana llegue a un valor umbral, la neurona genera un potencial de acción como resultado de un incremento en la permeabilidad de la membrana a los iones sodio y potasio.

En la membrana postsináptica existen ciertas sustancias que transforman al transmisor en otro compuesto para evitar que continúe su acción sobre la postsinapsis. Debido a la naturaleza vesicular de la transmisión sináptica en la placa neuromuscular, el transmisor se libera en cantidades discretas proporcionales a un valor mínimo dado, por lo que se le considera cuántica, donde el cuanto corresponde a  $10^4$  moléculas de acetilcolina<sup>7</sup>.

Este modelo básico, sin embargo, ha sufrido variaciones debido a estudios recientes y al descubrimiento de otras sustancias que pueden actuar como neurotransmisores. Así, se ha mostrado que el transmisor sináptico además de almacenarse en vesículas, existe en forma libre en el citoplasma, lo que sugiere que podría usarse otro tipo de mecanismos para la liberación del transmisor.

---

<sup>7</sup> Acetilcolina. Nombre que se da al compuesto orgánico que generan los extremos de muchas neuronas. Humor neural.

Se sabe que los neurotransmisores se pueden sintetizar en el soma y transportarse a la terminal sináptica por medio de microtúbulos en unos casos, pero también pueden sintetizarse en la terminal presináptica misma, donde pueden almacenarse en pozas disponibles para la liberación o en pozas de reserva.

Existen mecanismos que se encargan de eliminar el transmisor del espacio intersináptico, tales como la degradación del transmisor en el espacio intersináptico mismo, o la recaptación del transmisor por la terminal presináptica. El primer caso es un ejemplo de la degradación de la acetilcolina, mientras que el segundo es un ejemplo de la forma en que los aminoácidos transmisores son eliminados del espacio intersináptico.

Actualmente se han encontrado de diez a veinte sustancias que son consideradas neurotransmisores dado que cumplen con las siguientes características:

- a) La sustancia es sintetizada en la célula.
- b) Al ser estimulada fisiológicamente la célula, la sustancia es liberada al espacio intersináptico.
- c) Existen sistemas que eliminan la sustancia del espacio intersináptico.
- d) Tienen un receptor postsináptico específico que produce cambios en el potencial de la membrana.

Por tanto, se tiene que la cantidad de transmisor liberado y su recepción depende de la interacción de cinco sistemas y que cada uno de ellos puede ser regulado independientemente. Estos sistemas son los siguientes:

- a) El sistema de síntesis o movilización del transmisor de una poza no disponible para ser liberado a otra desde la cual puede ser liberado.
- b) La fracción de transmisor liberado que depende de la concentración tanto interna como externa de calcio, así como de la permeabilidad de la membrana a este ion.
- c) Sistemas de degradación o eliminación del transmisor del espacio intersináptico.
- d) Sistemas que regulen el efecto del potencial eléctrico.
- e) Sistemas que regulen la sensibilidad del receptor postsináptico.

### **1.1.2 Antecedentes de la Electroencefalografía**

Fue hace más de cien años, en 1875 cuando R. Caton de Liverpool registró la actividad eléctrica de la corteza cerebral al aire libre en los animales. Ciertamente existieron investigadores con anterioridad, que partiendo del descubrimiento de Galvani<sup>1</sup> en 1789, sospecharon que el sistema nervioso central, regidor de los músculos, podía producir impulsos eléctricos, pero Caton logró la demostración experimental de este hecho, así como más tarde lo

lograran Danielewsky, en 1876, en Charkow, Fleisch Von Markow, en 1883, en Viena y Beck, en 1890, en Cracovia.

Los resultados alcanzados hasta esos momentos fueron, sin embargo, un tanto inconstantes debido a la pantalla que constituyen las envolturas del cerebro y a las técnicas de detección no perfeccionadas utilizadas en aquellos tiempos. Dándose como consecuencia que se pusiera en duda durante largo tiempo la seguridad de que fueran procesos eléctricos los fenómenos que acompañaban a la función de las células del cerebro. Así según Tschieriev, en 1904 los efectos electromotores dependerían de la varia replicación de los vasos sanguíneos de la corteza cerebral. Pero entre los varios que refutaban aquella objeción, estuvo Prawdicz-Neminsky, quien con sus descripciones de los fenómenos eléctricos del córtex del perro de 1913 a 1925, se aproximó mucho a los actuales puntos de vista sobre estas cuestiones. Distinguía este investigador siete tipos de ondas eléctricas en el cerebro, aunque no llegó a sacar conclusión alguna para su empleo clínico en el hombre.

El privilegio de conseguir, en 1924, la primera imagen gráfica de las corrientes del cerebro en la piel intacta de la cabeza del hombre fue de Hans Berger, director de la Clínica Psiquiátrica de la Universidad de Jena en Alemania, quien registró primero pulsaciones eléctricas colocando electrodos sobre el pericráneo de un adulto joven trepanado, pero pronto demostró la posibilidad de derivar los potenciales a través del cráneo intacto, como lo había hecho Fleisch Von Markow anteriormente en perros. Berger obtenía la derivación, en aquellos primeros momentos, con dos agujas de plata, haciendo la observación en un galvanómetro de cuerda y encontrando en las regiones occipitales variaciones regulares de potencial que sólo alcanzaban la vigésima parte de la tensión de un electrocardiograma y que fueron denominadas por él "ondas alfa<sup>8</sup>". Para la obtención de los registros utilizó un equipo de amplificación cuyo objeto era transformar la corriente eléctrica generada por el cerebro, de una magnitud de millonésima de voltio (microvoltios), en voltios capaces de mover un galvanómetro al cual estaba fijada una pluma inscriptora. Berger retrasó la publicación de sus resultados hasta el año 1929, prosiguiendo con un trabajo tenaz hasta 1938 para anticipar con ello casi todas las observaciones fundamentales de esta materia. De esta forma, fue el fundador de la electroencefalografía clínica. Sus primeras publicaciones fueron rechazadas por fisiólogos y neurólogos. La electrocardiografía había demostrado, en 1910, que la actividad del músculo cardíaco va acompañada de fenómenos eléctricos que se hallan en una determinada relación con su energía. Los tejidos inductivos no desarrollan ninguna descarga eléctrica. Puesto que el "ritmo de Berger" (ritmo alfa) sólo aparece en un individuo en reposo y se esfumaría al abrir éste los ojos, estaría en franca oposición con la experiencia de que se disponía en aquellos momentos. Pero a partir de los trabajos del fisiólogo inglés Adrian, en 1934, realizados en

---

<sup>8</sup> El símbolo  $\alpha$  se emplea en los textos tanto para indicar alfa simplemente, como onda alfa. En círculos especializados se habla de actividad alfa, husos alfa, etc., en lugar de actividad de ondas alfa (actividad  $\alpha$ ), husos de ondas alfa (husos  $\alpha$ ). Lo mismo cabe decir de las restantes ondas y de sus símbolos griegos.

experimentos contundentes con la jaula de Faraday<sup>9</sup> con protección perfecta, no cupo ya la menor duda acerca de la existencia de una actividad rítmica en reposo del cerebro humano.

En Alemania fueron pocos los investigadores que, en los primeros momentos, aprovecharon las ideas de Berger. Kornmüller observó en 1932, la existencia de diferencias en la actividad eléctrica de distintos campos de la corteza cerebral, mediante derivación directa del córtex y, más tarde, en el cráneo cerrado en animales, sobre todo conejos y gatos. De él procede también la primera descripción de las "descargas de corrientes convulsivantes" en los epilépticos. Rohracher, en 1934, probó la utilidad del método en problemas psicológicos y desarrolló una teoría sobre las ondas alfa. Tönnies, así como Foerster y Altenburger comunicaron, en 1934 ó 1935, las primeras derivaciones directas de la corteza cerebral en el hombre durante operaciones del cráneo.

Después de los trabajos de Adrian, recibió la electroencefalografía, en 1934, mejor acogida en los países anglosajones que en la patria de Berger. A partir de entonces pudo Jasper, recoger corrientes cerebrales en muchas operaciones quirúrgicas. Gibbs, Davis y Lennox, siguiendo el concepto generalmente admitido de Kornmüller sobre las descargas de corrientes convulsivantes, encontraron una relación entre muestras punta-onda regulares, que aparecían en número de tres por segundo, con el *petit-mal* de los epilépticos. Grey Walter, en Londres, estableció en 1936, la técnica para la localización de los tumores con el EEG.

Durante la segunda guerra mundial se ensayó el empleo práctico de la electroencefalografía para establecer la aptitud para el vuelo de los aviadores y para diagnóstico en las heridas del cráneo. En Alemania fue durante largo tiempo el "neurógrafo" ideado por Tönnies, en 1932, el modelo de los primeros aparatos de EEG, pero dado que la industria americana sufrió menos a consecuencia de la guerra, algunas empresas de aquél país iniciaron la fabricación en serie de estos aparatos.

En los primeros electroencefalógrafos se disponían por separado amplificadores de entrada y generales, correspondientes a un solo canal, por lo que los aparatos de tres o cuatro canales con oscilógrafo de cinta, resultaban pesados, no eran movibles y eran delicados y susceptibles a fallar.

Los avances experimentados en la técnica amplificadora han permitido que los aparatos actuales resulten más sencillos, con menos peso y que ocupen menos espacio. Es posible, incluso, trasladar aparatos de ocho o más canales desde el laboratorio a la sala de operaciones o a la cabecera de la cama del enfermo. Así también, los progresos de la electrónica han hecho fácil actuar sobre registradores mecánicos o catódicos, de constantes de tiempo adecuadas, para cada tipo de investigación. A partir de todo esto, la electroencefalografía ha experimentado un rápido desarrollo. Aun así, hay bastante por hacer para satisfacer por completo los deseos de los psicólogos y psiquiatras, en miras a contar con un método experimental para el estudio de las relaciones entre mente y cerebro.

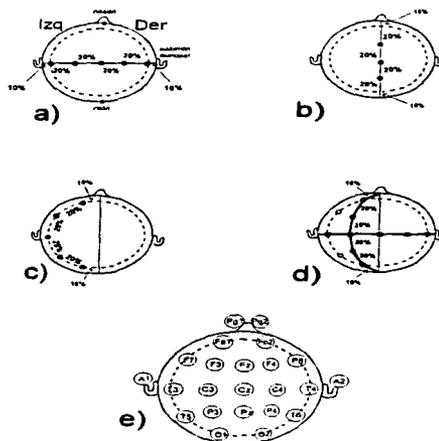
---

<sup>9</sup> La jaula de Faraday es una habitación cuyas paredes y techo están conformados por varias capas de madera, arena y malla de alambre, que aíslan el interior de la misma de la presencia de ondas electromagnéticas exteriores.

### 1.1.3 Técnicas de Registro del EEG

La técnica de registro del EEG es la técnica mediante la cual se sensa o recoge con electrodos el potencial eléctrico producto de la actividad eléctrica espontánea del cerebro.

En las técnicas de registro del EEG más comunes los electrodos se posicionan en la cabeza siguiendo un arreglo conocido como sistema 10-20. Este es un esquema de posicionamiento recomendado por la Federación Internacional de Sociedades de Encefalografía, y cuya colocación se basa en ciertas características anatómicas de la cabeza. Los puntos en los que se colocan los electrodos sobre la piel cabelluda son simétricos en los dos lados de la cabeza, y equidistantes unos de otros. Esto, con el objeto de medir actividades en ambos lados de la cabeza con fines de comparación.



**Figura 1.4. Líneas de referencia, figuras (a), (b), (c) y (d); en (e) se muestra la distribución final de los electrodos. Las letras indican las áreas o lóbulos del cerebro: F es frontal, C es central, P es parietal, O es Occipital y T es temporal. La designación Pg corresponde al punto nasofaríngeo y una A indica el pabellón auricular.**

En el sistema 10-20 se tienen cuatro líneas de referencia principales que se ilustran en la figura 1.4. Los puntos que marcan el inicio y final de las líneas son: los pabellones auriculares, el nacimiento de la nariz o *nasión* y la protuberancia de lóbulo occipital o *ión*. La primera línea une los pabellones auriculares, como se muestra en la figura 1.4 (a). La primera marca se coloca al 10% de la extensión total de la línea y las restantes marcas se colocan en segmentos de 20% , siendo el último segmento de la línea de 10% con lo que se completa el 100% de la

longitud total. La segunda línea une la *nasión* con el *inión*, y las marcas siguen el mismo patrón de 10%, seguido de segmentos de 20%, como se muestra en la figura 1.4 (b). La tercer línea es circular y se sitúa alrededor del cráneo, pasa por los segmentos de 10% de las dos líneas anteriores, como lo muestra la línea punteada de la figura 1.4 (c). La extensión total de la línea no es el perímetro de la circunferencia, sino la mitad de éste, ya que el cráneo se divide en dos porciones simétricas, y la división aplicada al lado izquierdo se aplica de igual forma al lado derecho. El inicio y fin de la línea lo constituye el cruce de la circunferencia con la línea que va de la *nasión* al *inión*. La totalidad de la longitud de la línea se divide al igual que las líneas anteriores en segmentos de 10% y 20%; por último, se tiene una línea que une la *nasión* con el *inión*, pero pasa por la primera marca de 20% de la línea que une los pabellones auriculares. Esta línea se muestra en la figura 1.4 (d), y tiene una línea simétrica del lado derecho de la cabeza.

La operación básica de un EEG mide diferencias de potencial, éstas se refieren a la diferencia de potencial obtenido entre dos electrodos. Existen tres formas de medir esta diferencia de potencial (figura 1.5): en la primera, la medición de voltajes puede ser tomada entre dos electrodos adyacentes, esta conexión se conoce como *derivación bipolar*; en la segunda, conocida como *derivación referencial*, las mediciones pueden ser tomadas de uno de los electrodos, a un punto de referencia común formado por un circuito resistivo conocido como la referencia media de Wilson, que promedia las actividades eléctricas de todos los electrodos de la piel cabelluda; por último, se tiene la *derivación unipolar*, en ésta, los electrodos de referencia deben ser colocados en regiones de actividad cerebral insignificante, tales como el pabellón auricular o las fosas nasales. Este esquema tiene el propósito de medir de forma individual el potencial eléctrico de cada electrodo.

Un montaje de electrodos es la selección de electrodos de los que el técnico requiere registrar la información, y constituye el conjunto de derivaciones registradas simultáneamente. Regularmente se utilizan montajes de 8 o 16 electrodos. Un montaje de ocho electrodos se muestra en la figura 1.6. En este montaje se utiliza la conexión llamada *derivación unipolar*, ya que todos los electrodos están referidos al pabellón auricular. Como se observa en la figura, se toman lecturas de electrodos simétricos a cada lado de la cabeza, en este montaje de ocho electrodos, éstos se comparan por pares, de modo que la lectura del electrodo F3 se compara con la lectura del electrodo F4, C3 con C4, P3 con P4, y finalmente O1 con O2.

Para el registro habitual de señales, las áreas de la piel cabelluda se limpian con alcohol, acetona o *redux* para provocar una mejor adherencia de los electrodos, y usualmente se marcan con un lápiz rojo. Esta adherencia se logra mediante el uso de tres técnicas: pasta, cemento colodión y agujas subdérmicas. La técnica del colodión es utilizada para obtener las formas de onda alfa, beta, delta y theta.

El cemento colodión es un pegamento de secado rápido que se aplica de dos formas principalmente: la primera es el método de la gasa, en este caso el electrodo se fija firmemente en la cabeza con una mano, y una pequeña cantidad de colodión es aplicada en un pedazo de gasa. La gasa empapada de colodión se coloca sobre el electrodo y se seca con aire comprimido. Entonces, una pequeña cantidad de gel electrolítico o gel conductor se inserta a través de un hoyo de la gasa y se introduce dentro del electrodo. Al mismo tiempo que el gel

electrolítico se inserta, la piel cabelluda se raspa con una jeringa sin filo. lo anterior para reducir la resistencia de contacto del electrodo. La otra aplicación del colodión es similar a ésta, pero con la diferencia de que se omite la gasa, y el cemento colodión se aplica directamente a la zona de contacto del electrodo con la piel cabelluda.

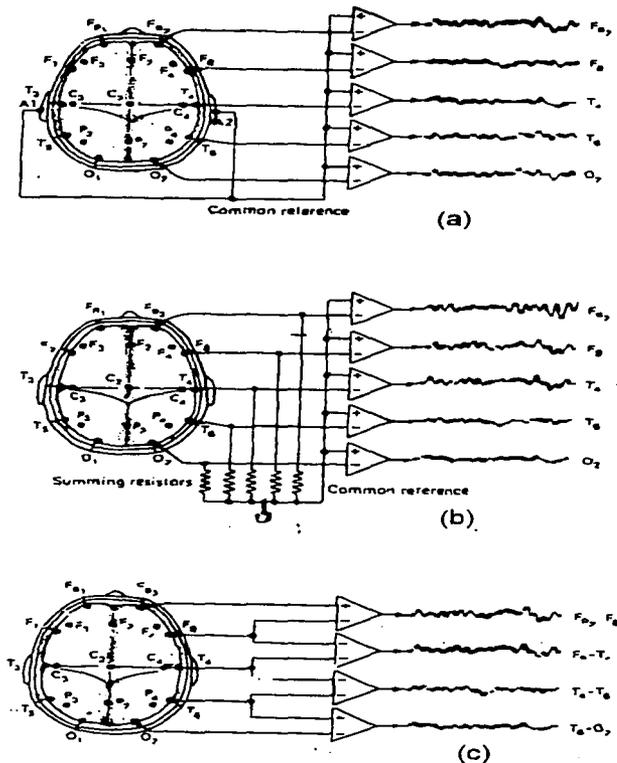
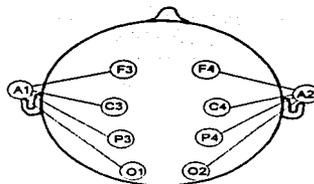


Figura 1.5. Tres configuraciones de amplificación del registro de la señal EEG: (a) Derivación unipolar, (b) Derivación referencial, (c) Derivación bipolar.



**Figura 1.6. Montaje de ocho electrodos.**

En la técnica de la pasta se llena el electrodo de forma cónica con una pasta adherente de electrodos. Después se coloca sobre el electrodo una pequeña cantidad de pasta y finalmente se cubre con una torunda (bola de algodón) o con una gasa.

El método de las agujas subdérmicas emplea una aguja llamada electrodo esfenooidal, que se coloca rompiendo la piel. Aunque este método mejora la resistencia de contacto del electrodo tiene implícito el peligro de infección.

Es preciso señalar que, una vez colocado el electrodo, la resistencia de contacto no debe ser mayor de  $10\text{ k}\Omega$ , aunque algunas fuentes citan que esta resistencia no debe superar los  $5\text{ k}\Omega$ . Es por esto que la piel, en la zona donde se coloca el electrodo, requiere de ser raspada. Este procedimiento, aunque es incomodo para el paciente, implica menos molestias y riesgos que la técnica de la aguja esfenooidal.

La amplitud de la actividad encefalográfica registrada por los electrodos en la piel cabelluda es generalmente del orden del 5 a 200 microvoltios, por lo que requiere de ser amplificada 50,000 veces para ser tomada por el sistema de adquisición de datos. Esta amplificación se lleva a cabo mediante un arreglo de amplificadores operacionales. Es deseable que la impedancia de entrada de los amplificadores diferenciales sea alta para minimizar el efecto (carga) del instrumento sobre la señal encefalográfica, detectada en el cuero cabelludo. Si la etapa de amplificación presenta una impedancia de entrada baja, cargará la señal, causando distorsión y provocando imprecisión en las lecturas de los registros. La impedancia de entrada del instrumento deberá ser de aproximadamente  $25\text{ M}\Omega$ . Esta alta impedancia de entrada es necesaria cuando la señal está presente en los electrodos. De esta forma, la señal se presenta inalterada y lista para la etapa amplificadora. Sin embargo, los cables provenientes de los electrodos son vulnerables a roturas. Además, surgen problemas con el pelo, mismo que tiende a incrementar la resistencia de contacto, además de producir *artefactos* que perturban la señal del cerebro.

De esta forma, los cables actúan como antena, de tal forma que se convierten en conductores para señales de radio, TV, y seguramente para la señal de 60 ciclos por segundo que se radia desde los cables de energía. La señal de 60 Hz. es el componente de señales parásitas más importante, puesto que cae en el espectro de frecuencia del la señal de EEG a registrar.

Estas interferencias pueden ser reducidas mediante el empleo de amplificadores operacionales, instrumentados en filtros paso-bajo, paso-alto y paso-banda. Aunque el uso de filtros supresores de banda pueden causar distorsión de fase y reducir la ganancia, este efecto se minimiza por el hecho de que las señales más importantes en EEG tienen una frecuencia por debajo de los 30 Hz.

### 1.1.4 Características de las señales de un EEG

Como se mostró anteriormente, las señales que constituyen la entrada de un EEG son originadas por potenciales bioeléctricos generados en el cerebro. Bajo el dominio del tiempo, dichos potenciales son conocidos comúnmente como ondas cerebrales.

Las ondas cerebrales se forman de la suma de las variaciones en los potenciales eléctricos producidos por las neuronas, debidas al estímulo provocado por los sentidos o bien por los procesos del pensamiento.

En la superficie del cerebro estas diferencias de potenciales eléctricos son del orden de los 10 mV; sin embargo, los electrodos que se encargan de hacer la medición no lo hacen directamente en la superficie del cerebro, sino que miden la actividad eléctrica que se propaga a través de los huesos del cráneo y el cuero cabelludo. Esta propagación provoca que la onda cerebral sea captada con una considerable atenuación, obteniendo lecturas de potencial no mayores a 200  $\mu$ V. Estos potenciales varían de acuerdo a la posición del electrodo sobre la superficie del cráneo, lo que hace necesario agrupar los electrodos de acuerdo con los diferentes lóbulos del cerebro (frontal, parietal, temporal y occipital).

En una señal electroencefalográfica, la característica importante es la frecuencia, a diferencia del electrocardiograma donde el rasgo importante es la forma de onda obtenida.

Las ondas cerebrales son señales no periódicas, es decir, su amplitud, frecuencia y fase cambian continuamente; es por esto que los muestreos de ondas cerebrales deben tomarse durante un periodo de tiempo considerablemente amplio.

El estudio de las ondas cerebrales ha permitido reconocer en ellas ciertos patrones determinados conocidos como ritmos, los cuales corresponden a estados fisiológicos o patológicos bien definidos.

Los diferentes ritmos eléctricos se designan con letras griegas, y todos ellos, en general, se caracterizan por:

- a) **Su frecuencia.**- Medida en Hz.
- b) **Su amplitud.**- Medida en  $\mu$ V.
- c) **Su morfología.**- Cada uno de los ritmos tiene formas de onda que son típicas.

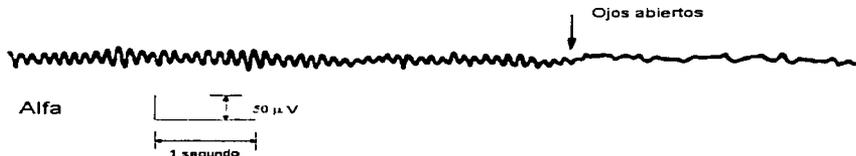
- d) **Su ubicación.**- Referida a la superficie del cuero cabelludo y en función de su origen y predominancia en una región de los hemisferios.
- e) **Su simetría o asimetría.**- Con respecto a cada lado de la cabeza.
- f) **Su simultaneidad o alternancia.**- También referida a cada lado de la cabeza.
- g) **Las circunstancias.**- Bien sean fisiológicas o patológicas las cuales determinan su aparición, mantenimiento, bloqueo o desaparición.

Existen cuatro tipos básicos de ritmos eléctricos los cuales se muestran en la tabla 1.1, donde se presentan los rangos de frecuencia y amplitud que se utilizan como criterio para diferenciarlos.

Tipo de Ritmo	Frecuencia del EEG. (Hz)	Amplitud del EEG. ( $\mu V$ )
Alfa $\alpha$	$8 \leq f < 13$	10 - 50
Beta $\beta$	$13 \leq f < 30$	5 - 10
Delta $\delta$	$0.5 \leq f < 4$	100 - 200
Teta $\theta$	$4 \leq f < 8$	10 - 50

**Tabla 1.1. Frecuencia y amplitud asociada a los cuatro tipos de ritmos eléctricos.**

El ritmo alfa fue el primero registrado por Hans Berger. Se presenta prácticamente siempre en adultos normales. Consiste de ondas sinusoidales regulares cuya frecuencia se presenta en el rango de 8 a 13 Hz y cuya amplitud varía desde 10 hasta 50  $\mu V$ . La amplitud raramente es constante, su variación es periódica, formando ondas intermitentes (llamadas ondas alfa) o bien, más a menudo, para formar valles. La presencia y ausencia de ondas y valles se presenta de manera simultánea en ambos hemisferios, siendo las primeras comúnmente más amplias en el hemisferio dominante (esto es, en el hemisferio derecho). El ritmo alfa se puede detectar primordialmente en la región occipital del cerebro. Una onda alfa tiene el aspecto que se muestra en la figura 1.7.



**Figura 1.7. Ejemplo de una onda alfa característica.**

Al ritmo alfa se le conoce también como ritmo de reposo, ya que se presenta precisamente cuando el sujeto está en reposo, recostado, con los ojos cerrados, sin estimulaciones sensoriales y en estado de relajación intelectual, pero sin llegar al estado de sueño.

Resulta fácil detener la presencia de ondas alfa, la simple apertura de los ojos es suficiente para que las ondas disminuyan de amplitud y desaparezcan en tan sólo algunas décimas de segundo. En resumen, el bloqueo y desincronización del ritmo alfa es producido por la atención visual y por toda circunstancia que origina fijar la atención y concentración en el dominio de la visión (tentativa de ver, recuerdo visual, representación visual, etc.)<sup>10</sup>.

El ritmo Beta debido a su débil amplitud normalmente no puede ser registrado de manera transcraneal, por lo que se le identifica siempre en las derivaciones corticales.

Se caracteriza por ondas sinusoidales de amplitud baja (de 5 a 10  $\mu\text{V}$ ), y una frecuencia del orden de los 13 a los 30 Hz. La forma de onda es normalmente de ondas o valles, presentándose de manera asimétrica en ambos hemisferios (ver figura 1.8). En un adulto normal estas ondas se presentan en la región frontal del cerebro.

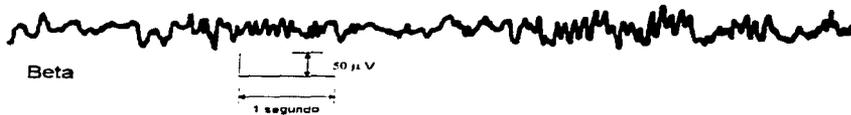


Figura 1.8. Ejemplo de onda beta.

El ritmo Beta desaparece durante el sueño; además, se ha llegado a la conclusión que este ritmo está altamente relacionado con la actividad motora voluntaria y la preparación para la acción, ya que para bloquearlo es necesario que se presenten condiciones tales como la ejecución voluntaria de un movimiento, cruzar los brazos, cerrar los puños o simplemente imaginar un movimiento.

El ritmo Delta se caracteriza por ser de ondas irregulares con una frecuencia baja del rango de los 0.5 a los 4 Hz y una amplitud mayor con respecto a los demás ritmos de entre los 100 y 200  $\mu\text{V}$ . Es una señal característica en individuos de la primera infancia.

Se distinguen cuatro tipos de ritmo delta, los cuales se describen a continuación:

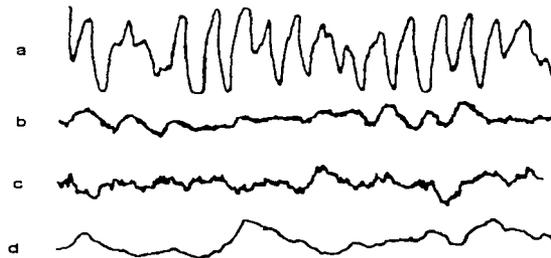
- **Delta monomorfo.**- Es aquel en el que las ondas lentas<sup>11</sup> son regulares, a menudo de gran amplitud.

<sup>10</sup> No sólo la atención visual genera un bloqueo de ondas alfa, en general, los estímulos de cualquier naturaleza que fijen la atención (interés, esfuerzo por percibir, etc.) han de bloquear este ritmo.

<sup>11</sup> Una onda lenta es aquella de duración superior a 1/8 de segundo.

- **Delta polimorfo.**- Es aquel en el que los elementos son más irregulares, más lentos y frecuentemente no son tan amplios como el delta monomorfo.
- **Delta regular.**- Presenta características intermedias al delta monomorfo y polimorfo con más variaciones en amplitud.
- **Delta irregular.**- Es una combinación entre el delta monomorfo y el delta polimorfo.

En la figura 1.9 se presentan ejemplos de los ritmos delta.



**Figura 1.9. Ritmos delta: a) Monomorfo, b) Regular, c) Irregular o mixto, d) Polimorfo.**

El ritmo Teta normalmente aparece “enmascarado” por el ritmo alfa, lo cual lo hace un tanto difícil de percibir en un individuo normal. Se caracteriza por ondas sinusoidales de 4 a 8 Hz y entre los 10 y 50  $\mu\text{V}$  de amplitud. Dichas ondas se manifiestan como breves accesos en forma de valles, los cuales son bilaterales, simultáneos y simétricos en ambos lados del cerebro. En la figura 1.10 se observa una onda teta típica.



**Figura 1.10. Ejemplo de onda Teta.**

Debido a que las ondas cerebrales son individuales y únicas, es decir, no se repiten entre un individuo y otro, es difícil clasificarlas; sin embargo, se han podido establecer patrones de normalidad para una señal de acuerdo a la edad y estado de alerta de los sujetos. La tabla 1.2 muestra un resumen de las características de una señal EEG de acuerdo a los criterios ya mencionados:

Edad	Aspecto del EEG	Descripto	Adormecido	Despierto
1 mes	Amplitud:	Baja	Sin cambio	Ligero aumento
	Tipo:	Delta y Teta	Sin cambio	Sin cambio
1 mes - 1 año	Amplitud:	Aumentando o alta	Igual	Igual
	Tipo:	Delta y Teta	Sostenido, rítmico	Valles asincrónicos; duración larga.
1 - 5 años	Amplitud:	Alta	Igual	Igual
	Tipo:	Teta, alfa aumentada, delta disminuida	Rítmica, brotes o sostenido	Valles sincrónicos asimétricos
6 - 10 años	Amplitud:	Disminuida o moderada	Atenuada o brotes	Valles y ondas V disminuidos.
	Tipo:	Alfa aumentada, teta disminuida, delta disminuida o ausente	Frecuencia aumentada de ondas lentas rítmicas	Sincrónica y simétrica.
11 - 20 años	Amplitud:	Moderada o aumentada	Atenuación	Disminución de ondas V y valles.
	Tipo:	Alfa, teta mínimo, delta posterior	Teta rítmica anterior	

**Tabla 1.2. Características de una señal EEG en función de la edad y estado de alerta.**

Hasta aquí se han contemplado las características de lo que se considera una señal electroencefalográfica normal (principalmente los ritmos alfa y beta); sin embargo, los EEG resultan de utilidad en el estudio de ondas cerebrales anormales, causadas por lesiones o disfunciones del cerebro.

### *Ritmos Anormales*

Es posible detectar ritmos anormales de acuerdo a las características de las ondas obtenidas en el EEG así como su distribución en la cabeza. Las ondas anormales se caracterizan por ser focales, bilaterales, hemisféricas o generalizadas, según se hable de alteraciones en un

solo lóbulo o en lóbulos homólogos, en un hemisferio o en ambos hemisferios. Los ritmos anormales se dividen en dos grandes grupos: la *disritmia* y la *arritmia* (ondas delta).

En la figura 1.11 se muestra un resumen de la clasificación de las ondas anormales considerando alteraciones focales y generalizadas. Las ondas mostradas son ejemplos simplificados de carácter ilustrativo y no representan la salida real de un EEG. Tanto las arritmias como las disritmias se gradúan de acuerdo a su intensidad siendo el estado I ligero, el II moderado y el III intenso. Esta graduación se fundamenta con la amplitud y persistencia de la onda, compensando las diferencias en frecuencia de acuerdo a la edad del individuo.

Las ondas anormales de la categoría delta se caracterizan por ser lentas (de 0.5 a 4 Hz), complejas e irregulares, con poca tendencia a repetir la misma forma de onda (de ahí el nombre de arritmia). Es una onda persistente que no es afectada de manera notable por cambios fisiológicos como la apertura de los ojos. Las lesiones que pueden producir estas alteraciones son básicamente los tumores (neoplasias, hematomas y abscesos), infartos agudos causados por insuficiente irrigación de sangre y contusiones agudas debidas a traumatismos externos.

Las ondas anormales del grupo de las disritmias son de frecuencia más elevada que las delta (de 4 a 7 Hz) y más intermitentes. Pueden presentarse en grupos cortos y rítmicos (espiga y onda, ondas sinusoidales) o bien como ondas aisladas y esporádicas con una forma bien definida. Normalmente se inhiben cuando se presenta un alertamiento mental en el individuo. Este tipo de onda tiende a sincronizarse en sitios similares de ambos hemisferios cerebrales.

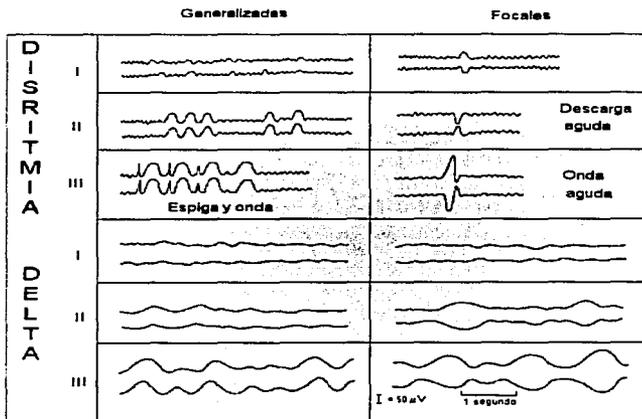


Figura 1.11. Clasificación general de las ondas cerebrales anormales.

### Estados del Sueño

Existen otro tipo de ondas cerebrales características que se presentan mientras el individuo duerme y cuyo estudio mediante un EEG permite establecer correlaciones entre los individuos y algunos estados fisiológicos o psicológicos, figura 1.12. A continuación se describen brevemente las distintas fases del sueño y el tipo de onda cerebral que le caracteriza a cada una.

**Vigilia.**- Esta es una fase previa al sueño caracterizada por un estado de relajación que en la señal EEG se manifiesta como ritmo alfa.

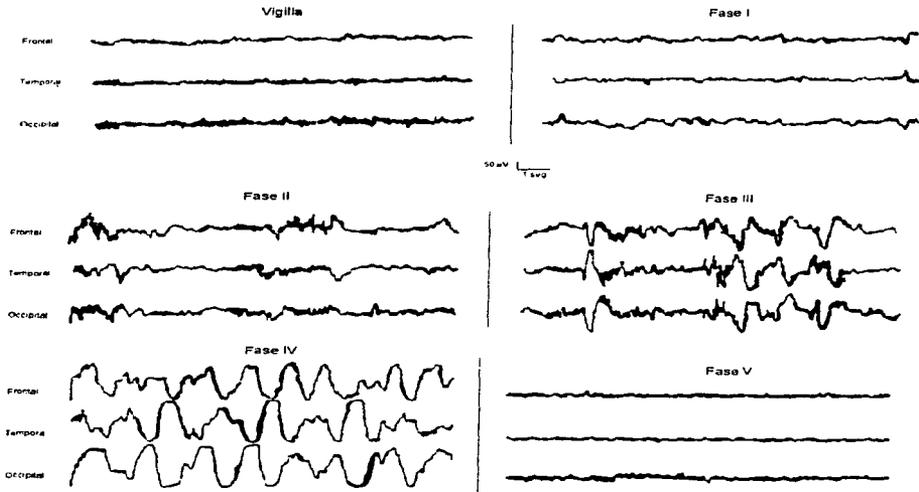


Figura 1.12. Señales EEG del sueño de un adulto mostrando las 5 fases.

**Fase I.**- Sigue inmediatamente al adormecimiento. Es la fase de sueño más ligero y por tanto más fácil de interrumpir; se caracteriza por un movimiento rápido de los ojos (REM por

sus siglas en inglés)<sup>12</sup>. En la señal EEG desaparecen las ondas alfa y se presenta una actividad<sup>13</sup> de amplitud baja. Esta fase es breve y corresponde a un sueño muy ligero.

**Fase II.-** La señal EEG se caracteriza por la aparición de “valles de sueño”, que son ondas sincronizadas, cuya frecuencia es de 14 a 16 Hz y su amplitud es de aproximadamente 50  $\mu$ V, la cual se produce periódicamente sobre una actividad de fondo rápida.

**Fase III.-** Es un sueño profundo. La señal EEG se caracteriza por la aparición de ondas lentas, ondas delta cuya amplitud es del orden de 100  $\mu$ V y su frecuencia está en el rango de los 0,5 a los 4 Hz, asociada a valles que desaparecen progresivamente.

**Fase IV.-** Es un sueño profundo cuya señal EEG contiene básicamente ondas lentas.

**Fase V.-** El sueño de un adulto normal es interrumpido varias veces por esta fase, cuya señal EEG presenta las siguientes características: Actividad compleja de amplitud baja sin valles de sueño ni ondas lentas, como un estado de vigilia sin ondas alfa en el que paradójicamente el sujeto duerme con un umbral de sueño<sup>14</sup> muy elevado.

#### *Estados de Sueño REM y NREM*

REM se deriva de “Rapid Eye Movement” o “Movimiento Rápido de Ojos” y es un estado de sueño en el que los músculos del ojo y el oído medio tienen mayor actividad, dirigidos por impulsos eléctricos regulares que causan el movimiento rápido de los ojos que da nombre a esta etapa de sueño. Al mismo tiempo, otros músculos del cuerpo son inhibidos, siendo ésta la razón por la cual la gente al soñar no puede realizar los movimientos que sueña que hace. En esta etapa, el cerebro disipa más energía que en cualquier etapa del sueño, de hecho, disipa más energía que cuando la persona se encuentra despierta.

NREM se forma de las iniciales de “Non-Rapid Eye Movement” o movimiento no rápido de ojos. Se presenta normalmente dentro de las fases II a IV, caracterizadas por un sueño más profundo. En esta fase es donde se presentan los sueños que la gente no recuerda normalmente, incluso las pesadillas son más comunes en esta etapa de sueño, debidas normalmente a la disminución en la cantidad de oxígeno que llega al cerebro, característica de este estado. Al inicio y al final de el sueño NREM, el individuo experimenta ajustes en la postura, debido a que los músculos no se inhiben durante esta etapa. Como resultado de lo anterior, es durante el sueño NREM que sea posible para el individuo el hecho de caminar dormido.

---

<sup>12</sup> El sueño puede ser dividido en etapas REM y NREM, las cuales se explican más adelante dentro de este apartado.

<sup>13</sup> Se define como actividad toda onda o secuencia de ondas.

<sup>14</sup> Se define como umbral de sueño el nivel de sueño en el que el sujeto es despertado por algún estímulo externo.

---

### 1.1.5 Aplicaciones del EEG

El electroencefalograma es considerado como una extensión del examen neurológico en la evaluación de aspectos de la función cerebral no siempre accesibles a las pruebas clínicas convencionales. La realización del EEG tiene la ventaja de no ser peligrosa y ser relativamente indolora, además de que puede llevarse a cabo en pacientes de cualquier edad.

Las aplicaciones del EEG están en el campo del diagnóstico clínico, y es usado extensivamente en las siguientes áreas:

**Neurología.** Como herramienta clínica, en conjunto con otras pruebas, como el electromiograma (EMG), ecocardiogramas, y exámenes neurológicos, donde es usado para definir patologías cerebrales del paciente.

**Anestesiología.** Uso del EEG para determinar niveles de anestesia administrados a pacientes. Esto es especialmente cierto para pacientes que fueron sometidos a cirugía cardíaca o pacientes que son difíciles de monitorear.

**Neurocirugía.** Neurocirujanos usan el EEG como ayuda para localizar anomalías patológicas, como tumores que se removerán del cerebro quirúrgicamente.

**Psiquiatría.** El EEG es usado para tratar de determinar la presencia o ausencia de una disfunción orgánica para diagnosticar un desorden mental con mayor exactitud.

**Pediatría.** El EEG, junto con otras pruebas, como la de potenciales provocados promediados, son usados para determinar problemas de visión y auditivos en el recién nacido.

Antes de proceder a estudiar las aplicaciones del EEG en el diagnóstico clínico, se introducirá el concepto de anomalías específicas y no-específicas de una señal. Las específicas son aquellas con morfología de onda distinta y correlación relativa alta con alteraciones particulares; las no-específicas son las formadas por ondas no descriptivas y producidas por muchas y variadas alteraciones neurológicas y sistémicas. Para valorar el significado clínico de estas anomalías encefalográficas se requiere amplio criterio. Una anomalía electroencefalográfica tan sólo por ser calificada de "no específica", no debe considerarse como insignificante, ya que en algunas circunstancias tendrá importancia clínica. Cualquier proceso patológico que afecte la función neuronal cerebral es capaz de producir anomalías electroencefalográficas específicas.

Se requiere establecer distinciones básicas entre las enfermedades que afectan primariamente áreas localizadas del cerebro, de aquellas que lo alteran más difusamente. Todo tipo de anomalías electroencefalográficas localizadas son en general significativas, ya que ocurren rara vez en personas asintomáticas, excepto por la senectud. No obstante, sobre la base de una anomalía focal en un EEG, no se distingue el tipo patológico de la lesión y, por lo tanto, con un sólo estudio no debe intentarse la diferenciación entre un tumor, un infarto o un absceso.

El EEG es un proceso dinámico fisiológico que comprende, conjuntamente, dimensiones de tiempo y espacio (biodinámica espacio-temporal). El EEG considerado de esta manera ayuda a entender las correlaciones clínicas y patológicas, además de emplearlo más adecuadamente para el diagnóstico.

En relación con la dimensión espacial, la extensión de la lesión es de importancia fundamental. Una lesión no siempre produce anomalías electroencefalográficas en el trazo habitual hasta que ésta alcanza suficiente tamaño. La densidad de la lesión o la concentración de sus efectos en la corteza adyacente también interviene en los cambios electroencefalográficos. La presión local producida por una masa congregada generalmente produce anomalía focal en el EEG más fácilmente que un tumor, aun aquel maligno infiltrado más difusamente entre las neuronas. Otro factor importante es la localización de la lesión. En el registro convencional, las fluctuaciones potenciales que conforman el EEG derivan de la actividad neuronal de las capas corticales más superficiales en la convexidad de los hemisferios cerebrales; la alteración de las estructuras situadas más profundamente afecta el EEG sólo indirectamente. Así, un tumor cerebral localizado superficialmente crea una alteración eléctrica focal con facilidad, mientras que un tumor de tamaño similar situado más profundo rara vez lo hace.

En la dimensión temporal se debe considerar la selección del tiempo en la cual se obtiene el EEG durante la evolución de la lesión. Un EEG registrado en un periodo temprano del desarrollo de un tumor puede ser normal, mientras que puede haber una anomalía focal si el registro se efectúa más adelante. Por esto los registros secuenciales en un paciente son frecuentemente de ayuda diagnóstica, al tiempo que elevan la posibilidad de registrar anomalías intermitentes que ocurren con poca frecuencia. El tipo y magnitud de la anomalía electroencefalográfica dependen también de la edad del paciente. La velocidad de desarrollo de una lesión y el balance entre las fuerzas destructoras y reparadoras tienen influencia sobre la actividad electroencefalográfica. Por ejemplo, un tumor rápidamente expansivo produce, típicamente, actividad lenta polimórfica persistente, en contraste con un tumor lento que puede causar como única manifestación en el EEG ondas en punta indistinguibles de los efectos causados a veces por una cicatriz.

### *Ejemplos de aplicaciones diagnósticas*

Los siguientes ejemplos de la utilidad del EEG para el diagnóstico clínico se consideran de acuerdo a categorías de enfermedad:

**Tumores.** EL EEG proporciona información importante para el diagnóstico de un tumor cerebral, por medio de la demostración de una anomalía focal cuando el grado de sospecha clínica es bajo<sup>15</sup>. Esta situación sucede cuando los síntomas consisten principalmente en dolor de cabeza, cambios de la personalidad o crisis generalizada, y cuando el examen

---

<sup>15</sup> Los registros secuenciales pueden ayudar a diagnosticar lesiones progresivas. Cuando los primeros trazos del EEG son normales, o anormales pero no localizados, la aparición subsiguiente de una anomalía focal, o bien, el empeoramiento de la misma, pueden sugerir fuertemente un tumor.

neurrológico es normal o ligeramente anormal. Más a menudo, estas circunstancias se encuentran en tumores originados en los lóbulos frontal o temporal.

Algunas veces un sólo trazo indica la posibilidad de una lesión expansiva si existe una discrepancia clara entre el tipo de hallazgos en el EEG y la evolución de la enfermedad. Por ejemplo, la asociación de anomalías focales y bilaterales proyectadas en el EEG de un paciente con una enfermedad crónica que simula una degenerativa, debe conducir a sospechar fuertemente un tumor intracraneal.

Después de la operación de un tumor el EEG es coadyuvante para evaluar el periodo postoperatorio y la posible recurrencia del tumor. Para tal efecto, es importante relizar un trazo poco después de la operación para compararlo con otros subsecuentes ya que el procedimiento quirúrgico y el defecto óseo del cráneo resultante generalmente producen alteraciones en el EEG (aumenta la amplitud de la actividad en las derivaciones por encima del lugar de la operación), lo cual hace diferente al EEG postoperatorio del preoperatorio. El EEG es también útil en la evaluación de los efectos de la radiación o la quimioterapia usados para el tratamiento de enfermedades intracraneales malignas.

**Trastornos Vasculares.** Los principios electroencefalográficos generales de diagnóstico y localización también se aplican a las lesiones cerebrovasculares. La hemorragia intracerebral produce anomalías en el EEG semejantes a las del tumor cerebral. Un infarto cerebral grande en estadio agudo con frecuencia produce manifestaciones similares (delta focal y disritmia bilateral proyectada) cuando se asocia con edema<sup>16</sup>. Sin embargo, en contraste con el tumor, en los EEG secuenciales durante la resolución de un infarto, generalmente se observa disminución progresiva de las anomalías. Unas horas o varios días después de un infarto cortical, el EEG a veces muestra descargas epileptiformes lateralizadas y periódicas (DELP) que frecuentemente se asocian con crisis clínicas. Sin embargo, las DELP se producen también por otros tipos de lesión y no deben ser consideradas siempre indicadores de infarto cerebral.

Hay que saber que un infarto pequeño de la cápsula interna, aun cuando sea agudo y se acompañe de parálisis motora grave, produce poca anomalía electroencefalográfica. Lo anterior no debe sorprender si se considera que aun lesiones infimas pueden provocar un déficit neurrológico acentuado, y que el EEG, por su parte, rara vez muestra anomalías en casos de lesiones pequeñas y profundas.

El EEG puede ayudar también al diagnóstico en pacientes con lesiones cerebrovasculares del tronco cerebral. El infarto o la hemorragia de la parte central del tronco cerebral se asocia con un EEG que contiene predominantemente actividad alfa parecida a la del EEG de vigilia normal (coma-alfa). Por el tipo y localización de la lesión, este patrón electroencefalográfico registrado en un paciente comatoso generalmente indica un pronóstico desfavorable para la recuperación clínica.

---

<sup>16</sup> Hinchazón producida por infiltración de líquido en el tejido celular.

**Traumatismo<sup>17</sup> y Anoxia<sup>18</sup>.** El EEG es generalmente anormal inmediatamente después de traumatismo craneal. Las anomalías pueden ser focales o difusas y variables en magnitud. En casos de traumatismos ligeros aparecen con más frecuencia en niños que en adultos.

Las anomalías ligeras no focales registradas después de un traumatismo craneal no se relacionan necesariamente a éste, y los complejos punta-onda difusos bien formados generalmente son independientes. La interpretación de muchos tipos de anomalías del electroencefalograma es difícil porque generalmente hace falta un trazo previo al traumatismo para la comparación.

El EEG es de utilidad en estudios secuenciales. La relación entre las anomalías electroencefalográficas y el traumatismo craneal se establece con más seguridad si las alteraciones disminuyen gradualmente junto con la recuperación clínica. Sin embargo, a menudo la mejoría electroencefalográfica y clínica no coinciden, ya que pueden aparecer largo tiempo después del traumatismo craneal, anomalías tales como ondas en punta focales que indican la formación de una cicatriz crónica. Una anomalía focal grave en aumento después de traumatismo craneal indica una lesión expansiva, como hematoma intracraneal.

Aunque el EEG rara vez es confiable para fines de pronóstico después de un traumatismo craneal, frecuentemente ayuda cuando el paciente comatoso por traumatismo craneal o por algunas otras enfermedades, donde el EEG muestra patrones de sueño normal (coma de ondas fusiformes) y cambios clínicos del sueño en registros nocturnos prolongados. Estos patrones de sueño y los cambios clínicos generalmente indican un pronóstico favorable para la recuperación clínica.

**Enfermedades Inflamatorias.** La mayoría de las enfermedades inflamatorias del sistema nervioso central que afectan el EEG producen predominantemente actividad lenta difusa no específica, independientemente del tipo de agente causal. La demostración de una alteración significativa en la actividad eléctrica cerebral puede ser importante, ya que las manifestaciones mentales son predominantes y, algunas veces, en etapas tempranas o en la fase crónica puede haber dudas para decidir si los síntomas son emocionales u orgánicos. Cuando la anomalía es focal debe sospecharse un absceso.

**Enfermedades Tóxicas, Metabólicas y Degenerativas.** Las anomalías electroencefalográficas de la mayoría de las enfermedades tóxicas, metabólicas y degenerativas consisten en ondas lentas con grados diferentes de intensidad. Generalmente estos cambios no tienen características distintivas. Ya que muchas de estas enfermedades producen disfunción cerebral, el EEG puede ser útil para descubrir la enfermedad orgánica cuando el origen de los síntomas es dudoso desde el punto de vista clínico.

---

<sup>17</sup> Trastorno causado por una herida.

<sup>18</sup> Ausencia de oxígeno en la sangre.

---

**Trastornos Convulsivos.** Las crisis son solamente síntomas y pueden ser causadas por muchas variadas alteraciones cuyo origen, con frecuencia, no es determinado. Sin embargo, una de las aplicaciones más importantes del EEG es en el diagnóstico de ataques epilépticos.

Existen patrones electroencefalográficos que se denominan epileptiformes, ya que tienen morfología distintiva y aparecen en alta proporción en EEG de pacientes con crisis epilépticas y rara vez en registros de personas asintomáticas. Son ejemplos de estos patrones algunos tipos de ondas esporádicas en punta, ondas agudas y complejos punta-onda lenta. Estos patrones electroencefalográficos permiten predecir, con alta probabilidad, una alteración convulsiva clínica. Sin embargo, no todos los patrones de ondas en punta o punta-onda tienen implicaciones similares. El significado de algunas descargas electroencefalográficas depende de la edad del paciente al tiempo del registro. Hay tipos de ondas en punta que pueden aparecer transitoriamente en la niñez sin manifestaciones clínicas aparentes. Sin embargo, una actividad parecida en el trazo de un adulto puede tener una mayor correlación con crisis clínicas.

El EEG ayuda para determinar si las crisis se originan en un área limitada del cerebro o afectan totalmente el cerebro desde el principio. Lo anterior es importante para distinguir las diversas causas posibles de estos dos tipos básicos de epilepsia, porque las manifestaciones clínicas, en ambas, pueden ser similares o idénticas. El EEG rara vez permanece inalterado durante una crisis epiléptica verdadera.

**Muerte Cerebral.** El cada vez más socorrido uso de los trasplantes de órganos ha conducido a la necesidad de investigar y utilizar signos de certidumbre de la abolición total e irreversible de las funciones cerebrales del donador con miras a tomar el órgano lo más pronto posible. Varios criterios han sido tomados en cuenta: la pérdida de toda vida de relación; la arreflexia y la atonía muscular totales; el paro de la respiración espontánea; el derrumbamiento de la presión arterial a partir del momento en que no es posible sostenerla artificialmente; y el trazado electroencefalográfico lineal absoluto (aun bajo estimulación) obtenido con técnicas definidas. Una importancia capital es atribuida a este último criterio.

**Procesamiento Avanzado.** Potenciales corticales provocados resultan de aplicar estímulos al sistema nervioso del cuerpo y están localizados en un área particular del cerebro. Estímulos típicos son un destello de luz o un sonido agudo. Esta técnica se usa para comparar los trazos de personas asintomáticas con los de aquellas que tiene alguna patología evidente y proporcionar un nuevo elemento de estudio.

Otra técnica avanzada de procesamiento EEG involucra el cálculo del espectro de frecuencia de señales EEG en intervalos de tiempo discreto y dibujar los resultados en un trazo de frecuencia-tiempo-amplitud conocido como arreglo espectral comprimido (*CSA compressed spectral array*). Al presente el CSA tiene aplicación en diagnóstico y monitoreo de anestesia profunda durante cirugía.

Otras dos técnicas usadas en el estudio clínico del sueño son: La polisomnografía, que es un complejo registro continuo multicanal de varias funciones fisiológicas durante un periodo extendido de sueño, usado para fines de diagnóstico. Un polisomnograma revela patrones de sueño del paciente y cantidad normal de sueño, permitiendo comparación con aquellos de otros

pacientes, y con otros patrones de insomnio. Y la cronobiología humana, que provee señales EEG para diagnóstico no psiquiátrico y tratamiento de desórdenes del sueño. Esta provee un ciclo de dormido-despierto sobre un periodo continuo de tiempo.

## 1.2 Conceptos Básicos de la Electroencefalografía

En este capítulo se destaca la aplicación de la electroencefalografía como una práctica diagnóstica de utilidad indubitable. La utilización adecuada de todos los conocimientos que se han presentado conducirá con mucha seguridad a la obtención de información útil al experto acerca de aspectos relevantes asociados a la fisiología y psicología del paciente en estudio. En el siguiente apartado se presentan los conceptos básicos que fundamentan la realización de un sistema de procesamiento de señales EEG.

### 1.2.1 Electroodos

Para medir y registrar potenciales y por consiguiente corrientes en el cuerpo, es necesario establecer alguna interfaz entre el cuerpo y los aparatos de medición electrónicos. Esta función de interfaz es realizada por los electrodos. La primera impresión puede ser que ésta es una función simple de realizar y que los electrodos deben ser relativamente cosas muy simples. Pero cuando se considera el problema con mayor detalle se tiene que los electrodos realizan en sí una función de transducción, pues la corriente es transportada en el cuerpo humano mediante iones, en tanto que los aparatos necesitan medir la corriente en electrones. De esta forma, los electrodos son transductores que cambian una corriente iónica en una corriente electrónica. Como es el funcionamiento de los electrodos, las diversas formas que pueden tener los mismos y sus características se verán a continuación.

#### *La Interfaz Electrodo-Electrólito*

Para explorar las características de la interfaz electrodo-electrólito, se considerará lo que pasa cuando se sumerge una pieza de metal (electrodo) en una solución conductora (electrólito) que contiene cationes<sup>19</sup> y aniones<sup>20</sup> en igual cantidad para mantener su equilibrio. Cuando el electrodo entra en contacto con el electrólito, inmediatamente se desencadena una de las dos reacciones mostradas en las figuras 1.13 (a) ó 1.13 (b), dependiendo de la naturaleza misma del material con que esté hecho el electrodo.

---

<sup>19</sup> **Catión.** Ion cargado positivamente.

<sup>20</sup> **Anión.** Ion cargado negativamente.

Como se observa en la figura 1.13, la concentración de cationes y aniones se ve alterada, lo que provoca que el electrólito que rodea al electrodo se encuentre a diferente potencial que el resto de la solución. Esto provoca una reacción de reducción<sup>21</sup> u oxidación<sup>22</sup> espontánea en el metal del electrodo. Por definición, el electrodo en el cual ocurre la oxidación se llama *ánodo*; el electrodo en el cual ocurre la reducción se llama *cátodo*.

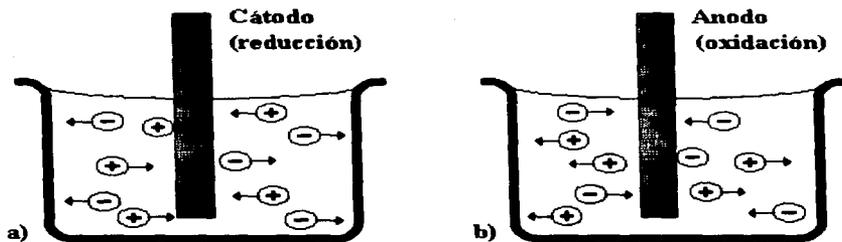


Figura 1.13 . Reacción desencadenada al introducir un electrodo dentro de un electrólito.

Es imposible medir el potencial de un solo electrodo, pero si arbitrariamente se asigna al potencial de un electrodo particular el valor de cero, puede usarse para determinar los potenciales relativos de otros electrodos. El electrodo de hidrógeno sirve como referencia para este propósito. Al potencial de un electrodo medido con referencia al electrodo estándar de hidrógeno bajo condiciones específicas de temperatura, concentración iónica, etc., se le llama *potencial estándar del electrodo* o *potencial de semirreacción*.

La tabla que se muestra a continuación lista los materiales más comúnmente utilizados como electrodos y muestra sus potenciales estándar. Por convención, se informan en la lista sólo potenciales estándar de reducción.

Cabe aclarar que los potenciales estándar se miden bajo condiciones donde no existe una corriente eléctrica entre el electrodo y el electrólito. De existir una corriente, por lo general se alteraría este potencial reportado debido a la polarización del electrodo. Esta alteración en el potencial estándar se llama *sobrepotencial*.

<sup>21</sup> **Reacción de reducción.** Reacción que implica la ganancia de electrones.

<sup>22</sup> **Reacción de oxidación.** Reacción que implica la pérdida de electrones.

<b>Metal y Semireacción</b>	<b>(V)</b>
$\text{Al} \rightarrow \text{Al}^{3+} + 3\text{e}^{-}$	-1.706
$\text{Zn} \rightarrow \text{Zn}^{2+} + 2\text{e}^{-}$	-0.763
$\text{Cr} \rightarrow \text{Cr}^{3+} + 3\text{e}^{-}$	-0.744
$\text{Fe} \rightarrow \text{Fe}^{2+} + 2\text{e}^{-}$	-0.409
$\text{Cd} \rightarrow \text{Cd}^{2+} + 2\text{e}^{-}$	-0.401
$\text{Ni} \rightarrow \text{Ni}^{2+} + 2\text{e}^{-}$	-0.230
$\text{Pb} \rightarrow \text{Pb}^{2+} + 2\text{e}^{-}$	-0.126
$\text{H}_2 \rightarrow 2\text{H}^{+} + 2\text{e}^{-}$	0.000
$\text{Ag} + \text{Cl}^{-} \rightarrow \text{AgCl} + \text{e}^{-}$	+0.233
$2\text{Hg} + 2\text{Cl}^{-} \rightarrow \text{Hg}_2\text{Cl}_2 + 2\text{e}^{-}$	+0.268
$\text{Cu} \rightarrow \text{Cu}^{2+} + 2\text{e}^{-}$	+0.340
$\text{Cu} \rightarrow \text{Cu}^{+} + \text{e}^{-}$	+0.522
$\text{Ag} \rightarrow \text{Ag}^{+} + \text{e}^{-}$	+0.799
$\text{Au} \rightarrow \text{Au}^{3+} + 3\text{e}^{-}$	+1.420
$\text{Au} \rightarrow \text{Au}^{+} + \text{e}^{-}$	+1.680

**Tabla 1.3. Potenciales estándar de reducción a 25°C.**

### *Electrodos externos*

#### *Electrodos De Placa De Metal*

Estos electrodos son unos de los más comúnmente utilizados para sensar potenciales bioeléctricos. Básicamente están formados por una placa de metal conductor que se pone en contacto con la piel. Se utiliza un gel electrolítico para establecer y mantener el contacto correcto. La figura 1.14 muestra distintas presentaciones para este electrodo. El electrodo del inciso (a) se utiliza mayormente en electrocardiogramas. El del inciso (b) es bueno tanto para electrocardiogramas, electromiogramas o electroencefalogramas. El del inciso (c) es un tipo de electrodo popular, ya que viene junto con el adhesivo y el gel integrados, de tal forma que su uso es más fácil y rápido, pues no se necesita aprender ninguna técnica especial para usarlos; por supuesto, su costo es mayor.

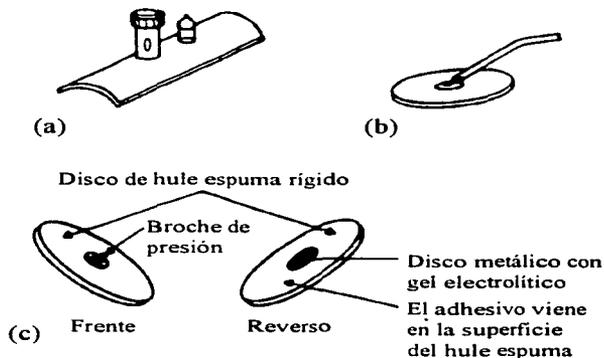


Figura 1.14. Electrodos de placa de metal.

### *Electrodos De Succión*

Es una modificación del electrodo de placa de metal que no requiere de correas o adhesivos para sujetarse a la piel (Figura 1.15). Estos electrodos frecuentemente utilizados en la electrocardiografía, están formados por un electrodo cilíndrico de metal ahuecado que hace contacto con la piel por la base, y por el otro extremo poseen una pequeña bomba de succión hecha de goma. Para instalar el electrodo se pone un poco de gel electrolítico sobre la superficie de contacto del electrodo con la piel, se oprime la bombilla y se coloca el electrodo sobre el punto deseado. La bombilla es liberada y aplica una leve succión sobre la piel de tal forma que el electrodo se mantiene firme en su lugar. Estos electrodos deben usarse sólo durante cortos periodos, pues la succión y presión ejercidas sobre la piel tienden a irritarla.

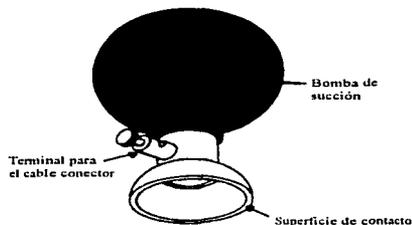


Figura 1.15. Electrodo de succión.

### Electrodos Flexibles

Los electrodos descritos hasta ahora son rígidos y planos o bien poseen una curvatura predefinida. La superficie del cuerpo, sin embargo, posee forma irregular y puede cambiar su curvatura con el movimiento. Por esto se desarrollaron los electrodos flexibles como los que se muestran en la figura 1.16. El inciso (a) muestra un electrodo formado por un pedazo de caucho relleno de carbón con forma de laminilla delgada el cual es usado en la misma forma que los electrodos de placa de metal. El inciso (b) muestra un electrodo flexible desarrollado por Neuman en 1973. Formado por una película extremadamente delgada de poliéster (de 13  $\mu\text{m}$ ) y recubierta por una capa de cloruro de plata, estos electrodos son útiles para monitorear bebés prematuros. El inciso (c) muestra una vista lateral de estos electrodos.

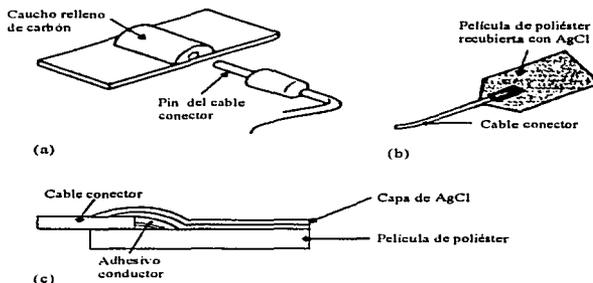


Figura 1.16. Electrodos flexibles.

### Electrodos internos

Los electrodos también pueden colocarse dentro del cuerpo para detectar los potenciales bioeléctricos. Estos pueden ser *electrodos percutáneos*, en donde el electrodo en sí atraviesa la piel desde el exterior, o bien pueden ser totalmente *electrodos internos*, en los que la conexión se hace a través de un implante que funciona como un transmisor de radio. Estos electrodos no requieren ningún gel electrolítico para mantenerse en contacto, pues en este caso, el líquido extracelular está directamente presente.

Existen diseños diferentes para los electrodos internos. Un investigador que esté estudiando un fenómeno bioeléctrico en particular utilizando electrodos internos, frecuentemente diseñará sus propios electrodos para ese fin específico. La figura 1.17 muestra los electrodos percutáneos más comúnmente utilizados, en tanto que la figura 1.18 hace lo mismo pero para los electrodos internos.

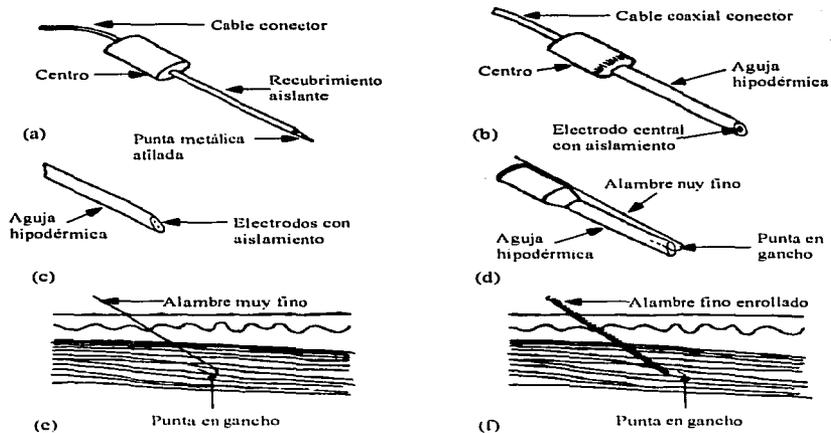


Figura 1.17. Electrodos percutáneos de aguja y de alambre.

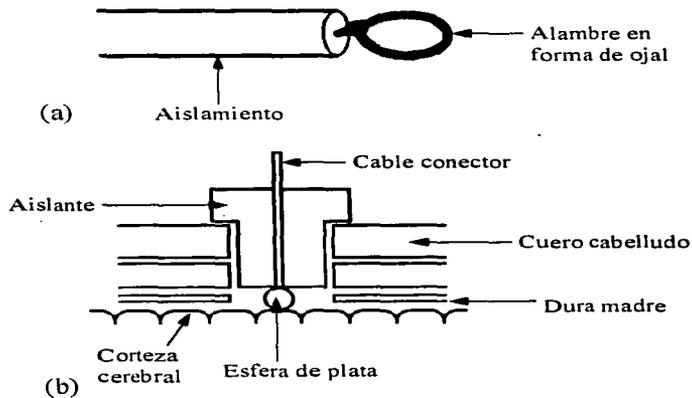


Figura 1.18. Electrodos internos.

### 1.2.2 Muestreo de Datos

Bajo ciertas condiciones una señal de tiempo continuo se puede representar y reconstruir completamente, partiendo del conocimiento de sus valores instantáneos, o *muestras*, igualmente espaciadas en el tiempo. Esta propiedad se deriva de un resultado básico que se conoce como el *teorema de muestreo*.

La importancia del teorema de muestreo reside en su papel de puente entre las señales de tiempo continuo y las señales de tiempo discreto. Como se verá más adelante, la habilidad para representar completamente una señal de tiempo continuo mediante una secuencia de muestras instantáneas proporciona un mecanismo para representar una señal de tiempo continuo mediante una señal de tiempo discreto. En varios contextos, el procesamiento de señales de tiempo discreto es más flexible y a menudo preferible al procesamiento de señales de tiempo continuo, debido a la creciente disponibilidad de sistemas digitales y de tiempo discreto de bajo costo, ligeros, programables y con facilidad de reproducir la señal. Esta tecnología también ofrece la posibilidad de explotar el concepto del muestreo para convertir una señal de tiempo continuo a una señal de tiempo discreto, y después de procesar la señal de tiempo discreto empleando un sistema de tiempo discreto, se puede convertir de nuevo a tiempo continuo.

El *muestreador* en un sistema de datos discretos o sistema digital, es un dispositivo que convierte una señal analógica (señal continua) en un tren de pulsos de amplitud modulada o una señal digital. Un dispositivo *retenedor* simplemente “sostiene” el valor del pulso o señal digital durante un tiempo de duración preestablecida. En la mayoría de las operaciones prácticas, el muestreo, y la retención se efectúan por una sola unidad y comercialmente el dispositivo se conoce como muestreador y retenedor, o S/H (*Sampler and Holder*).

Los dispositivos S/H son ampliamente usados en los sistemas digitales. Una de las principales aplicaciones del muestreador retenedor es la de sostener las señales de movimiento rápido durante todo tipo de operaciones de conversión. Otra aplicación común del S/H es para el almacenamiento de las salidas de un multiplexor mientras la señal está siendo convertida. La detección de picos de una señal es otra de sus aplicaciones y posteriormente se mostrará que el S/H se usa en el proceso de conversión A/D y D/A.

La forma más simple de un S/H se muestra en la figura 1.19, la apertura y cierre del interruptor (o muestreador) está controlada por un comando de muestreo. Cuando el interruptor se cierra, la salida del S/H muestrea y “sigue” a la señal de entrada  $e_i(t)$ .

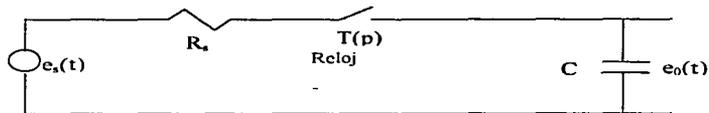
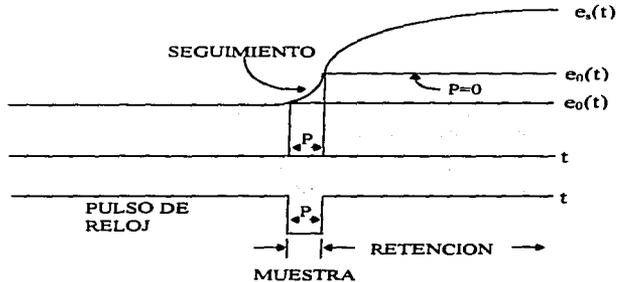


Figura 1.19. Circuito sencillo ilustrando el principio del muestreador-retenedor.

Cuando el interruptor se abre, la salida se mantiene al voltaje al que se cargó el capacitor. En la figura 1.20 se muestran las señales de entrada y salida típicas del S/H simple mostrado en la figura 1.19.

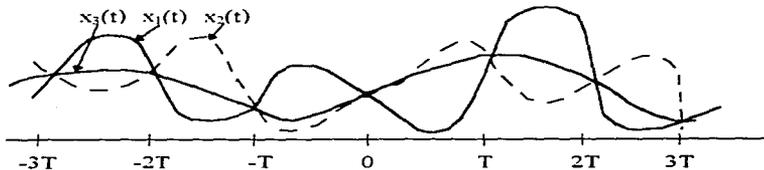
El intervalo de tiempo durante el cual permanece cerrado el muestreador es llamado duración de muestreo  $p$ . Prácticamente la resistencia  $R_s$  no es cero y el capacitor se cargará hasta el valor de la señal de entrada muestreada con una constante de tiempo  $R_s C$ . Aún más, la operación del muestreador no es instantánea ya que requiere de cierto tiempo para responder a los comandos de muestreo. En la práctica, un muestreador y retenedor tiene muchas imperfecciones y errores, por lo que la salida del dispositivo puede desviarse considerablemente de la forma ideal mostrada en la figura 1.20.



**Figura 1.20. Señales simplificadas de muestreo y retención.**

### El teorema de muestreo

En general, no es de esperarse que en la ausencia de cualquier condición o información adicional una señal se especifique unívocamente por una secuencia de muestras igualmente espaciadas. Por ejemplo, en la figura 1.21 se muestran tres diferentes señales de tiempo continuo, que tienen valores idénticos en múltiplos enteros de  $T$ .  $x_1(kT) = x_2(kT) = x_3(kT)$



**Figura 1.21. Tres señales de tiempo continuo con valores idénticos en múltiplos enteros de  $T$ .**

En general hay una cantidad infinita de señales que pueden generar un conjunto dado de muestras. Sin embargo, como se vera más adelante, si una señal es de banda limitada y si las muestras son tomadas lo suficientemente cercanas unas de las otras, en relación con la frecuencia más alta presente en la señal, entonces las muestras especifican *unívocamente* a la señal y se puede por tanto reconstruir perfectamente dicha señal. Específicamente, si una señal  $x(t)$  de banda limitada se modula en amplitud con un tren de pulsos periódicos, lo que corresponde a extraer segmentos de tiempo igualmente espaciados, se puede recuperar exactamente mediante un filtrado pasa bajas si la frecuencia fundamental del tren de pulsos modulador es mayor que el doble de la frecuencia más alta presente en  $x(t)$ . Además, la habilidad para recuperar  $x(t)$  es independiente de la duración en tiempo de los pulsos individuales. Entonces, como se sugiere en la figura 1.22 y en la figura 1.23, conforme esta duración se hace arbitrariamente más pequeña, la modulación en amplitud de los pulsos está, en efecto, representando a la señal  $x(t)$  mediante muestras instantáneas espaciadas igualmente en el tiempo. En el sistema de modulación de amplitud de pulsos de la figura 1.23, se ha escalado la amplitud del tren de pulsos para que sea inversamente proporcional al ancho del pulso  $\Delta$ . En cualquier sistema práctico de modulación de amplitud de pulsos, es particularmente importante mantener constante una potencia promedio en el tiempo para la señal modulada cuando se hace que  $\Delta$  sea pequeño. Como se ilustra en la figura 1.23, conforme  $\Delta$  se aproxima a cero, la señal modulada se convierte en un tren de impulsos en el que los impulsos individuales tienen valores correspondientes a muestras instantáneas de  $x(t)$  en valores de tiempo espaciados  $T$  segundos.

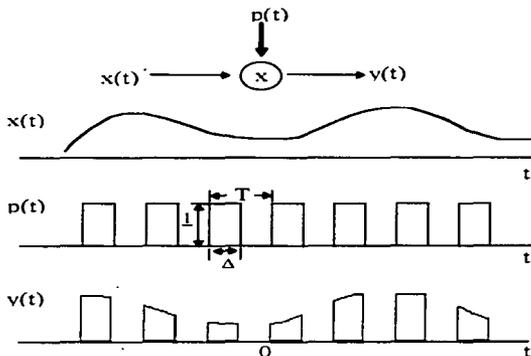


Figura 1.22. Modulación por amplitud de pulsos. Conforme  $\Delta \rightarrow 0$ ,  $p(t)$  se aproxima a un tren de impulsos

### Muestreo con tren de Impulsos

De una manera idéntica a la empleada para analizar el caso más general de la modulación de amplitud de pulsos, Teniendo en cuenta el caso específico del muestreo con un

tren de impulsos como el mostrado en la figura 1.22. El tren de impulsos  $p(t)$  se conoce como la *función de muestreo*, el periodo  $T$  como el *periodo de muestreo* y la frecuencia fundamental de  $p(t)$ ,  $\omega_s = 2\pi/T$ , como la *frecuencia de muestreo*. En el dominio del tiempo se tiene:

$$x_p(t) = x(t)p(t) \quad (1.1)$$

donde

$$p(t) = \sum_{n=-\infty}^{+\infty} \delta(t - nT) \quad (1.2)$$

$x_p(t)$  es un tren de impulsos cuyas amplitudes son iguales a las muestras de  $x(t)$  en intervalos espaciados por  $T$ ; esto es;

$$x_p(t) = \sum_{n=-\infty}^{+\infty} x(nT)\delta(t - nT) \quad (1.3)$$

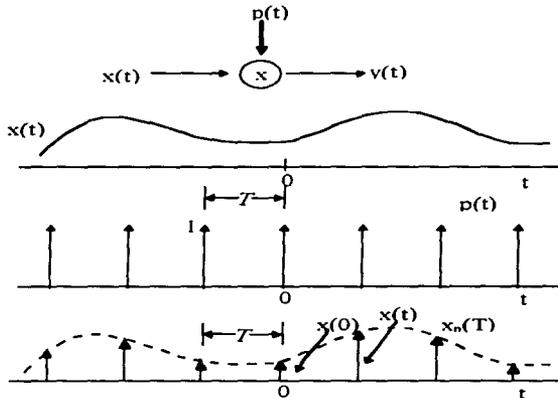


Figura 1.23. Modulación de amplitud de pulsos con un tren de impulsos.

Como la multiplicación de una señal por otra es considerado como el empleo de una señal para escalar o modular la amplitud de la otra, la multiplicación de las dos señales se refiere frecuentemente como *modulación en amplitud*. Por esta razón utilizando la propiedad de modulación:

$$X_p(\omega) = \frac{1}{2\pi} [X(\omega) * P(\omega)] \quad (1.4)$$

y utilizando transformada de Fourier:

$$P(\omega) = \frac{2\pi}{T} \sum_{k=-\infty}^{+\infty} \delta(\omega - k\omega_s) \quad (1.5)$$

de manera que:

$$X_p(\omega) = \frac{1}{T} \sum_{k=-\infty}^{+\infty} X(\omega - k\omega_s) \quad (1.6)$$

$$\omega_p(\omega) = \frac{1}{T} \sum_{k=-\infty}^{+\infty} X(\omega - k\omega_s) \quad (1.7)$$

Esto es,  $X_p(\omega)$  es una función periódica en el dominio de la frecuencia que consiste de una suma de réplicas de  $X(\omega)$  desplazadas y escaladas por  $1/T$ , tal como se ilustra en la figura 1.24.

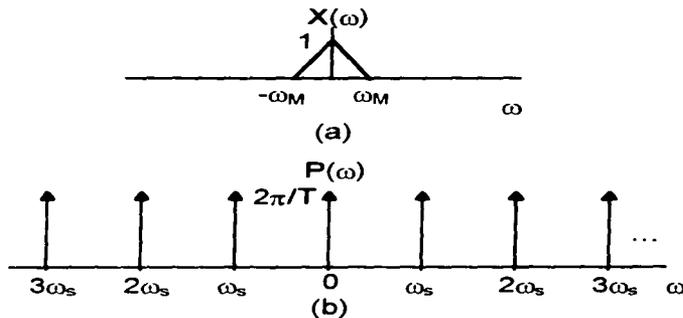
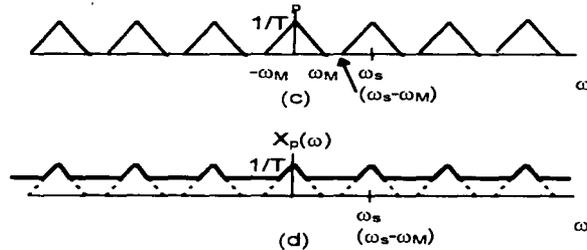


Figura 1.24. Efecto en el dominio de la frecuencia del muestreo en el dominio del tiempo.  
 (a) espectro de la señal original; (b) espectro de la función de muestreo.  
 (Continúa)



**Figura 1.24. Efecto en el dominio de la frecuencia del muestreo en el dominio del tiempo.**

(c) espectro de la señal muestreada con  $\omega_s > 2\omega_M$ ; (d) espectro de la señal muestreada con  $\omega_s < 2\omega_M$ .

En la figura 1.24 (c),  $\omega_M < (\omega_s - \omega_M)$ , lo que equivale a  $\omega_s > 2\omega_M$ , por lo que no hay traslape entre las réplicas desplazadas de  $X(\omega)$ , mientras en la figura 1.24 (d) en donde  $\omega_s < 2\omega_M$ , sí hay traslape. Para el caso ilustrado en la figura 1.24 (c),  $X(\omega)$  se reproduce fielmente en múltiplos enteros de la frecuencia de muestreo. Por lo tanto, si  $\omega_s > 2\omega_M$ ,  $x(t)$  se puede recuperar exactamente a partir de  $X_p(t)$  por medio de un filtro pasa bajas con ganancia  $T$  y una frecuencia de corte mayor que  $\omega_M$  y menor que  $\omega_s - \omega_M$ . Este resultado básico conocido como el *teorema de muestreo* se expresa como sigue:

#### Teorema del muestreo:

Sea  $x(t)$  una señal de banda limitada con  $X(\omega) = 0$  para  $|\omega| > \omega_M$ . Entonces  $x(t)$  está determinada unívocamente por sus muestras  $x(nT)$ ,  $n = 0, \pm 1, \pm 2, \dots$  si

$$\omega_s > 2\omega_M \quad (1.8)$$

sabiendo que  $\omega_s$  está definida por:

$$\omega_s = \frac{2\pi}{T} \quad (1.9)$$

Dadas estas muestras, se puede reconstruir  $x(t)$  generando un tren de impulsos periódicos en el que los impulsos sucesivos tienen amplitudes que corresponden a valores de muestras sucesivas. Este tren de impulsos es entonces procesado a través de un filtro pasa bajas ideal con ganancia  $T$  y cuya frecuencia de corte es mayor que  $\omega_M$  y menor que  $(\omega_s - \omega_M)$ . La salida resultante será exactamente igual a  $x(t)$ .

La frecuencia de muestreo  $\omega_s$ , también se conoce como la *frecuencia de Nyquist*. La frecuencia  $2\omega_N$ , que bajo el teorema de muestreo, debe ser excedida por la frecuencia de muestreo, se denomina por lo común como la *velocidad de Nyquist*.

### 1.2.3 Amplificación de Señales

La mayoría de las señales bioeléctricas son de amplitudes pequeñas, y las señales EEG, como se mencionó anteriormente, no son la excepción y por ende requieren amplificación. Los amplificadores también son utilizados como interfaz a los sensores. En adición a simple amplificación, los amplificadores pueden modificar la señal para producir filtrado de frecuencias y efectos no lineales.

Un amplificador diferencial, abreviado *Amp Dif*, es un amplificador electrónico en el cual el voltaje de salida es proporcional a la diferencia entre los dos voltajes de entrada. Los amplificadores diferenciales son particularmente útiles para medir biopotenciales, ya que muchos potenciales biológicos de significado en el diagnóstico clínico y médico consisten en las diferencias de voltaje en dos sitios del cuerpo. El electrocardiograma, por ejemplo, es medido como la diferencia de potenciales de superficie entre dos extremidades. El electromiograma registra la diferencia entre dos potenciales medidos en un músculo. El electroencefalograma, por otro lado, es la diferencia de potencial registrada entre dos sitios del cuero cabelludo. El amplificador diferencial es ideal para medir estas diferencias de potencial y es por ende utilizado en diversas instrumentaciones médicas.

Los amplificadores diferenciales son ampliamente usados también dados los avances en la fabricación de circuitos integrados de semiconductores. En 1970, el número de elementos activos que podía ser fabricado en un circuito integrado era de alrededor de mil. Para 1987 este número se había incrementado a más de un millón. El incremento en eficiencia ha guiado también a la reducción del costo del amplificador diferencial, de alrededor de trescientos cincuenta dólares en los años cincuenta, cuando los tubos de vacío (bulbos) eran usados, a alrededor de treinta y cinco centavos de dólar en 1987. Además de la reducción en el costo de mil a uno, ha habido una reducción de tamaño igual de dramática. Más aún, amplificadores diferenciales en circuitos integrados tienen un consumo tan bajo de energía que es posible tener instrumentos médicos alimentados por baterías, como marcapasos. Estas ventajas de bajo costo, bajo consumo de energía, y pequeño tamaño han estimulado el uso de amplificadores diferenciales en equipo médico, y por ende ha contribuido significativamente al incremento del número de instrumentos médicos que se utilizan hoy en día.

El símbolo del amplificador diferencial se muestra en la Figura 1.25, y su definición matemática es:

$$V_{OUT} = A_d(V_2 - V_1) \quad (1.10)$$

donde  $V_1$  y  $V_2$  son caídas de voltaje de entrada referenciadas a tierra.  $V_{OUT}$  es el voltaje de salida, referenciado a tierra, y  $A_d$  es la ganancia de voltaje diferencial.

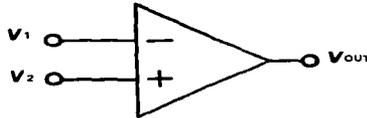


Figura 1.25. Símbolo del amplificador diferencial.

Un amplificador diferencial es usualmente fabricado como un circuito integrado. Sin embargo, para ilustrar el concepto y el entendimiento del amplificador diferencial, se considera el circuito de transistor mostrado en la Figura 1.26. Este amplificador es un circuito balanceado con transistores  $Q_1$  y  $Q_2$ , y las resistencias de colector  $R_L$  idénticas. Usa dos fuentes de suministro con voltajes  $V_{BB}$  y  $-V_{BB}$ , respectivamente. El voltaje de salida es medido como la caída de voltaje más-a-menos de uno de los colectores a tierra. Este hecho ( $V_{OUT}$  referenciado a tierra), es una ventaja, dado que el amplificador puede ser usado como entrada a etapas sucesivas de amplificador para subsecuente proceso.

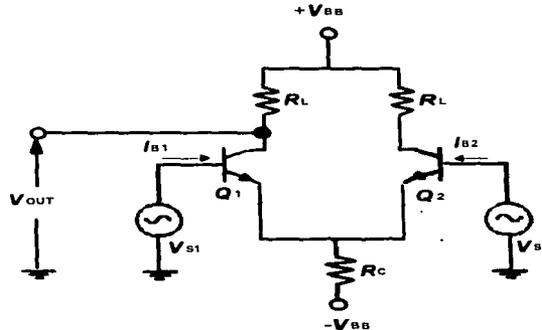


Figura 1.26 Amplificador diferencial de circuito de transistor.

Una característica particularmente conveniente del amplificador diferencial es que éste tiende a eliminar la interferencia de *voltaje modo-común*. Los voltajes de modo-común son aquellos que tienen el mismo valor en cada terminal de entrada del amplificador diferencial. Esto significa que si  $V_1$  y  $V_2$  son voltajes modo-común, entonces  $V_1 = V_2$  y el voltaje de salida debido a éstos es cero, por la ecuación (1.10). Esto es, el voltaje de salida debido a la interferencia de modo común tiende a cero en un amplificador diferencial. Este hecho es de notable importancia pues es el fundamento que permite reducir la interferencia en los

amplificadores biopotenciales al momento de hacer las mediciones, ya que en condiciones normales se pueden esperar valores de capacitancia parásita entre uno y diez picofaradios con la que estará cargado un paciente debido a las líneas de energía [Asto90]. Así, la interferencia de las líneas de energía, por la capacitancia parásita con que cargan al paciente, excede por mucho la magnitud de la señal medida, la cual es del orden de microvoltios para el caso del EEG. Esta mala noticia es a menudo cancelada por el hecho de que la señal de interferencia aparece igualmente intensa en las dos terminales de entrada del amplificador diferencial, y es por eso llamada señal de modo-común.

Los amplificadores diferenciales pueden ser producidos en circuitos integrados a bajo costo. Estos circuitos integrados a menudo tienen una construcción MOS (*metal-oxide-semiconductor*), en el cual el metal de la compuerta de entrada está separada del semiconductor por vidrio ( $\text{SiO}_2$ ), como aislante. Esto hace posible producir una impedancia de entrada alta en amplificadores hechos de esta forma. Más aún, la ganancia puede ser muy alta, todo a bajo costo cuando los requerimientos de potencia son bajos y el rango de frecuencia está dentro del usado en amplificadores biopotenciales. Por ello se vuelve económico y práctico definir un amplificador diferencial ideal como un componente para análisis de circuitos. Esto es en adición a los otros componentes de circuitos, resistores,  $R$ , capacitores,  $C$ , e inductores,  $L$ . Todos ellos definidos en el sentido ideal. La razón de que estas definiciones ideales son prácticas es que es posible diseñar económicamente dispositivos que realmente se aproximen a este desempeño ideal.

Un amplificador diferencial ideal es definido como aquel para el cual (1) la impedancia de entrada para cada entrada es infinita, (2) la ganancia es infinita ( $A_d$  tiende a infinito), y (3) el voltaje de salida está dado por la ecuación (1.10)

Las implicaciones inmediatas de asumir esto idealmente es que  $V_2$  es aproximadamente igual a  $V_1$ . También, dado que la impedancia de entrada es infinita en ambos puertos, el voltaje de salida  $V_{OUT}$  permanece constante sin importar la carga a la que sea conectado el puerto de salida. Esta es exactamente la propiedad de una fuente de voltaje ideal, y tales fuentes tienen una impedancia de salida cero. Por lo anterior se concluye que un amplificador diferencial ideal tiene impedancia de salida cero.

Amplificadores que usan circuitos de amplificadores diferenciales ideales como componentes, junto con otros elementos  $R$  y  $C$ , son llamados amplificadores operacionales, o *amp op*. El término "operacional" es usado porque estos circuitos ejecutan operaciones matemáticas en el voltaje de entrada. Tales operaciones pueden incluir multiplicación o integración.

Como se mencionó, los amplificadores se requieren para incrementar la fuerza de la señal manteniendo alta fidelidad. Los amplificadores son una parte importante de los sistemas modernos de instrumentación para medir biopotenciales. Tales mediciones involucran voltajes de bajo nivel y a menudo de fuentes de impedancia alta. Aquellos amplificadores que han sido diseñados específicamente para procesar potenciales biológicos, son conocidos como amplificadores biopotenciales.

La función esencial de un amplificador biopotencial es tomar una señal débil de origen biológico e incrementar su amplitud para que pueda ser procesada, grabada o desplegada. Usualmente tales amplificadores son en forma de amplificadores de voltaje, porque son capaces de incrementar el nivel de voltaje de una señal. Sin embargo, los amplificadores de voltaje también sirven para incrementar el nivel de potencia, por ello pueden ser considerados también como amplificadores de potencia. Todos los amplificadores biopotenciales deben tener una alta impedancia de entrada, tal que provean una mínima carga a la señal siendo medida. Las características de los electrodos biopotenciales pueden ser afectadas por la carga eléctrica que ven, la cual, combinada con una excesiva carga pueden resultar en distorsión de la señal. Los efectos de la carga pueden ser minimizados haciendo la impedancia de entrada del amplificador tan alta como sea posible, reduciendo esta distorsión. Amplificadores biopotenciales modernos tienen impedancia de entrada de al menos diez megohms.

El circuito de entrada de un amplificador biopotencial también debe de proveer protección al organismo en estudio. Cualquier corriente o potencial que aparezca en las terminales de entrada del amplificador es capaz de afectar los potenciales biológicos siendo medidos. El amplificador debe tener circuitos de aislamiento y protección tal que las corrientes a través del circuito del electrodo se mantengan a niveles seguros y cualquier artefacto<sup>23</sup> generado por dicha corriente sea minimizado. La manera en que se obtiene una relación señal-a-ruido óptima es limitando el ancho de banda del amplificador, de tal modo que sea solo lo suficientemente grande para procesar la señal adecuadamente. Los amplificadores biopotenciales deben de tener amplias ganancias, del orden de mil o más.

Finalmente los amplificadores biopotenciales que son usados en aplicaciones médicas y de laboratorio deben hacer posible una calibración rápida.

#### 1.2.4 Filtros Analógicos

Existen muchos tipos de instrumentos médicos en los que es necesario seleccionar componentes de frecuencia de la señal de entrada. En encefalografía, varios estados del cerebro tales como estado de alerta, estado de sueño ligero y profundo, producen bandas distintivas de frecuencia. En un EEG, las ondas *teta* en la banda de frecuencia de 4 a 8 Hz. indican dormido, mientras que las señales *beta* en la banda de frecuencia de 13 a 22 Hz. indican un estado alto de interés o alerta.

Los filtros pueden ser utilizados para dirigir esas bandas de frecuencia hacia diferentes canales para facilitar el procesamiento de la señal y el diagnóstico. En este caso, el filtrado es esencial para la función diagnóstica del instrumento. En caso todo monitoreo, el filtrado es útil en la reducción del ruido y frecuentemente en la reducción de los efectos de la señal de interferencia de 60 Hz. En general los filtros son utilizados ya sea para seleccionar o rechazar frecuencias.

---

<sup>23</sup> Artefacto: Toda diferencia de potencial de origen extracerebral registrada en el EEG.

Un filtro ideal es aquel que deja pasar la señal deseada sin distorsión de amplitud o fase, y rechaza cualquier señal innecesaria o parásita.

La forma en la que la función del filtro varía con la frecuencia es llamada respuesta en frecuencia. Esto es representado matemáticamente por medio de la *función de transferencia  $H$*  del filtro, o gráficamente por medio de las *gráficas de frecuencia*. Esta frecuencia se divide a su vez en *respuesta en magnitud* y *respuesta en fase*. La respuesta en magnitud, denotada como  $|H|$ , muestra la cantidad de amplificación o atenuación que una señal sinusoidal experimenta cuando pasa a través del filtro; La respuesta de fase, denotada como  $\angle H$ , muestra la cantidad del cambio de la fase experimentada por la misma señal. En general, ambas respuestas, la magnitud y la fase son dependientes de la frecuencia, aunque en un caso particular una de estas puede ser constante.

### *Respuestas Comunes de Frecuencia*

En lo que se refiere a respuesta en magnitud, los filtros se clasifican en paso-bajo, paso-alto, paso-banda, y supresores de banda. Las respuestas en magnitud ideales a entradas específicas se presentan en la figura 1.27. Las características de respuesta de los filtros se definen a continuación:

1. La respuesta paso-bajo se caracteriza por una frecuencia  $f_0$  llamada *frecuencia de corte*, en la que  $|H| = 1$  para  $f < f_0$ , y  $|H| = 0$  para  $f > f_0$ , indicando que las entradas sinusoidales con frecuencia menor que  $f_0$  atraviesan el filtro sin cambios de amplitud, mientras que aquellas con frecuencia menor a  $f_0$  se someten a una atenuación completa. Una aplicación común de los filtros paso-bajo es la eliminación del ruido de alta frecuencia de una señal.
2. La respuesta paso-alto es complementaria a la respuesta paso-bajo. Las entradas sinusoidales con frecuencias mayores que la frecuencia  $f_0$  de corte pasan por el filtro sin cambios de amplitud, mientras que aquellas con frecuencia menor que  $f_0$  muestran una atenuación completa.
3. La respuesta paso-banda se caracteriza por una banda de frecuencia  $f_L < f < f_H$  tal que las entradas sinusoidales con frecuencia dentro de la banda pasan sin cambios, mientras que aquellas con frecuencia fuera de la banda son atenuadas. La diferencia  $f_H - f_L$  es conocida como *ancho de banda* del filtro, y el punto del espectro de frecuencia en que la banda se centra es conocido como *frecuencia central*. un ejemplo familiar de un filtro paso-banda es el circuito de sintonización de un radio AM/FM, el que permite al usuario elegir una estación en particular y bloquear todas las demás.
4. La respuesta del filtro supresor de banda es complementaria a la respuesta paso-banda, ya que atenúa los componentes de frecuencia de la banda  $f_L < f < f_H$ , mientras que deja pasar todas las demás. Cuando la banda de rechazo es lo suficientemente estrecha, la respuesta del filtro se conoce como *respuesta de filtro*

en V (Notch). Una aplicación común de los filtros en V es la eliminación del componente parásito de 60 Hz.

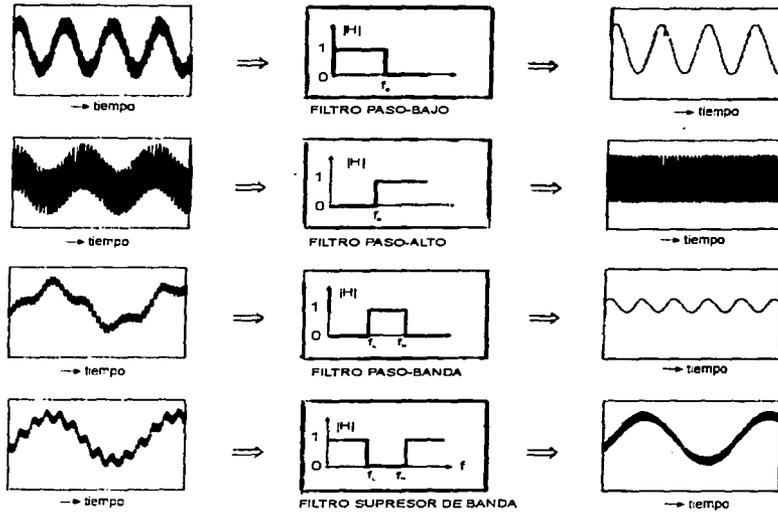


Figura 1.27. Respuestas ideales en el dominio del tiempo de los cuatro tipos de filtros más comunes.

La figura 1.27 ilustra el efecto de varios tipos de filtro en el dominio del tiempo. La señal que se muestra a la izquierda es la forma de onda de entrada como una función del tiempo; a la derecha está la forma de onda que emerge del filtro. Sin embargo, un filtro puede ser mejor entendido en términos de su efecto en el espectro de la señal de entrada, esto es, en el dominio de la frecuencia. Este punto de vista se ilustra en la figura 1.28.

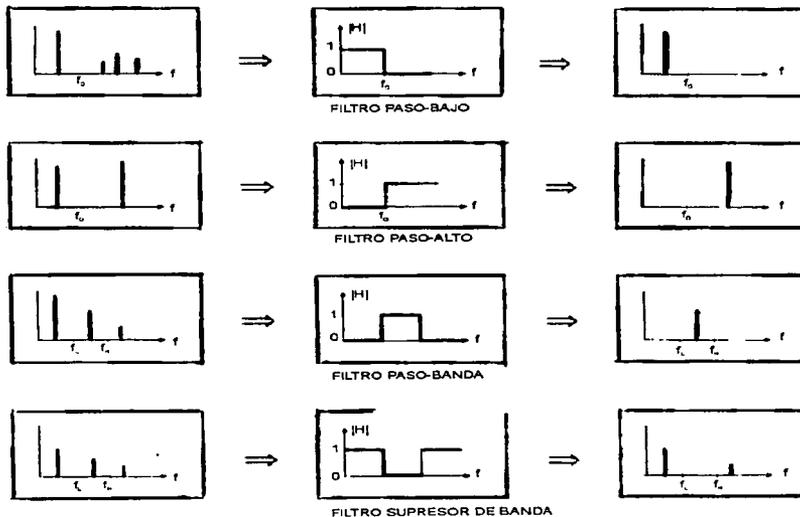


Figura 1.28. Respuestas ideales en el dominio de la frecuencia de los cuatro tipos de filtros más comunes.

Por ejemplo, el diagrama superior izquierdo de la figura 1.28 muestra el espectro correspondiente a la forma de onda de la figura 1.27. Esta forma de onda consiste de la suma de cuatro componentes sinusoidales, cada uno con su amplitud particular, frecuencia y fase. En la representación espectral de la figura 1.28, esos componentes son señalados por cuatro segmentos cuya altura coincide con la magnitud y cuya abscisa coincide con la frecuencia del componente. Después de atravesar el filtro paso-bajo, el primer componente, cuya abscisa es menor que la frecuencia de corte  $f_0$ , emerge sin cambio ya que el filtro presenta ganancia unitaria en esa frecuencia, mientras que los restantes componentes son eliminados debido a que se encuentran en la región del espectro de frecuencia donde el filtro presenta ganancia cero. Los otros tipos de filtros producen resultados correspondientes.

La teoría de los filtros es un tema vasto por sí mismo, y los amplificadores operacionales juegan un importante papel en el desarrollo de esta disciplina, particularmente en la teoría de los filtros activos.

## Filtros Activos

Los filtros pueden construirse exclusivamente con resistencias, inductores y capacitores (filtros RLC), que son componentes pasivos. Sin embargo, después de la aparición del concepto de realimentación se comprendió que al incorporar un amplificador en el circuito de un filtro se podía lograr cualquier respuesta posible, sin el uso de inductores. En otras palabras, amplificadores más resistores más capacitores pueden hacer lo que inductores más resistores más capacitores. El cómo el amplificador desplaza el uso de inductores en el diseño de filtros no es tema de este trabajo, pero sin embargo es importante hacer notar que el amplificador puede tomar energía de su fuente de poder e inyectarla al circuito circundante, previendo la pérdida de energía de los elementos resistivos. Los inductores y capacitores son dispositivos no disipadores que pueden almacenar energía durante parte del ciclo y liberarla posteriormente. Un amplificador, respaldado por su fuente de poder, hace lo mismo y más, a diferencia de los inductores y capacitores, libera más energía que la que es absorbida por las resistencias. Los amplificadores son llamados elementos *activos* por esta causa, y los filtros que incorporan amplificadores son, por extensión, llamados *filtros activos*.

La eliminación de los inductores es una ventaja ya que su desempeño es el menos ideal de los elementos básicos de circuitos. Más aún, tienden a ser voluminosos y pesados, y debido a que no se prestan para la producción en masa de circuitos integrados, son por tanto caros. Sin embargo, el rompimiento hacia el amplificador operacional (*amp op*) vino con el advenimiento del *amp op* monolítico de bajo costo, que hizo los filtros activos comercialmente factibles.

Un filtro activo trabajará apropiadamente en el rango que el *amp op* lo haga. El comportamiento no ideal de un *amp op* a altas frecuencias, restringe la aplicación de filtros activos a rangos debajo de los Mhz. A frecuencias más altas que los *amp op* pueden manejar, los inductores son la opción factible, de tal forma que los filtros para altas frecuencias son implementados con componentes pasivos RLC. Aquí los tamaños y pesos de los inductores son más manejables, ya que los valores de inductancias y capacitancias decrecen en la medida que la frecuencia de operación se incrementa.

### Función de Transferencia

Los filtros son implementados con dispositivos con características dependientes de la frecuencia, como son, capacitores e inductores. Estos dispositivos, cuando son alimentados con una señal senoidal oponen una resistencia al flujo de corriente que depende de la frecuencia e introducen un cambio de fase de  $90^\circ$  entre la corriente y el voltaje. Para explicar este comportamiento recurrimos a las *impedancias complejas*  $Z_L = sL$  y  $Z_C = 1/(sC)$ , donde  $s$  es la variable compleja ( $s = \sigma + j\omega$ ;  $\omega$  es la frecuencia angular, en rad/s;  $j = \sqrt{-1}$  es la unidad imaginaria)

El comportamiento de un filtro se caracteriza de forma única por su función de transferencia  $H(s)$ . Su papel es similar al de la ganancia para un amplificador ordinario, excepto que en aplicaciones de filtrado se debe considerar la fase, y ambos, la ganancia y la fase, en general, varían con la frecuencia. Para hallar esta función, se tiene una función para  $V_o$  en términos de  $V_i$ , utilizando herramientas tales como la ley de Ohm, Ley de Voltajes de Kirchoff,

Ley de Corrientes de Kirchoff, fórmulas para los Divisores de Corriente y Voltaje, y el Principio de Superposición. En términos prácticos, todas las funciones de transferencia deben tomar la forma de funciones racionales de  $s$ :

$$H(s) = \frac{N(s)}{D(s)} \quad (1.11)$$

Donde  $N(s)$  y  $D(s)$  son los polinomios adecuados para  $s$  con coeficientes reales, y el orden de  $N(s)$  nunca excede el de  $D(s)$ . El orden de  $D(s)$  es de un significado especial y es conocido como *orden del filtro* (primer orden, segundo orden, etc.). Los ceros de  $N(s)$  y  $D(s)$  son llamados, respectivamente, *ceros y polos* de  $H(s)$ . Un principio fundamental de la teoría de sistemas es que *todos los polos deben caer en la mitad izquierda del plano complejo* para que el circuito sea estable. Si este no es el caso, las señales llegan a su nivel de saturación y el circuito comienza a oscilar.

Una vez que  $H(s)$  se conoce, la respuesta  $V_o(t)$  del circuito para una entrada arbitraria  $V_i(t)$ , puede en principio ser determinada como  $V_o(t) = L^{-1}[H(s)V_i(s)]$ , donde  $V_i(s)$  denota la transformada de Laplace para  $V_i(t)$  y  $L^{-1}$  denota la transformada inversa de Laplace. La ganancia del filtro,  $A_v$ , es la tasa o relación entre el voltaje de salida del filtro y el voltaje de entrada. La ganancia se expresa frecuentemente en decibeles y se calcula de la fórmula:

$$A_{dB} = 20 \log A_v \quad (1.12)$$

### 1.2.5 Convertidores A/D - D/A

Cada día las computadoras digitales encuentran más y más áreas de utilización gracias al rápido desarrollo de microprocesadores cada vez más potentes y al mismo tiempo de menor costo. Actualmente es común utilizar una microcomputadora para muchas actividades de control así como procesamiento de señales campos que anteriormente eran dominados por circuitos lineales.

Los sistemas y dispositivos digitales están diseñados para trabajar con datos expresados en forma discreta, esto es, niveles de voltaje que representan números binarios. Sin embargo, la mayoría de los fenómenos del mundo tienen una naturaleza continua, por lo que para ser interpretados por una computadora digital necesitan ser representados de una manera discreta. Por el contrario, cuando una computadora se encarga de controlar un dispositivo que utiliza información de manera continua, la primera debe representar la información discreta de una forma analógica (voltaje o corriente eléctrica). En este apartado describimos las características de los dispositivos que se encargan de realizar la interfaz entre el mundo real y los sistemas digitales, dichos dispositivos se llaman convertidores analógico-digital y digital-analógico. Se comenzara por describir los convertidores D/A ya que éstos son utilizados por algunos tipos de convertidores A/D.

### *Convertidores Digital-Analógico*

La función de un convertidor D/A (de ahora en adelante CDA), es aceptar un grupo de bits de algún dispositivo digital y convertir dicho patrón de bits en un nivel de voltaje analógico equivalente. Generalmente se interpreta el patrón de bits mencionado como un número binario.

La salida del CDA debe tener la capacidad de tomar un nivel diferente para cada entrada digital única que se aplique. La salida del CDA puede ser un voltaje o una corriente, dependiendo de su construcción interna. Las características principales de un CDA son:

#### *Resolución y salida a plena escala*

El número de niveles de salida diferentes que puede dar un CDA está relacionado con el número de líneas de entrada que tiene:

$$N = 2^n \quad (1.13)$$

donde  $N$  es el número de niveles de salida diferentes que el CDA puede producir y  $n$  es el número de bits de entrada que tiene el convertidor. El número de niveles de salida diferentes se utiliza para definir la resolución del dispositivo. Mientras más entradas tiene el convertidor la resolución que tiene es más alta. La resolución puede ser expresada como una parte en  $N$ , por ejemplo, si un convertidor consta de 10 entradas (10 bits), el convertidor tendrá una resolución de 1 en 1024 ( $2^{10} = 1024$ ). La resolución puede ser expresada también como un porcentaje mediante la siguiente ecuación:

$$R = \frac{1}{2^n} \times 100\% \quad (1.14)$$

Para el caso presente, el convertidor tiene una resolución del 0.098 %, que se interpreta como sigue: el cambio más pequeño posible en la salida del convertidor causado por un cambio en la entrada será del 0.098% de la salida a plena escala.

La salida a plena escala es el nivel de voltaje ( $V_{FS}$ ) o corriente ( $I_{FS}$ ) producido a la salida de un CDA hipotético cuya resolución es infinita (con un número de entradas infinito) aplicando un 1 binario a cada entrada. El cambio más pequeño que se puede producir en la salida, producido por las entradas se denomina 1 LSB o incremento. Se denomina así porque el cambio más pequeño ocurre cuando el bit de menos significancia de la entrada (*Least Significant Byte*) cambia de estado. Lo anterior se expresa como:

$$1LSB = \frac{V_{FS}}{2^n} \quad (1.15)$$

#### *Exactitud*

Exactitud y resolución no son la misma cosa, ya que no necesariamente la salida es una representación exacta para una entrada dada. Bajo condiciones normales, la salida de un CDA

debe tener un rango de exactitud de  $\pm 0.5$  LSB. Sin embargo, existen muchas posibles causas para que se produzca un error en la salida de un convertidor D/A, algunas situaciones se presentan a continuación:

**Error de desplazamiento.** Un error de este tipo causa que la salida de un convertidor no sea cero cuando a la entrada se aplica un cero binario; esto provoca que la salida esté desplazada hacia arriba o hacia abajo de una manera constante, por lo que el error absoluto es constante también.

**Error de ganancia.** Un error de este tipo producirá incrementos más largos o bien, más cortos que el esperado LSB. El error absoluto se incrementa cuando la entrada aumenta.

**Error de linealidad.** Este tipo de error es causado por no-linealidades inherentes al convertidor. Esto provoca que la salida del mismo no sea proporcional a la entrada como se presentaría en un CDA ideal. En otras palabras, el LSB puede incrementarse o decrementarse cuando la entrada aumenta. Las principales causas de la no-linealidad son la temperatura y las influencias externas.

#### *Tiempo de Estabilización*

Un convertidor D/A ideal responderá inmediatamente a una entrada. En la práctica, un CDA requiere una cantidad de tiempo para cambiar el estado de la salida y estabilizarse en un nivel de salida. Este intervalo de tiempo se le conoce como tiempo de estabilización.

#### *Circuitos de conversión D/A*

Los convertidores D/A pueden encontrarse en circuitos integrados o bien pueden ser diseñados utilizando amplificadores operacionales. A continuación se mencionan los circuitos típicos que implementan un convertidor D/A.

#### *CDA de Resistencias Ponderadas*

Este es uno de los convertidores D/A más simples. El corazón de este tipo de convertidor está formado por un amplificador sumador. En la figura 1.29 se muestra un CDA de este tipo de 4 bits. Las resistencias de entrada se escogen de tal manera que, partiendo de LSB (interruptor 1), cada entrada sucesiva es amplificada al doble que su predecesora; de esta forma las ganancias relativas a las entradas se incrementan en potencias de 2.

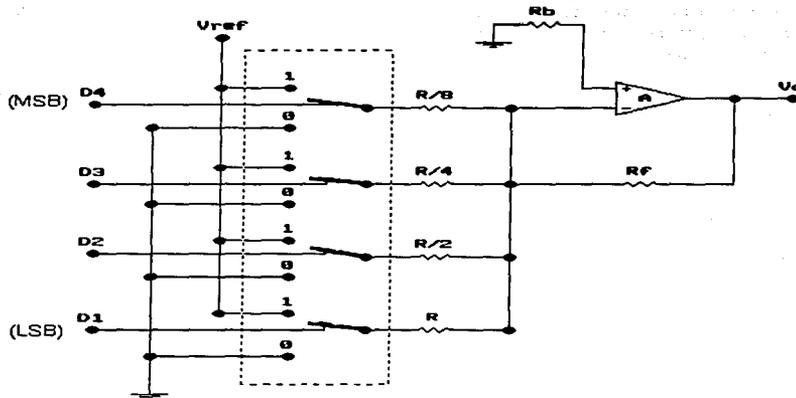


Figura 1.29. Convertidor D/A de resistencias ponderadas.

Analizando el circuito se obtiene la siguiente expresión para el voltaje de salida:

$$V_o = -V_{ref} \left( \frac{D_1 R_F}{R_1} + \frac{D_2 R_F}{R_2} + \dots + \frac{D_n R_F}{R_n} \right) \quad (1.16)$$

Donde  $D_1, D_2, \dots, D_n$  representan el estado de los interruptores (0 = abierto y 1=cerrado).

Este tipo de circuitos no es práctico para aplicaciones que requieren alta resolución; la razón es que se necesitaría un rango de valores de resistencia amplio y que no se encuentra disponible en el mercado. Además, la linealidad depende de la precisión de las mismas debiendo ser ésta del orden de 5%.

#### Convertidor D/A R-2R

Es uno de los más populares convertidores basado en una configuración de escalera R-2R. Un ejemplo se muestra en la figura 1.30. La ventaja de este circuito sobre el convertidor de resistencias ponderadas es que solo se requieren definir con precisión dos valores de resistencias para cualquier número de entradas binarias.

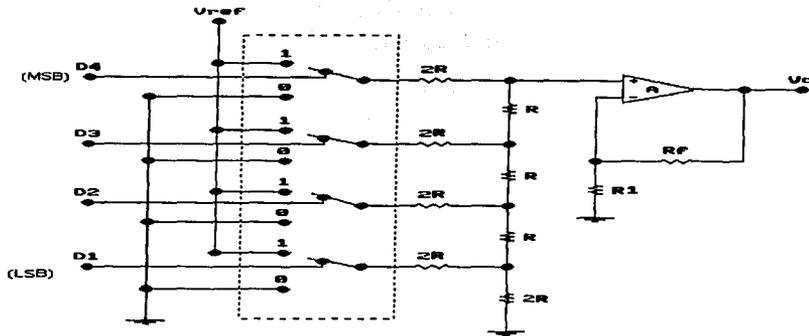


Figura 1.30. Convertidor D/A R-2R

Analizando el circuito anterior se llega a la siguiente expresión para el voltaje de salida:

$$V_o = (V_{ref}) \left( 1 + \frac{R_F}{R_1} \right) \left( \frac{D_4}{2} + \frac{D_3}{4} + \frac{D_2}{8} + \frac{D_1}{16} \right) \quad (1.17)$$

Se tiene entonces un convertidor que podrá ser expandido a cualquier resolución arbitraria. Hablando en términos de las resistencias, lo anterior es cierto; sin embargo, al aumentar la resolución exige mayor exactitud por parte del amplificador operacional.

#### Convertidores D/A en Circuito Integrado

Son la mejor elección para la mayoría de las aplicaciones de un CDA. Los hay disponibles para 8, 10, 12, 14 y 16 bits y vienen en una de las siguientes tres configuraciones:

1. Salida de tipo voltaje
2. Salida de tipo corriente
3. Ambos tipos de salida (voltaje y corriente)

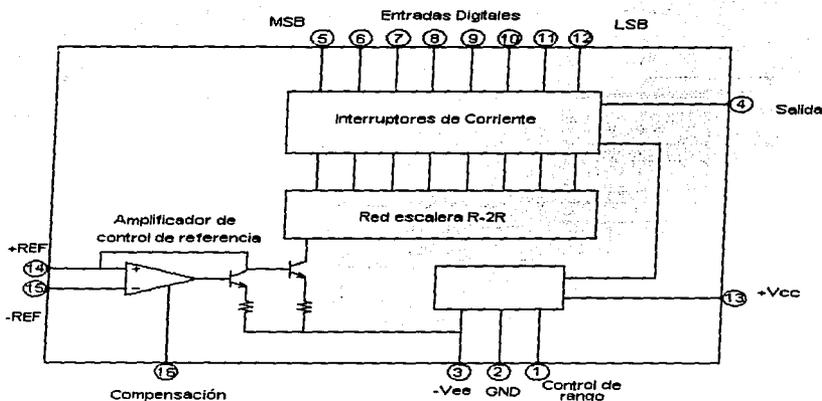


Figura 1.31. Diagrama lógico del CDA en circuito integrado DAC-IC8B de 8 bits.

En la figura 1.31 se observa un diagrama de bloques de un circuito integrado CDA de 8 bits. Este circuito consiste de ocho interruptores rápidos de corriente, una red de resistencias R-2R, un circuito de polarización y un amplificador de control de referencia. Una corriente de 2 mA externa programa el factor de escalamiento del CDA. Esta normalmente va acompañada de una fuente de voltaje de referencia (como un diodo Zener) y una resistencia.

### Convertidores Analógico-Digital

La función de un convertidor analógico-digital (CAD de ahora en adelante) es muestrear alguna señal analógica (normalmente un voltaje) y producir una representación digital cuantificada (normalmente binaria) de tal señal a la salida. Como en los convertidores D/A, el número de bits en la salida define la resolución y el potencial de exactitud del convertidor A/D. Las características que muestra un CAD son las siguientes:

#### Rango a plena escala

La entrada de un CAD comprende un rango de valores limitados, por ejemplo, 0 a 10 Volts. El abanico de valores de entrada que pueden ser resueltos por el convertidor se le conoce como *rango a plena escala (FSR)*. El rango de entrada del convertidor se divide en  $2^n$  partes, donde n es el número de bits de la salida codificada. Este valor es análogo a la resolución de los convertidores D/A, y se conoce como División de Rango de Entrada (DRE). Se expresa como:

$$DRE = 1LSB = \frac{FSR}{2^n} \quad (1.18)$$

### *Especificación de errores*

Aun dentro de un rango finito, una cantidad analógica puede tomar un número infinito de valores diferentes. El convertidor A/D muestrea una entrada analógica y produce una salida cuantificada en forma de código de palabra que representa tal entrada. Esto en sí introduce la posibilidad de un error a la salida.

**Error de cuantificación.** Debido a que la salida tiene un número finito de bits, siempre habrá una incertidumbre acerca del valor real de la entrada. Lo más cerca que se puede estar de la entrada es  $\pm 0.5$  LSB. Esta incertidumbre asociada con una conversión dada es conocida como *error de cuantificación* o *ruido de cuantificación*. Como mínimo, todo convertidor A/D tiene un error de cuantificación de 0.5 LSB.

En aplicaciones de procesamiento digital de señales se acostumbra expresar el ruido de cuantificación en decibeles, llamado entonces la *razón señal - error de cuantificación (SNQR)*, definido como:

$$SNQR = 20 \log \frac{FSR(V)}{LSB(V)} \text{ dB} \quad (1.19)$$

**Error de desplazamiento.** Debido a la naturaleza de los dispositivos utilizados para construir convertidores A/D, se presenta este tipo de error, el cual producirá básicamente un desplazamiento por una cantidad fija de las divisiones del rango de entrada.

**Error de ganancia.** Un error de este tipo resultará en una salida binaria no relacionada con la entrada. La presencia de un error de ganancia provoca un FSR más pequeño que el esperado si la ganancia es muy alta, o un FSR mayor al esperado si la ganancia es baja.

**No-linealidad diferencial.** Es la diferencia entre la división del rango de entrada real y la teórica. Esto normalmente produce un código saltado a la salida.

### *Tiempo de Conversión*

Un convertidor A/D real requiere un tiempo finito mayor a cero para completar una conversión. Este intervalo de tiempo es conocido como *tiempo de conversión*  $T_c$ . No todos los CAD tienen el mismo tiempo de conversión y la mayoría de las veces se especifica bajo las condiciones de el peor de los casos. El inverso de el tiempo de conversión se denomina *razón de conversión*.

### *Circuitos de conversión A/D*

Existen muchas formas de implementar un circuito de conversión A/D. Sin embargo, muchas de estas implementaciones están disponibles en circuitos integrados.

### Convertidores de rampa

Deriva su nombre de la forma en que realiza la conversión. Tomando como base la figura 1.32, describimos a continuación su operación:

1. La entrada **START** se pone a cero. Esto limpia el contador, causando que la salida del convertidor D/A sea cero.
2. Debido a que el voltaje de entrada es mayor al voltaje de salida del CDA, la salida del comparador es cero, habilitando la compuerta **OR** para pasar pulsos de reloj para el contador. La salida del comparador es usada también para indicar que una conversión está en progreso ( $EOC=0$ ).
3. El contador se incrementa en cada pulso de la señal de reloj, forzando la salida del convertidor D/A a incrementarse en 1 LSB en cada pulso. Esta acción continúa hasta que la salida del convertidor D/A es mayor al voltaje de entrada que se muestrea.
4. Una vez que la salida del CDA es mayor al voltaje de entrada, la salida del comparador es uno, lo que deshabilita la entrada del reloj del contador. Un estado alto en la salida **EOC** indica que la conversión ha sido concluido y la cuenta a la salida es proporcional a la entrada analógica.

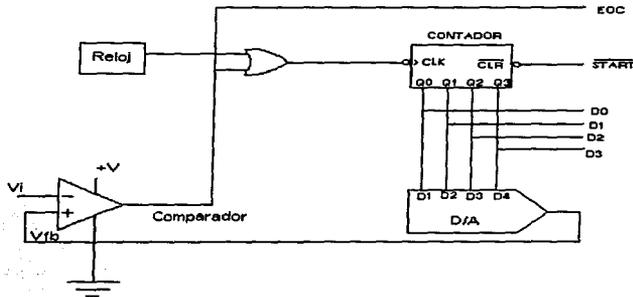


Figura 1.32. Convertidor A/D de rampa de 4 bits simplificado.

### Convertidor de aproximaciones sucesivas

Los convertidores A/D de propósito general más usados se diseñan alrededor de un circuito digital llamado registro de aproximaciones sucesivas (SAR), el cual ofrece ciertas ventajas sobre el convertidor de rampa.

En la figura 1.33 se muestra un diagrama de bloques de este tipo de convertidor. El SAR se utiliza para producir estimaciones del voltaje de entrada, comenzando con  $FSR/2$  y trabajando hacia abajo hasta LSB, un bit a la vez. La salida del CDA interno se compara con el voltaje de entrada después de cada cambio interno de estado. Si al encender un bit en particular resulta que el voltaje de salida del CDA es mayor al de entrada, entonces el bit es puesto a cero. Si sucede lo contrario, ese bit se deja encendido.

Este proceso de mantener/descartar se repite para cada bit sucesivo hasta que, después de la octava prueba, la conversión está completa. Esta técnica es similar a aquella que utilizaría una persona para adivinar un número entre 0 y 255. Bajo ésta aproximación no toma más de ocho preguntas para encontrar el número correcto.

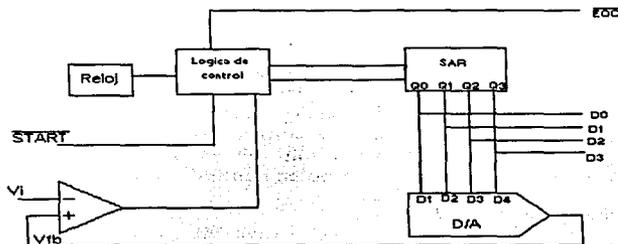


Figura 1.33. Convertidor A/D SAR de 8 bits simplificado.

### Convertidor A/D de doble rampa

Este es un ejemplo de convertidor de los de tipo integrador. A continuación se muestra un diagrama simplificado del mismo:

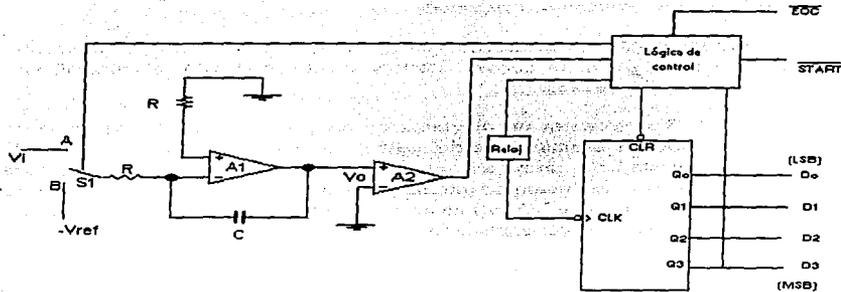


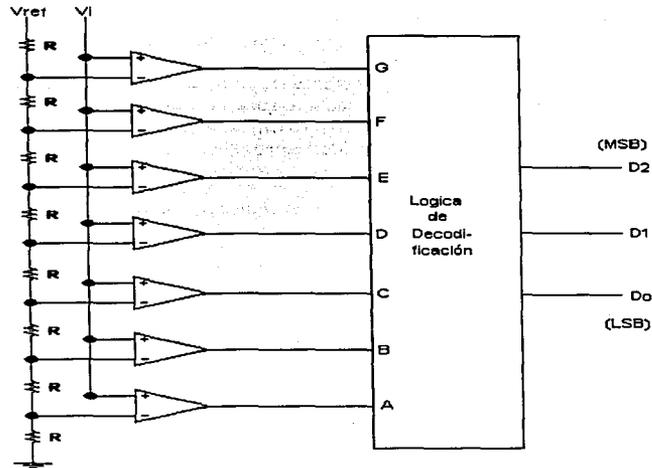
Figura 1.34. Convertidor A/D de doble rampa.

Este circuito está diseñado para trabajar con voltajes de entrada opuestos en polaridad al voltaje de referencia. El núcleo de este circuito es un integrador conformado por los amplificadores  $A_1$  y  $A_2$ . El segundo amplificador es un comparador que utiliza como referencia la tierra. La salida del circuito es tomada por un contador ascendente.

En general, los convertidores de doble rampa pueden ser muy exactos. Además, debido al circuito integrador, el ruido de entrada tiende a ser mínimo. Esto porque lo más común es que un voltaje aleatorio de ruido tiende a tener un área igual a cero Volts, por lo que al integrarlas, resulta cero. Este tipo de convertidores son lentos en comparación con los de aproximaciones sucesivas, por lo que son utilizados en dispositivos en donde la velocidad de conversión no es un factor crítico, como en multímetros digitales.

#### *Convertidores A/D Paralelos*

También conocidos como convertidores flash, son utilizados en aplicaciones que requieren tiempos de conversión extremadamente rápidos. En la figura 1.35 se observa una implementación de 3 bits de este tipo de convertidor.



**Figura 1.35. Convertidor A/D paralelo de 3 bits simplificado.**

La sección principal de este circuito es la cadena de comparadores, los cuales tienen conectadas sus entradas no inversoras en paralelo a la entrada, de ahí el nombre de este convertidor. El voltaje de referencia se aplica al divisor de voltaje que proporciona la entrada inversora para cada comparador con una fracción del voltaje de referencia por una potencia de dos. Debido a que la entrada paralela es muestreada de manera simultánea por todos los comparadores, el único retardo para producir una salida binaria es el tiempo de interrupción de cada comparador y el retardo de propagación de la lógica de decodificación. Comparadores de alta velocidad y lógica rápida como dispositivos ECL y GaAs (Galio-Arsénico), hacen posible una conversión extremadamente rápida. Se han producido convertidores paralelos con tiempos de conversión del orden de los cientos de picosegundos. La mayor desventaja de estos convertidores es la mayor cantidad de comparadores requeridos para obtener una alta resolución.

#### *Resumen de convertidores A. D*

*Los convertidores basados en técnicas de integración presentan las siguientes características:*

- Excelente linealidad
- No requieren técnicas de muestreo y retención (*Sample and Hold*)

- **Bajo costo y simplicidad**
- **La sección contadora puede estar alejada del circuito de entrada permitiendo enviar la información por dos cables**
- **La respuesta se hace independiente de los elementos que lo constituyen**
- **Buen rechazo a señales alternas y al ruido en general**
- **Generalmente el tiempo de conversión es largo**
- **Funcionan con una sola polaridad normalmente**
- **Se aplican en multímetros digitales y donde no se requiere rapidez de conversión**

*Los convertidores de aproximaciones sucesivas se distinguen por:*

- **Tiempo de conversión corto**
- **Alta resolución**
- **Información de salida en serie o en paralelo**
- **Buena linealidad**
- **La exactitud no depende de la linealidad del reloj**
- **Requiere de circuitos de muestreo y retención**
- **Complejidad en los circuitos**

En este capítulo se han presentado detalles que van desde cómo funcionan las neuronas, el cerebro y las señales que en este órgano se generan. Esto permitió observar el por qué de la electroencefalografía y sus aplicaciones. Se presentaron los conceptos que son la base técnica, eléctrica y electrónica que indican como trabajar las señales electroencefalográficas. Ahora, con todo esto presente, se entrará de lleno al *Procesamiento Digital de Señales*, que es el tema tratado en el siguiente capítulo.

## Capítulo 2

# El Procesamiento Digital de Señales y sus Aplicaciones

Dentro de las áreas de la ingeniería eléctrica se encuentra el Procesamiento Digital de Señales (PDS). Esta área se ha desarrollado notoriamente en los últimos veinticinco años, en paralelo con los avances de la ciencia y de la tecnología en disciplinas tales como la electrónica, las comunicaciones (tanto analógicas como digitales), la teoría de control y la computación.

Entre las diferentes áreas en que se apoya esta disciplina de la ingeniería se tienen las siguientes: análisis de sistemas y señales, análisis y síntesis de filtros digitales, análisis espectral, estimación espectral, modelado de señales, identificación de sistemas, la ingeniería de *software*, la arquitectura de microcomputadoras, el diseño de circuitos a muy grande escala de integración VLSI (*Very Large Scale Integration*), etc.

En el presente capítulo será presentada brevemente la teoría básica en que se sustenta el procesamiento digital de señales, en donde se introducirá la noción de sistemas discretos, enfocándose en los sistemas lineales invariantes en el tiempo. Posteriormente se mostrarán las técnicas de procesamiento, en donde se expondrá la transformada de Fourier discreta, así como la transformada rápida de Fourier (FFT) y la transformada rápida de Hartley (FHT). También al final de el presente capítulo se explican la función de correlación y la estimación de la densidad espectral. Así entonces a lo largo del capítulo se muestran las técnicas del procesamiento digital, sus aplicaciones y sus usos, con la finalidad de proporcionar un conocimiento básico al lector de lo que hace cada una de estas técnicas y cómo es que lo hace.

---

## 2.1 Análisis de Sistemas y Señales

Los principios en los que se fundamenta el Procesamiento Digital de Señales (PDS) están dados por la teoría de señales y sistemas. Por ello, para lograr un entendimiento de las diferentes áreas del PDS es imprescindible conocer estos principios. Es por eso que a continuación se presentara en forma concisa el análisis de sistemas y señales como fundamento del PDS.

Para facilitar el estudio y la comprensión de los fenómenos naturales el hombre se ha propuesto conceptualizarlos a través de abstracciones o representaciones matemáticas. Se puede representar un fenómeno de cualquier tipo, ya sea físico, biológico, económico, social, etcétera, como una caja negra que tiene una entrada y una salida, como muestra la figura 2.1. Esta es la más abstracta definición de un sistema. El fenómeno, sistema o caja negra realiza una transformación de la entrada (excitación) y arroja una salida (reacción).



Figura 2.1. Representación abstracta de un sistema.

Dado que una de las variables más importantes en la vida del hombre es el tiempo, las características de las señales y los sistemas son principalmente dadas en función de la variable temporal. Sin embargo, se observara a lo largo de esta tesis que la ayuda que brinda el trabajar en función de la variable frecuencial es altamente valiosa.

La noción de sistema es la siguiente: Dados dos conjuntos de señales  $C_x$  y  $C_y$ , se asignará a cada elemento  $x$  de  $C_x$  un elemento  $y$  de  $C_y$ . Esto se hace de acuerdo con una regla, esta regla definirá un sistema con entrada  $x$  y salida  $y$ . Por tanto, un sistema  $S$  definido de esta manera es una correspondencia o una transformación del conjunto de entrada  $C_x$  al conjunto de salida  $C_y$ .

Esta caracterización de un sistema no es única. También pueden ser caracterizados por ecuaciones diferenciales, ecuaciones en diferencias, función de transferencia, polinomios, etcétera.

En el presente capítulo se dará especial interés en la representación del sistema por su función de transferencia, porque permitirá la representación del sistema en función de la variable frecuencial.

Una función de transferencia es una representación del sistema paralela a la representación temporal (variable  $k$ ). La función de transferencia caracteriza al sistema en el dominio de la frecuencia (variable  $z = e^{j\omega}$ ). La herramienta que permite pasar del dominio temporal al frecuencial y viceversa es la transformada  $Z$  y la transformada  $Z$  inversa,  $Z^{-1}$ , respectivamente.

Dentro de la clasificación de los sistemas sólo se escogieron para este análisis los sistemas discretos lineales e *invariantes en el tiempo*, cuya definición se expone a continuación.

### 2.1.1 Señales Discretas en el Tiempo

El explicar las señales discretas en el tiempo<sup>1</sup> es con el fin de utilizar una computadora digital para realizar los cálculos de los algoritmos provenientes del análisis.

Una señal discreta en el tiempo,  $x$ , es representada por una secuencia de números reales o complejos definidos para todo número entero  $k$ .

$$x = \{ x[kT] \} \quad (2.1)$$

$k$  es el conjunto de todos los enteros.

$T$  es el intervalo de tiempo entre muestras.

#### *Señales Discretas Especiales*

Existen una serie de señales discretas características que en muchas ocasiones permiten representar la entrada y salida de un sistema y de las cuales ya se tienen definidos parámetros para su análisis. Estas se conocen como *señales discretas especiales* y son:

- Impulso unitario
- Escalón unitario
- Señales sinusoidales discretas en el tiempo
- Exponenciales complejas relacionadas armónicamente

A continuación se presenta una explicación de cada una de estas señales:

Impulso unitario. La señal impulso unitario, también llamada señal delta, es una secuencia que vale la unidad cuando su argumento es cero y vale cero para cualquier otro valor del argumento.

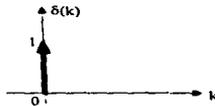
$$\delta[k] = \begin{cases} 1 & k = 0 \\ 0 & k \neq 0 \end{cases} \quad (2.2)$$

Gráficamente tenemos:

---

<sup>1</sup> Es ampliamente utilizado el término *señales en el tiempo discreto*, haciendo referencia a lo que aquí se han llamado *señales discretas en el tiempo*. Sin embargo, en este trabajo se homogeneizará la nomenclatura a ésta última forma, que es más exacta, pues la característica de *discreta* es propia de la señal y no de la variable. Análogo es el caso de las señales continuas en el tiempo.

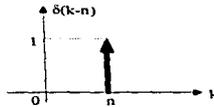
---



El impulso unitario permitirá caracterizar la salida del sistema para cualquier entrada. Lo anterior debido a que los valores tomados por el impulso unitario, cuando el tiempo es desplazado de más o menos un número entero  $n$ , están dados por:

$$\delta[k-n] = \begin{cases} 1 & k = n \\ 0 & k \neq n \end{cases} \quad (2.3)$$

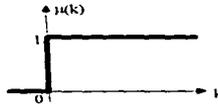
en forma gráfica esto es:



Escalón unitario. La señal escalón unitario toma el valor de la unidad para todos los argumentos más grandes que o igual a cero, y es igual a cero para los demás valores del argumento.

$$\mu[k] = \begin{cases} 1 & k \geq 0 \\ 0 & k < 0 \end{cases} \quad (2.4)$$

Gráficamente se representa como:



El escalón unitario se relaciona con el impulso unitario por:

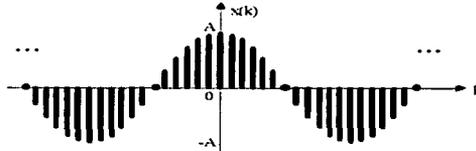
$$\mu[k] = \sum_{n=-\infty}^k \delta[n] = \sum_{n=0}^k \delta[k-n] \quad (2.5)$$

Señales sinusoidales discretas en el tiempo. Una señal sinusoidal discreta en el tiempo se define como sigue:

$$x[k] = A \cos [\Omega k + \theta] \quad (2.6)$$

Donde  $k$  es la variable entera, llamada número de muestra, y varía de menos infinito a más infinito,  $A$  es la amplitud de la señal,  $\Omega$  es la frecuencia en radianes por muestra y  $\theta$  es la fase en radianes.

En forma gráfica:



Si en lugar de  $\Omega$  se usa la variable frecuencial  $f$ , definida por:

$$\Omega = 2\pi f \quad (2.7)$$

la relación (2.6) se vuelve:

$$x[k] = A \cos [2\pi f k + \theta] \quad (2.8)$$

La frecuencia  $f$  tiene dimensiones de ciclos por muestra.

A diferencia de las sinusoidales continuas en el tiempo, las sinusoidales discretas en el tiempo tienen las siguientes propiedades:

a) Las sinusoidales discretas en el tiempo son periódicas sólo si  $f$  es un número racional.

Por definición, si la señal  $x[k]$  es periódica con período  $N$  mayor que cero, entonces:

$$x[k+N] = x[k], \quad \text{para todo } k \quad (2.9)$$

El valor más pequeño para el cual la expresión anterior es válida, es llamado período fundamental.

$$f_0 = \frac{m}{N} \quad \text{o bien,} \quad N = \frac{m}{f_0} \quad (2.10)$$

Para determinar el período fundamental,  $N$ , de una señal periódica se expresa su frecuencia  $f$  como en la relación (2.10), hasta que se obtenga una mínima expresión de tal forma que  $m$  y  $N$  no tengan factores comunes. Así el período fundamental será igual a  $N$ .

b) Aquellas señales sinusoidales discretas en el tiempo cuyas frecuencias estén separadas por un múltiplo entero de 2, son idénticas.

c) Las tasas de oscilación más altas en señales sinusoidales discretas en el tiempo son alcanzadas cuando de manera equivalente,  $f = 1/2$  (o  $f = -1/2$ )

**Exponenciales Complejas Relacionadas Armónicamente.** Las señales sinusoidales y las exponenciales complejas juegan un papel importante en la teoría de señales y sistemas. Es interesante trabajar con funciones exponenciales (o sinusoidales) relacionadas armónicamente. Estas forman conjuntos de exponenciales complejas periódicas, con frecuencias fundamentales que son múltiplos de una frecuencia positiva.

Puesto que una señal sinusoidal discreta en el tiempo es periódica, si su frecuencia relativa es un número racional, se puede escoger  $f = 1/N$  y así definir a los conjuntos de exponenciales complejos relacionados armónicamente, de la siguiente manera:

$$S_n(k) = e^{j2\pi f_0 kn} \quad (2.11)$$

para  $n = 0, \pm 1, \pm 2, \pm 3, \dots$

Es evidente que se puede escoger cualquier número  $N$  de exponenciales complejas consecutivas, de  $n=n_0$  a  $n=n_0+N-1$ , para formar un conjunto relacionado armónicamente, con frecuencia fundamental  $f_0=1/N$ . Si  $n_0=0$ , el conjunto será:

$$S_n[k] = e^{j2\pi kn/N}, \quad n = 0, 1, 2, \dots, N-1. \quad (2.12)$$

### *Muestreo de una Señal Continua en el Tiempo*

Sea la señal continua  $x(t)$  a banda limitada,  $|X(\omega)| = 0$  para  $|\omega| > \beta$ , si se toman muestras lo suficientemente espaciadas con respecto a la frecuencia más alta de la señal  $x(t)$ , entonces las muestras determinarán únicamente a esa señal y ésta podrá ser reconstruida a partir de sus muestras.

Las muestras  $x(nT)$  para  $n = 0, \pm 1, \pm 2, \pm 3, \dots$  podrán reconstruir  $x(t)$  mediante:

- La generación de un tren de impulsos cuyas amplitudes sean las muestras sucesivas.
- Pasando este tren de impulsos por un filtro paso bajas con ganancia  $T$  y frecuencia de corte  $\omega_c$ , donde  $\beta < \omega_c < \omega_s - \beta$ . En este caso, la salida resultante será exactamente igual a  $x(t)$ .

Donde  $\omega_s > 2\beta$

Lo anterior es el enunciado del teorema del muestreo, y a  $\omega_s > 2\beta$  se le conoce como la frecuencia de Nyquist o de Shannon.

El teorema del muestreo permite representar una señal continua  $x(t)$ , en términos de los valores  $x(kT)$  que toma la señal mediante una secuencia de puntos equidistantes. Los puntos equidistantes pueden ser representados por la secuencia  $x[kT]$ , la señal discreta. Lo anterior es válido siempre y cuando se satisfaga el teorema del muestreo para una señal  $x(t)$  continua.

Para resaltar un detalle importante que se relaciona con el muestreo de una señal es importante considerar que al muestrear una señal se está obteniendo una señal discreta a partir de una señal continua. Esto cambia la esencia de la señal en el sentido siguiente: Ahora sólo se tiene el valor asociado de la señal para valores del tiempo  $kT$ , donde  $k$  es un número entero.

Ahora bien, esta característica de una *señal discreta en el tiempo* tiene su análogo en lo que se introduce a continuación como *señal digital*.

Una *señal digital*<sup>2</sup> es aquella que está representada en el eje de las ordenadas por valores no continuos, o sea discretos. Es decir, la amplitud de la señal sólo toma algunos valores *finitos*. Por ejemplo, una señal binaria es digital, pues sólo toma valores de cero y uno. Así, no importa si hay dos o miles de valores para asignar una amplitud, si ese número de valores es finito, entonces se trata de una representación digital de una señal, o simplemente una *señal digital*. Si se piensa con detenimiento lo que establece este párrafo, se entenderá por qué existe a menudo la confusión de usar, por error, el término *señal digital* como sinónimo a *señal discreta*.

Ahora bien, es absolutamente cierto que la representación en computadora de una señal obliga a que la señal sea *discreta y digital*. Esto debido al error de representación intrínseco de la computadora por tener un número de *bits*<sup>3</sup> limitado para representar una cantidad, por lo que se tiene que efectuar redondeo y truncamiento. Esto es lo que se conoce como el problema de cuantificación. Por eso, aún la más poderosa super computadora, con la palabra de memoria más grande, introduce errores en la representación de la cantidad.

### 2.1.2 Sistemas Discretos Lineales e Invariantes en el Tiempo

Un sistema lineal discreto es una correspondencia o transformación entre la secuencia  $x[k]$  y la secuencia  $y[k]$  [GaRo75]. Para representar la relación entre  $x[k]$  y  $y[k]$  se utiliza comúnmente la notación siguiente:

$$y[k] = L\{x[k]\} \quad (2.13)$$

Donde  $L$  es una transformación.

A continuación se señalan las propiedades que se deben de observar en un sistema para ser considerado sistema discreto lineal y/o invariante en el tiempo:

Sistemas con memoria o sin memoria. Un sistema se dice sin memoria, si su salida para cada valor de la variable independiente  $k$ , es únicamente dependiente de la entrada al mismo tiempo  $k$ .

Por otro lado, un sistema con memoria es aquel para el cual la salida depende de los valores de la entrada anterior al tiempo  $k$ .

Inversibilidad de un sistema. Un sistema es inversible cuando observando su salida se puede determinar su entrada.

---

<sup>2</sup> Recordemos que *digital* viene de *dígito* (0..9)

<sup>3</sup> BIT es un acrónimo de Binary digIT (dígito binario)

**Sistema Causal.** Un sistema causal es aquel cuya salida no depende de valores futuros de la entrada, esto es, para que la salida producida por el sistema cambie, se requiere que la entrada aplicada a éste cambie con anterioridad.

**Linealidad de un sistema.** Un sistema es lineal si satisface las propiedades de:

a) superposición:  $y_1[k] + y_2[h] = L\{x_1[k] + x_2[h]\}$

b) escalamiento:  $a y[k] = L\{a x[k]\}$

**Sistemas invariantes en el tiempo.** Un sistema es invariante en el tiempo si para una entrada  $x[k-n]$  dada a una transformación  $L$  se obtiene a la salida  $y[k-n] = L\{x[k-n]\}$ . En otras palabras, un desplazamiento en la entrada se traduce por un desplazamiento igual en la salida.

Recordando que las características de linealidad e invariancia en el tiempo son independientes entre sí, es decir, que el que se cumpla una de ellas no obliga a que se cumpla la otra. También hay otras propiedades importantes de los sistemas que aquí no se han mencionado, y dentro de este universo de diferentes características se expondrán básicamente las correspondientes a los sistemas discretos lineales e invariantes en el tiempo.

En lo sucesivo se abreviara el nombre de sistemas discretos lineales invariantes en el tiempo simplemente como *sistemas lineales*.

*Respuesta de un sistema lineal al impulso unitario  $\delta[k]$*

Si  $x[k] = \delta[k]$  entonces la salida,  $y[k]$ , del sistema será la transformación de  $\delta[k]$ , o

$$y[k] = L\{\delta[k]\}.$$

Suponiendo que se conoce la respuesta  $y[k]$ , es decir, definiendo a  $h[k]$  como la respuesta al impulso unitario  $\delta[k]$ .

$$L\{\delta[k]\} = h[k] \quad (2.14)$$

Si la función  $h[k]$  es igual a cero para  $n < 0$  entonces  $h[k]$  representa un sistema causal.

*Respuesta de un sistema lineal a cualquier señal discreta  $x[k]$*

*Suma de convolución*

Una vez estimada la respuesta al impulso unitario,  $h[k]$ , la salida del sistema para cualquier entrada  $x[k]$  estará dada por:

$$y[k] = \sum_{n=-\infty}^{\infty} x[n]h[k-n] \quad (2.15)$$

Se define a la suma de convolución por el operador \*, con esto la ecuación anterior se puede escribir de la siguiente forma:

$$y[k] = x[k] * h[k] \quad (2.16)$$

Se puede demostrar también que:

$$y[k] = x[k] * h[k] = h[k] * x[k] = \sum_{n=-\infty}^{\infty} h[n]x[k-n] \quad (2.17)$$

### *Respuesta de un sistema lineal a una exponencial compleja*

#### *Función de transferencia*

Es de interés lograr representar una secuencia de muestras, señal discreta, como una combinación lineal de exponenciales complejas para aproximar la señal discreta mediante una serie de Fourier.

A continuación se verá que la exponencial compleja es una función propia de los sistemas lineales.

Suponiendo que la entrada de un sistema lineal está dado por:

$$x[k] = z^k \quad , \text{donde} \quad z = e^{j\omega} \quad (2.18)$$

para esta entrada, la salida se puede calcular a partir de la convolución (2.17).

$$y[k] = \sum_{n=-\infty}^{\infty} h[n]x[k-n] = \sum_{n=-\infty}^{\infty} h[n]z^{k-n} = z^k \sum_{n=-\infty}^{\infty} h[n]z^{-n}$$

por lo que  $y[k]$  se puede escribir de la siguiente manera:

$$y[k] = H(z) z^k \quad (2.19)$$

donde  $H(z)$  se define como:

$$H(z) = \sum_{n=-\infty}^{\infty} h[n]z^{-n} \quad (2.20)$$

De la ecuación (2.19) se observa que si la entrada del sistema es una exponencial compleja, la salida es la misma exponencial multiplicada por la constante  $H(z)$ , que depende del valor de  $z$ .

$H(z)$  es el valor propio asociado a la función propia  $z^k$  del sistema. Al factor  $H(z)$  se le llama *función de transferencia del sistema*.

Existe una correspondencia entre la secuencia  $h[k]$  y la función  $H(z)$ ,

$$h[k] \leftrightarrow H(z)$$

que es válida para cualquier secuencia discreta  $x[k]$ , y su correspondiente función  $X(z)$ ,

$$x[k] \leftrightarrow X(z) \quad (2.21)$$

A la función  $X(z)$  se le llama transformada Z de la secuencia  $x[k]$  y se define como:

$$X(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n} \quad (2.22)$$

### *Teorema de Convolución*

Utilizando la definición de la transformada Z de una secuencia, se puede demostrar [GaRo75] fácilmente el siguiente Teorema:

Si las ecuaciones:

$$X(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n} \quad \text{y} \quad Y(z) = \sum_{n=-\infty}^{\infty} y[n]z^{-n}$$

son respectivamente, la transformada Z de la entrada  $x[k]$  y de la salida  $y[k]$  del sistema lineal con función de transferencia  $H(z)$ , se tiene entonces que:

$$Y(z) = H(z) X(z) \quad (2.23)$$

### *Respuesta en frecuencia de un sistema lineal*

La respuesta en frecuencia de un sistema lineal al impulso  $h[k]$ , se obtiene evaluando la función  $H(z)$  para  $z = e^{j\omega}$ ,

$$H(e^{j\omega}) = \sum_{n=-\infty}^{\infty} h[n]e^{-j\omega n} \quad (2.24)$$

puesto que,

$$H(e^{j\omega}) = |He^{j\omega}| e^{j\theta} h(\omega) \quad (2.25)$$

donde  $|He^{j\omega}|$  y  $\theta_H(\omega)$  son la magnitud y la fase, respectivamente, de la respuesta en frecuencia del sistema.

Si la excitación al sistema representado por  $h[k]$  es

$$x[k] = a e^{j(\omega k + \theta)}$$

donde  $a$  es una constante, se tendrá que:

$$y[k] = a|H(e^{j\omega})| e^{j[\omega k + \theta + \theta_h(\omega)]}$$

La expresión anterior indica que la respuesta en frecuencia del sistema es modificada cuando se tendrá una señal de entrada  $x[k]$ , cuyas características frecuenciales pueden ser definidas de la siguiente manera:

$$x(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n] e^{-j\omega n} = |X(e^{j\omega})| e^{j\theta_x(\omega)} \quad (2.26)$$

de la expresión anterior y evaluando la ecuación (2.23) para  $z=e^{j\omega}$ , se puede escribir:

$$Y(e^{j\omega}) = H(e^{j\omega}) X(e^{j\omega})$$

y ahora, reemplazando las ecuaciones (2.24) y (2.25) en la ecuación anterior, obtenemos

$$Y(e^{j\omega}) = |H(e^{j\omega})| |X(e^{j\omega})| e^{j[\theta_h(\omega) + \theta_x(\omega)]} \quad (2.27)$$

La función  $Y(e^{j\omega})$  también puede ser escrita como

$$Y(e^{j\omega}) = \sum_{n=-\infty}^{\infty} y[n] e^{-j\omega n} = |Y(e^{j\omega})| e^{j\theta_y(\omega)} \quad (2.28)$$

Igualando estas últimas dos ecuaciones, se tiene que la magnitud y la fase de  $Y(e^{j\omega})$  están dadas, respectivamente por:

$$|Y(e^{j\omega})| = |H(e^{j\omega})| |X(e^{j\omega})| \quad (2.29)$$

y

$$\theta_y(\omega) = \theta_h(\omega) + \theta_x(\omega) \quad (2.30)$$

Tanto la magnitud como la fase de  $Y(e^{j\omega})$  están representadas en función de la variable frecuencial  $\omega$ . Si se representan gráficamente  $|Y(e^{j\omega})|$  y  $|\theta_y(\omega)|$  contra la variable frecuencial se obtienen el espectro en frecuencia de magnitud y espectro en frecuencia de fase respectivamente, conocidos como la respuesta en frecuencia del sistema.

La noción de espectro puede ser utilizada para caracterizar gráficamente a cualquier señal en el dominio de la frecuencia, sin importar si está representada la señal como discreta o continua en el tiempo.

### *Estabilidad de los sistemas lineales*

A partir de la representación temporal un sistema lineal se dice estable, en el sentido de respuestas acotadas a entradas acotadas, si su respuesta a cualquier entrada acotada, es también acotada. Una secuencia  $x[k]$  será acotada, si existe un escalar finito  $M$ , tal que:

$$|x[k]| \leq M \text{ para todo } k \quad (2.31)$$

Estos es, que las magnitudes de los elementos de la señal nunca son mayores a un valor finito preestablecido.

Un sistema lineal caracterizado por su respuesta al impulso unitario  $h[k]$ , es estable en el sentido establecido, si, y sólo si:

$$\sum_{n=-\infty}^{\infty} |h[n]| < \infty \quad (2.32)$$

Dicho en otras palabras, un sistema lineal será estable si lo elementos de la secuencia  $h[k]$  tienden a cero, lo suficientemente rápido como  $k$  se aproxima a más y menos infinito.

Ahora bien, a partir de la representación frecuencial,  $H(z)$ , un sistema es estable si los polos del sistema están localizados dentro de  $|z| = 1$  [RoMu87].

### Filtros Digitales

Existen dos clases principales de filtros digitales: los filtros a respuesta impulsional finita (RIF) y los filtros a respuesta impulsional infinita (RII).

*Filtros o Sistemas RIF.* Un filtro RIF es aquel que tiene una respuesta al impulso que vale cero fuera de algún intervalo de tiempo finito. En general, si el sistema RIF es causal:

$$h[k] = 0 \quad \begin{cases} k < 0 \\ k \geq M \end{cases}$$

La suma de convolución de tal sistema se reduce a:

$$y[k] = \sum_{n=0}^{M-1} h[n] \times [k-n] \quad (2.33)$$

La secuencia  $y[k]$  depende únicamente del valor actual de la señal de entrada y de un número finito de valores pasados de la señal de entrada. La salida del sistema es la combinación lineal ponderada de la señal de entrada.

*Filtros o Sistemas RII.* En contraste con los filtros RIF, un sistema RII es aquel que tiene una respuesta al impulso de duración infinita.

$$y[k] = \sum_{n=-\infty}^{\infty} h[n] \times [k-n] \quad (2.34)$$

Es decir, la respuesta del filtro RII es una función de los valores presentes y pasados de la señal de entrada y de los valores pasados de las muestras de la señal de salida.

## 2.2 Técnicas de Procesamiento

La mayoría de las técnicas para el procesamiento de señales dan por hecho que la señal misma puede describirse en términos estadísticos. De la señal se obtiene parámetros importantes, tales como la media, desviación estándar, la autocorrelación y la distribución de potencia [IWZe81]. Algunos métodos asumen que los parámetros anteriormente citados no cambian durante ciertos intervalos de tiempo. Por tanto podemos dividir los métodos en que se basan la mayoría de las técnicas de procesamiento en dos grupos:

- Métodos Paramétricos
- Métodos No Paramétricos

Los métodos paramétricos parten de la idea de que la señal puede representarse como un modelo estocástico, envolviendo parámetros específicos.

Por otro lado, los métodos no paramétricos hacen suposiciones generales acerca de la señal analizada, especialmente su calidad estacionaria en pequeños intervalos [IWZe81].

El estimar la Densidad Espectral es en general importante para el procesamiento de una señal, pues arroja información de cómo se distribuye la potencia con respecto a la frecuencia. Estos es particularmente útil para cierto tipo de señales, por ejemplo, al analizar una señal de EEG durante el sueño da la posibilidad de asociar intervalos de la señal a los diferentes estados del sueño si se conoce esta distribución de la potencia. Esto debido a que los estados del sueño poseen características que tiene que ver con la frecuencia.

### 2.2.1 Transformada de Fourier Discreta (DFT), FFT y FHT

#### *Transformada de Fourier Discreta (DFT)*

Si una señal discreta en el tiempo es periódica la transformación apropiada es la transformada de Fourier discreta (DFT, *Discrete Fourier Transform*). Así, uno puede pensar en ésta como la versión discreta de las series de Fourier. Si bien la clase de señales discretas periódicas no parece ser de mucha utilidad, la DFT se vuelve tremendamente importante en el procesamiento digital de señales. La publicación de Cooley y Tukey [CoTu65] del algoritmo rápido para la DFT dio origen a un conjunto completamente nuevo de aplicaciones del procesamiento digital de señales. Este algoritmo se volvió conocido como la transformada rápida de Fourier (FFT *Fast Fourier Transform*) [HJBu84].

Hay tres razones principales para utilizar la DFT. Primero, puede ser eficientemente calculada. Segundo, tiene un gran número de aplicaciones incluyendo la aproximación para otras transformadas, diseño de filtros, y convolución rápida para el filtrado RIF. Tercero, es la única transformada de Fourier que puede ser finitamente parametrizada. Uno puede analizar la DFT y todas sus propiedades usando álgebra matricial.

Sea  $f(k)$  una señal discreta periódica de periodo  $N$ , y sea  $W_N$ :

$$W_N = e^{j2\pi/N} \quad (2.35)$$

La DFT se define por

$$F(n) = \sum_{k=0}^{N-1} f(k)W_N^{-kn} \quad (2.36)$$

La sumatoria se realiza sobre un periodo de la señal, dado que la señal es periódica en el tiempo y está caracterizada por sus valores sobre un periodo. Los valores de  $F(n)$  también se repiten dado que:

$$\begin{aligned} W_N^N &= 1, & \text{y por ende} \\ F(n+N) &= \sum_{k=0}^{N-1} f(k)W_N^{-kn} [W_N^{-N}]^k \\ &= \sum_{k=0}^{N-1} f(k)W_N^{-kn} = F(n) \end{aligned}$$

Así, tanto  $f(k)$  como  $F(n)$  están caracterizadas por  $N$  valores cada una. Por esto, se introducirá la siguiente notación:

$$\mathbf{f} = \begin{bmatrix} f(0) \\ \vdots \\ f(N-1) \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} F(0) \\ \vdots \\ F(N-1) \end{bmatrix} \quad (2.37)$$

El elemento sencillo en negrita  $\mathbf{f}$  significa cualquiera de los vectores  $N$ -dimensionales en las ecuaciones (2.37) o la señal periódica completa, dependiendo del contexto.

En forma matricial, la DFT en la ecuación (2.36) se vuelve

$$\mathbf{F} = \mathbf{V} \mathbf{f} \quad (2.38)$$

donde  $\mathbf{V}$  es la matriz  $N \times N$  con elementos

$$V_{nk} = W_N^{-kn}, \quad 0 \leq k, n \leq N-1 \quad (2.39)$$

$\mathbf{V}$  es la representación matricial de la DFT.

De entre las propiedades de la DFT se encuentran la ortogonalidad y la propiedad de desplazamiento. Así mismo, como ya se mencionó anteriormente, la DFT da una aproximación a la transformada de Fourier [RoMo87].

### *Transformada Rápida de Fourier (FFT)*

Uno de los aspectos importantes de la teoría del análisis de Fourier de sistemas y señales discretas en el tiempo es el cálculo real asociado con la DFT, dado que ésta es la transformada

que en la práctica se calcula. En este apartado se explicará la FFT, que es parte de los globalmente llamados "algoritmos rápidos" para la DFT y la convolución.

Recordando que la DFT de la secuencia  $\{x(k)\}$ ,  $k=0, 1, \dots, N-1$  está dada por

$$X(n) = \sum_{k=0}^{N-1} x(k) W_N^{-nk} \quad , n = 0, 1, \dots, N-1 \quad (2.40)$$

donde

$$W_N = e^{j\frac{2\pi}{N}} \quad (2.41)$$

La DFT inversa (o IDFT) del espectro  $\{X(n)\}$ ,  $n=0, 1, \dots, N-1$  está dada por

$$X(k) = \frac{1}{N} \sum_{n=0}^{N-1} X(n) W_N^{nk} \quad , k = 0, 1, \dots, N-1 \quad (2.42)$$

En notación matricial, la secuencia  $\{X(n)\}$  está dada por

$$\mathbf{X} = \mathbf{V} \mathbf{x} \quad (2.43)$$

donde  $\mathbf{V}$  es la matriz DFT dada por

$$\mathbf{V} = [V_{nk}] \quad n, k = 0, 1, \dots, N-1 \quad (2.44)$$

teniendo elementos

$$V_{nk} = W_N^{-nk} = e^{-j\frac{2\pi}{N}nk} \quad (2.45)$$

En general, el cálculo de  $\mathbf{X}$  a partir de  $\mathbf{x}$  en la ecuación (2.43) requiere  $N^2$  multiplicaciones dado que  $\mathbf{V}$  es una matriz  $N \times N$ . Sin embargo, por el carácter de  $W_N^{-nk}$ , hay simetrías contenidas en  $\mathbf{V}$  que reducen dramáticamente las  $N^2$  multiplicaciones implicadas por la ecuación (2.43). El elemento en el  $n$ -ésimo renglón y la  $k$ -ésima columna de  $\mathbf{V}$  es la  $n$ -ésima raíz de unidad que se da de la potencia  $-nk$ . Porque  $W_N^{-nk}$  es idéntico a  $W_N^{-l}$  donde,

$$l = nk \bmod N \quad (2.46)$$

Usando la ecuación (2.46) en el caso de  $N=8$ , por ejemplo, resulta en una matriz  $\mathbf{V}$  con potencias de  $W_N^{-1}$  dadas en la siguiente matriz  $\mathbf{P}$ :

$$\mathbf{P} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 0 & 2 & 4 & 6 & 0 & 2 & 4 & 6 \\ 0 & 3 & 6 & 1 & 4 & 7 & 2 & 5 \\ 0 & 4 & 0 & 4 & 0 & 4 & 0 & 4 \\ 0 & 5 & 2 & 7 & 4 & 1 & 6 & 3 \\ 0 & 6 & 4 & 2 & 0 & 6 & 4 & 2 \\ 0 & 7 & 6 & 5 & 4 & 3 & 2 & 1 \end{bmatrix} \quad (2.47)$$

Los elementos en  $\mathbf{P}$  son valores de  $l$  de la ecuación (2.46) requeridos en la matriz  $\mathbf{V}$  de la DFT para calcular una DFT con  $N=8$ .

En si se considera a los algoritmos rápidos para la DFT como una factorización de la matriz  $\mathbf{V}$  de la DFT.

#### Algoritmos para la FFT Potencia-de-2

Suponiendo que el número de puntos dato en  $\mathbf{x}$  es una potencia de 2, por ejemplo,  $N = 2^y$ . Lo siguiente es una simple derivación de un algoritmo rápido para la DFT llamado transformada rápida de Fourier por *decimalización-en-tiempo* (FFT).

Si se divide el índice original de datos en índices impares y pares (dado que  $N$  es divisible por 2). Se escribe la DFT, ecuación (2.40), en la forma

$$\begin{aligned} X(n) &= \sum_{k=0}^{N-1} [x(2k)W_N^{-2kn} + x(2k+1)W_N^{-(2k+1)n}] \\ &= \sum_{k=0}^{N-1} x(2k)W_N^{-2kn} + W_N^{-n} \sum_{k=0}^{N-1} x(2k+1)W_N^{-2kn} \end{aligned} \quad (2.48)$$

para  $n = 0, 1, \dots, N-1$ . Ahora  $W_N^{-2kn}$  puede ser escrito en la forma

$$W_N^{-2kn} = \left[ e^{j\frac{2\pi}{N}} \right]^{-2kn} = \left[ e^{j\frac{2\pi}{N/2}} \right]^{-kn} = W_{N/2}^{-kn} \quad (2.49)$$

La ecuación (2.49) es un paso clave pues permite escribir la ecuación (2.48) como

$$X(n) = G(n) + W_N^{-n}H(n) \quad , n = 0, 1, \dots, \frac{N}{2} - 1 \quad (2.50)$$

Aquí  $G(n)$  y  $H(n)$  son DFT de  $N/2$  puntos. Estas DFT de  $N/2$  puntos son de periodo  $N/2$ . Esto permite calcular  $X(n)$  para  $n = N/2, N/2+1, \dots, N-1$  por medio de

$$X(n) = G(n - \frac{N}{2}) + W_N^{-n}H(n - \frac{N}{2}) \quad , n = \frac{N}{2}, \frac{N}{2} + 1, \dots, N - 1 \quad (2.51)$$

Se puede ver la descomposición en las ecuaciones (2.50) y (2.51) esquemáticamente como se muestra en la figura 2.2 (asumiendo  $N=2^3=8$ ). Se necesita sólo combinar  $G(n)$  y  $H(n)$  usando los pesos  $W_N^n$  apropiados o "valores twiddle".

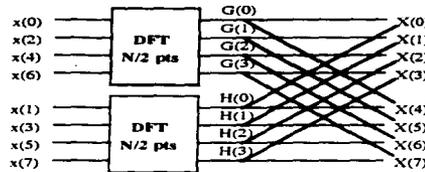


Figura 2.2. Primera etapa de descomposición-en-tiempo de una FFT con  $N=2^3$ -puntos.

¿Cuántos cálculos se requieren usando una etapa de descomposición? Cada DFT de  $N/2$ -puntos requiere  $(N/2)^2$  multiplicaciones. Multiplicaciones por  $W_N^n$  en las ecuaciones (2.50) y (2.51) requieren  $N$  multiplicaciones. Así que el total es  $N + 2(N/2)^2 = N + (N^2/2)$  multiplicaciones contra los  $N^2$  cálculos de la DFT original.

Dado que  $N$  es potencia de 2, se puede repetir el proceso de descomposición para las secuencias  $g$  y  $h$ , donde,

$$g(l) = x(2l), \quad h(l) = x(2l + 1), \quad l = 0, 1, \dots, N/2-1 \quad (2.52)$$

Así queda  $G(n)$  en la forma

$$\begin{aligned} G(n) &= \sum_{k=0}^{N/2-1} g(k)W_{N/2}^{-kn} = \sum_{k=0}^{N/4-1} [g(2k)W_{N/2}^{-2nk} + g(2k+1)W_{N/2}^{-(2k+1)n}] \\ &= \sum_{k=0}^{N/4-1} g(2k)W_{N/2}^{-kn} + W_{N/2}^{-n} \sum_{k=0}^{N/4-1} g(2k+1)W_{N/4}^{-nk} \\ &= \begin{cases} R(n) + W_{N/2}^n S(n), & n = 0, 1, \dots, \frac{N}{4} - 1 \\ R(n - \frac{N}{4}) + W_{N/2}^n S(n - \frac{N}{4}), & n = \frac{N}{4}, \frac{N}{4} + 1, \dots, \frac{N}{2} - 1 \end{cases} \end{aligned} \quad (2.53)$$

( $H(n)$  puede expresarse análogamente).  $G(n)$  está ahora expresada como una sumatoria de dos DFT de  $N/4$ -puntos. El grafo de flujo de la señal para calcular  $G(n)$  se muestra en la figura 2.3 para el caso de  $N = 8$ . Note que se uso de nuevo la relación  $W_{N/2}^{-2nk} = W_{N/4}^{-nk}$ .

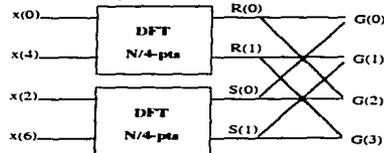


Figura 2.3. Una porción de la 2a descomposición para la FFT por descomposición-en-tiempo de potencias-de-2.

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

¿Cuántos cálculos están involucrados usando descomposición en dos etapas? Cada DFT de  $N/4$ -puntos requiere  $(N/4)^2$  multiplicaciones, y hay cuatro de ellas. La multiplicación por  $W_{N/2}^{-n}$ ,  $n = 0, 1, \dots, N/2-1$  para  $G(n)$  y  $H(n)$  resulta en  $N$  multiplicaciones. Combinando  $G$  y  $H$  para formar  $X$  también requiere  $N$  multiplicaciones. El total es entonces  $2N + 4(N/4)^2$  contra  $N^2$ . se puede continuar este proceso para  $\log_2 N = \log_2 2^v = v$  etapas. La etapa final es una DFT de dos-puntos de la forma mostrada en la figura 2.4 y no requiere multiplicaciones. Así que las únicas multiplicaciones requeridas son para combinar las DFT de más-pequeños-puntos. Hay  $\log_2 N$  etapas, cada una requiere  $N$  multiplicaciones para un total de  $N \log_2 N$  multiplicaciones para la FFT de potencias-de-2. (Estas multiplicaciones son complejas y corresponden a cuatro multiplicaciones reales).

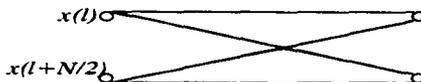


Figura 2.4. Grafo de flujo de señal para una DFT dos-puntos.

Se puede construir un grafo de flujo de señal para el caso  $N = 2^3$ . No es difícil deducir la matriz de factorización de  $V$  que este algoritmo implica, equivalente al grafo citado. Tal factorización se da en la ecuación (2.54). La matriz  $E$  representa las permutaciones de los datos de entrada necesarias para la multiplicación de la matriz siguiente, que consiste de 4 DFT 2-puntos contenidas en  $V_{2T}$ .

$$\begin{aligned}
 V &= V_{8T} V_{4T} V_{2T} E \\
 &= \begin{bmatrix} 1 & 0 & 0 & 0 & W_8^0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & W_8^{-1} & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & W_8^{-2} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & W_8^{-3} \\ 1 & 0 & 0 & 0 & W_8^{-4} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & W_8^{-5} & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & W_8^{-6} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & W_8^{-7} \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & W_8^{-0} & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & W_8^{-2} & 0 & 0 & 0 & 0 \\ 1 & 0 & W_8^{-4} & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & W_8^{-6} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & W_8^{-0} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & W_8^{-2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & W_8^{-4} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & W_8^{-6} \end{bmatrix} \\
 &\cdot \begin{bmatrix} 1 & W_8^0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & W_8^{-1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & W_8^0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & W_8^{-1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & W_8^0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & W_8^{-1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & W_8^0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & W_8^{-1} \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.54)
 \end{aligned}$$

Las matrices  $\mathbf{V}_{8T}$ ,  $\mathbf{V}_{4T}$ ,  $\mathbf{V}_{2T}$  representan las tres etapas de los llamados cálculos mariposa (*butterfly*). Note que cada columna contiene sólo dos multiplicaciones no cero, lo que implica que cada salida es ejecutada de sólo dos valores de entrada como se mostró en la figura 2.4. Hay  $NW_N^n$  factores multiplicando en cada componente de la matriz y un total de  $\log_2 N$  etapas, resultando en  $N \log_2$  multiplicaciones complejas ( $\mathbf{V}_{2T}$  contiene sólo  $\pm 1$ 's y 0's).

### Transformada Rápida de Hartley (FHT)

La transformada rápida de Hartley (FHT, *Fast Hartley Transform*) es similar a la FFT de Cooley-Tukey pero su desempeño es más rápido porque requiere sólo cálculos de aritmética real, comparados con los cálculos de aritmética compleja requeridos por la FFT. Se puede decir que la FHT es dos veces más rápida que la FFT y utiliza sólo la mitad de los recursos de cómputo.

La transformada discreta de Hartley (DHT) usa la variable real  $\cos(2\pi kn/N) + \text{sen}(2\pi kn/N)$  como kernel de la transformación, mientras que la DFT utiliza exponenciales complejas como kernel de la transformación. Esto es, realiza un mapeo de la función real en el tiempo  $x(kt)$ , a una función real en la frecuencia  $\mathbf{H}(k\omega)$ , mientras que la transformada de Fourier ejecuta un mapeo de una señal real en el tiempo  $x(kt)$ , a una señal compleja en la frecuencia  $\mathbf{F}(k\omega)$ . Intuitivamente la DHT es más simple (no es compleja, sino real) y por ende más rápida que la FFT, dado que la aritmética compleja implica al menos el doble de operaciones de la aritmética real (una suma o resta compleja son dos operaciones reales y una multiplicación o división compleja son cuatro operaciones). Más aún, los arreglos reales requieren sólo la mitad de memoria que los arreglos complejos. Esto significa que la transformada de Hartley requerirá considerablemente menos memoria para un conjunto de datos que la transformada de Fourier. Se presentara el atractivo algoritmo recursivo llamado FHT. Este algoritmo es una generalización del algoritmo FFT de Cooley-Tukey.

### Definición de la DHT

De acuerdo a la definición de la DHT [Brac83], [Brac84], para una secuencia de datos dada  $\{x_n; n = 0, 1, 2, \dots, N-1\}$ , la secuencia de datos DHT  $\{y_k; k = 0, 1, 2, \dots, N-1\}$  está dada por la siguiente relación

$$y_k(N, x) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x_n [\cos(2\pi kn / N) + \text{sen}(2\pi kn / N)] \quad (2.55)$$

para  $k = 0, 1, 2, \dots, N-1$

y dicha transformación es llamada la DHT. Escrita en forma matricial, (2.55) se vuelve

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_{N-1} \end{bmatrix} = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \begin{bmatrix} \cos(2\pi kn/N) \\ + \\ \sin(2\pi kn/N) \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{N-1} \end{bmatrix} \quad (2.56)$$

Como se ve en (2.55) el cálculo de la DHT presenta un problema análogo al cálculo de la DFT. Esto es, se tiene efectuar  $N^2$  operaciones aritméticas para calcular la DHT de un conjunto de datos de  $N$  elementos.

### El Algoritmo Rápido

En esencia, la FFT usa un proceso de permutaciones para bisectar los datos hasta que se llega a pares de datos. Calcular la transformada de tales pares de datos es trivial y rápido. La idea atrás del proceso de permutación es que es más fácil separar los datos en pares, calcular la transformada de los pares, y recombinar esos pares para obtener la transformada completa, en vez de calcular la transformada del conjunto de datos completo. La permutación es particularmente rápida cuando la cantidad de datos es grande. Si se superponen tales pares de elementos usando el proceso *butterfly* (llamado así por la apariencia del diagrama de flujo de datos), se puede calcular la transformada de Fourier del conjunto de datos de entrada.

Ahora bien, Bracewell [Brac83] ha demostrado que se puede emplear una metodología similar en el caso de la transformada de Hartley. De nuevo, se utilizan permutaciones para bisectar los datos hasta que se obtienen pares de ellos. La transformada Hartley de un par de datos  $(a, b)$  es  $\frac{1}{2}(a + b, a - b)$ , y el cálculo de tal par es trivial. También se puede sobreponer esta secuencia de dos elementos para la transformada Hartley del conjunto de datos de entrada. Sin embargo, para hacer eso se necesita una fórmula que exprese una transformada Hartley completa (DHT) en términos de sus subsecuencia de media longitud  $(N/2)$ .

Bracewell muestra que, aplicando los teoremas de desplazamiento y similitud, la ecuación (2.57) expresa la fórmula de la descomposición general para la DHT. Esta fórmula genera la deseada DHT bisectando los datos:

$$H(k) = H_1(k) = H_2(k) \cos(2\pi kn / N_s) + H_2(N_s - kn) \sin(2\pi kn / N_s) \quad (2.57)$$

donde  $N_s$  es el número de elementos de la mitad de la secuencia,  $N_s = N/2$  para un conjunto de datos de  $N$  elementos.

La fórmula de descomposición para la FHT difiere de la FFT en un aspecto importante: Los elementos multiplicados por los términos trigonométricos no son simétricos. Esta asimetría se manifiesta cuando se expresa la transformación en forma matricial [HouH87],[Brac86]: Los términos de la matriz FFT son simétricos con respecto a la diagonal principal, mientras que los correspondientes términos de la FHT son asimétricos. Esto introduce problemas de cálculo, porque el procesamiento de una matriz asimétrica es difícil de implantar. Se puede tratar esta asimetría usando una variable independiente como índice para los elementos multiplicados por

los coeficientes seno. Este índice decrece mientras los otros índices crecen; este comportamiento es llamado indexamiento retrógrado.

### *FHT Inversa*

Una característica sobresaliente de la FHT es que se puede obtener la FHT inversa aplicando el algoritmo FHT a la salida dada, lo que regenera los datos originales de entrada. Esto significa que se puede usar el mismo código de programa para calcular la transformada directa y la inversa. Hay, sin embargo, una pequeña asimetría entre la FHT y su inversa. En el caso de la transformada directa se debe escalar el resultado del cálculo *butterfly*. Es decir, se debe dividir la salida del *butterfly* por N para obtener la transformada de Hartley. Esto no se necesita hacer en la transformación inversa, consecuentemente, el cálculo *butterfly* constituye la transformación inversa. Es trivial codificar<sup>4</sup> el control del algoritmo para que realice o no dicho escalamiento.

### *Comparando la FFT con la FHT*

Se puede usar la FHT para muchas de las aplicaciones para las cuales se usa la FFT. Esto incluye la convolución, deconvolución, generación de espectro de potencia. También se puede obtener la Transformada de Fourier de la FHT, de hecho, es a menudo más rápido generar la transformada de Fourier y el espectro de potencia con la FHT que con la FFT. Se pueden construir la parte real y la imaginaria de la FFT usando las siguientes ecuaciones:

$$F(k)_{\text{real}} = H(k) + H(N-k) \quad \text{y} \quad F(k)_{\text{imag}} = H(k) - H(N-k).$$

Igualmente se puede obtener la convolución, el espectro de potencia y la transformada coseno partiendo de la secuencia entregada por la FHT. [HouH87], [O'Ne88].

El uso y desarrollo de algoritmos rápidos para la DFT y otras transformadas discretas es un área de aplicación e investigación grande y creciente. Los algoritmos rápidos han llevado a muchas aplicaciones en campos como el análisis espectral, filtrado digital, procesamiento de datos de radar y sonar, procesamiento de datos biomédicos y procesamiento sísmico.

El desarrollo de la FFT y algoritmos relacionados con ella han hecho la formulación de filtros digitales en el dominio de la frecuencia competitivos con el acercamiento en el dominio temporal. La FFT ha también logrado eficientar el cálculo de la correlación y el análisis espectral, que tiene aplicación en diversos campos de la ciencia y de la ingeniería. Así, la DFT (y los algoritmos rápidos para la DFT) son extremadamente poderosos y útiles como herramientas en el procesamiento digital de señales.

---

<sup>4</sup> En el Apéndice 4 se lista un programa en object pascal que contiene esta FHT. La unidad se llama spectrum y la función que realiza el algoritmo se llama Hartley.

### 2.2.2 La Función de Correlación

La correlación es una cantidad estadística útil para saber cuál es el comportamiento de la señal en diferentes tiempos. Si se calcula la correlación de la señal con ella misma se llama la autocorrelación y es una descripción de un proceso aleatorio que expresa estadísticamente la relación lineal entre las amplitudes de la señal en dos instantes. Es también una indicación de qué tan rápidamente están ocurriendo variaciones en el proceso. La autocorrelación se define como:

$$R_{XX}(t_1, t_2) = E\{x(t_1)x^*(t_2)\} \quad (2.58)$$

Siendo  $x(t_1)$  y  $x(t_2)$  funciones del proceso aleatorio en los instantes  $t_1$  y  $t_2$  respectivamente. Si se toma  $t_2 = t_1 + \tau$  y el proceso es estacionario, el valor esperado es el mismo para cualquier valor de  $t_1$  y sólo depende de  $\tau$ .

$$R_{XX}(\tau) = E\{x(t)x^*(t+\tau)\} \quad (2.59)$$

Los valores de  $R_{XX}(\tau)$  son usualmente estimados a partir de datos muestreados y son obtenidos por promedio. Por lo tanto, si la señal se muestrea a  $f_s = 1/\Delta t$  un estimador de la correlación estaría dado por:

$$\hat{R}_{xx}(k \Delta t) = \frac{1}{N-k} \sum_{i=1}^{N-k} x_i x_{i+k} \quad (2.60)$$

A la expresión anterior se le llama estimador insesgado de  $R_{XX}(k\Delta t)$  [McAu81]. Sin embargo, no es el más usado. El mejor estimador es aquel que minimiza el error cuadrático medio [McAu81], como el siguiente:

$$\hat{R}_{xx}(k \Delta t) = \frac{1}{N} \sum_{i=1}^{N-k} x_i x_{i+k} \quad (2.61)$$

Este estimador tiene poco sesgo pero su error cuadrático medio (ECM) es reducido. Si se multiplica (2.61) por (2.62) se obtiene (2.60):

$$W_1 = \frac{N}{N-k} \quad (2.62)$$

Logrando con esto una disminución del sesgo del estimador.

### 2.2.3 Estimación de la Densidad Espectral

Si se desea saber cuál es el contenido frecuencial de una señal, se estará formulando uno de los cuestionamientos que más se presentan en aplicaciones de procesamiento de señales. A menudo se pregunta esto cuando sólo una parte pequeña de la señal está disponible, lo que significa que no puede ser respondida correctamente. *Estimación espectral* es el nombre que se le da a la colección de métodos que suministran respuestas imperfectas a esta pregunta.

Se definirá primero qué significa "contenido frecuencial". Para una banda dada de frecuencias  $\Omega$ , se puede aislar la parte de la señal que se encuentra en la banda  $\Omega$  haciendo pasar la señal por un filtro ideal paso banda, filtro cuya banda de paso es  $\Omega$ .

$$x \xrightarrow{H(z)} y \quad (2.63)$$

$$H(e^{j\omega}) = \begin{cases} 1 & , \omega \in \Omega \\ 0 & , \omega \notin \Omega \end{cases} \quad (2.64)$$

La señal de salida  $y$  en este experimento idealizado es lo que quiere decir la frase "la parte de  $x$  que se encuentra en la banda  $\Omega$ ". Si  $x$  tiene energía finita, entonces tal energía es "la energía de  $x$  contenida en la banda  $\Omega$ ". Si  $x$  y  $y$  son estacionarias en sentido amplio<sup>5</sup> con potencia finita, entonces la pot. promedio de  $y$  es "la potencia que  $x$  contiene en la banda  $\Omega$ ".

Se tendrá una respuesta completa al cuestionamiento acerca del contenido frecuencial, si se pueden calcular estas cantidades para cualquier  $\Omega$  dada. Este es el propósito de la *función de densidad espectral*, que es una función de la variable frecuencial  $\omega$  con la siguiente propiedad:

La potencia de la señal en la banda  $\Omega$  es la integral de la función de densidad en  $\Omega$ .

Con base en esta propiedad se construirá la función de densidad. Hay dos clases de señales de interés, señales de energía finita, y señales aleatorias estacionarias en sentido amplio. El primer tipo tendrá *densidad espectral de energía*. El segundo tipo tendrá *densidad espectral de potencia*.

### Densidad Espectral de Energía (DEE)

Se asume que la señal

$$x(k) \xleftrightarrow{\text{TFTD}} X(e^{j\omega}) \quad (2.65)$$

tiene energía finita

$$\|x\|^2 = \sum_{-\infty}^{\infty} |X(k)|^2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} |X(e^{j\omega})|^2 d\omega \quad (2.66)$$

Para extraer la porción de energía que está en la banda  $\Omega$ , se hace pasar  $x$  por el filtro ideal paso banda  $H(z)$  de la ecuación (2.64) para obtener una salida  $y$ . Luego la energía total de  $y$  es la parte de la energía de  $x$  que está en  $\Omega$  y será:

$$\|y\|^2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} |X(e^{j\omega})|^2 d\omega \quad (2.67)$$

<sup>5</sup> Cualquier secuencia aleatoria que satisface las siguientes tres condiciones:

$$\begin{aligned} \mu &= E[u(k)] \quad , \forall k \\ r_{uu}(l) &= E[u(k+l)u(k)] \quad , \forall k \\ r_{uu}(0) &= E[u^2(k)] < \text{infinito} \end{aligned}$$

pero

$$y = h * X \xrightarrow{\text{TFTD}} Y(e^{j\omega}) = H(e^{j\omega})X(e^{j\omega}) \quad (2.68)$$

y usando la ecuación (2.64), la energía de  $x$  en  $\Omega$  es:

$$\|y\|^2 = \frac{1}{2\pi} \int_{\Omega} |X(e^{j\omega})|^2 d\omega \quad (2.69)$$

Esta es la propiedad de la función de densidad que se necesita, y por tanto, la función  $|X|^2$  es la densidad espectral de energía para la señal  $x$ .

### Densidad Espectral de Potencia (DEP)

Una señal estacionaria en sentido amplio no trivial no tendrá energía finita, pero puede tener potencia promedio finita. Si una señal  $x$ , estacionaria en sentido amplio con media cero, es ergódica, los promedios de tiempo se acercarán a valores esperados. En particular,

$$\text{potencia promedio} = \lim_{L \rightarrow \infty} \frac{1}{L} \sum_{k=0}^{L-1} |x(k)|^2 = E|x(k)|^2 \quad (2.70)$$

Así, la potencia promedio es la varianza de  $x(k)$ , o el valor inicial (cero-ésimo) de la secuencia de autocorrelación

$$r_{xx}(k) = E x(k+l)x^*(l) \xrightarrow{\text{TFTD}} S_{xx}(\omega) \quad (2.71)$$

Para hallar la potencia promedio que está en la banda  $\Omega$ , se pasa la señal  $x$  a través del filtro paso banda  $H(z)$  de la ecuación (2.64). El teorema Wiener-Khintchine [RoMo87] establece:

$$r_{yy}(k) \xrightarrow{\text{TFTD}} S_{yy}(\omega) = S_{xx}(\omega) |H(e^{j\omega})|^2 \quad (2.72)$$

por lo tanto, la potencia promedio en  $x$  en la banda  $\Omega$  es igual a:

$$r_{yy}(0) = \frac{1}{2\pi} \int_{-\pi}^{\pi} S_{yy}(\omega) d\omega \quad (2.73)$$

$$r_{yy}(0) = \frac{1}{2\pi} \int_{\Omega} S_{xx}(\omega) d\omega \quad (2.74)$$

Por lo tanto,  $S_{xx}(\omega)$ , que es la TFTD de la secuencia de autocorrelación para  $x$ , es la función de densidad espectral. Es decir, se puede obtener la potencia promedio en la banda  $\Omega$  integrando esta función en  $\Omega$ .

Los modelos de espectro de señales más populares son los de ruido blanco, ARMA y líneas, o una combinación lineal de ellos [RoMo87]. En particular, el caso de espectro de línea más ruido blanco ha sido un popular modelo para muchas aplicaciones.

### Acercamientos a la estimación espectral

Sea  $y$  una señal ESA<sup>6</sup> con DEP  $S(\omega)$ . Suponiendo que una porción de esta señal ha sido

<sup>6</sup> ESA: Estacionaria en Sentido Amplio

grabada, con  $L$  muestras.

$$\mathbf{y}[0, L-1] = \{y(0), y(1), \dots, y(L-1)\} \quad (2.75)$$

Esta parte de la señal es llamada los "datos". El problema es entonces estimar la función  $S(\omega)$  dados los datos. Así, un *estimador* mapéa los datos al espectro de potencia:

$$\hat{S}(\omega) = \hat{\Delta}(\omega; \mathbf{y}[0, L-1]) \quad (2.76)$$

Aquí,  $\hat{\Delta}$  es llamado el estimador y  $\hat{S}$  es llamado el estimado. Dado que los datos consisten de variables aleatorias, el estimado es aleatorio. Se debe juzgar el estimador con esto en mente. A continuación se consideran varios detalles de el problema de la estimación.

### *Estimación usando la DEE*

Conociendo sólo los datos  $\mathbf{y}[0, L-1]$ , Se puede construir una señal de energía finita extendiendo los datos con ceros. Sea

$$w(k) = \begin{cases} 1 & , 0 \leq k \leq L-1 \\ 0 & , \text{en otro caso} \end{cases} \quad (2.77)$$

y

$$\mathbf{x}(k) = w(k)y(k) \quad , -\infty < k < \infty \quad (2.78)$$

Entonces  $\mathbf{y}$  es una señal de potencia y energía infinita,  $\mathbf{w}$  es una ventana limitada en tiempo (ventana uniforme), y  $\mathbf{x}$  es una señal limitada en tiempo. Dado que  $\mathbf{x}$  debe tener energía finita, tiene densidad espectral de energía. Se usa esta DEE para estimar la DEP de  $\mathbf{y}$ , con el escalamiento apropiado. El estimado

$$\hat{S}(\omega) = \frac{1}{L} |\mathbf{X}(e^{j\omega})|^2 \quad (2.79)$$

es el resultado, y es llamado *periodograma* en honor a Schuster (1898).

### *Estimación indirecta con los valores de la autocorrelación*

Se da a menudo el caso de que los elementos de bajo orden de la secuencia de autocorrelación son estimados independientemente. Un estimado tal es

$$\hat{r}(k) = \frac{1}{L} \sum_{l=0}^{L-1-k} y(k+l)y(l) \quad , 0 \leq k < L \quad (2.80)$$

Suponiendo luego que  $\hat{r}(0)$  hasta  $\hat{r}(n)$  han sido obtenidos ( $n < L$ ). Se puede hallar un espectro único de bajo orden en un espectro de clase paramétrico que concuerde con estos valores. Tal espectro se convierte en el estimado.

Así, por ejemplo, si es espectro se asume que es MA, entonces se debe tomar el estimado de Blackman-Tukey. Que es el espectro MA de más bajo orden que concuerda con  $\hat{r}(0)$  hasta  $\hat{r}(n)$ .

Si el espectro es asumido que es AR, entonces se toma

$$\hat{S}(\omega) = \frac{\alpha}{\left| \sum_{k=0}^n a_k e^{-jk\omega} \right|^2}, \quad a_0 = 1 \quad (2.81)$$

donde  $(\alpha, a_1, \dots, a_n)$  son obtenidos de  $(\hat{r}(0)$  hasta  $\hat{r}(n)$ ) por el algoritmo de Levinson [RoMo87]. Este espectro AR reproducirá los valores dados de la secuencia de autocorrelación. Denominándose esto como el *estimado de Levinson*.

Existe un espectro único que consiste de ruido blanco de la varianza apropiada, además de a lo mucho  $n$  líneas en el intervalo  $-\pi$  a  $\pi$ . Este es llamado el *estimado Pisarenko*.

Cada estimación del espectro producido por estos medios debe corresponder a los valores de  $\hat{r}(k)$  para  $0 \leq k \leq n$ . Condiciones que no siempre es posible cumplir.

### *Ajuste de datos sinusoidales*

Si se asume que el espectro es de línea, entonces los datos son paramétricos, así como el espectro. La señal y estará caracterizada por tres parámetros: frecuencia, fase y amplitud. El problema es entonces estimar estos parámetros dados los datos. La forma clásica de hacer esto es ajustar un problema de mínimos cuadrados Gaussianos:

Dado  $y[0, L-1]$  y  $m$ , minimizar

$$V(A_1, A_2, \dots, A_m, \omega_1, \omega_2, \dots, \omega_m) = \sum_{k=0}^{L-1} \varepsilon(k)^2$$

donde

$$\varepsilon(k) = y(k) - \sum_{i=1}^m A_i \cos(k\omega_i + \phi_i) \quad (2.82)$$

La función  $V$  es una suma de cuadrados y es por ello no negativa. Si ésta vale cero, entonces todos los errores  $\varepsilon(k)$  deben ser cero, y se han ajustado los datos exactamente. No se espera que esto pase, pero para la minimización de los valores los errores deben ser pequeños. Luego se reescribe la ecuación (2.82) para modelar los datos como una sinusoidal más ruido blanco (la secuencia de error se aproximará a ruido blanco).

### *Características de una buena estimación*

¿Cuáles son las fortalezas y debilidades de un acercamiento dado? Se muestran aquí una serie de características generalmente aceptadas en relación a la calidad de un estimador.

**Positividad.** Se refiere a que el estimador tenga la propiedad que se su estimado sea siempre no negativo.

**Carga.** Un estimado está "cargado" si sus valores esperados difieren de los valores reales, y la "carga" del estimado es la diferencia entre estos dos.

**Varianza.** La varianza de un estimador es una función de la frecuencia.

**Consistencia.** Un estimador es consistente si fielmente reproduce el espectro real cuando se le da una cantidad infinita de datos.

**Estabilidad.** Esta característica sólo se presenta en el acercamiento de análisis/síntesis de estimación del espectro. Se requiere que el modelo de síntesis sea estable para que el estimado sea apropiado.

**Computación.** Esto se refiere no tanto con la calidad del estimado, sino con lo caro en cuestión de cálculo que resulta obtenerlo. Es decir cual es más fácil de calcular.

Con esto se concluye la exposición de la teoría básica para el procesamiento digital de señales. Se ha dicho ya que el PDS es ampliamente utilizado para resolver problemas de diversa índole. Al diseñar el sistema propuesto, se pondrán en práctica algunos de los conceptos descritos a lo largo del presente capítulo.

## Capítulo 3

### Diseño del Sistema

Durante el desarrollo de este capítulo se explica la etapa de diseño del sistema de procesamiento digital de señales electroencefalográficas. Se ha dividido este capítulo en dos partes: diseño del hardware y diseño del software; esto con objeto de resaltar cada una de ellas y describirlas con la profundidad que merecen. En la primera parte de este capítulo se describe la solución propuesta para traer la señal generada en el cerebro hasta una computadora y dejarla lista para ser procesada; se ha dividido la explicación del diseño de hardware en etapas funcionales como son: registro de la señal, amplificación de la señal, filtrado de la señal y adquisición de datos; todo esto con el fin de darle mayor claridad a esta parte. En la segunda parte se aborda el tema del diseño del software; a partir de la señal que se tiene digitalizada, se propone aquí una solución que satisfaga los requerimientos que se plantean desde un enfoque de ingeniería de programación, partiendo de un diseño conceptual, donde se establece que se desea; proponiendo luego un diseño técnico, o sea los pasos con los que se solucionara cada procedimiento y su estructura jerárquica; y finalmente se presenta un diseño de los programas es decir una aproximación en *pseudocódigo* a lo que será la codificación final.

### 3.1 Diseño del Hardware

El diseño del hardware comprende la especificación del sistema propuesto para las etapas de registro, condicionamiento de la señal y adquisición de datos, como se muestra en el diagrama de la figura 3.1. La primera etapa consiste de obtener la señal EEG del cerebro por medio de electrodos, la segunda trata sobre la amplificación de la señal a un nivel manejable por la siguiente etapa, así como el filtrado de dicha señal con el fin de eliminar los componentes de frecuencia no deseados; cuando se diseña esta etapa, se considera también el diseño de una fuente de poder que le proporcione el voltaje necesario para operar. La última etapa, la adquisición de datos, es el puente entre el hardware y el software y consiste en el muestreo y digitalización de la señal EEG de cada uno de los dieciséis canales, dejándola de esta manera lista para ser utilizada por la aplicación de software, la cual conforma la segunda parte del sistema.



Figura 3.1. Diagrama de bloques para el diseño de la etapa de hardware.

#### 3.1.1 Registro de la Señal

La primera etapa a contemplar dentro del diseño del hardware es precisamente el registro de la señal. Como ya se dijo en el capítulo uno, registrar la señal se refiere a sensar o recoger con electrodos el potencial eléctrico producto de la actividad eléctrica espontánea del cerebro. Además, como se mostró en otro apartado del mismo capítulo, existen distintos tipos de electrodos que pueden ser usados tanto para registrar señales EEG, como para sensar otros tipos de señales biológicas. Todo esto establece que esta etapa del diseño consiste simplemente en elegir aquél tipo de electrodos que permita registrar adecuadamente las señales EEG que van a alimentar el sistema. Para realizar dicha elección, se tomará como base lo expuesto con anterioridad en el capítulo uno, específicamente en el apartado que trata sobre los electrodos.

De los dos grupos que existen de electrodos (externos e internos), se puede observar que para cubrir los objetivos de este trabajo será mejor utilizar electrodos externos, ya que no requieren de una intervención quirúrgica para ser colocados, como es el caso de los electrodos internos.

Dentro de los electrodos externos, quedarían definitivamente eliminados los electrodos de succión, pues éstos sólo pueden usarse durante periodos cortos, y además provocan irritación en la piel, y esto no serviría para registrar señales EEG, pues muchas veces se requiere estar sensando las señales continuamente y por largos periodos para lograr un registro adecuado.

Todo lo anterior reduce la elección a los electrodos de placa de metal o a los electrodos flexibles. El funcionamiento y usos de estos dos tipos de electrodos es bastante similar, no así su costo, siendo los electrodos flexibles más costosos que los de placa de metal. Como se mencionó anteriormente, los electrodos flexibles poseen la ventaja de que, al no ser rígidos, se adaptan mejor a la forma irregular que posee el cuerpo humano. Esto sería un factor determinante de decisión si se fuera a realizar algún estudio en donde se tuviesen que colocar los electrodos sobre una superficie irregular, que no es el caso de un EEG. Como se indicó en el capítulo uno, para un EEG, los electrodos se colocan sobre el cuero cabelludo del paciente, y en este caso, no es una ventaja el que los electrodos sean rígidos o flexibles. Debido a esto y considerando la diferencia en costos, se eligieron para este sistema electrodos de placa de metal.

### 3.1.2 Acondicionamiento de la Señal

La señal obtenida directamente de la etapa anterior es una señal no apta aún para ser tratada digitalmente. Por esto se tiene dentro del diseño del sistema una etapa de acondicionamiento de la señal, la cual realizará dos tareas específicas, primeramente la adecuación en amplitud y posteriormente la adecuación en frecuencia. La adecuación en amplitud se traduce en una etapa de amplificación del nivel de voltaje de la señal, mientras que la adecuación en frecuencia será un filtrado que limitará el ancho de banda a las frecuencias de interés, eliminando también de paso la interferencia producida por la señal de 60 ciclos. En este apartado se presenta entonces el diseño de los amplificadores y filtros que van a acondicionar las señales provenientes de cada uno de los 16 canales considerados, así como el diseño de la fuente de poder que va a alimentar estos circuitos.

#### *Amplificación de la señal*

La etapa de amplificación tendrá como entrada la señal proveniente de los electrodos usados para el registro de la señal, mismos que se describieron en la sección anterior; como se recordará, las señales con las que se trabajará son del orden de los microvolts en su amplitud, por lo que se puede observar que se necesita una etapa de amplificación de ganancia alta para llevar las señales a los niveles adecuados que pueda leer la tarjeta de adquisición.

Los niveles de voltaje de la señal a amplificar están en el rango de 5 a 200  $\mu\text{V}$ . Si se parte del hecho de que se desea que los niveles ya amplificados lleguen al rango de 0 a 10 V, que es el rango que se utilizará en la tarjeta que realizará la adquisición, quiere decir que se requiere una ganancia general de 50,000. Esto quiere decir que si se ocupa una amplificación de la tarjeta de adquisición de 50, queda un remanente de amplificación de 1,000, que es la ganancia que se necesita obtener de la etapa de amplificación a base de operacionales, es decir  $A_r = 50,000$ .

Si se tiene una señal de máximo 10 V a la salida del amplificador, quiere decir que el escalón de voltaje mínimo medible, tomando en cuenta una resolución de 12 bits en la tarjeta de adquisición, es de:

$$\frac{10\text{V}}{2^{12}} = \frac{10\text{V}}{4096} = 2.44 \text{ mV}$$

$\left. \begin{array}{l} 0.00 \text{ mV} \\ 2.44 \text{ mV} \\ 4.88 \text{ mV} \\ \vdots \\ 0,000 \text{ mV} \end{array} \right\} = 4,096 \text{ niveles de voltaje a la salida.}$

Debido a que la señal a la entrada del amplificador es muy débil, y por lo tanto más susceptible al ruido que señales de mayor nivel, se tiene que utilizar una configuración tal, que permita rechazar dicho ruido, es decir se necesita que tenga una etapa de amplificación diferencial que permita amplificar sólo la señal de interés y no el ruido indeseable, el cual estará presente a la entrada de ambas terminales del amplificador. Además se requiere que la impedancia de entrada de la etapa de amplificación sea alta, lo cual se logra con unos seguidores de voltaje, que arreglados de manera adecuada permitirán obtener una ganancia manipulable y evitará una demanda alta de corriente a los electrodos de registro. Cuando se conjuntan estas etapas en una sola configuración se llega a lo que se conoce como *amplificador de instrumentación*, que es ampliamente usado en aplicaciones médicas [Asto90].

A continuación se presenta el análisis de las etapas que componen el amplificador de instrumentación, primeramente se verá el amplificador diferencial (ver figura 3.2) y luego la etapa formada por los seguidores de voltaje, en la cual se obtiene la ganancia.

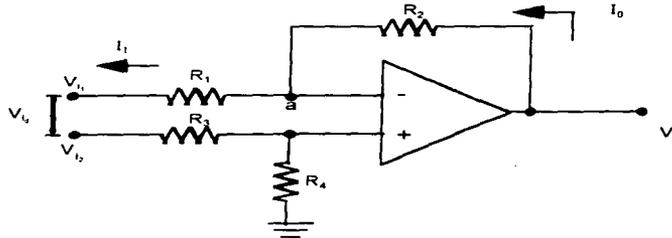


Figura 3.2. Amplificador diferencial.

Haciendo sumatoria de corrientes en el punto "a", se tiene:

$$\sum I_a$$

$$I_0 - I_1 = 0$$

$$\frac{V_0 - V^+}{R_2} = \frac{V^+ - V_1}{R_1}$$

$$V_O = -\frac{R_2}{R_1} V_{i_1} + \frac{R_2}{R_1} V^+ + V^+$$

$$V_O = -\frac{R_2}{R_1} V_{i_1} + \left[ 1 + \frac{R_2}{R_1} \right] V^+ \quad (3.1)$$

Además:

$$V^+ = \frac{R_4}{R_3 + R_4} V_{i_2}$$

Substituyendo esta última ecuación  $V^+$  en (3.1) se tiene:

$$V_O = -\frac{R_2}{R_1} V_{i_1} + \left[ 1 + \frac{R_2}{R_1} \right] \left[ \frac{R_4}{R_3 + R_4} \right] V_{i_2}$$

También

$$V_{i_d} = V_{i_2} - V_{i_1}$$

Si se hace que  $\frac{R_2}{R_1} = \frac{R_4}{R_3}$  se tiene finalmente que:

$$V_O = \frac{R_2}{R_1} V_{i_d} = \frac{R_4}{R_3} V_{i_d}$$

La idea fundamental de este circuito es eliminar el voltaje de modo común además de poder medir un voltaje a la salida del amplificador con respecto a tierra, en vez de medir una carga flotante.

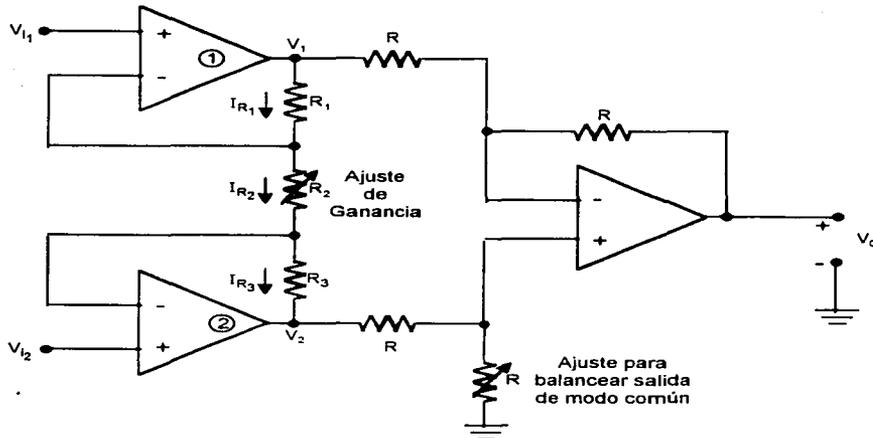


Figura 3.3. Amplificador de Instrumentación.

Del análisis del circuito de la figura 3.3, se tiene lo siguiente:

$$I_{R1} = I_{R2} = I_{R3} = I$$

$$I_{R1} = \frac{V_1 - V_1}{R_1} \Rightarrow V_1 = I_{R1} R_1 + V_1 \quad (3.2)$$

$$I_{R2} = \frac{V_1 - V_2}{R_2} \quad (3.3)$$

$$I_{R3} = \frac{V_2 - V_2}{R_3} \Rightarrow V_2 = V_2 - I_{R2} R_3 \quad (3.4)$$

Del análisis del amplificador diferencial y considerando  $R_3 = R_4 = R$  se tiene:

$$V_O = \frac{R}{R} (V_2 - V_1) = V_2 - V_1 \quad (3.5)$$

si se resta (3.4) a (3.2)

$$V_2 - V_1 = VI_2 - IR_2 R_3 - IR_1 R_1 - VI_1$$

$$V_2 - V_1 = VI_2 - VI_1 - I(R_3 + R_1) \quad (3.6)$$

de (3.3), (3.5) y (3.6)

$$V_o = VI_2 - VI_1 - \left[ \frac{VI_1 - VI_2}{R_2} \right] (R_3 + R_1)$$

$$V_o = VI_2 - VI_1 + \frac{VI_2 - VI_1}{R_2} (R_3 + R_1)$$

$$V_o = \left[ 1 + \frac{(R_3 + R_1)}{R_2} \right] (VI_2 - VI_1) \quad (3.7)$$

considerando  $R_3 = R_1 = R$  y  $R_2 = aR$ , se tiene:

$$V_o = \left[ 1 + \frac{2}{a} \right] (VI_2 - VI_1)$$

de donde fácilmente se puede manipular la ganancia del arreglo.

Esta configuración, como se ve, puede construirse a partir de amplificadores operacionales que cuenten con una razón de rechazo en modo común alta y usando resistencias de precisión. Sin embargo, en la práctica es difícil que se logre en conjunto el resultado deseado, debido principalmente a las características únicas de cada amplificador operacional, es decir, no es fácil encontrar varios que tengan exactamente las mismas características, por lo que estas diferencias hacen que resulte un error en la razón de rechazo de modo común deseada. A su vez, las resistencias aun siendo de precisión, no serán exactamente iguales, lo cual también inducirá un error. Todo esto es altamente importante por las características de la señal con que se trabaja.

De lo anterior se desprende como una mejor alternativa el uso de un encapsulado que ya es en sí un amplificador de instrumentación, lo que conlleva que están ya adecuadamente acoplados, cuentan internamente con esas resistencias de precisión deseadas y por lo tanto se puede confiar en que se comportarán como se espera. Hay un número considerable de amplificadores de instrumentación en el mercado, ¿cuál elegir? La respuesta tiene que ver con los mismos parámetros que hasta ahora se han manejado. Uno de ellos es la razón de rechazo en modo común, la cual es necesario que sea al menos de 120 dB, que tenga una alta impedancia de entrada para evitar que los electrodos sean demandados de demasiada corriente y una ganancia alta, que, como se vio, tiene que ser de 1000.

Esto va cerrando el panorama y permite llegar a elegir un encapsulado. Por ejemplo, analizando las características del LM363 y el LH0038, se observa que si bien son de características similares, ambos cumplen con las expectativas, el primero, es decir, el LM363 tiene una razón de

rechazo en modo común y una razón de rechazo a variaciones de la fuente típicas más altas (130 dB). Además que no requiere partes externas para una ganancia de 100 y 1000 (ver Apéndice 2). Estas son buenas razones para elegir el primer encapsulado mencionado, si bien debe quedar claro que en caso de adolecer de presupuesto se puede dar más peso a la parte costo, con lo cual quizá se pueda prescindir de unos diez decibeles en las razones de rechazo y con un poco de esfuerzo extra todavía lograr una buena amplificación. A continuación, en la figura 3.4, se muestra la conexión del encapsulado a usar. El corto circuito entre las patas 2 y 4 es para fijar la ganancia en 1,000.

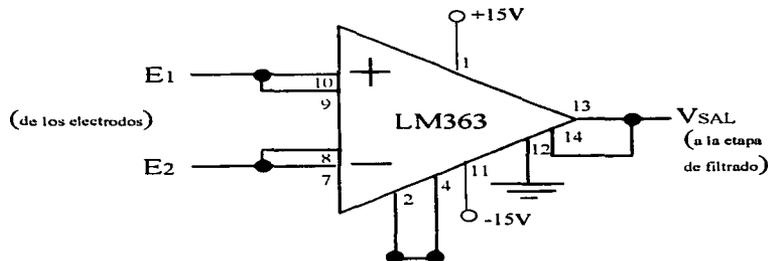


Figura 3.4. Conexión usada para el LM363 de 16 pines.

Una vez elegido el amplificador a usar, se debe tener en cuenta cómo se van a reflejar sus necesidades de polarización en el diseño. Este integrado tiene un consumo de corriente típico de 1.5 mA y de 2.8 mA máximo. Además, el fabricante sugiere que los voltajes de alimentación sean de +15 V y -15 V, ya que esto logra que se optimen las razones de rechazo que se han manejado. Estas consideraciones delimitan con más claridad las características de la fuente de alimentación.

### Diseño del filtro

Como se desprende del capítulo 1.1.4, las señales que se sensan en la superficie del cuero cabelludo están clasificadas en diferentes bandas de frecuencia. Estas bandas de frecuencia van desde los 0.5 Hz, donde inicia el ritmo delta, hasta los 30 Hz, frecuencia terminal del ritmo beta. Por lo anterior, el filtro de elección sólo debe pasar los componentes inferiores a 30 Hz y eliminar las frecuencias superiores, constituidas principalmente por la componente de 60 Hz y todas las frecuencias superiores que se agregan a la banda de interés. Como se mencionó anteriormente, estas señales se deben a interferencias de transmisión de radio, tv, motores, etc.

Considérese otra característica importante de la señal que se maneja a la entrada: el rango de su amplitud inicia desde cero y termina hasta los 200  $\mu$ V. Por tanto tomando en cuenta que la señal de 60 Hz de las líneas de alimentación eléctrica es la que genera la interferencia de

mayor consideración, es de esperar que la interferencia producida por la señal de 60 Hz sea de una magnitud mucho mayor la de la señal de estudio. Así entonces, el diseño del filtro debe de contemplar eliminar la señal de 60 Hz al máximo.

Para ello, la etapa de filtrado consistirá de dos secciones principales: la primera estará formada por un filtro supresor de banda (filtro *Notch*) cuya función será única y exclusivamente disminuir lo más posible la componente de 60 Hz que se encuentre junto con la señal EEG. La segunda etapa consistirá de un filtro paso bajas el cual se encargará de seleccionar las frecuencias de interés (0.5 a 30 Hz).

#### *Primera sección: Filtro Notch*

Un filtro Notch consiste de manera general de un filtro paso banda de banda estrecha (se explica este término más adelante) cuya salida es restada de la señal original; por ello, se analizarán primero los parámetros fundamentales de los filtros paso banda.

Un filtro paso banda, como su nombre lo indica, permite pasar un rango de frecuencias comprendido entre un límite inferior y un límite superior. La ganancia máxima de este tipo de filtro se encuentra a una frecuencia llamada *de resonancia*  $f_r$ . Para un filtro normalizado, la ganancia máxima será igual a uno (0 dB). La frecuencia por debajo de  $f_r$  en la cual la ganancia cae a 0.707 se denomina frecuencia de corte inferior  $f_{L1}$ ; así mismo, la frecuencia por encima de  $f_r$  en donde la ganancia tiene un valor de 0.707 recibe el nombre de frecuencia de corte superior  $f_{H1}$ . El intervalo de frecuencias comprendido entre  $f_{L1}$  y  $f_{H1}$  se conoce como *ancho de banda B* el cual está dado por:

$$B = f_{H1} - f_{L1} \quad (3.8)$$

la frecuencia de resonancia se obtiene a partir de:

$$f_r = \sqrt{f_{L1} f_{H1}} \quad (3.9)$$

despejando  $f_{H1}$  de (3.8):

$$f_{H1} = f_{L1} + B \quad (3.10)$$

sustituyendo (3.10) en (3.9) se tiene:

$$f_r = \sqrt{f_{L1} (B + f_{L1})} \quad (3.11)$$

despejando  $f_{L1}$  de (3.11):

$$f_{L1} = \sqrt{\frac{B^2}{4} + f_r^2} - \frac{B}{2} \quad (3.12)$$

El *factor de calidad Q* es la relación entre la frecuencia de resonancia y el ancho de banda:

$$Q = \frac{f_r}{B} \quad (3.13)$$

El factor  $Q$  indica selectividad, un valor elevado indica que el filtro selecciona una banda de frecuencias pequeña. Cuando  $Q$  es menor o igual a 0.5, al filtro se le conoce como de banda ancha, mientras que cuando  $Q$  es mayor a 0.5 se considera que el filtro es de banda estrecha. Un filtro paso banda de banda estrecha se puede implementar utilizando solamente un amplificador operacional [CoDr93] valiéndonos de la configuración que se muestra en la figura 3.5.

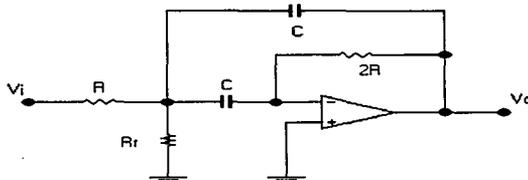


Figura 3.5. Circuito del filtro paso banda de banda estrecha.

Se trata de una configuración de ganancia unitaria gracias a que el valor de la resistencia de retroalimentación es del doble de la resistencia de entrada; la resistencia  $R_r$  sirve para darle un ajuste fino a la frecuencia de resonancia. Para el circuito de la figura 3.5 se tienen las siguientes relaciones [CoDr93]:

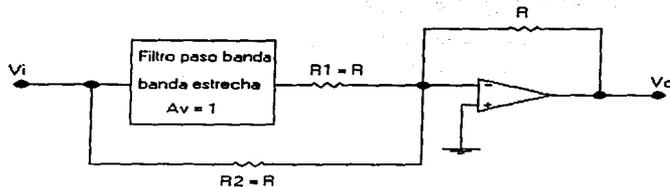
$$B = \frac{0.1591}{RC} \quad (3.14)$$

$$R_r = \frac{R}{2Q^2 - 1} \quad (3.15)$$

$$f_r = \frac{0.1125}{RC} \sqrt{1 + \frac{R}{R_r}} \quad (3.16)$$

Con las ecuaciones (3.8) a (3.16) es posible diseñar un filtro paso banda dependiendo de los datos de entrada que se especifiquen ( $f_r$ ,  $Q$ ,  $B$ ).

Una vez que se tienen los elementos para diseñar un filtro paso banda, se podrá continuar entonces con el diseño de el filtro Notch. Como se mencionó anteriormente, éste resulta de restar la salida de un paso banda a la señal original utilizando el circuito que se muestra en la figura 3.6.



**Figura 3.6. Filtro Notch compuesto por un filtro paso banda y un sumador inversor.**

Cuando la frecuencia de la señal de entrada se encuentra dentro de las bandas de rechazo del filtro paso banda, la salida de este filtro se aproxima a cero, la entrada que se aplica entonces al sumador inversor es  $V_i$  que pasa por  $R_2$ , teniendo a la salida  $-V_i$ . Por otro lado, cuando la frecuencia de la señal de entrada se aproxima a la frecuencia de resonancia del filtro paso banda, se tendrá a la salida de este  $-V_i$ , el cual se invierte al pasar por el sumador, teniendo a la salida  $+V_i$ ; sin embargo, la entrada también pasa por  $R_2$ , teniendo a la salida  $-V_i$  por lo que al responder el sumador a las dos entradas se tiene que  $V_o = +V_i - V_i = 0$ . Nótese que lo anterior sucede cuando la frecuencia de la señal de entrada corresponde a la banda de paso del filtro paso banda, por lo que se concluye que la frecuencia de resonancia para el filtro Notch es la misma que para el filtro paso banda. De hecho, todos los parámetros del filtro paso banda ( $Q$ ,  $B$ ,  $f_r$ ,  $f_{H1}$ ,  $A_v$  y por supuesto  $f_r$ ) corresponden al filtro Notch.

#### Procedimiento de diseño [CoDr93]

- Definir la frecuencia que se desea rechazar, es decir, la frecuencia de resonancia  $f_r$ .
- Definir el ancho de banda del filtro  $B$  y con el dato del punto anterior, obtener  $Q$ .
- Hacer un filtro paso banda de ganancia unitaria con  $f_r$ ,  $B$  y  $Q$  definidos anteriormente.
  - Escoger un valor de capacitancia para  $C$  y calcular en base a ello el valor de la resistencia  $R$  utilizando la ecuación (3.14).
  - Calcular  $R_f$  utilizando la ecuación (3.15)
  - Verificar utilizando la ecuación (3.16) el valor de  $f_r$  obtenido con valores de resistencia comerciales, si no es satisfactorio, usar un potenciómetro para  $R_f$  y ajustar su valor utilizando esta misma ecuación.
- Conectar el sumador inversor al filtro paso banda obtenido. En general se escogen resistencias iguales de  $10\text{ k}\Omega$  con el menor valor de tolerancia posible (1%) ya que de esto depende el valor de la ganancia justo en  $f_r$ .

**Cálculo de los valores del filtro**

La frecuencia que se desea rechazar es 60 Hz, por lo que  $f_r = 60$  Hz.

Es deseable tener curvas de respuesta en frecuencia muy pronunciadas, por lo que se escoge un ancho de banda estrecho con un valor de  $B = 6$  Hz.

Utilizando la ecuación (3.13) se tiene que:

$$Q = \frac{60}{6} = 10.$$

Empleando la ecuación (3.12):

$$f_L = \sqrt{\frac{36}{4} + 3600} - 3 = 57.075 \text{ Hz.}$$

Se obtiene  $f_H$  a partir de (3.10):

$$f_H = 57.075 + 6 = 63.075 \text{ Hz.}$$

Se tiene entonces que el filtro rechazará todas las frecuencias entre 57.075 y 63.075 Hz.

A continuación se determinan los valores de los componentes:

Se escoge primeramente  $C = 0.47 \mu\text{F}$ .

Se obtiene el valor de R despejándolo de la ecuación (3.14):

$$R = \frac{0.1591}{BC} = \frac{0.1591}{6(0.47 \times 10^{-6})} = 56418.44 \Omega \approx 56 \text{ k}\Omega.$$

Se obtiene  $R_r$  de la ecuación (3.15):

$$R_r = \frac{56 \times 10^3}{2(100) - 1} = 281.407 \Omega \approx 276 \Omega.$$

Se verifica el valor de  $f_r$  utilizando valores comerciales mediante (3.16):

$$f_r = \frac{0.1125}{(56 \times 10^3)(0.47 \times 10^{-6})} \sqrt{1 + \frac{56 \times 10^3}{276}} = (4.274)(14.279) = 61.0 \text{ Hz.}$$

Escogiendo valores de resistencias y capacitores con tolerancias mínimas (1%), la frecuencia de resonancia puede ir desde 59.83 hasta 62.27 Hz, que se considera aceptable, sin embargo, en caso de requerir mayor exactitud en este valor, puede utilizarse un potenciómetro en  $R_r$  para compensar estas desviaciones.

Finalmente, se acopla el filtro paso banda al sumador inversor y se obtiene el filtro Notch que se muestra en la figura 3.7.

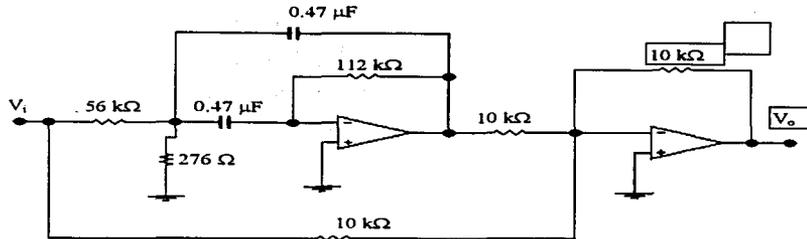


Figura 3.7. Configuración final del filtro Notch con  $f_r=60\text{Hz}$ .

*Segunda sección : Filtro paso bajas*

La *frecuencia de corte* que se considera para el diseño del filtro paso bajas es  $f_c=30\text{ Hz}$ . El orden implica un análisis más exhaustivo, ya que cuando se diseñan filtros cuya banda de rechazo se halla próxima a la frecuencia de corte, se requiere que el filtro se aproxime más al comportamiento ideal, lo que implica que el *roll-off* del filtro sea muy pronunciado.

Un análisis previo sobre las condiciones de la señal, dan una orientación sobre el tipo y orden del filtro que se debe utilizar para filtrar adecuadamente la señal. El orden de un filtro indica que tan pronunciada es la atenuación de la señal fuera de la banda de paso. Esta atenuación, conocida como *roll-off*, se mide en dB/octava o dB/década, y es una función del orden del filtro. Así, encontramos filtros de orden 2 y atenuación a 12 dB/octava, de orden 3 y atenuación de 18 dB/octava, orden 4 y atenuación a 24 dB/octava y así sucesivamente.

Se verá a continuación el caso para el filtro deseado. Una de las frecuencias que se requiere eliminar es la de 60 Hz, la cual se encuentra a 30 Hz de la frecuencia de corte que se necesita. Esta frecuencia se sitúa justamente a una octava (el doble de la frecuencia), por lo que las atenuaciones marcadas por el orden del filtro se aplican exactamente sobre la frecuencia que se desea rechazar. La ganancia en decibeles del filtro está dada por la relación  $A_{dB} = 20\log(V_o/V_i)$ . El voltaje de salida en función de la pendiente de atenuación *roll-off* se muestra en la tabla 3.1:

Orden del Filtro	Roll-off (dB/octava)	Voltaje de Salida $V_o$
1	-6	$0.501 V_i$
2	-12	$0.251 V_i$
3	-18	$0.125 V_i$
4	-24	$0.063 V_i$
5	-30	$0.031 V_i$

Tabla 3.1. Voltaje de salida en función del roll-off.

Como se observa, un filtro de orden tres proporciona una atenuación de -18 dB. Este valor es aún insuficiente ya que aporta a la salida un 12% de la amplitud de la señal de 60 Hz. Es por esto que se requiere un filtro de orden superior, esto es, de cuarto orden o mayor, según se requiera la atenuación. Se utilizará en este diseño un filtro de orden cinco el cual atenúa la señal de 60 Hz hasta un 3.1%, valor que es adecuado para la aplicación para la que se diseña el filtro.

Como se demostró anteriormente, es necesario el uso de un filtro de orden superior para satisfacer las necesidades de filtrado. Existen dos configuraciones para la síntesis de filtros de orden superior: La aproximación cascada, en la que la respuesta deseada se obtiene al colocar en cascada etapas de filtros de primero y segundo orden; y la aproximación simulación de escalera RLC, que utiliza convertidores activos de impedancia, para simular un prototipo de filtro RLC pasivo que satisfaga el objetivo deseado.

La aproximación cascada es por mucho el método más simple y popular. Esta aproximación genera la función de transferencia deseada como un producto de funciones de transferencia de segundo orden, y posiblemente de primer orden, cuyas frecuencias características y valores de factores de calidad  $Q$ , son tabulados en manuales de filtros o pueden ser determinados con la ayuda de programas computarizados.

### Aproximaciones de Respuesta

Existen diferentes tipos de respuesta que dan los filtros de orden superior, conocidas como *aproximaciones* de tipo Butterworth, Chebyshev, Elíptico y Bessel. De los filtros anteriores, los tres primeros optiman el *roll-off* o atenuación del filtro, y el último, el filtro Bessel optimiza el cambio de fase de la señal de salida, esto es, provoca que el cambio de fase varíe linealmente con la frecuencia. Si la fase de una señal no varía linealmente con la frecuencia, entonces diferentes componentes de frecuencia obtendrán diferentes retrasos, y las componentes no senoidales tales como las ondas cuadradas tendrán una distorsión significativa. Como en encefalografía todas las componentes son senoidales, sólo se toman en cuenta los tres primeros filtros, cuya respuesta en frecuencia para un orden 5 se presenta en la figura 3.8.

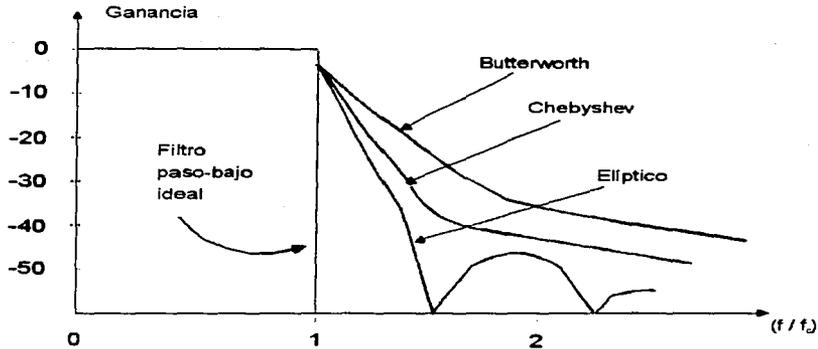


Figura 3.8. Comparación de respuestas Butterworth, Chebyshev y elíptico para  $n=5$ .

El rango de frecuencias en donde la amplitud de las señales presenta poca o ninguna atenuación se llama *banda de paso*, misma que para un filtro paso-bajo se extiende desde el componente directo *dc* hasta una frecuencia  $f_c$ , llamada *frecuencia de corte*. La ganancia no es estrictamente constante dentro de la banda de paso. El máximo cambio de ganancia permisible se denota por  $A_{max}$  y se expresa en decibeles. Esta variación de ganancia se presenta como un variación ondulatoria dentro de la banda de paso (véase figura 3.9), y se conoce como *ondulación paso-banda máxima*.

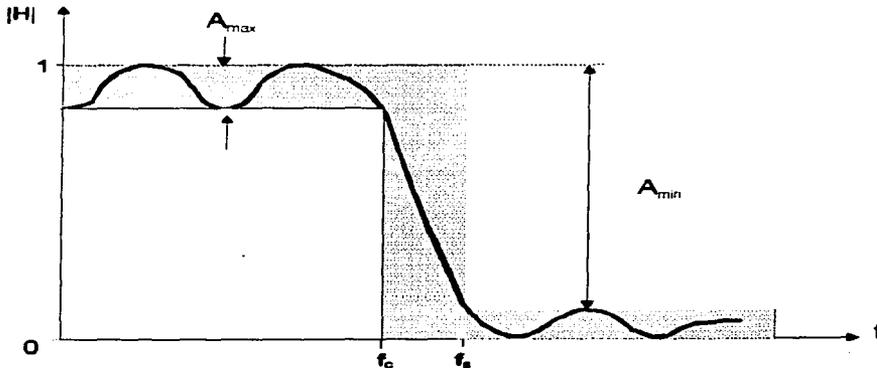


Figura 3.9. Respuesta en magnitud para una aproximación práctica de un filtro paso-bajo.

Pasando la frecuencia de corte  $f_c$ , la respuesta decae hasta la *banda de paro*,  $f_s$ , misma que marca el inicio de la banda de atenuación más significativa. La *banda de paro* se especifica generalmente en términos de una atenuación mínima permisible, por ejemplo  $A_{\min} = 50$  dB. La región delimitada por  $f_c$  y  $f_s$  se conoce como *banda de transición*.

Como se observa en la figura 3.8, de los tres filtros, el filtro Elíptico presenta la máxima atenuación en la banda de transición. Esto no es motivo suficiente para su elección ya que una atenuación que aproxime al comportamiento ideal del filtro se obtiene a expensas de una mayor *ondulación paso-banda máxima* y lo que interesa es que la ganancia sea uniforme dentro de la banda de paso para identificar señales de amplitud específica como son las señales alfa, beta, delta y teta. Esto se puede obtener mediante el uso de un filtro Butterworth, que maximiza la uniformidad de la ganancia dentro de la banda de paso.

### Cálculo del filtro

Especificaciones del filtro paso-bajo:

$f_c = 30$  Hz, atenuación mínima a 60 Hz, 33 dB (representa un 2.2% del componente a esta frecuencia).

n	$f_{01}$	$Q_1$	$f_{02}$	$Q_2$	$f_{03}$	$Q_3$	$f_{04}$	$Q_4$	$f_{05}$	$Q_5$	at. a 2F (dB)
2	1	0.707									15
3	1	1.000	1								21
4	1	0.541	1	1.306							27
5	1	0.618	1	1.620	1						33
6	1	0.518	1	0.707	1	1.932					39
7	1	0.555	1	0.802	1	2.247	1				45
8	1	0.510	1	0.601	1	0.900	1	2.563			51
9	1	0.532	1	0.653	1	1.000	1	2.879	1		57
10	1	0.506	1	0.561	1	0.707	1	1.101	1	3.196	63

Tabla 3.2. Tabla de especificaciones para filtros Butterworth.

La tabla 3.2 muestra las especificaciones para filtros Butterworth. Esta tabla señala que para la atenuación de 33 dB se requiere un filtro de orden  $n=5$ , que se implementa al conectar en cascada dos etapas de segundo orden y una de primer orden del tipo que se muestra en la figura 3.10. La especificaciones para cada etapa se muestran en la parte sombreada de la tabla, y éstas son:

- Primera Etapa: Filtro de segundo orden con  $f_{01} = 1 \times 30$  Hz y  $Q_1 = 0.618$ .
- Segunda Etapa: Filtro de segundo orden con  $f_{02} = 1 \times 30$  Hz y  $Q_2 = 1.620$ .
- Tercer Etapa: Filtro de primer orden con  $f_{03} = 1 \times 30$  Hz.

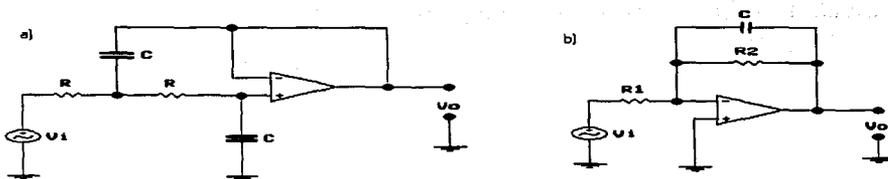


Figura 3.10. Plantilla para filtros. a) filtro de 2º orden, b) filtro de primer orden.

El tipo de circuito que se está considerando es una *fente de voltaje controlada por voltaje* (VCVS por sus siglas en inglés). Además de este tipo de circuito, existen también los de realimentación múltiple con ganancia infinita e incluso los de realimentación simple con ganancia infinita [ToGr85]. Para este diseño se utilizarán los de fuente controlada debido principalmente a que su puesta a punto es más sencilla sin afectar los parámetros de la red, por ejemplo el factor de calidad puede modificarse sin afectar la frecuencia de corte; además, los circuitos se pueden conectar en cascada sin que se afecten entre ellos.

#### Cálculo de los componentes de circuito paso-bajo de segundo orden

El cálculo del filtro debe realizarse con elementos RC de valores comerciales, por tanto, es necesario hacer un análisis previo del filtro. Para simplificar el álgebra, el circuito del filtro de segundo orden se dibuja parametrizado como lo muestra la figura 3.11.

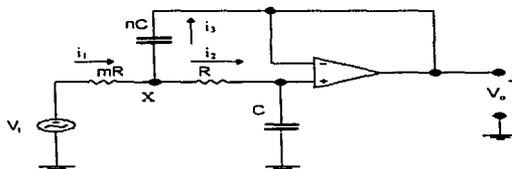


Figura 3.11. Filtro paso-bajo de ganancia unitaria parametrizado.

En esta simplificación se expresan los valores del primer par RC en términos del segundo par por medio de los multiplicadores  $m$  y  $n$ , respectivamente. Se denota  $V_x$  al voltaje en que se unen las dos resistencias.  $V_x$  y  $V_0$  se encuentran relacionadas por la función de un filtro paso-bajo de primer orden.

$$V_0 = \frac{V_x}{1 + j\omega RC} \quad (3.17)$$

por LCK en el nodo X se tiene

$$\frac{V_i - V_x}{mR} = \frac{V_x - V_o}{R} + \frac{V_x - V_o}{1/(j\omega nC)} \quad (3.18)$$

multiplicando ambos lados por  $mR$  y agrupando,

$$V_i = (1 + m + j\omega mnRC)V_x - (m + j\omega mnRC)V_o \quad (3.19)$$

de (3.17):

$$V_x = (1 + j\omega RC)V_o \quad (3.20)$$

(3.20) en (3.19):

$$V_i = V_o(1 + j\omega RC)(m + j\omega mnRC + 1) - V_o(m + j\omega mnRC)$$

$$\frac{V_o}{V_i} = \frac{1}{1 + j\omega RC(m + 1) - \omega^2 mR^2 nC^2} \quad (3.21)$$

Se sabe que:

$$\omega = 2\pi f \quad (3.22)$$

(3.22) en (3.21):

$$\frac{V_o}{V_i} = \frac{1}{1 - 4\pi^2 f^2 mR^2 nC^2 + j2\pi f RC(m + 1)} \quad (3.23)$$

por otro lado, la frecuencia de corte del filtro está dada por:

$$f_o = \frac{1}{2\pi\sqrt{mn}RC} \quad (3.24)$$

y el factor de calidad del mismo está dado por:

$$Q = \frac{\sqrt{mn}}{m + 1} \quad (3.25)$$

(3.24) y (3.25) en (3.21) y simplificando:

$$H = \frac{V_o}{V_i} = \frac{1}{1 - (f/f_o)^2 + (j/Q)(f/f_o)} \quad (3.26)$$

Como se puede observar, el factor de calidad del filtro depende exclusivamente de los factores  $m$  y  $n$ , mientras que  $f_0$  depende de  $R$  y  $C$  así como de  $m$  y  $n$ .

#### Procedimiento de diseño

Una labor común en el diseño de filtros es encontrar los valores de componentes  $RC$  para lograr valores de  $f_0$  y  $Q$ . Los valores estándar de resistencias ofrecen más variedad que los capacitores, por tanto, es razonable utilizar algún valor de capacitancia disponible, y después afinar los valores de resistencia. Este es el procedimiento de diseño:

1. Iniciar con dos resistencias iguales ( $m=1$ ) en el rango de 10 a 100 kOhms. Llamando a este valor  $R^*$ .
2. Calcular  $C^* = 1 / (4\pi Q f_0 R^*)$ .
3. Calcular  $n^* = 4Q^2$ .
4. Encontrar los valores comerciales de capacitancia  $C$  y  $nC$ , de tal forma que  $C = C^*$  y  $n \geq n^*$
5. Utilizar el nuevo valor de  $n$  encontrado para calcular  $k = n / Q^2 - 2$ . Entonces, con este valor calcular  $m = [k + \sqrt{k^2 - 4}] / 2$
6. Encontrar  $R = \frac{1}{2\pi\sqrt{mn}f_0C}$ .
7. Seleccionar de la tabla de resistencias comerciales los valores más cercanos a  $R$  y  $mR$ .

#### Cálculo de los componentes de circuito paso-bajo de primer orden

El circuito de la figura 3.10(b) tiene la siguiente función de transferencia:

$$H = H_0 \frac{1}{1 + j(f / f_0)} \quad (3.27)$$

Donde:  $H_0 = -\frac{R_2}{R_1}$  (3.28)

y:  $f_0 = \frac{1}{2\pi R_2 C}$  (3.29)

#### Procedimiento de diseño

1. Fijar la ecuación (3.29) con un valor de capacitancia comercial, para calcular  $R_2$ .
2. Se calcula  $R_2$ .

3. Como el circuito de primer orden se comporta como un filtro paso-bajo con ganancia, esta se fija como unitaria, de tal forma que los valores de  $R_1$  adquieren el mismo valor que los de  $R_2$ .

**Valores específicos de las etapas del filtro**

**Primera etapa:** Filtro de segundo orden con  $f_{01} = 30$  Hz y  $Q_1 = 0.618$

1.  $R^* = 18 \text{ K}\Omega$
2.  $C^* = 1 / (4\pi \times 0.618 \times 30 \times 18 \times 10^3) = .23 \mu\text{F}$
3.  $n^* = 4 \times 0.618^2 = 1.53$
4. Escogemos  $C = 0.22 \mu\text{F}$  y  $nC = 2.2 \mu\text{F}$ ;  $n = (2.2 / 0.22) = 10$ .  $n > n^*$
5.  $k = 10 / 0.618^2 - 2 = 24.18$ ;  $m = [24.18 + \sqrt{24.18^2 - 4}] / 2 = 24.13$
6.  $R = 1 / [2\pi \sqrt{(24.13 \times 10)} \times 30 \times 0.22 \times 10^{-6}] = 1,552.37 \Omega$ ;  $mR = 37458.88 \Omega$
7. Escogemos  $R = 1.5 \text{ k}\Omega$ ;  $mR = 37 \text{ k}\Omega$

**Segunda Etapa:** Filtro de segundo orden con  $f_{02} = 30$  Hz y  $Q_2 = 1.620$ .

1.  $R^* = 33 \text{ k}\Omega$ .
2.  $C^* = 1 / (4\pi \times 1.620 \times 30 \times 33 \times 10^3) = 49 \text{ nF}$
3.  $n^* = 4 \times 1.620^2 = 10.5$
4. Escogemos  $C = 47 \text{ nF}$  y  $nC = 820 \text{ nF}$ ;  $n = (820 / 47) = 17$ .  $n > n^*$
5.  $k = 17 / 1.620^2 - 2 = 4.5$ ;  $m = [4.5 + \sqrt{4.5^2 - 4}] / 2 = 4.27$
6.  $R = 1 / [2\pi \sqrt{(4.27 \times 17)} \times 30 \times 47 \times 10^{-9}] = 13,248 \Omega$ ;  $mR = 56,570 \Omega$
7. Escogemos  $R = 13.3 \text{ k}\Omega$ ;  $mR = 57 \text{ k}\Omega$

**Tercer Etapa:** Filtro de primer orden con  $f_{03} = 30$  Hz

1. Fijamos  $C = 0.47 \mu\text{F}$
2.  $R_2 = 1 / (2\pi \times 30 \times 0.47 \times 10^{-6}) = 11287.58$ ; Entonces  $R_2 = 11 \text{ k}\Omega$
3. Por la ganancia unitaria  $R_1 = R_2 = 11 \text{ k}\Omega$

Finalmente, el filtro con las etapas en cascada queda como lo indica la figura 3.12, con los valores de capacitancia y resistencia indicados. Para toda la etapa de filtrado se utilizarán amplificadores operacionales TL084 (Apéndice 2) debido a su bajo costo, alta impedancia de entrada y al tipo de encapsulado que incluye cuatro amplificadores por circuito integrado (tómese en cuenta que se necesitarán 5 A.O. por canal dando un total de  $5 \times 16 = 80$  A.O.).

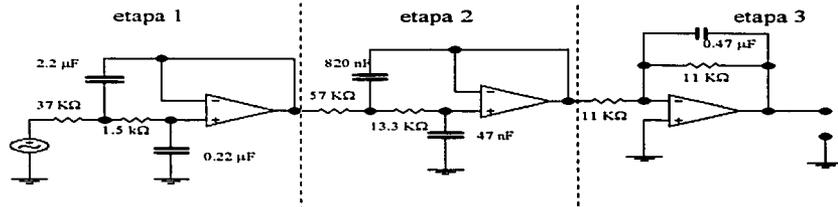


Figura 3.12. Filtro de tipo Butterworth para  $n=5$ .

### Fuente de Poder Regulada a $\pm 15V$

El diagrama de bloques en general para una fuente de poder regulada es el que se muestra en la figura 3.13. Como se observa, la primera etapa está formada por un transformador cuya función es reducir el voltaje en el contacto de pared de 120V a un valor de voltaje más bajo que requieren los transistores, C.I. y otros dispositivos electrónicos. La siguiente etapa se integra con un rectificador que convierte el voltaje de corriente alterna de salida del transformador en un voltaje pulsante de corriente directa. Este voltaje pulsante no es en sí una señal de corriente directa, de modo que se coloca un filtro a la salida del rectificador para suavizar los pulsos y producir un voltaje de DC con variaciones mínimas. La última etapa, la etapa de regulación, hace que el voltaje de DC no disminuya conforme se incrementa la corriente demandada por la carga, esto es, hace que el voltaje de salida entregado por la fuente permanezca casi constante ante variaciones en la corriente de carga. Además de esto, la etapa de regulación disminuye el voltaje de rizo de la señal.

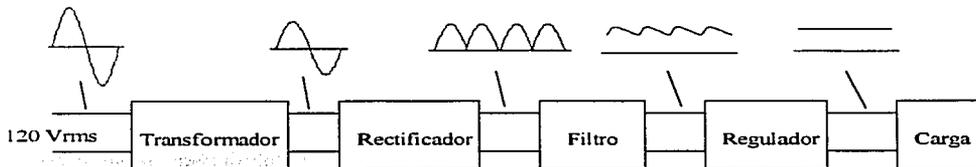


Figura 3.13. Diagrama de bloques para una fuente de poder.

La fuente de poder que requiere este sistema debe proporcionar un voltaje de  $\pm 15V$  y debe ser capaz de entregar una corriente de hasta 100.8 mA. El valor del voltaje de polarización se

eligió de  $\pm 15V$  debido a la recomendación del fabricante para el amplificador de instrumentación LM363, como se mencionó anteriormente. La corriente máxima de 100.8 mA se determinó bajo las siguientes consideraciones: Para la etapa de amplificación se utilizarán 16 encapsulados LM363 (uno para cada canal), cuyo valor de corriente de alimentación es de 2.8 mA, por tanto,  $2.8 \times 16 = 44.8$  mA. Para la etapa de filtrado se utilizarán 5 A.O. por canal, teniendo un total de 80 A.O.; para esta etapa se utilizarán encapsulados TL084 con 4 A.O. en cada circuito integrado, por lo que se requieren 20 encapsulados en total. La corriente de alimentación para este circuito integrado es también de 2.8 mA por lo que el requerimiento es  $2.8 \times 20 = 56$  mA. Se tiene entonces un requerimiento de 44.8 mA para la etapa de amplificación y 56 mA para la etapa de filtrado, lo que da un total de 100.8 mA.

#### *El Transformador*

Se va a utilizar un transformador a 30V con derivación central para tener  $15V_{rms}$  en cada mitad del secundario. Esto quiere decir que el voltaje pico  $V_m$  presente en cada mitad del secundario del transformador va a ser:

$$V_m = \sqrt{2}V_{rms} = \sqrt{2}(15) = 21.21V \quad (3.30)$$

Se eligió este transformador debido a que en la etapa de regulación se utilizarán reguladores que especifican en sus hojas de datos que para operar correctamente deben tener como voltaje de entrada mínimo 17.7V, así que se selecciona el mínimo transformador comercial que cubra esta recomendación.

En cuanto a la especificación de la corriente del transformador, dado que se va a utilizar un rectificador de puente de onda completa, se recomienda que dicha corriente sea como mínimo 1.8 veces la corriente máxima que demandará la carga [CoDr93], que para este caso es de 100.8 mA (esta corriente se requiere en el secundario).

Entonces:

$$I_{transformador} \geq 1.8(100.8mA)$$

$$I_{transformador} \geq 181.44mA \quad (3.31)$$

Aproximando a un valor comercial queda:

$$I_{transformador} = 300mA \quad (3.32)$$

#### *El Rectificador*

Se utilizará un rectificador de puente de onda completa que debe tener las siguientes características [CoDr93]:

- 1) Una corriente promedio mayor o igual a la corriente máxima que demandará la carga.
- 2) Un voltaje de pico inverso, PIV, mayor que  $V_m + 20\% V_m$ .

De la ecuación (3.30) se tiene que  $V_m = 21.21V$ , entonces  $PIV > 25.45V$ . Tomando en cuenta los valores comerciales, se eligió un puente rectificador con una corriente promedio de 1A y  $PIV = 50V$ .

### El Filtro

El filtro que se va a utilizar en la fuente está formado por un capacitor conectado en paralelo a la salida del rectificador. La figura 3.14 muestra el voltaje a través del capacitor  $C_1$ , que es el filtro de la fuente diseñada.

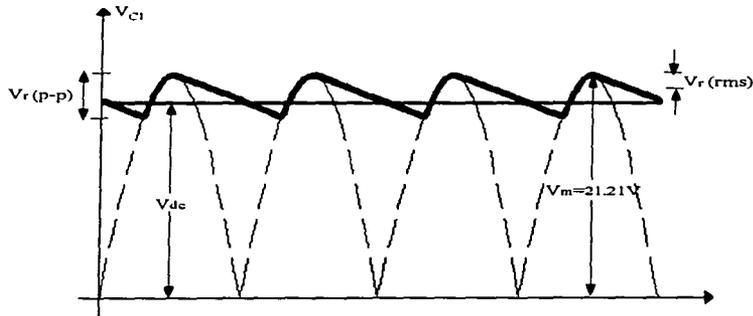


Figura 3.14. Voltaje a través del capacitor del filtro.

donde  $V_r(p-p)$  = Voltaje de rizo pico a pico  
 $V_r(rms)$  = Voltaje de rizo en rms  
 $V_{dc}$  = Componente de voltaje directo presente.

Además

$$V_r(p-p) = 2\sqrt{3}V_r(rms) \quad (3.33)$$

y

$$V_r(rms) = \frac{2.4I_{dc}}{C_1} \quad (3.34)$$

donde  $I_{dc}$  está en mA y C en  $\mu F$ .

El regulador en C.I. 7815 especifica en su hoja de datos que para regular el voltaje correctamente debe tener como voltaje de entrada mínimo 17.7 V, esto es:

$$V_{IN} \geq 17.7V \quad (3.35)$$

De la gráfica de la figura 3.9, se tiene que el mínimo voltaje que se puede presentar a la entrada del regulador es:

$$V_{IN(min)} = V_m - V_r(p-p) \geq 17.7 \quad (3.36)$$

sustituyendo las ecuaciones (3.18) y (3.19) en (3.21) se tiene

$$V_m - 2\sqrt{3} \frac{2.4I_{dc}}{C_1} \geq 17.7 \quad (3.37)$$

Como ya se explico, la máxima corriente que puede demandar la carga aplicada es de 100.8mA. Sustituyendo este valor en la ecuación (3.37) se tiene:

$$V_m - 2\sqrt{3} \frac{2.4(100.8)}{C_1} \geq 17.7$$

$$V_m - \frac{838.04}{C_1} \geq 17.7 \quad (3.38)$$

Sustituyendo el valor de la ecuación (3.30) en (3.38), el valor del capacitor  $C_1$  queda:

$$21.21 - \frac{838.04}{C_1} \geq 17.7,$$

$$\frac{838.04}{C_1} \leq 21.21 - 17.7,$$

$$C_1 \geq \frac{838.04}{21.21 - 17.7}$$

$$C_1 \geq 238.76 \mu F \quad (3.39)$$

Para aproximar el valor del capacitor obtenido en (3.39) a un valor comercial, se debe tener en cuenta la tolerancia del capacitor que se va a utilizar. Dado que se utilizarán capacitores electrolíticos y considerando que éstos en general tienen una tolerancia del 20%, se establece  $C_1$  al mínimo valor comercial que cubre las necesidades del diseño, esto es:

$$C_1 = 470 \mu F \quad (3.40)$$

En cuando a la especificación de voltaje, al igual que para el puente rectificador se recomienda que éste sea mayor que  $V_m + 20\% V_m$  [CoDr93]. Se establece entonces  $C_1$  de 1000 $\mu$ F a 50V.

### El Regulador

Para la etapa de regulación se seleccionaron los reguladores de voltaje fijo con tres terminales 7815 para la parte positiva y el 7915 (TO-220) para la parte negativa (ver Apéndice 2). Como ya se ha mencionado en párrafos anteriores, estos reguladores requieren un voltaje mínimo de entrada de +17.7V y -17.7V respectivamente para mantener la regulación correctamente. Estos encapsulados requieren además de un disipador de calor. Finalmente, el fabricante recomienda que se coloque un capacitor de tantalio de 0.1µF a la salida de estos reguladores para mantener la estabilidad de los mismos.

El voltaje de entrada en los reguladores es de 21.21 V mientras que la corriente que demandará el circuito es de 100.8 mA, lo que implica que la potencia que deberá disipar cada uno de ellos es de:

$$P = 21.21 \times 100.8 \times 10^{-3} = 2.14 \text{ W}$$

De la hoja de especificaciones de los reguladores, se tiene que su resistencia térmica es de 50 °C/W sin disipador de calor. Por otro lado, el rango de temperatura que pueden soportar es de 0 °C a 125 °C para el LM7915 y de 0°C a 150°C para el LM7815.

La temperatura que alcanzarán los reguladores es entonces:

$$50 \text{ °C/W} \times 2.14 \text{ W} = 106.9 \text{ °C.}$$

Este valor se encuentra dentro del rango de temperatura que éstos pueden soportar, por lo que se llega a la conclusión de que pueden operar sin necesidad de un disipador de calor.

La figura 3.15 muestra en conjunto todas las etapas diseñadas para la fuente de poder.

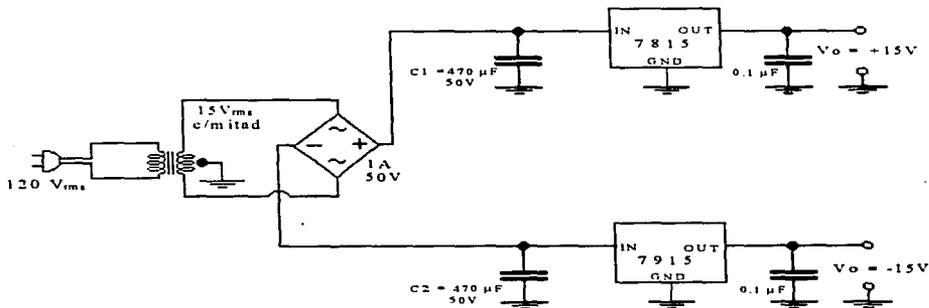


Figura 3.15. Fuente de poder regulada a ±15V.

### 3.1.3. Adquisición de Datos

Dentro del diseño del sistema, la etapa de adquisición de datos constituye el último segmento de la sección correspondiente al hardware. En términos generales, esta etapa se encarga de convertir la señal analógica de interés en una secuencia de palabras<sup>1</sup>, que representan de manera digital a dicha señal, y además coloca dichas palabras en el sitio adecuado (puede ser directamente la memoria de la computadora o bien, un registro del hardware que realiza la adquisición), lugar donde el software del sistema puede tomar los datos pertinentes sin ningún problema.

En el ámbito de ingeniería, se conoce por *adquisición de datos (DAQ<sup>2</sup>)* en su concepción más general, al proceso en el que una señal continua es muestreada, digitalizada y almacenada, normalmente en una computadora para ser procesada digitalmente, ya sea en tiempo real o en algún momento posterior. La adquisición de datos encuentra aplicación en diversas áreas de la ingeniería como medición, monitoreo y control de procesos, instrumentación, medición de señales biomédicas, etc.

Actualmente existe en el mercado una gran variedad de productos para adquisición de datos, su precio es accesible y son flexibles de adaptarse a diferentes tipos de aplicaciones específicas. Por otro lado, el diseño de una etapa de adquisición de datos envuelve una complejidad extrema, considérese que un equipo que realiza estas funciones debe contener circuitos para muestreo, de conversión analógico-digital, de temporización y conteo, de aislamiento de RFI y EMI<sup>3</sup>, de protección contra sobrevoltaje, de interfaz con la señal de entrada, de interfaz con el bus de la computadora, así como características especiales tales como alta impedancia de entrada, alta resolución, tiempo de conversión rápido, exactitud aceptable y una elevada integración de sus componentes. Cumplir con todas estas especificaciones al diseñar una etapa de adquisición de datos es, como se dijo, en extremo complejo, provocando así que se excedan los alcances de este trabajo tanto en tiempo como en recursos, de hecho, el diseño de un sistema de hardware que realice solamente esta función constituye en sí un proyecto de investigación suficientemente profundo.

Por todo lo mencionado en el párrafo anterior, se utilizara para la etapa de adquisición de datos del sistema PDS, un producto existente en el mercado que cumpla con las características necesarias, de acuerdo a la naturaleza de la señal que se manejará a la entrada y a la forma en que se requieren los datos a la salida, es decir, la forma en que la etapa de software tomará la señal ya muestreada y digitalizada.

De entre los productos existentes en el mercado (de ahora en adelante serán llamados tarjetas DAQ), se tomaron como conjunto de selección los productos que aparecen en la tabla 3.3 teniendo como base características fundamentales de los mismos; características que serán analizadas a continuación.

---

<sup>1</sup> En el vocabulario utilizado en computación, "palabra" denota una secuencia de bits de longitud fija que representa un dato.

<sup>2</sup> DAQ es el acrónimo de "Data Acquisition", equivalente en idioma inglés al término "adquisición de datos".

<sup>3</sup> RFI = Interferencia de Radio Frecuencia; EMI = Interferencia Electromagnética.

**Resumen de Características de tarjetas de adquisición de datos**

Modelo Tarjeta	No. de canales analógicos NI / Dig.	Ganancia Mín. / Máx.	Resolución (bits)	Exactitud Relativa	Frecuencia Máxima de Muestreo	Rango de Entrada		Marca	Precio (USD)	Software de tarjeta (precio en USD)
						Signaler (Volts)	Dispositivo (Volts)			
ACL-8112HG	16 / 08	x 5 / x1000	12	± 1 LSB	90 KHz	± 005 / ± 10	0 - 001 / 0 - 10	C.I.	389 00	ACLS - DLL2 89 00
ACL-8112DG	16 / 08	x1 / x8	12	± 1 LSB	90 KHz	± 1 25 / ± 10	0 - 1 25 / 0 - 10	C.I.	369 00	ACLS - DLL2 89 00
ACL-8112PG	16 / -	x1 / x16	12	± 1 LSB	100 KHz (1 ch.)	± 3125 / ± 10	N / D	C.I.	229 00	ACLS - DLL2 89 00
AR-83001	16 / 08	N / D	12 + 1 paridad	N / D	30 S/sec	± 5	N / D	C.I.	89 00	N / D
PCI-8112	16 / 08	x 5 / x8	12	± 1 LSB	110 KHz	± 1 25 / ± 10	0 - 1 25 / 0 - 10	C.I.	479 00	ACLS 79 00
ACL-8216	16 / 08	x1 / x8	16	N / D	87 KHz	± 1 25 / ± 10	0 - 1 25 / 0 - 10	C.I.	469 00	ACLS - DASK 139 00
ACL-8113	32 / -	N / D	12	± 1 LSB	25 KHz	± 1 25 / ± 10	0 - 1 25 / 0 - 10	C.I.	265 00	ACLS - DLL2 89 00
PCI-2042BW - 1	16 / 08	x1 / x100	12	N / D	100 KHz	N / D	N / D	I.I.	N / D	Master Link N / D
PCI-2009BC	16 / 08	N / D	12	N / D	100 KHz	N / D	N / D	I.I.	N / D	Master Link N / D
PC30 F / G	16 / 08	N / D	12	N / D	330 / 100 KHz	N / D	N / D	ET	325 00	N / D
ATMIO-16E-1	16 / 08	x 5 / x100	12	± 1.5 LSB	1 25 MS/s	± 05 / ± 10	0 - 1 / 0 - 10	N.I.	1 975 00	NI-DAQ Sin Costo
ATMIO-16E-2	16 / 08	x 5 / x100	12	± 1.5 LSB	500 KS/s	± 05 / ± 10	0 - 1 / 0 - 10	N.I.	1 755 00	NI-DAQ Sin Costo
ATMIO-64E-3	64 / 32	x 5 / x100	12	± 1.5 LSB	500 KS/s	± 05 / ± 10	0 - 1 / 0 - 10	N.I.	1 975 00	NI-DAQ Sin Costo
ATMIO-16E-10	16 / 08	x 5 / x100	12	± 1 LSB	100 KS/s	± 05 / ± 10	0 - 1 / 0 - 10	N.I.	1 095 00	NI-DAQ Sin Costo
ATMIO-16XE-10	16 / 08	x1 / x100	16	± 1 LSB	100 KS/s	± 1 / ± 10	0 - 1 / 0 - 10	N.I.	2 185 00	NI-DAQ Sin Costo
AT-AL-16XE-10	16 / 08	x1 / x100	16	± 1 LSB	100 KS/s	± 1 / ± 10	0 - 1 / 0 - 10	N.I.	1 645 00	NI-DAQ Sin Costo
ATMIO-16XE-50	16 / 08	x1 / x100	16	± 1 LSB	20 KS/s	± 1 / ± 10	0 - 1 / 0 - 10	N.I.	1 095 00	NI-DAQ Sin Costo
PC-LPM-16	16 / -	N / D	12	± 1.5 LSB	50 KS/s	N / D	N / D	N.I.	435 00	NI-DAQ Sin Costo

C.I. = Circuit Specialist Inc  
 I.I. = Intelligent Instrumentation  
 ET = Eagle Technology  
 N.I. = National Instruments

N / D = No Disponible

**Tabla 3.3. Características principales de las tarjetas DAQ del primer conjunto de selección.**

**Criterio de Selección**

Como se verá en los siguientes párrafos, las características de prácticamente todas las tarjetas cumplen o exceden los requerimientos del proyecto, además, obsérvese que mientras más sofisticada es una tarjeta, su precio se incrementa, por lo que uno de los criterios de selección que se utilizará consiste en desechar las tarjetas cuyas características sobrepasen por mucho los requerimientos para el tipo de señal que se manejará, disminuyendo así el rango de precios a considerar; para ello, se analizará característica por característica.

**No. de canales de entrada analógica.**- Como ya se mencionó anteriormente, dentro de este mismo capítulo, se manejarán un total de 16 electrodos para analizar un total de 16 señales EEG; así mismo, todas las señales estarán referenciadas a un solo punto en común, por lo que no se requerirán entradas diferenciales. Como se observa en la tabla, casi todas las tarjetas tienen 16 entradas analógicas sencillas, eliminando solamente a los modelos ACL-8113 y ATMIO-64E-3.

**Ganancia Mínima / Máxima.**- Como se vio en el apartado de acondicionamiento de la señal, la entrada original se amplifica previamente desde el orden de los microvolts hasta un nivel máximo de 200 mV. Se escogió esta magnitud de amplificación a propósito debido a que, como se mencionó en el apartado 2.2.5, la resolución de un convertidor A/D se incrementa cuando el rango

de entrada (escala completa) disminuye; por tanto, tomando como base los rangos más pequeños de entrada que pueden manejar las tarjetas (véase tabla 3.3), se tiene que la señal ya entra a la tarjeta con un nivel de voltaje adecuado, no siendo necesario un nivel de ganancia alto por parte de la tarjeta. Por tanto, se desecha la tarjeta que figura por su nivel de ganancia extremadamente alto, la ACL-8112HG.

**Resolución.**- Prácticamente todas las tarjetas manejan una resolución de 12 bits, que, como se recordará del apartado 3.1.2 da un cambio mínimo a la salida de  $24.41 \mu\text{V}$ , que comparado con los 200 mV de salida máxima, representa un valor aceptable. Aquí se podrían eliminar las tarjetas de 16 bits de resolución, pero se considerarían otros factores antes de hacerlo.

**Exactitud Relativa.**- Este dato es prácticamente el mismo para todo el conjunto de selección, siendo  $\pm 1 \text{ LSB}$  un valor normal de exactitud para un convertidor A/D. Por lo anterior, se eliminan solo aquellas tarjetas que manejan niveles inferiores de exactitud, como las ATMIO-16E1 y E2 así como la PC-LPM-16.

**Frecuencia Máxima de Muestreo.**- La frecuencia de la señal que se manejará no excede los 30 Hz, por otro lado, el teorema de muestreo pide al menos el doble de la frecuencia máxima de la señal para obtener una muestra significativa; por lo tanto, se necesita una frecuencia de muestreo de al menos 60 Hz, que como se observa en la tabla 3.3, es un valor por debajo de lo que las tarjetas comerciales pueden manejar a excepción del modelo AR-B3001, por lo que se elimina del conjunto. De igual manera, se eliminan las tarjetas que manejan niveles superiores y por tanto sobrados, como la PC30/F y las ATMIO-16E-1, 16E2 y 64E3. El modelo ACL-8112PG se elimina porque el dato disponible se refiere a un solo canal, no a los 16 requeridos.

**Rango de Entrada.**- El tipo de señal que se maneja es unipolar, lo que significa que todos los valores que puede tomar son superiores a cero. Así las cosas, se descarta para la selección el rango de entrada bipolar. Como se mencionó anteriormente, se utilizará un rango de entrada de 0 a 200 mV. No se eliminan posibilidades de acuerdo a este criterio debido a que primero se observaron las capacidades de las tarjetas y posteriormente se diseñó la etapa de amplificación, sin embargo, se pueden eliminar aquellas tarjetas que, no proporcionan este valor, el cual es importante (la PCI-20428W-1, PCI-20098 y PC 30/G).

El segundo criterio de selección y no menos importante es la consideración de la marca, el precio y el software de interfaz. De acuerdo a la marca y tomando como base la variedad de modelos, la documentación y soporte de los fabricantes, junto a otros aspectos por el estilo, se consideraron las mejores marcas a *National Instruments* y a *Circuit Specialist Inc.*, siendo la primera definitivamente superior teniendo en cuenta su presencia en el mercado, su nivel de estandarización y el soporte hacia sus productos.

En cuanto al precio de las tarjetas de adquisición de datos, se observa que las de la marca *Circuit Specialist* están por debajo de las de *National Instruments*, sin embargo, como se verá más adelante, existen criterios de más peso que el costo para seleccionar una tarjeta. Además, hay que considerar que para las tarjetas de *Circuit Specialist* el software de interfaz tiene un costo extra, mientras que para *National Instruments* este software se incluye con la tarjeta sin costo adicional.

Finalmente, se considerara el software de interfaz. Este es un aspecto importante de tomar en cuenta antes de elegir una tarjeta DAQ. El software de interfaz es el encargado de explotar al máximo las capacidades de una tarjeta, además determina en gran manera el alcance de las aplicaciones que se le puede dar a la misma. Por principio de cuentas, hay que hacer notar que existen cinco clases distintas de software de interfaz para tarjetas DAQ, dichas clases se definen a continuación:

**Tipo TurnKey.**- Es la clase de software más fácil de utilizar. No necesita programación y requiere poco tiempo de configuración. Controlan bien una aplicación, no más. Frecuentemente la interfaz emula un instrumento de medición. No se necesita un lenguaje de programación para desarrollar una aplicación.

**Interfaz con lenguaje de programación.**- Consisten de una colección de rutinas de adquisición de datos o llamadas a funciones para lenguajes de programación como Pascal y C. Es necesario escribir código para utilizar las llamadas a funciones y obtener los datos necesarios. Con estos datos en memoria, ya se puede seguir programando para analizar y graficar los mismos.

**Herramientas Add-on.**- Adjunta las capacidades de DAQ a herramientas conocidas como hojas de cálculo. VBX diseñados para Visual Basic pueden ser considerados como este tipo de herramientas, difiriendo de las interfaz de lenguaje en que son archivos DAQ añadidos a una aplicación Visual Basic; aunque muchas veces se utilizan las dos modalidades, confundándose una con otra.

**Código Fuente.**- Es código usualmente en lenguaje C que puede ser compilado junto con una aplicación para adquirir y controlar datos. Es la clase más compleja de software DAQ que hay debido a la programación necesaria para obtener resultados.

**Instrumentos Virtuales.**- Permite al desarrollador preparar un instrumento para visualizar la señal que le interese de la mejor manera. Ejemplo de este tipo de software es LabView. Tienen la desventaja de tener un costo elevado (LabView se cotiza por encima de los \$2,500.00 USD).

Debido a razones de diseño de software que se explicarán a detalle en el apartado 3.2, se llegó a la conclusión de que el tipo más conveniente para utilizar en el diseño del sistema es la interfaz con lenguajes de programación, debido entre otras cosas a la flexibilidad que se obtendrá con ello además de una relativa facilidad de uso. En la tabla 3.3 se muestra listado el tipo de software de interfaz con lenguajes que cada tarjeta puede utilizar, de la misma tabla se observa que básicamente se trata de dos posibilidades: *ACLS-DLL2* y *NI-DAQ*, de *Circuit Specialist* y *National Instruments* respectivamente. En general, los dos tienen características parecidas, los dos son bibliotecas de funciones en forma de DLL's<sup>4</sup> utilizables por diferentes lenguajes de programación, sin embargo las características de *NI-DAQ* son en conjunto superiores a las de *ACLS-DLL2*.

---

<sup>4</sup> DLL significa "Dynamic Link Library" o Biblioteca de Ligado Dinámico. Contiene funciones que son llamadas por los programas en Windows al tiempo de ejecución.

En especial, las características que provocaron la decisión por *NI-DAQ* es su amplia documentación, su disponibilidad y precio, su compatibilidad con gran número de tarjetas y sobre todo, que contiene funciones especiales que resultan ser adecuadas para el tipo de aplicación en que se utilizara, tales como funciones de *DAQ* en doble buffer, característica muy importante para adquirir los datos y visualizarlos en tiempo real; otro tipo de funciones son las de escritura a disco, lo cual como se verá en el siguiente apartado, resultará de gran utilidad para el análisis posterior de la señal.

Como se podrá ver, se ha escogido definitivamente una tarjeta de la marca *National Instruments*; de entre los modelos que quedan por escoger (*ATMIO-16E-10*, *16XE10*, *16XE-50* y *AT-AI-16XE-10*) se considera que tienen características similares y cualquiera de ellas es adecuada para las necesidades del sistema, sin embargo se inclina la decisión por el modelo de tarjeta ***ATMIO-16E-10***, cuya característica es que maneja 12 bits de resolución, más que suficiente para las necesidades señaladas anteriormente. Otro criterio de mucho peso para elegir esta tarjeta es el precio de la misma, que como se puede observar en la tabla 3.3 es la más económica del grupo final de selección. En la figura 3.16 se muestra un diagrama de la tarjeta elegida en el que se observa donde se realizan las funciones básicas de muestreo y conversión A/D.

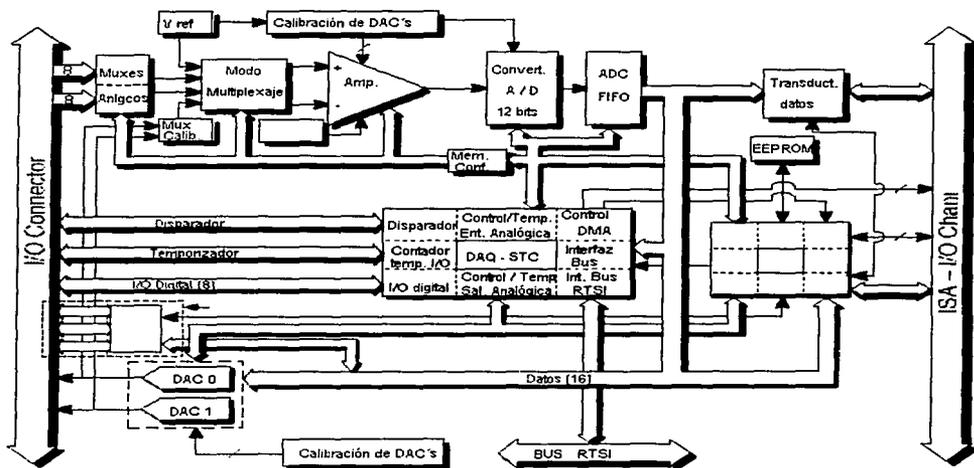


Figura 3.16. Diagrama de bloques de la tarjeta DAQ ATMIO-16E-10

Para acoplar la etapa de acondicionamiento de la señal con la tarjeta de adquisición de datos, se deberá utilizar un conector tipo DIN modelo **CB68-LP** de *National Instruments*. En la

figura 3.17 se muestra la designación de cada pin del conector. Nótese que esta misma designación se aplica para la entrada de la tarjeta DAQ.

ACH8	34	58	ACH0
ACH1	33	57	AIGND
AIGND	32	56	ACH9
ACH3	31	55	ACH2
AIGND	30	54	AIGND
ACH4	29	53	ACH11
AIGND	28	52	AISENSE
ACH13	27	51	ACH12
ACH6	26	50	ACH5
ACH10	25	49	AIGND
AIGND	24	48	ACH14
ACH15	23	47	ACH7
DAC0OUT	22	46	AIGND
DAC1OUT	21	45	AOGND
EXTREF	20	44	AOGND
DIO4	19	43	DGND
DGND	18	42	DIO0
DIO1	17	41	DIO5
DIO6	16	40	GND
DGND	15	39	DIO2
5 V	14	38	DIO7
DGND	13	37	DIO3
DGND	12	36	SCANLCK
PF10/TRIG1	11	35	EXTSTROBE*
PF11/TRIG2	10	34	DGND
DGND	9	33	PF12/CONVERT*
5 V	8	32	PF13/GPCTR1_SOURCE
DGND	7	31	PF14/GPCTR_GATE
PF15/UPDATE*	6	30	GPCTR1_OUT
PF16/AFTRIG	5	29	DGND
DGND	4	28	PF17/STARTSCAN
PF19/GPCTR0_OUT	3	27	PF18/GPCTR0_SOURCE
GPCTR0_OUT	2	26	DGND
FREQ_OUT	1	25	DGND

Figura 3.17. Conector I/O de la tarjeta DAQ ATMIO-16E-10.

### 3.2 Diseño del Software

Es bien conocido que las representaciones gráficas sirven para mejorar la comunicación de información. Es este hecho el que se utiliza para transmitir información acerca del comportamiento eléctrico del cerebro. El sistema a desarrollar tendrá la capacidad de presentar en pantalla la información proveniente del sistema de adquisición de datos, el desplegado selectivo de canales con información original o aplicando algún proceso de tipo digital, y permitir la impresión de

zonas de interés. Esta presentación incluye la señal en dominio del tiempo y de la frecuencia, es decir la señal original y una transformación de la misma; y presentación de la señal en el montaje mismo, con una gama de colores indicando en que zonas se encuentra presente un determinado ritmo, identificando con colores previamente convenidos, las diferentes magnitudes que pudiera tener el mismo. Lo anterior tiene el propósito de ayudar al experto con información adicional que no es evidente en el dominio del tiempo.

Resumiendo, se identifica que el sistema debe realizar las siguientes funciones:

- Verificar, mediante visualización en pantalla, el montaje de electrodos.
- Graficar las señales.
- Realizar anotaciones a la señal.
- Análisis espectral.
- Mapeo electroencefalográfico.
- Almacenar la información en memoria secundaria.
- Impresión.

El diagrama de la figura 3.18 muestra la forma en que interactúan entre si todas las funciones del sistema y a continuación, con base en la misma figura, se presenta una descripción del diseño conceptual, del diseño técnico y del diseño de los programas para cada una de estas funciones.

### **3.2.1 Diseño conceptual**

#### *Montaje de Electrodos*

Previo a la visualización en tiempo real de las señales EEG, el usuario podrá verificar que el montaje de electrodos realizado sobre el paciente sea correcto. Esta opción permitirá visualizar en pantalla un esquema prototipo mostrando un cráneo bidimensional con círculos que muestren las posiciones para los electrodos de acuerdo a la técnica 10-20 y destacando sobre ésta el montaje, de tal forma que el técnico responsable pueda darse cuenta, mediante mensajes específicos del sistema, si omitió colocar algún electrodo o bien si alguno de ellos está haciendo mal contacto. Además, para el caso de señales reproducidas, es decir, aquellas que hayan sido previamente almacenadas, el sistema permitirá visualizar el montaje que dio origen a dichas señales.

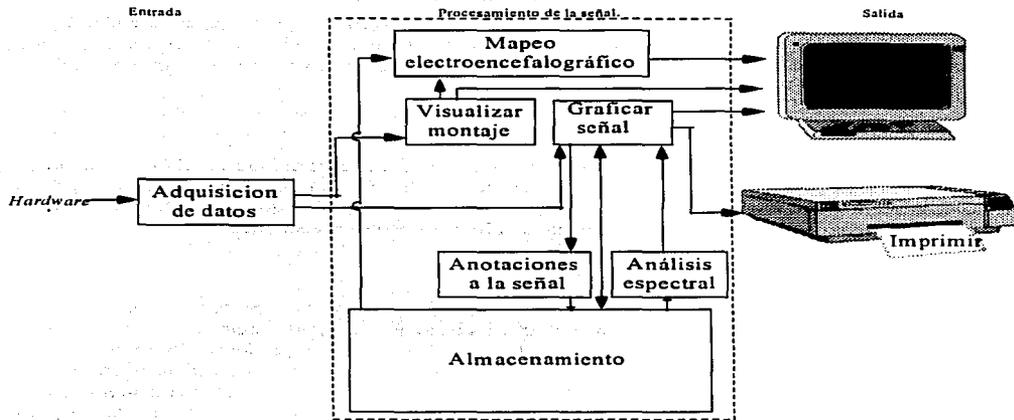


Figura 3.18. Diagrama de bloques para el diseño del software.

### *Graficar las Señales*

El sistema será capaz de graficar las señales EEG tanto en tiempo real como almacenadas previamente en disco. En ambos casos, será posible visualizar en una misma pantalla todos los canales que se encuentren captando la señal o bien, aquellos que el usuario elija visualizar de manera específica. Para iniciar la graficación de las señales en tiempo real, el usuario deberá activar el módulo de graficación eligiendo la opción en el menú respectivo. En el caso de las señales previamente almacenadas, el mecanismo es similar, sólo que el usuario deberá cargar la señal desde disco antes de visualizarla. De la misma forma, el detener la visualización estará bajo control del usuario. El sistema dará al usuario la facilidad de modificar la escala en el eje horizontal para el caso de señales graficadas en el dominio del tiempo, y modificar la escala en ambos ejes, **X** y **Y**, para el caso de señales graficadas en el dominio de la frecuencia.

El sistema también será capaz de graficar señales producto de algún proceso aplicado a la señal original, como es el caso del análisis espectral (gráfica de la señal en el dominio de la frecuencia) o el mapeo electroencefalográfico por colores.

### *Anotaciones a la Señal*

Estas anotaciones se refieren a poder establecer algún comentario sobre un punto específico del conjunto de señales graficadas en tiempo real o reproducidas desde algún archivo. Esta necesidad surge debido a que durante un EEG, llega a haber factores o estímulos externos que pueden alterar al paciente y provocar distorsión en las señales de salida. Estos estímulos

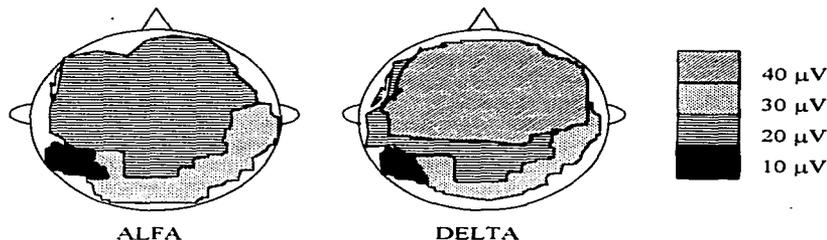
pueden ser artefactos o bien estímulos visuales de ritmos específicos provocados por el técnico. De cualquier forma, se requiere llevar nota de todo lo que acontece en la sala, relacionándolo con determinados instantes de muestreo a efecto de contar con información complementaria del EEG que evite conclusiones inadecuadas por parte del médico.

### *Análisis Espectral*

El análisis espectral consiste básicamente en estudiar una determinada onda en el dominio de la frecuencia, esto es, se trabaja con el espectro de la señal. El sistema mostrará dicho espectro en una gráfica de frecuencia contra ganancia. Es condición necesaria que la señal a analizar se encuentre almacenada en disco, es decir que haya sido registrada previamente.

### *Mapeo Electroencefalográfico*

El sistema permitirá visualizar una representación gráfica de la cantidad y localización de cada uno de los ritmos EEG que se presenten en cierto periodo, es decir, mostrar la cabeza dividida en zonas según la magnitud en volts de la frecuencia que se represente, como se muestra en la figura 3.19. Para esto se tendrá una escala de colores predeterminada identificando cada magnitud con un color específico. Para construir una gráfica de este tipo se debe considerar como entrada un conjunto de señales EEG, correspondientes cada una a un electrodo de una muestra llevada a cabo con anterioridad y previamente almacenada en disco.



**Figura 3.19.** Mapeo electroencefalográfico.

### *Almacenamiento en Memoria Secundaria*

El sistema almacenará las señales EEG que estén siendo muestreadas en periodos de tiempo que el usuario indique, en algún medio de almacenamiento secundario. Para iniciar o detener el almacenamiento de las señales, el usuario deberá activar o desactivar el módulo de almacenamiento eligiendo la opción en el menú respectivo. Junto con la información proveniente de todos los canales, se almacenarán las anotaciones asociadas a dichas señales, el número de expediente del paciente, fecha y hora del estudio y el montaje utilizado para el EEG.

Además el usuario podrá almacenar si lo desea, las señales producto de haber aplicado el análisis espectral a las señales EEG originales.

El sistema contará con un módulo para facilitar la reproducción de señales previamente grabadas, permitiendo al usuario utilizar criterios de selección, tales como fecha del estudio o número de expediente para localizar un estudio en particular.

### Impresión

Eventualmente, el usuario puede elegir imprimir alguna señal visualizada de forma tabular o como se presenta en pantalla; para esto, el sistema permitirá imprimir zonas selectivas de alguna señal, ya sea una parte de la señal original previamente almacenada o la resultante de aplicar una función específica sobre ésta. También se podrán imprimir las anotaciones realizadas a las señales y las imágenes resultantes del mapeo electroencefalográfico por colores.

### 3.2.2 Diseño técnico

Con base en un diagrama a bloques, figura 3.20, se presenta la estructura jerárquica que tendrán las funciones del sistema, en donde se puede ver cómo interactúan entre sí dichas funciones, para posteriormente mostrar por cada una de estas funciones como será el flujo de información en las mismas y las sub-funciones o rutinas que intervendrán en cada una de ellas.

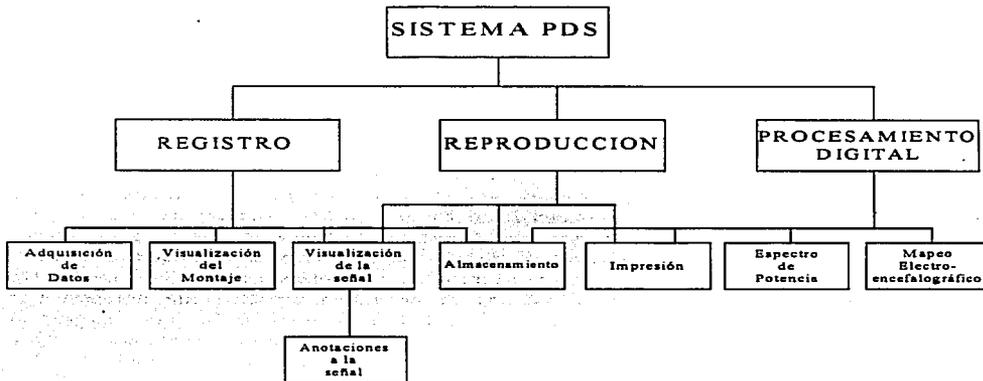


Figura 3.20. Jerarquía de las funciones del sistema.

En la figura 3.21 se muestra como está conformada la función de registro de la señal proveniente del hardware. Es en esta función donde se leen las muestras de la señal, mismas que se almacenan temporalmente en *buffers* de memoria de la computadora de donde se toman para

visualizar, primero que nada, su presencia en cada uno de los electrodos montados. De aquí también se deriva el procedimiento de tomar los datos y conformar una gráfica de amplitud contra tiempo en la pantalla de la computadora para cada una de las señales que se estén registrando. Estas mismas muestras que se obtienen de la memoria pueden ser tomadas para realizar el proceso de almacenar en memoria secundaria la señal. Así también para la ocasión que el usuario del sistema lo determine, se contará en esta parte con el proceso que permitirá almacenar anotaciones de texto junto con la señal en el dispositivo de almacenamiento.



Figura 3.21. Flujo de la información en la etapa de registro de la señal.

A continuación en el diagrama de la figura 3.22 se muestra cómo estará conformada la función de procesamiento digital de las señales. Aquí se observa que es a partir de una señal previamente almacenada de donde se tomarán los datos o muestras discretas de la señal para aplicarles primeramente el algoritmo numérico de la transformada rápida de Hartley, para obtener otra serie numérica de datos que representen a la misma señal pero en el dominio de la frecuencia -espectro de potencia-, de la cual también se mostrará una representación gráfica en pantalla. De manera semejante mediante algoritmos numéricos se obtendrán espectros de potencia por separado de los diversos rangos de frecuencias -anchos de banda- determinados, que representan los ritmos electroencefalográficos característicos (alfa, beta, teta, etc.), representando la posición y cantidad de cada uno de ellos en la cabeza, por colores, en la pantalla de la computadora.

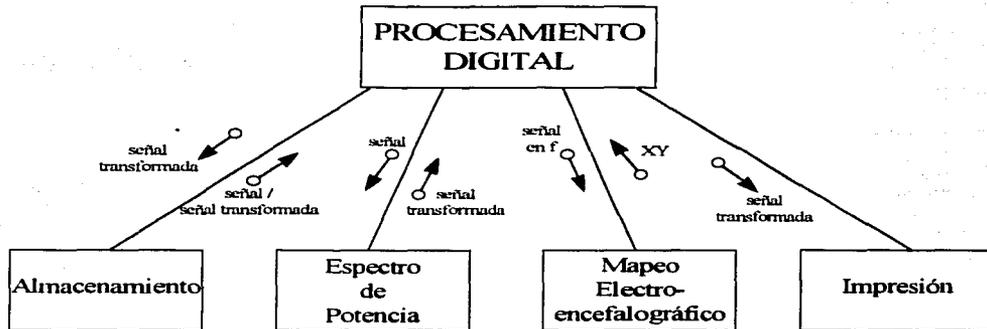


Figura 3.22. Flujo de la información en la etapa de procesamiento digital de la señal.

En el diagrama de flujo de información mostrado en la figura 3.23 se puede ver cómo estará conformada la función de reproducción de una señal almacenada. Esto es, que una señal almacenada, ya sea en su estado original o la resultante de una transformación, será leída por la función y puesta en forma de gráfica (ya sea de tiempo o de frecuencia) en la pantalla de la computadora, o si así se desea en una salida de impresión en papel.

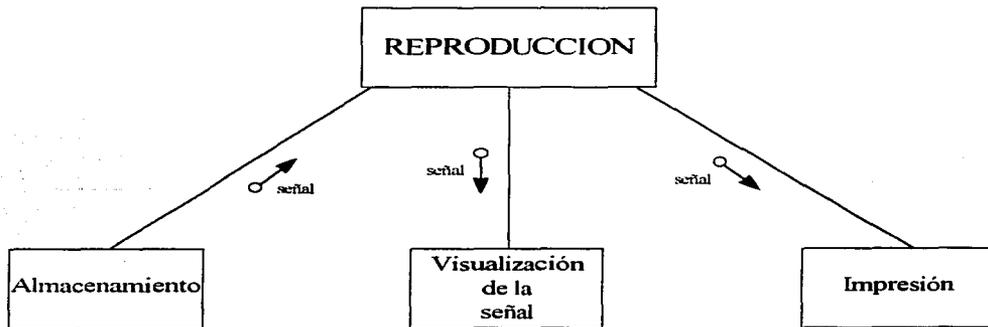


Figura 3.23. Flujo de la información en la etapa de reproducción de la señal.

### 3.2.3 Diseño de Programas

En esta parte se muestra el desglose de las funciones a programarse en el sistema y sus aproximaciones en *pseudocódigo* a la codificación final.

#### *Función: Adquisición de Datos*

La adquisición de datos constituye en sí una subfunción de la función de registro definida anteriormente. Su utilidad principal es servir de enlace entre la tarjeta de adquisición de datos y el sistema de software mediante el control del muestreo, la obtención de los datos (la tarjeta los coloca en un *buffer* en memoria) y la transferencia de éstos a la función que los haya solicitado. Para lograr esto, esta función necesita tener una buena interacción con las funciones de adquisición de datos que provee el programa de interfaz NI-DAQ, en otras palabras, desde esta función se gobierna el inicio y final de los muestreos, así como la configuración y las propiedades de éstos, utilizando en gran medida las funciones que para tal efecto proporciona el programa de interfaz antes mencionado.

Se comenzará el diseño de esta función con la determinación y jerarquización de sus funciones, en la figura 3.24 se observa dicha jerarquía de una manera global.

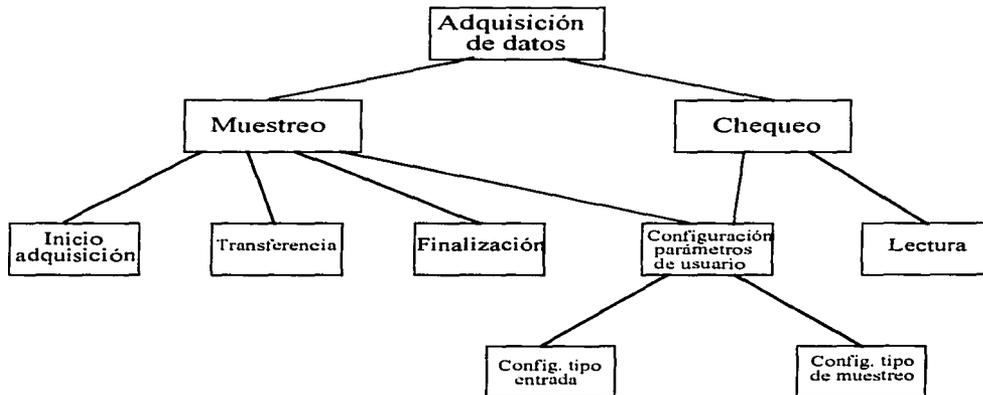


Figura 3.24. Descomposición de la función de adquisición de datos.

Con el fin de darle mayor claridad al diseño y de simplificar la explicación de todas estas funciones, se presentan a continuación los diagramas de estructura de cada una de ellas mostrando sus entradas y salidas principales, tanto datos como señales de control. Se describen dichos diagramas partiendo de lo general a lo particular siguiendo la convención de diseño adoptada, que es la aproximación "top-down".

En la figura 3.25 se muestra el diagrama de estructura de nivel 0, es decir, el más general. En éste se observa que la función de adquisición de datos realiza dos funciones principales: muestreo y chequeo. La entrada que recibe la función de adquisición es una indicación de inicio o final de adquisición y la salida es una señal en formato binario de 8 bits (FB).

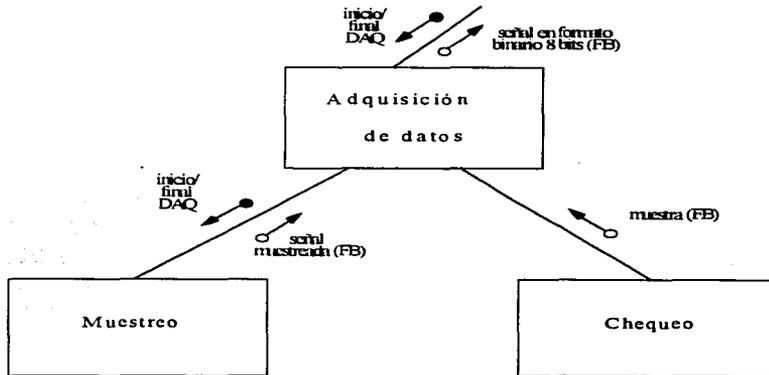


Figura 3.25. Diagrama de estructura a un nivel para la función de adquisición de datos.

La función de muestreo es la que realiza todas las funciones necesarias para llevar a cabo la adquisición de datos; al hacer uso de esta función, la función que la llama debe indicar a ésta que debe iniciar la adquisición de datos, de la misma manera, se le debe especificar cuándo debe detener el proceso. La función entregará a la salida la señal digitalizada en formato binario de 8 bits (FB), para ello se utilizará un arreglo de bytes de longitud fija determinada, en el que cada byte representa una muestra de la señal. Nótese que lo anterior implica que el procedimiento que utiliza a esta función debe realizar al menos tres llamadas: una para iniciar la adquisición de datos, un número variable de llamadas, para obtener el arreglo lleno de datos digitalizados mientras dure la adquisición, y finalmente, una para detener la adquisición una vez que no sea necesario seguir analizando la señal.

Mientras tanto, la función de chequeo se encarga de hacer lecturas individuales de datos con el fin de "chequear" si existe señal en algún canal en específico, no importando la frecuencia de la misma; esta función será utilizada por la función de visualización del montaje, la cual será descrita más adelante. Esta función regresa un solo valor, es decir, una muestra de la señal en formato binario de 8 bits.

La función de muestreo está compuesta de 4 funciones básicas, como se muestra en el diagrama de estructura de la figura 3.26. La función de configuración de parámetros de usuario se encarga de inicializar el proceso de adquisición de datos con los valores adecuados antes de llevar

a cabo cualquier operación de muestreo de señal en alguno de los canales. Para ello requiere como entrada el número de canales a muestrear y la velocidad de muestreo.

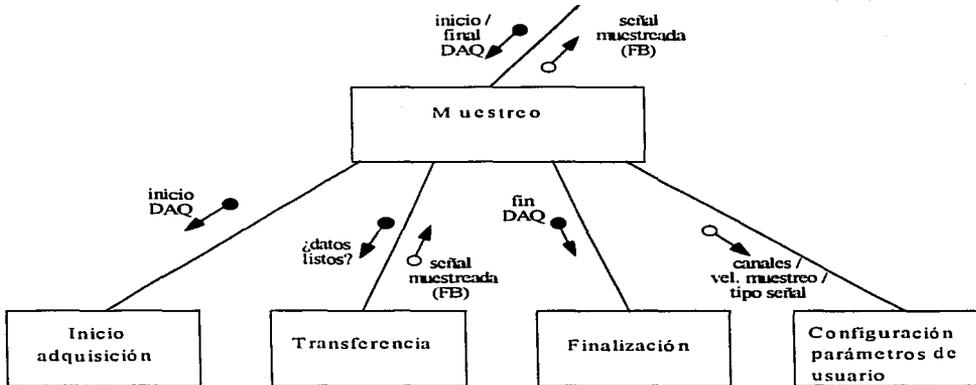


Figura 3.26. Diagrama de estructura de la función de muestreo.

La función de inicio de adquisición se encarga de indicarle a la tarjeta de adquisición de datos que debe comenzar a muestrear y almacenar los datos en un *buffer*. Como entrada debe recibir ciertos datos requeridos como número de canales a muestrear, ganancia en cada uno, etc. La función llamada transferencia se encarga de vaciar el *buffer* entregado por la tarjeta de adquisición de datos y proporcionar los valores obtenidos en un arreglo de bytes de longitud fija a la función que lo solicite, previa verificación de éste, si es que los datos se encuentran listos para ser transferidos. La función de finalización se encarga de indicarle a la tarjeta que debe finalizar el muestreo de la señal previamente solicitado, además, se encarga de reinicializar las banderas y el circuito en general. Finalmente en esta etapa se encuentra la función de configuración de parámetros de usuario para cada estudio en particular y es aquí donde se inicializan los parámetros que se necesitan para la operación en general como cuántos y cuáles electrodos se desean conectar.

La otra función que contiene la función de adquisición de datos es la de chequeo, cuyo diagrama de estructura se muestra en la figura 3.27. Dos son las funciones que conforman esta función: configurar los parámetros de usuario para este registro y realizar la lectura. La primera función ya fue descrita en párrafos anteriores, sólo que en la primera sólo se especifica cómo y qué canales se van a leer, mientras que la segunda función se encarga de indicarle a la tarjeta de adquisición de datos que debe realizar una sola lectura en los canales indicados en la función de configuración, y devolver la muestra leída en un arreglo de bytes, donde cada elemento representa una lectura en el canal correspondiente.

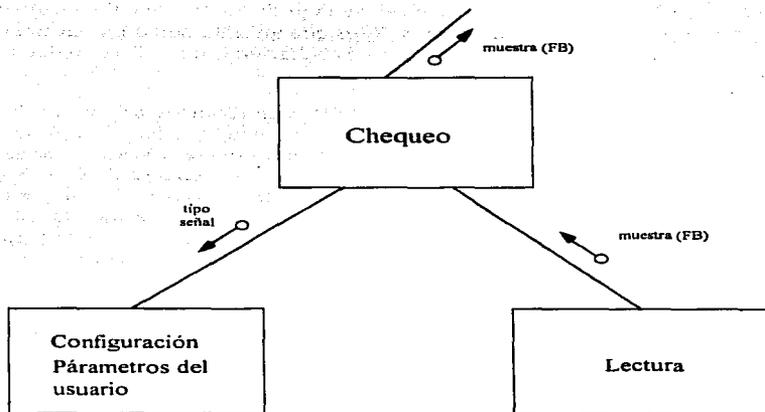


Figura 3.27. Diagrama de estructura de la función de chequeo.

Analizando el siguiente nivel de la jerarquía definida para la función de adquisición de datos, se tiene, que la función de configuración de parámetros consta de una función principal: la configuración para el tipo de muestreo. Esto se muestra en el diagrama de estructura de la figura 3.28.

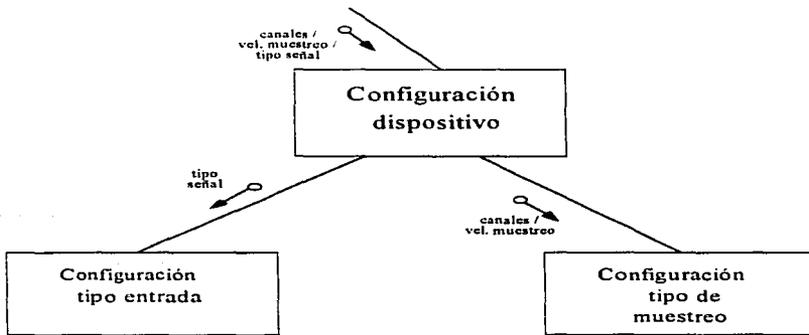


Figura 3.28 Diagrama de estructura de la función de configuración del dispositivo.

La función asociada a esta función principal de adquisición es la configuración del tipo de muestreo, la cual proporciona al sistema ciertos valores que utilizará como por ejemplo el tipo de disparo a utilizar, el origen de los pulsos de reloj y la habilitación o inhabilitación del muestreo en modo de doble buffer.

Una vez descrita la función de adquisición de datos en términos de sus funciones que la conforman, a continuación se muestra el diseño de cada una de éstas. La descomposición funcional indica qué es lo que hace la función principal, mientras que el diseño mostrado indica los mecanismos que se utilizarán para llevar a cabo las funciones dependientes de ésta. Nuevamente, con el fin de proporcionar mayor claridad se muestra un diagrama general de las dos funciones principales de esta función y posteriormente el desglose de cada función de manera individual utilizando una aproximación "top-down". En la figura 3.29 se muestra el punto de partida de este análisis, con las 2 funciones básicas que realiza la función de adquisición de datos.

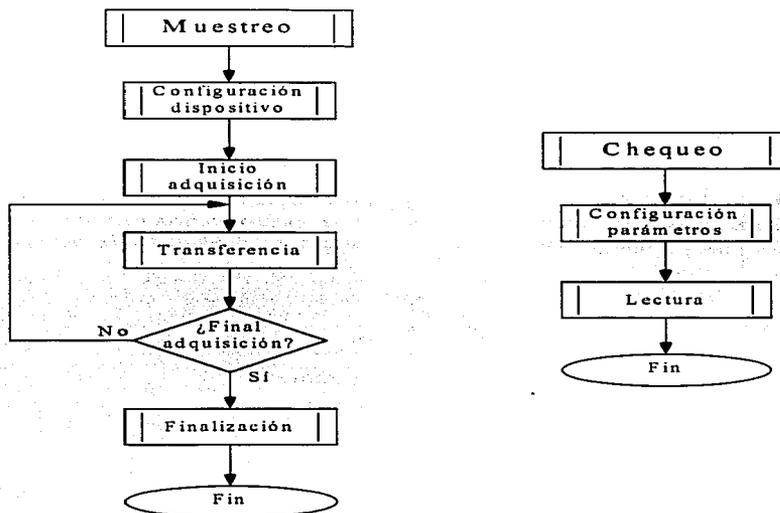


Figura 3.29. Diagrama de flujo de las funciones de muestreo y chequeo.

En la figura 3.30 se muestra el diagrama de flujo de la función básica de configuración.

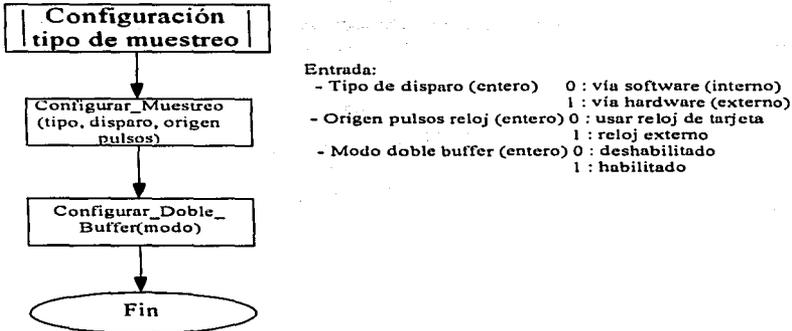


Figura 3.30. Diagrama de flujo para la configuración del tipo de muestreo.

La función de inicio de adquisición ha sido diseñada como lo muestra la figura 3.31.

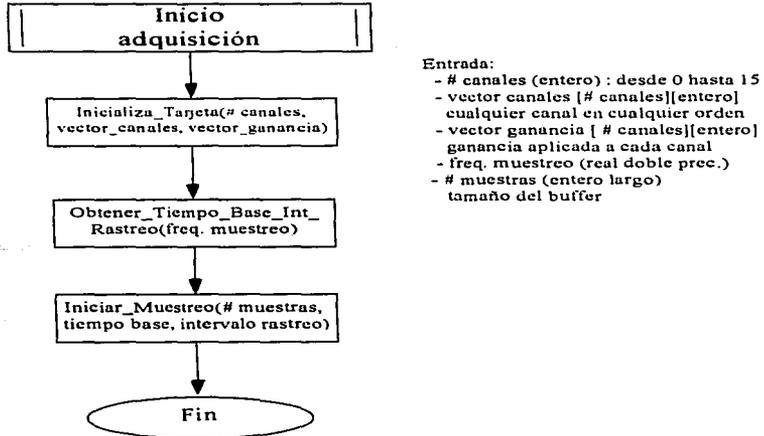
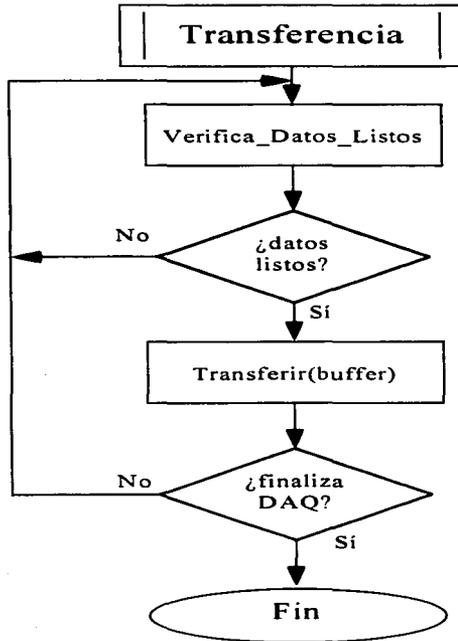


Figura 3.31. Diagrama de flujo de la función de inicio de adquisición.

La función de transferencia tiene gran importancia en la operación del sistema, ya que se encarga de proporcionar los datos muestreados a la función que lo solicite. El mecanismo para transferirlos se muestra en la figura 3.32.



Salida:  
- Buffer [# muestras][byte]

Figura 3.32. Diagrama de flujo para la transferencia de datos.

Cuando concluye el proceso de muestreo, es necesario indicarlo explícitamente a la tarjeta mediante una función cuyo diagrama de flujo se muestra en la figura 3.33.

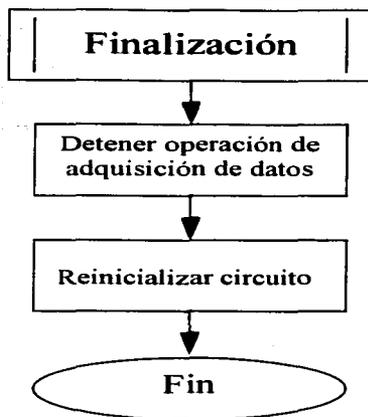


Figura 3.33. Diagrama de flujo de la función de finalización.

Finalmente, sólo resta describir el mecanismo utilizado para caracterizar la función de lectura de la función superior de chequeo, lo cual se representa en el diagrama de la figura 3.34.

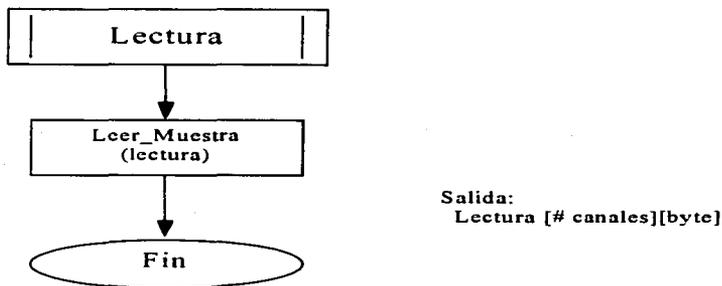
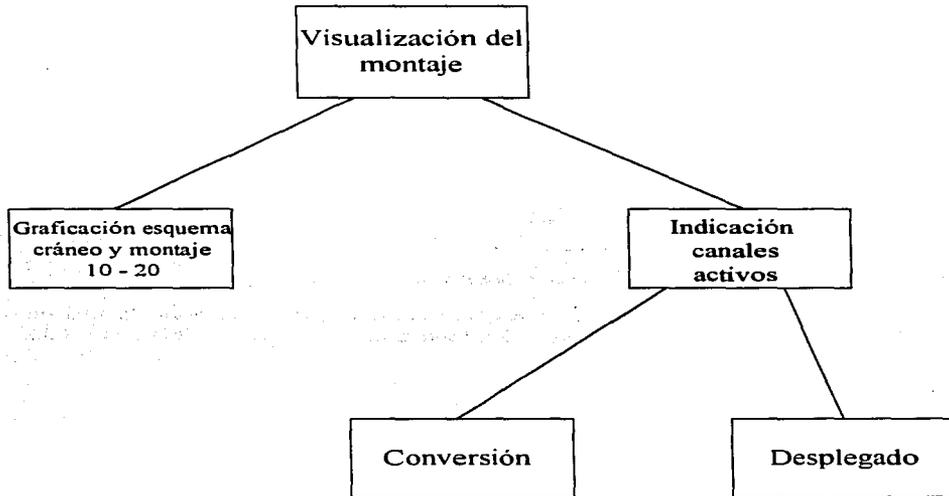


Figura 3.34. Diagrama de flujo para la función de lectura de muestra.

***Función: Visualización del montaje de electrodos.***

Esta función se encarga de proporcionarle al usuario un medio visual para corroborar el estado del montaje de electrodos utilizado por él. Mediante una indicación luminosa en una representación gráfica del cráneo, se establece cuáles electrodos están correctamente conectados y proporcionando señal y cuáles no lo están haciendo. En la figura 3.35 se muestra una estructura jerárquica de las funciones necesarias para que esta función principal cumpla con su cometido.



**Figura 3.35. Descomposición funcional de la función principal de visualización del montaje.**

Dos son las funciones básicas de la función principal de visualización del montaje, en el diagrama de estructura de la figura 3.36 se observan dichas funciones: graficación del esquema del cráneo y montaje 10-20 e indicación de canales activos.

La graficación del esquema del cráneo y del montaje 10-20 se encarga de cargar una imagen previamente almacenada, la cual representa la vista superior de un cráneo humano, así como una indicación del lugar donde deben conectarse los electrodos de acuerdo al montaje 10-20, que es el estándar en mediciones electroencefalográficas. La función de indicación de canales activos se encarga de representar cromáticamente aquellos electrodos que están recibiendo señal correctamente y aquellos que no lo están haciendo.

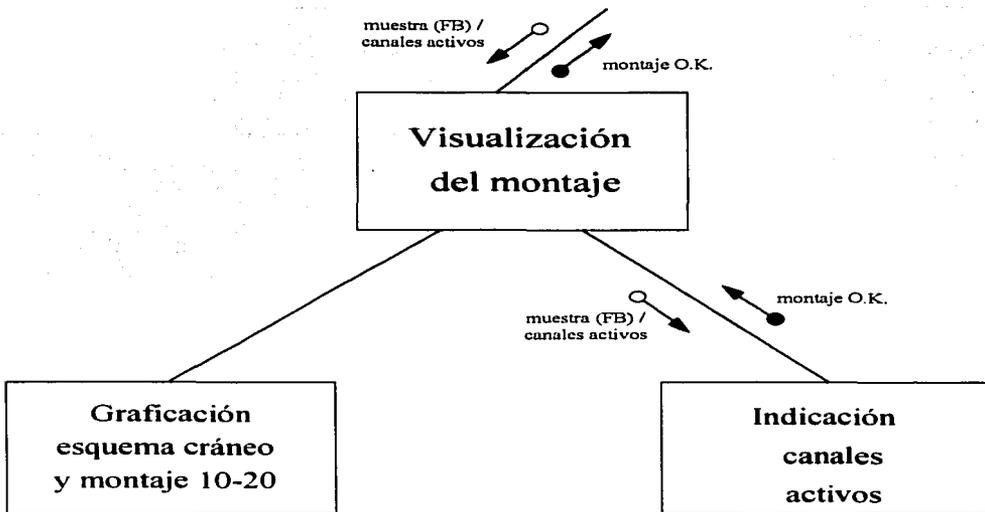


Figura 3.36. Diagrama de estructura de la función de visualización del montaje.

Para conocer el estado del montaje, esta función solicita una muestra de la señal en cada canal activo dado en intervalos de tiempo regulares (sin considerarse esto un muestreo por importar solamente la existencia de señal y no la naturaleza de la misma); a cambio, entrega mediante una bandera el estado del montaje (correcto o incorrecto).

A su vez, la función de indicación de canales activos se descompone en dos funciones, como se observa en la figura 3.37.

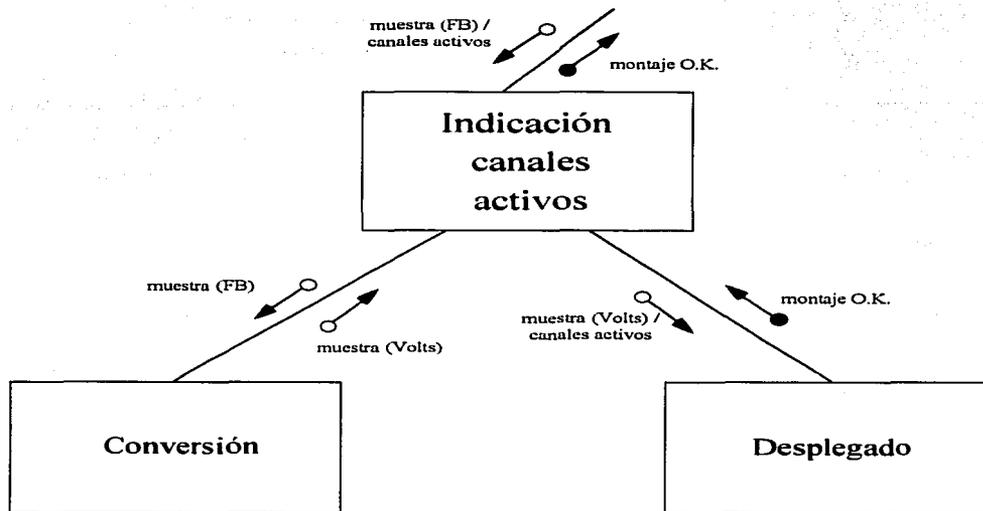


Figura 3.37. Diagrama de estructura de la función de indicación de canales activos.

La primera de estas funciones es la de conversión que se encarga de transformar la muestra de señal obtenida en formato binario a su voltaje equivalente de acuerdo a ciertos parámetros globales como la resolución de conversión utilizada (mínimo valor de voltaje representable en un byte).

La función de desplegado muestra iluminando con color verde la colocación correspondiente en el esquema del cráneo aquellos electrodos que presentan un nivel de señal. Como una de las entradas a esta función es una máscara indicando los canales activos, el sistema es capaz de identificar aquellos canales que están señalados como activos pero que no reciben señal, indicándolo con un "parpadeo" en la posición correspondiente en el esquema del cráneo. Finalmente, indica al usuario sobre aquellos canales inactivos que sí están recibiendo señal, advirtiéndolo sobre una posible equivocación al realizar el montaje.

A continuación se presenta el diseño de las funciones descritas anteriormente. En la figura 3.38 se describe el mecanismo utilizado para graficar el esquema del cráneo y montaje 10-20.

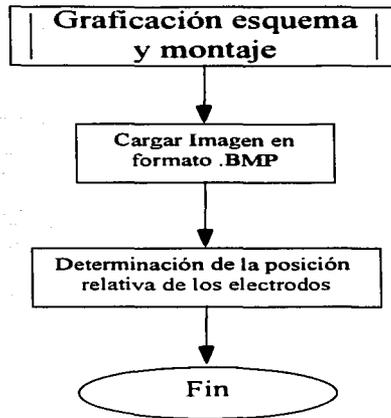


Figura 3.38. Diagrama de flujo de la graficación del esquema craneal.

Cuando la función de indicación de canales activos recibe las muestras de la señal en cada canal, necesita convertirlas a su correspondiente valor de voltaje para determinar si se está recibiendo señal. La función de conversión es bastante simple y se ilustra en la figura 3.39.

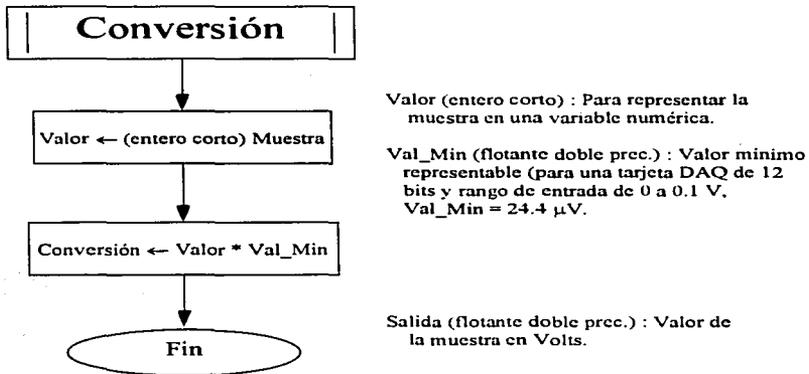


Figura 3.39. Diagrama de flujo para la función de conversión.

Finalmente, se presenta en la figura 3.40 el diseño de la función de desplegado.

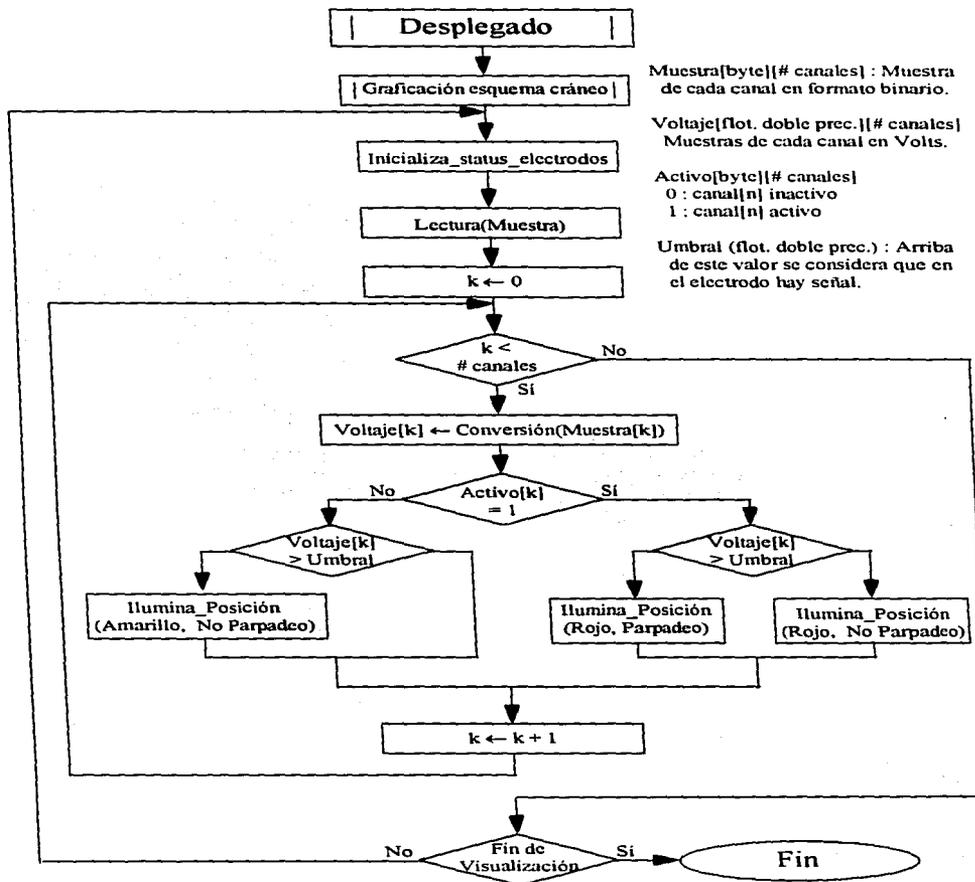


Figura 3.40. Diagrama de flujo de la función de desplegado.

***Función: Graficación de la señal***

A continuación se muestra la serie de instrucciones de programación que se implementará para lograr graficar la señal muestreada, en forma de gráfica de tiempo continuo.

**If** ¿se desea graficar una señal ya almacenada? Entonces

archivo = Nombre archivo

input("Nombre Archivo a graficar")

;preguntar nombre de archivo donde se encuentra la señal

**open**(archivo.eeg) = nombre de archivo de señal

**open**(archivo.hdr) = nombre de archivo de encabezados

**read** Nombre canal 1

;lectura del montaje

Nombre canal 2

Nombre canal 3

:

Nombre canal n

**elseif**

;leer de el arreglo de parámetros de graficación definido por el usuario, el número de canales n que desea visualizar en pantalla según el montaje y asignarles el nombre especificado para cada uno según su colocación.

n = núm. de canales

canal1 = Nombre canal 1

canal2= Nombre canal 2

:

canalN = Nombre canal n

**while** LeeSeñal() = falso

**if** canal 1 LeeSeñal() = falso entonces

mensaje "Señal "<Nombre Canal 1> no conectada según lo especificado"

**elseif** canal 2 LeeSeñal() = falso

mensaje "Señal "<Nombre Canal 2> no conectada según lo especificado"

:

**elseif** canal n LeeSeñal() = falso

mensaje "Señal "<Nombre Canal n> no conectada según lo especificado"

**endif**

**endwhile**

**endif**

t= escala o rango de graficación máximo en segundos

;leer de el arreglo de parámetros definido por el usuario el rango máximo en segundos (t) que se visualizara en una pantalla

mmax=60xt

;Considerando como 60 hz la frecuencia de muestreo

;Inicializar la pantalla en el modo gráfico colocando ejes cartesianos de acuerdo a los parámetros de escala definidos por el usuario en los parámetros de graficación

xmax = rango máximo en 'x'

ymax = rango máximo en 'y'

Inicializar x = xmax

m = mmax - 1

**while** x > 0 ejecutar:

;la coordenada x de la ventana de graficación no llegue al punto 0 (extremo derecho de la pantalla de graficación.

**for** l a n ejecutar:

;para obtener cada uno de los valores de la muestra que da la tarjeta para cada uno de los canales montados de manera que queden almacenados en un arreglo ar

**LeeDato(n,ar[n,m])**

;donde n es el numero de canal a leer, m es el numero total de datos que pueden estar

graficados en pantalla al mismo tiempo y ar es un arreglo de datos en memoria.

**endfor**

;for l a n ejecutar:

**dibujapunto(x,y,ar[n,m]).**

;dibujo en pantalla de cada uno de los n valores obtenidos considerando un origen distinto para las n ordenadas "y" de tal manera de dividir la pantalla en n secciones de graficación, y dejando fija "x".

**endfor**

x = x-1

;Decrementar en una unidad el valor de 'x' y de 'm'

m = m-1

**endwhile**

;fin del ciclo de graficar una pantalla completa, vuelve a iniciar con la lectura de datos para graficar el siguiente punto avanzando la gráfica de esta manera hacia la izquierda.

**while** no se teclee un mandato de finalizar la graficación por parte de usuario ejecutar:

x = xmax

m = mmax

**while** x>0 ejecutar:

;la coordenada x de la ventana de graficación no llegue al punto 0 (extremo derecho de la pantalla de graficación)

**if** localizar Núm. de muestra m en archivo.hdr entonces: ;verifica que existan anotaciones a la señal para esta muestra a graficar

**mensaje read**(archivo.hdr,"anotación de la muestra")

**endif**

**for** l a n ejecutar:

;para obtener cada uno de los valores almacenados en el arreglo ar

**LeeDato(n,ar[n,m])**

**endfor**

**for** l a n ejecutar:

---

```
dibujapunto(x,y,ar[n,m]).  
endfor  
;Decrementar en una unidad el valor de 'x' y de 'm'  
x = x-1  
m = m-1  
  if petición de usuario de anotación a la señal entonces: ;sí el usuario pulso una  
  tecla solicitando la ventana de anotaciones.  
    ejecutar función: anotaciones(m) ;m= núm. de muestra  
  endif  
  endwhile ;Termina ciclo de graficar una pantalla completa.  
for l a n ejecutar:  
  LeeDato(n,ar[n,m]) ;leer el siguiente arreglo de datos de la tarjeta  
  colocándolo en m = 0 y repetir el ciclo principal para graficar de nuevo  
  toda la señal y dar con esto la impresión de que esta camina hacia la derecha.  
endfor  
endwhile ;Termina ciclo principal
```

A continuación en la figura 3.41 se ilustra en forma de diagrama de flujo la función de graficación.

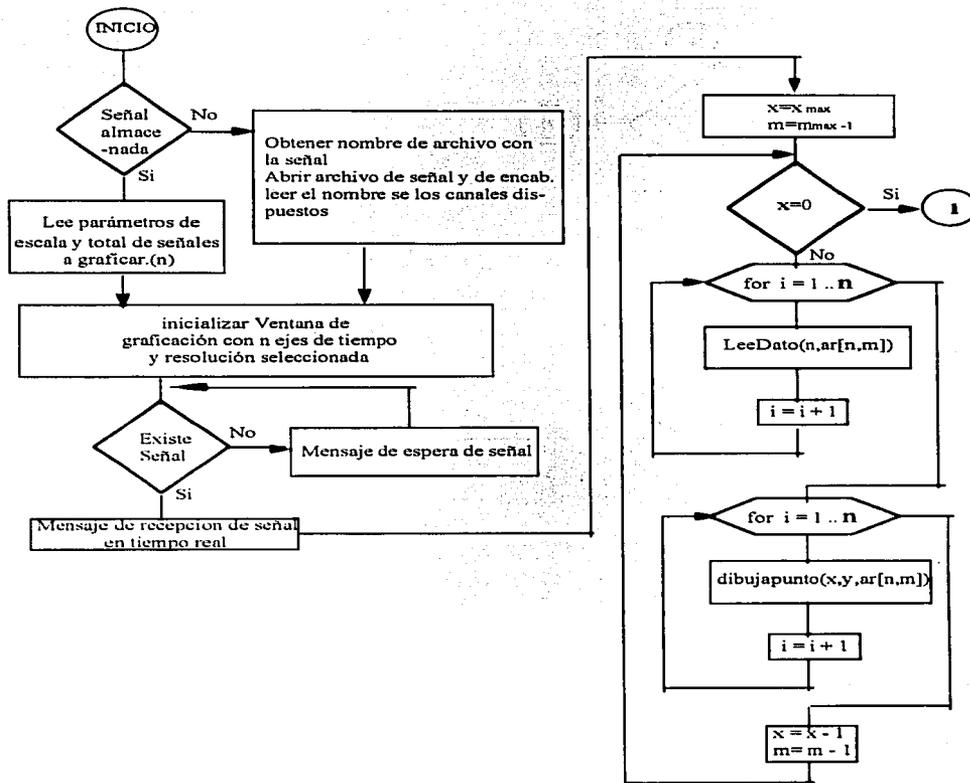


Figura 3.41. Diagrama de flujo para la función de Graficación. (Continúa).

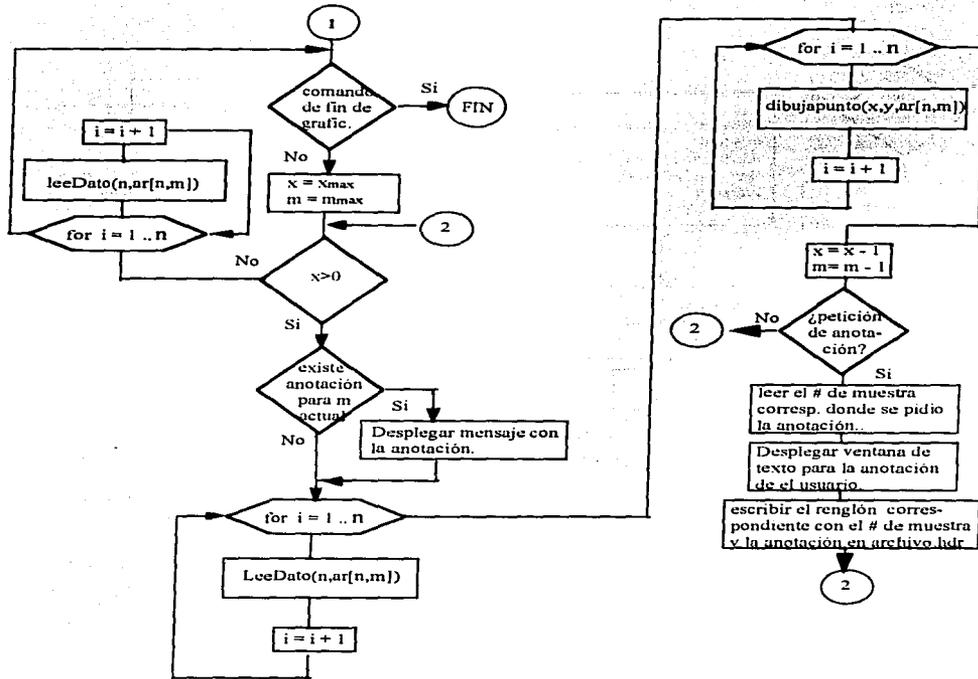


Figura 3.41. Diagrama de flujo para la función de Graficación.

**Función: Almacenamiento**

Para almacenar las señales EEG originales, se van a utilizar dos archivos por cada estudio a realizar; uno de ellos va a contener las señales provenientes de los distintos canales muestreados y otro va a ser un archivo de encabezado, el cual contendrá información general del estudio, además de las anotaciones que se vayan haciendo a las señales muestreadas. Estos archivos tendrán un nombre estándar, que es:

- Nombre\_Archivo.EEG   ⇒  para el archivo con señales EEG.
- Nombre\_Archivo.HDR   ⇒  para el archivo de encabezado.

donde

*Nombre\_Archivo* = # de expediente (6 caracteres) + # de estudio (2 caracteres).

El archivo *Nombre\_Archivo.EEG* es un archivo binario en el que se representa cada lectura con un byte. Las lecturas se almacenan de forma intercalada, es decir, el orden en que se almacenan es:

lectura 1 canal 1  
lectura 1 canal 2  
.  
.  
.  
lectura 1 canal n  
lectura 2 canal 1  
lectura 2 canal 2  
.  
.  
.

Por otra parte, el archivo *Nombre\_Archivo.HDR* es un archivo en formato ASCII el cual guarda datos de una muestra de *n* canales con el formato que se presenta a continuación, considerando que cada dato se almacena en un renglón diferente:

- Nombre del paciente.
- Frecuencia de muestreo.
- Número de canales muestreados (*n*).
- Los siguientes *n* renglones contienen el nombre de los canales muestreados.
- Posteriormente se almacenan las anotaciones a la señal en un sólo renglón por anotación, siguiendo el siguiente formato:

Número de muestra (10 posiciones) + 245 caracteres para el texto de la nota.

Para almacenar las señales que contengan el espectro de potencia producto de haber aplicado el análisis espectral sobre las señales EEG originales, se utilizará un sólo archivo binario de nombre:

*Nombre\_Archivo.PXX*

donde *XX* informa el nombre del canal. Este archivo binario contendrá un byte por renglón.

En la figura 3.42 se presenta el diagrama de flujo correspondiente al diseño del programa que va a hacer posible el almacenamiento descrito en el diseño conceptual, y a continuación se presenta el pseudocódigo correspondiente:

Almacena: **if** (almacenar montaje)

Obtiene no. de expediente y nombre del paciente

---

```

for (x = 1 .. 99)
    Arma nombres de archivos *.EEG y *.HDR como: no. expediente + x
    if (ya existen los archivos)
        x = x + 1
    else
        Crea archivos *.EEG y *.HDR
        goto Sigue
    endif
endifor
    Error: "Ha alcanzado el número máximo de estudios permitidos por no. de
    expediente."
goto Fin
Sigue:
    Escribe el nombre del paciente en *.HDR
    N = No. de canales muestreados
    Escribe N en *.HDR
    for (i = 1 .. N)
        Escribe el nombre del canal i en *.HDR
        i = i + 1
    endifor
goto Fin
Obtener:
    else if (almacenar señal EEG original)
        Obtener datos de una muestra
        N = No. de canales muestreados
        for (k = 1 .. N)
            Escribe dato del canal k en *.EEG
            k = k + 1
        endifor
        if (terminar almacenamiento)
            goto Fin
        else
            goto Obtener
        endif
    else if (almacenar anotación)
        Obtiene el no. de la muestra
        Obtiene la anotación
        Escribe un renglón con el no. de muestra y la anotación
        goto Fin
    else if (almacenar espectro de potencia)
        Obtiene nombre del archivo *.PXX donde se desea almacenar información
        if (existe archivo)
            if (desea reemplazar información)
                Limpiar archivo *.PXX
            else
                goto Nombre1
            endif
        endif

```

```

else
  Crear archivo *.PXX
endif
Obtiene datos del espectro de potencia
Escribe datos en archivo *.PXX
goto Fin
endif
endif
endif
endif
Salir

```

Fin:

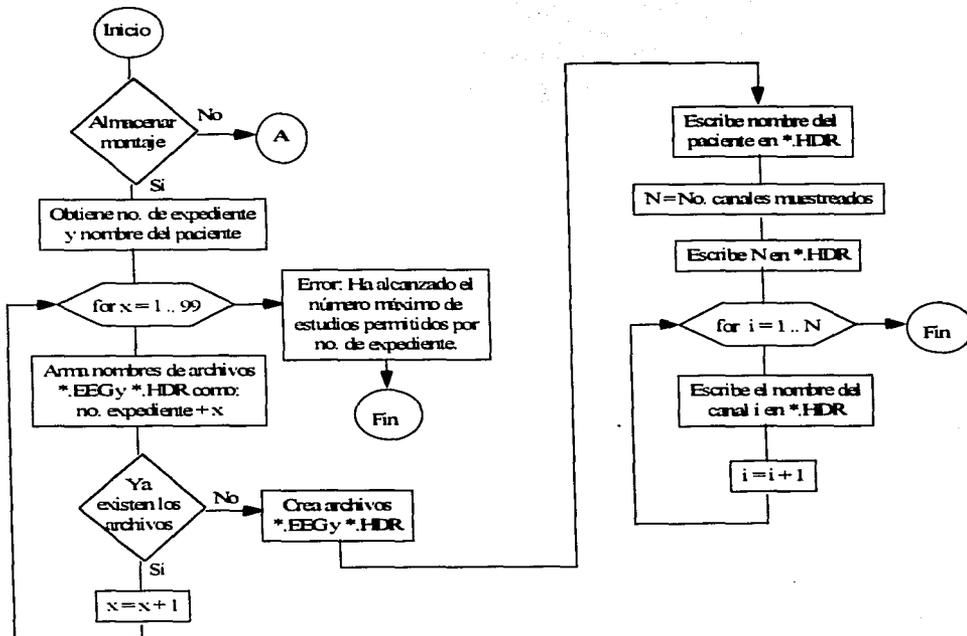


Figura 3.42. Diagrama de flujo para la función de almacenamiento, (Continúa).

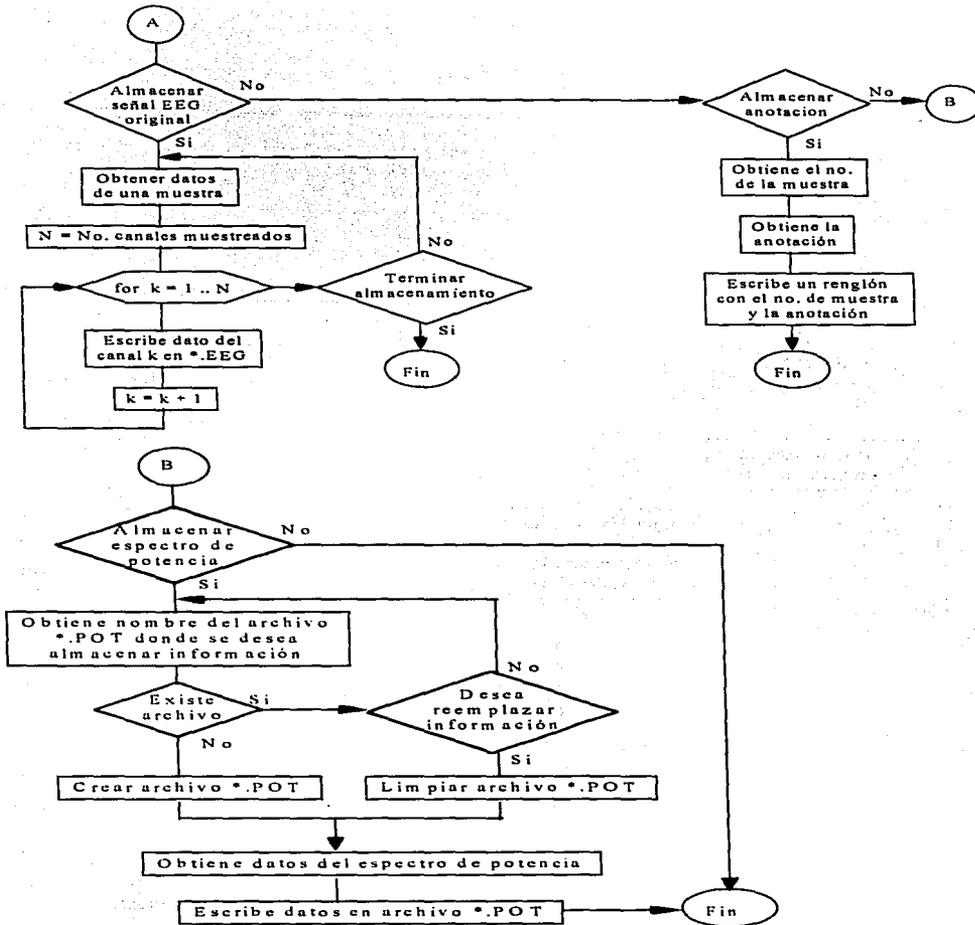
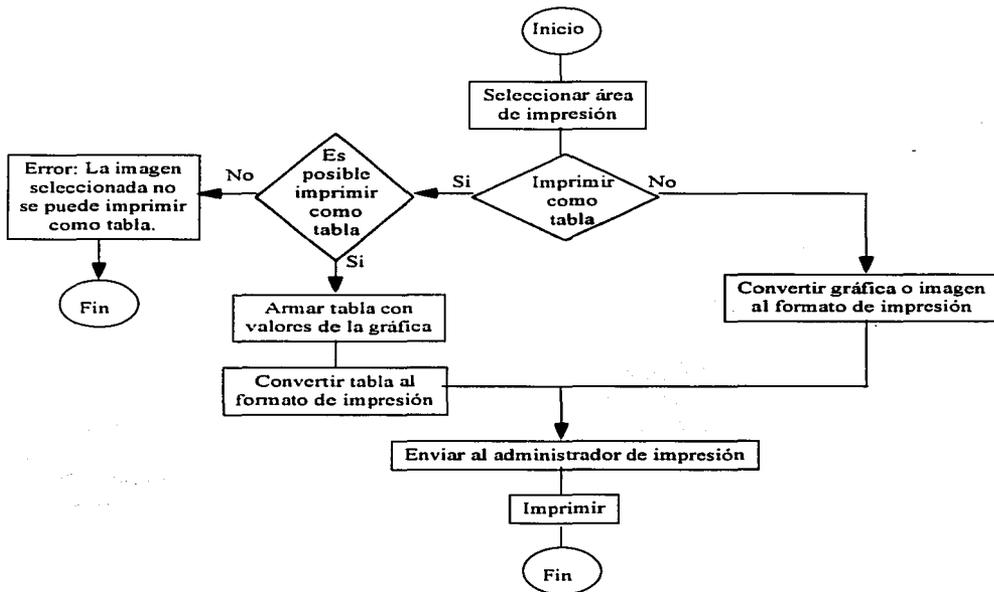


Figura 3.42. Diagrama de flujo para la función de almacenamiento.

**Función: Impresión**

La función de imprimir considera, además de lo descrito en el diseño conceptual, que sólo se pueden imprimir en forma de tabla las señales EEG originales y los espectros de potencia, esto es, las imágenes resultantes del mapeo electroencefalográfico, no se pueden imprimir en forma tabular. La figura 3.43 muestra el diagrama de flujo para esta función y a continuación se escribe el pseudocódigo correspondiente.



**Figura 3.43. Diagrama de flujo para la función impresión.**

```

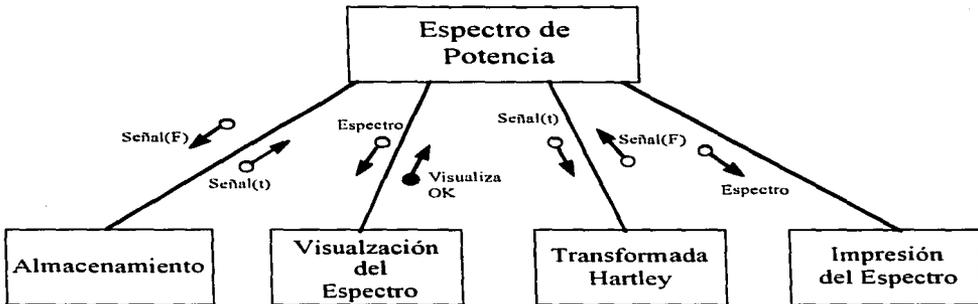
Imprime:  Seleccionar área de impresión
          if (imprimir como tabla)
            if (es posible imprimir como tabla)
              Armar tabla con valores de la gráfica
              Convertir tabla a formato de impresión
            else
              Error: "La imagen seleccionada no se puede imprimir como tabla."
          
```

```

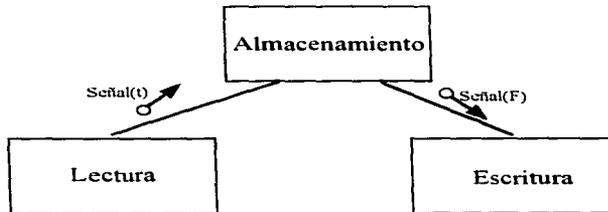
    goto Fin
  endif
else
  Convertir gráfica o imagen a formato de impresión
endif
Enviar al administrador de impresión
Imprimir
Salir
Fin:
  
```

*Función: Espectro de potencia*

Para llegar a un detalle adecuado en el diseño se descompondrá la etapa del cálculo del espectro de potencia en sus funciones, tal como se muestra en las figuras 3.44, 3.45, 3.46, 3.47 y 3.48.



**Figura 3.44** Flujo de la información en la etapa de Cálculo de Espectro de Potencia.



**Figura 3.45.** Funciones de almacenamiento.

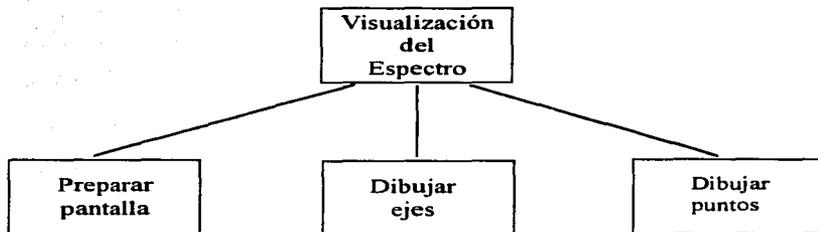


Figura 3.46. Funciones de la Visualización del Espectro.

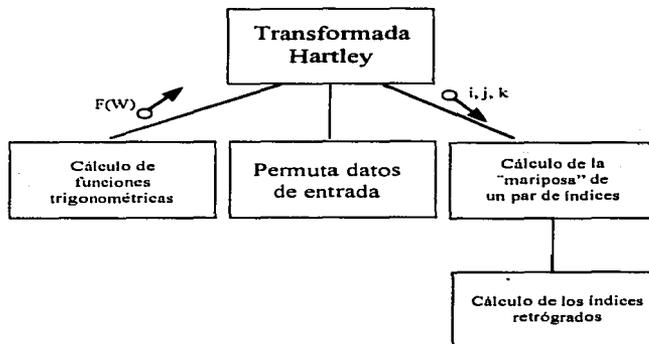


Figura 3.47. Funciones básicas de la transformada de Hartley.

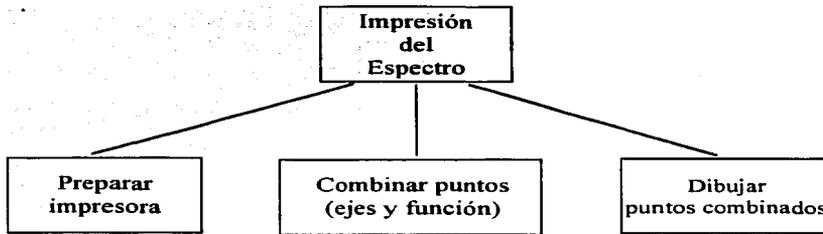


Figura 3-48. Funciones de la Impresión del Espectro.

En base a lo anterior se continuará con el diseño de los programas. Para esto se utilizará la técnica de *español estructurado*. Al mismo tiempo se explicarán algunas estructuras de datos básicas, las cuales se utilizarán a lo largo del desglose de las diferentes funciones.

#### *Almacenamiento del espectro*

##### Lectura

```
Inicio
Pide nombre de archivo
Abre archivo de señal para lectura
For i=1 to N
    Lee de disco V[i]
Cierra archivo
endFor
Fin
```

##### Escritura

```
Inicio
Pide nombre de archivo
Abre archivo de señal para escritura
For i=1 to N
    Escribe a disco V[i]
endFor
Cierra archivo
Fin
```

*Visualización del Espectro*

```
Inicio
Llamada a Lectura
Define área para despliegue
Inicializa área definida para despliegue
Dibuja ejes coordenados
Dibuja títulos de despliegue
For i=1 to N-1
    Calcula punto i
    Calcula punto i+1
    Dibuja línea del punto i al punto i+1
endFor
Fin
```

*Transformada de Hartley*

*Cálculo de funciones trigonométricas*

```
Inicio
PI=3.14159265
Angulo=0
Omega=2*PI/Npts ;Npts es el número de puntos de la señal, previamente obtenido.
For i=1 to Npts
    sen[i]=seno(Angulo)
    cos[i]=coseno(Angulo)
    Angulo=Angulo+Omega
endFor
Fin
```

*Permuta Datos de Entrada*  
(Recibe como dato: Indice)

```
Inicio
j=0
Indice = Indice+1 ;Indice está previamente calculado.
for i=1 to IndicePotencia ;IndicePotencia es el número al que hay que
    s=Indice DIV 2 ;elear 2 para obtener un número >= Npts
    j=2*j+Indice-2*s
    Indice=s
endFor
Regresa j+1 ;El valor a regresar es j+1
Fin
```



```

Repeat
  ind_trig = 1
  s_inicio = Sección*Potencia+1
  s_final = (Sección+1)*Potencia
  For k=1 to Potencia
    Llama a Cálculo de los índices retrógrados con:
      Potencia, s_inicio, s_final, j+Potencia) Regresa IR
    Llama a Cálculo de la Mariposa de un par de Índices con:
      ind_trig, j, j+Potencia, IR
    ind_trig = ind_trig + IncrementoTrig
    j = j+1
  endFor
  j=j+Potencia
  Sección=Sección+2
until j > Npts
  Potencia = Potencia*2
  Intercambia t con f
endFor

For i=1 to Npts
  EntSal[i] = Señal[f,i] / Npts
endFor
Fin

```

Finalmente, el cálculo del Espectro de Potencia queda como sigue:

#### *Espectro de Potencia*

```

Inicio
  Llama a Lectura de la señal
  Llama a Transformada Hartley de la señal, la señal
    transformada queda en Datos[1..Npts]
  i=1
  While i < Npts
    EspectroPotencia[i] = (Datos[i]*Datos[i] + Datos[Npts-i+1]*
      Datos[Npts-i]) / 2
    i = i + 1
  endWhile
  Llama a Escritura de la señal para guardar Datos[1..Npts]
Fin

```

***Función: Mapeo Electroencefalográfico***

Para invocar esta función, el usuario debe haber realizado antes el análisis espectral para todos los electrodos (canales) activos, ya que dicho análisis se toma como base para realizar el mapeo. De no hacerlo, el sistema enviará un mensaje de error indicando la anomalía. A continuación se escribe el pseudocódigo correspondiente a esta función.

```

Mapeo:   Establecer una zona constante asociada a cada uno de los electrodos que conforman
           la técnica 10-20
           Establecer una escala de colores predeterminados para representar distintos rangos
           de amplitud
           N = No. de canales muestreados
           Obtiene cuáles con los canales (electrodos) activos
           if (ya se realizó análisis espectral para todos los canales activos)
               for (i = 1 .. N)
                   Lee archivo con análisis espectral para el canal i
                   Calcula el promedio de amplitud para cada ritmo
                   Asigna un color a cada ritmo dependiendo de la amplitud promedio
                   obtenida
                   Ilumina el área asignada al canal i con el color antes determinado para
                   cada ritmo
                   i = i + 1
               endfor
               Despliega resultados en pantalla
           else
               Error: "Para realizar el mapeo EEG debe haber realizado antes el análisis
               espectral para todos los canales activos."
           endif
Fin:     Salir
  
```

Finalmente, en la figura 3.49 se muestra el diagrama de flujo para la función de mapeo electroencefalográfico..

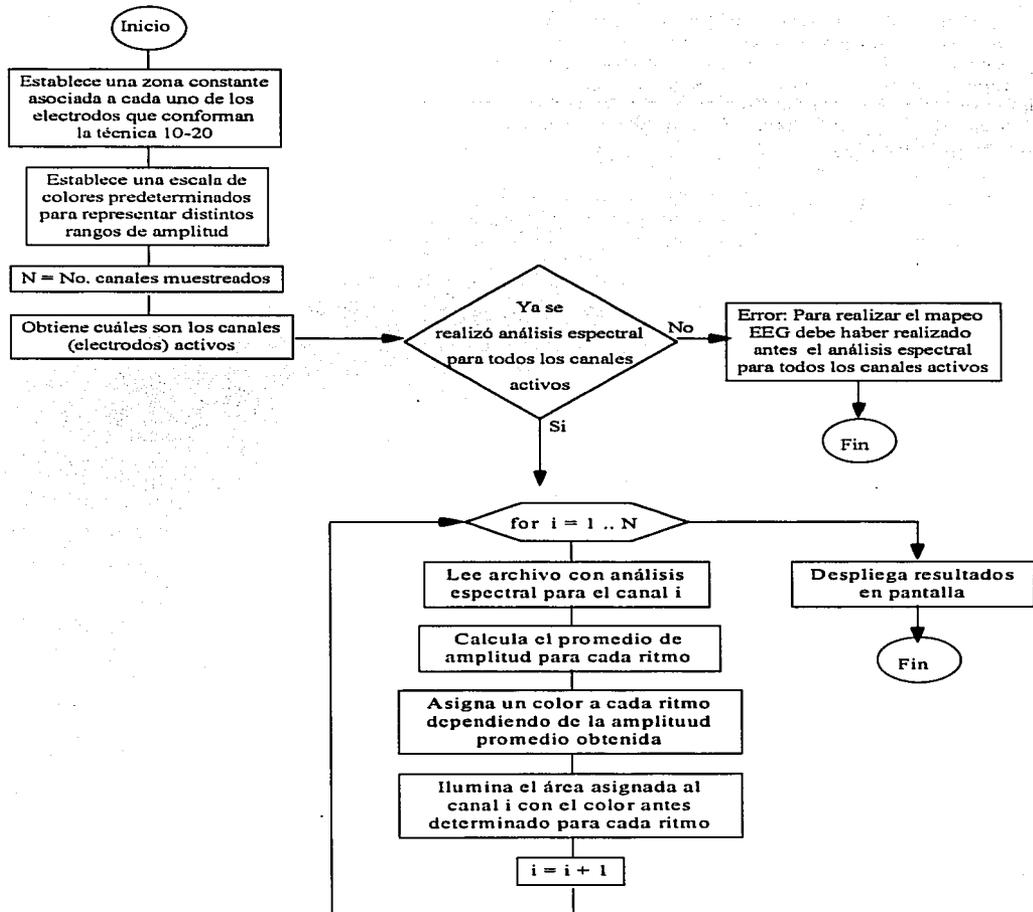


Figura 3.49. Diagrama de flujo para la función de mapeo electroencefalográfico.

Como se pudo observar, en la segunda parte del capítulo, se diseñó de manera integral el software que se desea para el sistema, desde su diseño conceptual, su diseño en forma técnica y finalmente el diseño en forma de diagramas de flujo y pseudocódigo para cada una de las funciones propuestas, en forma de programas. Por tanto en el siguiente capítulo se mostrará la implementación real de las funciones consideradas, formando un sistema de ambiente gráfico que en general interactúe con el usuario y realice el procesamiento digital de las señales electroencefalográficas.

## Capítulo 4

# Desarrollo del Sistema

En este capítulo se presenta el desarrollo de la etapa de software del sistema de procesamiento digital de señales EEG, de acuerdo a las especificaciones establecidas en el capítulo 3.2 (Diseño del Software). El contenido de este capítulo consiste en la explicación de la plataforma de desarrollo (sistema operativo, herramientas, lenguaje de programación, etc.). También se incluye una explicación de los detalles de desarrollo y operación de cada uno de los módulos que componen el sistema. Cabe hacer mención que aun cuando el alcance de este proyecto se limita al diseño del sistema (tanto en su etapa de hardware como de software) la etapa de software si se desarrolla debido a que complementa e ilustra adecuadamente los objetivos planteados en este proyecto.

### 4.1 Plataforma de Desarrollo

Como se ha venido manejando desde el principio de esta tesis, el sistema de procesamiento digital de señales EEG está diseñado para trabajar con computadoras tipo PC, por lo que la plataforma de hardware queda definida así: computadora con procesador *Intel Pentium*, trabajando de manera independiente (no necesita conectarse en red) con al menos un puerto de expansión tipo ISA para conectar la tarjeta de adquisición de datos y monitor a color SVGA con una resolución de 800 x 600 pixeles.

El sistema operativo a utilizar será MS-DOS versión 6.2x o posterior, utilizando la interfaz gráfica de usuario Windows 3.1x de 16 bits o posterior. La razón por la cual se utilizará una plataforma de 16 bits en lugar de 32 a nivel de sistema operativo es debido a que en el caso de este

país, la mayoría de las instalaciones actuales utiliza todavía MS-DOS como sistema operativo y Windows 3.1x como interfaz de usuario. Además, desde la selección de la tarjeta de adquisición de datos se ha contemplado una orientación a los 16 bits, por eso es que la tarjeta finalmente es tipo ISA, ya que para la aplicación considerada, la velocidad que brinda este equipo para colocar los datos en memoria es más que suficiente (recuérdese que se procesarán señales de no más de 30 Hz, lo que lleva a utilizar frecuencias de muestreo por debajo de los 100 KHz, que es la capacidad máxima de la tarjeta utilizada).

La herramienta de desarrollo a utilizar será *Borland Delphi 1.0*. Las razones de esta decisión se exponen a continuación. Primeramente, es una herramienta que permite programar en la interfaz gráfica de usuario elegida. Entre las posibilidades consideradas se tuvieron Borland/Visual C++, Visual BASIC y el mismo Delphi, las cuales son hoy por hoy las herramientas más populares para desarrollos de propósito general bajo el ambiente Windows.

Con el objeto de enfocarse en la programación de las funciones del sistema más que en la apariencia del mismo, se resolvió utilizar una herramienta de tipo RAD<sup>1</sup>, siendo Visual BASIC y Delphi herramientas de este tipo. Por esta razón, se elimina a Borland/Visual C++ aun cuando es el lenguaje de programación que genera aplicaciones más rápidas de entre los tres considerados.

Teniendo entonces sólo dos posibilidades, la inclinación final fué hacia Delphi debido a que presenta las mismas características de facilidad de uso y rapidez de elaboración de pantallas de interfaz con el usuario que Visual BASIC, pero como lenguaje de programación es un producto superior en todos los aspectos. Primeramente, el lenguaje utilizado dentro de Delphi es Object Pascal, y como se sabe, Pascal siempre ha sido un lenguaje más rápido y robusto que BASIC. Por otro lado, Visual BASIC sigue siendo un intérprete aun cuando genera aplicaciones con extensión .EXE, no así Delphi, que es un compilador real.

Además, tradicionalmente los turbo compiladores de Borland siempre se han caracterizado por la generación de código ejecutable rápido, robusto y compacto, y Delphi no es la excepción. De hecho, esta es la razón por la que no hay mayor problema de eliminar a Borland/Visual C++, sabiendo que el código resultante de Delphi será casi tan rápido como el generado en aquel pero sin la complejidad inherente dentro de la programación en C++ para Windows.

Finalmente Pascal siempre ha favorecido las técnicas de programación ordenadas, no así BASIC, por lo que resultaría más sencillo trasladar el diseño del sistema a una implementación en Pascal que una en BASIC conservando así los principios de modularidad establecidos en dicho diseño.

Cabe mencionar que debido a que el diseño está orientado hacia los procesos y no hacia los objetos, la implementación conservará estas características. Sin embargo Delphi es una herramienta que se basa consistentemente en la teoría de objetos, por lo que al crear un ambiente para un programa en particular, se están aprovechando las ventajas de la programación orientada a objetos para manipular componentes visuales, así como sus métodos y sus propiedades.

---

<sup>1</sup> RAD son las iniciales de Rapid Application Development o Desarrollo Rápido de Aplicaciones

## 4.2 Interfaz Gráfica de Usuario

Una interfaz gráfica de usuario (GUI<sup>2</sup> por sus siglas en inglés) es un formato de exhibición que permite al usuario escoger comandos, iniciar programas y observar listas de archivos y otras opciones mediante el señalamiento de representaciones gráficas (iconos) y listas de menús sobre la pantalla de la computadora. Por lo general dichas opciones son accesibles vía teclado y ratón.

El principal objetivo de utilizar una interfaz gráfica es que actualmente el mercado de la computación personal ha desechado prácticamente las aplicaciones basadas en DOS, por lo que para seguir esta tendencia, se plantea el desarrollo del sistema bajo este tipo de interfaz. Sin embargo, no es la única razón, a continuación se muestran varias de las características principales de una interfaz gráfica, lo que las ha llevado a dominar en el campo de las aplicaciones de software. En particular, refiriéndose a Microsoft Windows 3.1X.

Por principio de cuentas, todas las aplicaciones que corren bajo Windows presentan un aspecto y forma de utilización similar, por lo que al usuario le resulta fácil e intuitivo aprender a utilizar una nueva aplicación que se parezca a aquellas que ha venido utilizando.

Al diseñar una aplicación bajo una interfaz gráfica de usuario, el programador se beneficia de ciertas características que debe cumplir dicha aplicación proporcionando así al usuario una herramienta que le va a funcionar de verdad. Dichas características se describen a continuación [OsGB96].

### Control

El usuario debe siempre sentir que tiene el control de todo lo que sucede en la pantalla. Debe tener la sensación de que él es quien inicia la acción en lugar de reaccionar a las acciones de la computadora. Además, la aplicación debe informar siempre al usuario la actividad que está realizando en un momento dado para darle una guía del avance de un cierto proceso.

### Direccionalidad

Significa permitir que los usuarios manipulen directamente los objetos de su medio, es decir, representar objetos propios de la computadora como objetos del mundo en donde se desenvuelve el usuario, ejemplo de ello son las carpetas de trabajo, que se sabe son en realidad subdirectorios; otro ejemplo es el "bote de basura" en Macintosh, que significa eliminar documentos.

---

<sup>2</sup> GUI son las siglas de Graphical User Interface

## **Consistencia**

Este es uno de los aspectos más importantes en el desarrollo de una aplicación bajo una interfaz gráfica de usuario como Windows. La consistencia, como se mencionó anteriormente, permite a los usuarios familiarizarse más rápido con las nuevas aplicaciones debido a que se parecen a otras que ya dominan. De hecho no solamente hay que ser consistentes con otras aplicaciones y el mismo sistema operativo, sino que hay que ser consistente dentro de los módulos de una misma aplicación, de modo que el usuario no encuentre confusión al utilizar ciertas pantallas de manera distinta que otras.

## **Rectificabilidad**

Una aplicación siempre debe de proporcionar una salida inmediata a un usuario para cualquier acción que éste emprenda, es decir, si el usuario se pone a explorar su aplicación, lo hará con toda la confianza puesta en que no se presentarán resultados desastrosos. Por otro lado, si comete alguna acción equivocada debe siempre tener un medio de anularla o regresar.

## **Retroalimentación**

Una aplicación debe permitir siempre al usuario visualizar lo que está sucediendo proporcionándole una retroalimentación periódica. Esta retroalimentación puede ser mediante indicadores visuales o sonoros los cuales dan la impresión al usuario que la computadora está siempre "al pendiente" de ellos.

## **Estética**

Visualmente, una aplicación debe resultar atractiva al usuario, después de todo, se supone que el usuario pasará largo tiempo frente a frente con ella. Además de utilizar los colores adecuados (normalmente los mismos del sistema operativo para que la aplicación se integre adecuadamente), la colocación y número de objetos en la pantalla son aspectos muy importantes. Existe una regla denominada "regla del siete" que dice que se le debe dar al usuario en todo momento sólo siete opciones (más o menos dos) de tal manera que no se confunda con más de este número y no se sobrecargue cerebralmente.

## **Sencillez**

Una aplicación debe ser fácil de aprender, pero sobre todo, debe ser fácil de utilizar, cuestión muy relacionada con el cumplimiento de los objetivos de diseño ya que en la mayoría de las aplicaciones de software que se construyen, va implícita una ventaja sustancial en la forma de operar que tendrá el usuario.

### 4.3 La Aplicación

Dentro de este apartado se presentan los resultados del desarrollo del sistema bajo el ambiente y herramientas previamente comentadas. Se trata de una aplicación de software que se ejecuta bajo Windows y que presenta las características comunes a las aplicaciones de este tipo. Este programa le proporciona al usuario los medios necesarios para que mediante una secuencia lógica de pasos, lleve a cabo un estudio EEG obteniendo información para analizar con fines específicos. Dichos pasos consisten de la configuración inicial, la preparación para el muestreo, el muestreo en sí junto con la visualización de la señal, el almacenamiento y finalmente el análisis de dicha señal. Cada uno de estos pasos se relaciona con uno o varios módulos funcionales, relaciones que se explican a lo largo de este apartado.

Primeramente, se observa en la figura 4.1 el aspecto que presenta el sistema durante su ejecución. Como la mayoría de las aplicaciones Windows, consta de una barra de título en la parte superior, icono para el menú del sistema en la esquina superior izquierda, iconos para maximizar y minimizar en la esquina superior derecha, menú principal en la parte superior del área del cliente, barra de estado en la parte inferior del área de cliente y botones de acceso rápido justo por debajo del menú principal.

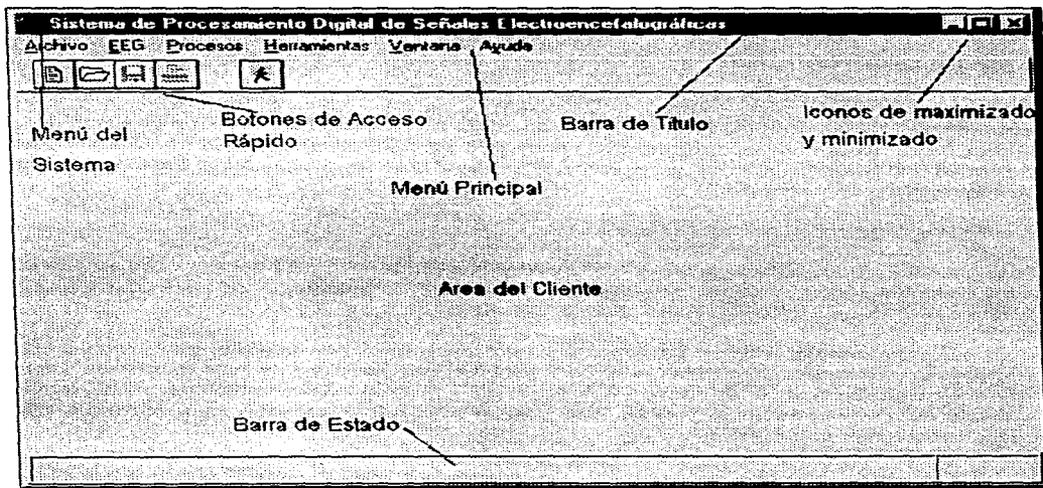


Figura 4.1. Pantalla principal del sistema de procesamiento digital de señales EEG.

Como se observa en la barra de menú, existen cuatro grupos principales de funciones, las relacionadas con el manejo de archivos: Nuevo, Abrir, Cerrar, Guardar, Imprimir, (opción Archivo en el menú); las relacionadas con la adquisición y visualización de los datos (opción EEG); el tratamiento numérico de la señal (opción de Procesos); y las herramientas del sistema (opción herramientas). Las otras dos opciones, (Ventana y Ayuda) corresponden a parte de lo que se llama el "overhead"<sup>1</sup> del sistema.

Debajo de la barra de menú se observan unos botones de acceso rápido a las funciones más comunes relacionadas con los archivos: Nuevo, Abrir, Guardar e Imprimir, los cuales tienen el mismo efecto que las opciones del menú. Adicionalmente, se cuenta con un botón para salir de manera rápida.

Para explicar las funciones principales del sistema, éstas se dividirán en dos grupos principales: las funciones relacionadas con el registro de la señal y las relacionadas con el tratamiento de la señal:



#### 4.3.1 Registro de la Señal

Las funciones relacionadas con el registro de la señal comprenden todo lo que es la adquisición y visualización de un EEG en pantalla, ya sea en tiempo real, leído directamente de la tarjeta de adquisición, o uno previamente almacenado en disco. A continuación se describen las funciones que proporciona el sistema para llevar a cabo dicha visualización en el orden lógico que se debe seguir para tener éxito.

##### *Configuración de Electroodos Disponibles*

Como se recordará de la teoría explicada en el capítulo uno, para llevar a cabo un estudio electroencefalográfico se utiliza un estándar internacional de colocación de electrodos llamado montaje 10-20 estándar, bajo el cual se basa el sistema.

Obsérvese que dicho estándar define veintiún posiciones posibles para colocar los electrodos, diecinueve se conectan al cráneo para obtener señal y dos más se utilizan como referencia conectándolos a las orejas o a la nariz. Sin embargo, cabría preguntarse cómo es que teniendo diecinueve posiciones posibles para colocar electrodos, se diseñó un sistema que sólo puede analizar dieciséis de ellos.

---

<sup>1</sup> En términos computacionales, se conoce como "overhead" a todo lo relacionado con la autoadministración de un programa o un sistema operativo.

La razón es muy sencilla, el estándar 10-20 sólo define las diecinueve posibles posiciones, pero en la práctica nunca se llegan a utilizar todas, de hecho, muchos estudios particulares utilizan tan solo ocho de ellas. De la investigación se concluyó que dieciséis electrodos es un número más que suficiente para prácticamente la totalidad de los estudios posibles que se realizan actualmente.

Por tanto, debe haber un mecanismo en el cual se discriminen tres de las diecinueve posiciones posibles para conectar los electrodos. Debido a la naturaleza de los estudios, debe ser posible para el usuario cambiar en un momento dado las posiciones que se discriminan. Esta es precisamente la función de éste módulo, el cual se muestra en la figura 4.2 y se accede mediante las opciones Herramientas/Configuración del menú.

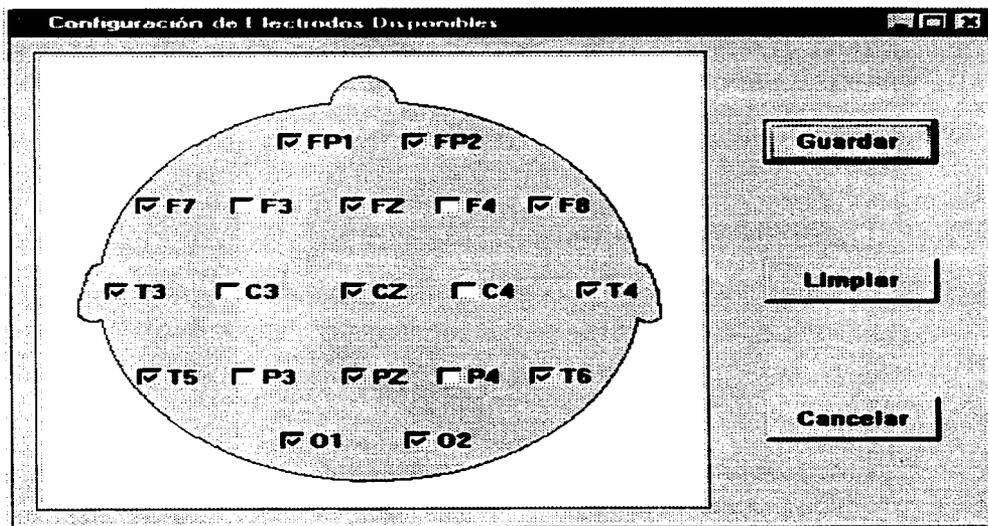


Figura 4.2. Pantalla para la selección de los electrodos disponibles.

En esta pantalla, el usuario debe escoger las dieciséis posiciones que se encontrarán disponibles en todo momento para conectar electrodos hasta que la configuración sea cambiada en esta misma pantalla. Una vez seleccionadas las posiciones, el usuario deberá guardar la configuración con el botón correspondiente.

Cabe hacer mención que cambiar los electrodos disponibles es una actividad esporádica, de hecho, en la mayoría de los casos, es una acción que se debe llevar a cabo cuando se instala el sistema por primera vez y normalmente no se deberá utilizar esta pantalla nuevamente.

### Canales Activos

Es común que para cada estudio de electroencefalografía se utilicen diferentes electrodos y en diferente número. Para visualizar una señal EEG, el sistema debe saber cuáles son los canales a los que se les conectarán electrodos y cuáles no. A los primeros se les denominará como canales activos y la manera en que el usuario le indica al sistema cuáles son se muestra en la figura 4.3, pantalla que se accede mediante las opciones EEG/Canales Activos del menú principal.

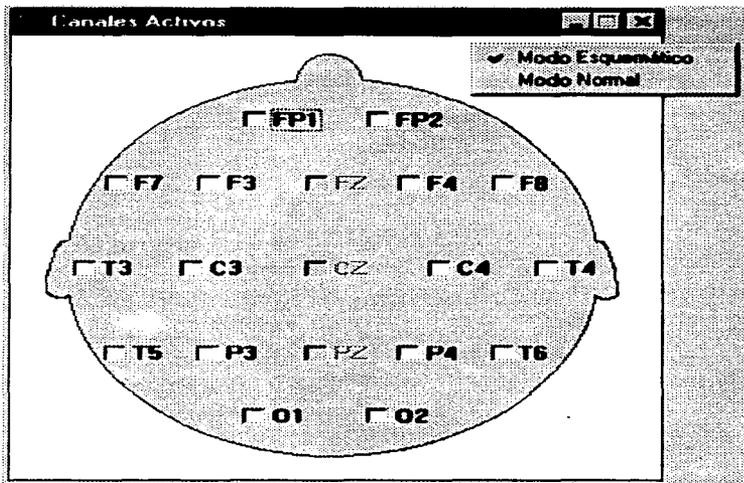


Figura 4.3. Pantalla de selección de canales activos (modo esquemático).

Obsérvese que algunos canales están deshabilitados, éstos corresponden a los electrodos discriminados en la pantalla de selección de electrodos disponibles. Para los dieciséis canales que sí están habilitados, el usuario debe marcar en esta pantalla aquellos que estarán activos, así el sistema sabrá qué canales deberán ser muestreados por la tarjeta y desplegados por la pantalla de visualización en tiempo real.

No confundir la pantalla de electrodos disponibles con la de canales activos, la primera indica las dieciséis posibles posiciones que se podrán utilizar para conectar electrodos y sólo se

realizará esta configuración esporádicamente, mientras que la segunda sirve para indicar aquellos canales que se van a utilizar para un estudio en especial, por lo que la entrada a esta pantalla debe ser siempre un paso previo a la de visualización en tiempo real de la señal EEG.

En esta pantalla se muestran los canales en la posición en la que se conectarían los electrodos en la cabeza del paciente, llamando a esta forma de visualizar los canales activos *modo esquemático*. Sin embargo, mediante el menú flotante (popup) que se activa con el botón derecho del ratón, el usuario puede elegir otra forma de visualización de los canales activos que se denominó *modo normal*, esta forma se visualiza en la figura 4.4. La característica de esta ventana es que puede haber otras ventanas abiertas y ésta permanecerá por encima de todas siempre visible. Al igual que en el caso anterior, se puede regresar al otro modo de visualización mediante el botón derecho del ratón.

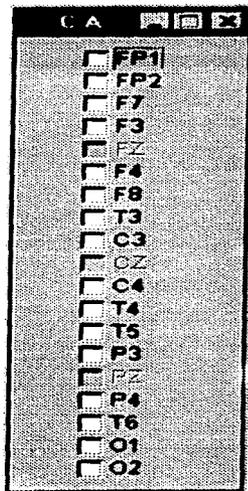


Figura 4.4. Canales Activos (modo normal).

### *Visualización del Montaje*

Una vez que el usuario eligió un juego de canales como activos, éstos forman entonces una configuración que se conoce como *montaje de los electrodos*, el cual debe verificarse antes de proceder a realizar la adquisición de la señal. Para visualizar el montaje utilizar la opción EEG/Visualización del Montaje dentro del menú principal.

La visualización del montaje es una representación esquemática en la pantalla de la computadora, de las conexiones realizadas al individuo de acuerdo a la configuración de canales activos determinada por el usuario. Todos aquellos canales que el usuario marcó como activos y en los cuales se está recibiendo una señal se considera que están correctamente montados y se visualizan en la pantalla como leds encendidos de color verde (ver figura 4.5).

Sin embargo, existe la posibilidad de que el usuario haya marcado algún o algunos canales como activos y que no están montados correctamente en el cráneo del paciente o bien, se olvidó conectarlos, en este caso al no detectar señal el sistema, se visualizará dicho canal como un led amarillo encendido. Por otro lado, el usuario puede conectar incorrectamente un electrodo sin haber marcado el canal como activo, en tal caso, el sistema lo indicará mostrando un led amarillo que parpadea. En la figura 4.5 se observa que los canales CZ y O2 no están recibiendo señal adecuadamente.

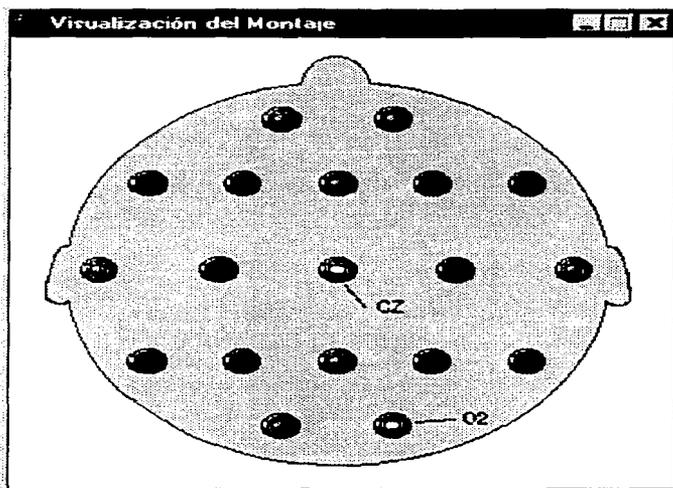
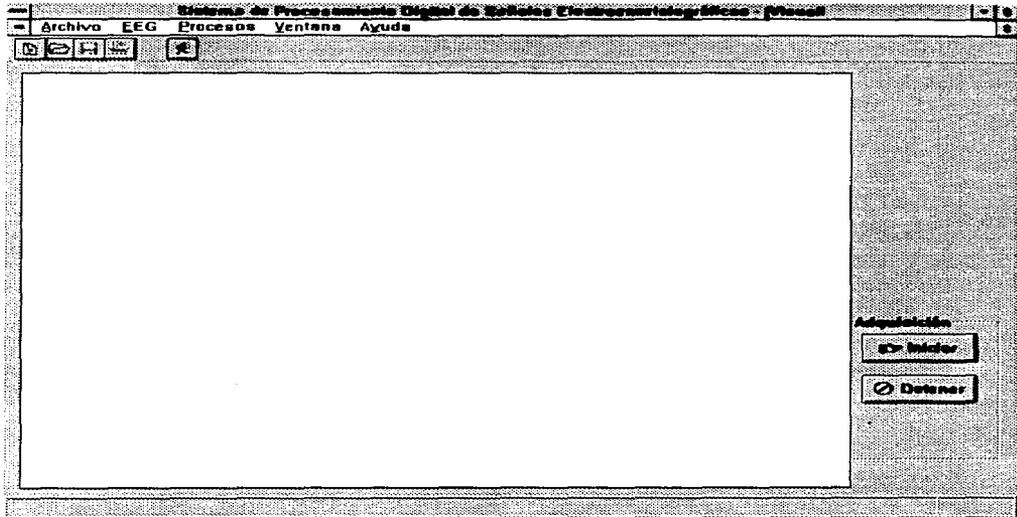


Figura 4.5. Pantalla para la visualización del montaje.

### *Visualización EEG en tiempo real*

Al seleccionar la opción *visualización en tiempo real* del menú *EEG*, se da inicio al proceso de adquisición de señales electroencefalográficas. Primero que nada se deben de haber seleccionado los canales que se desean utilizar como activos en el panel de selección previamente visto de canales activos. A continuación, teniendo la pantalla como muestra la figura 4.6, se podrá

presionar el botón de iniciar, arrancando así el proceso para visualizar la representación gráfica de las señales electroencefalográficas que se estén registrando en los canales activos.



**Figura 4.6. Pantalla de visualización del EEG en tiempo real.**

En la figura 4.7 se observa la misma pantalla de la figura 4.6 pero con el proceso ya en marcha de visualizar la gráfica de las señales electroencefalográficas de los canales que se estén registrando.

Aquí se mostrara siempre una gráfica continua de distinto color para cada uno de los canales activos, ajustándose automáticamente el tamaño de su escala de graficación en el espacio de graficación disponible, consiguiendo con esto que si se desea ver uno o dos canales con mayor resolución, se podrán reseleccionar estos únicamente en el panel de selección de canales activos y con esto verlos a escala completa de la pantalla de graficación. Lo anterior lo ilustra la figura 4.8.

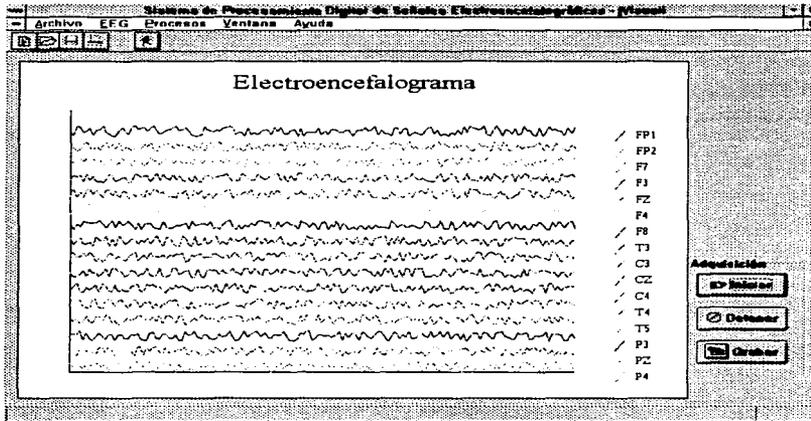


Figura 4.7. Arranque del proceso de Visualización del EEG en tiempo real con los 16 canales activos.

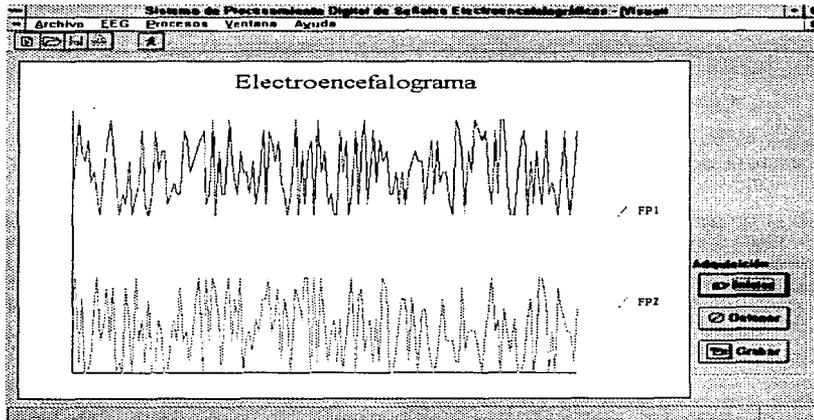


Figura 4.8. Proceso de visualización del EEG en tiempo real con 2 canales activados.

Como se observa en las figuras 4.7 y 4.8. Al dar inicio al proceso se habilita el botón "Grabar" en la parte inferior derecha de la pantalla, con el cual se puede dar inicio al proceso de almacenar la señal en disco.

Al arrancar el proceso de grabar se crea el archivo en disco donde se almacenarán las señales. Este archivo tendrá un nombre construido a partir de los datos dados por el usuario en el cuadro de diálogo que se muestra en la figura 4.9, mismo que aparece cuando se le indica al Sistema iniciar un nuevo estudio, es decir, con la secuencia Archivo/Nuevo, o con el botón de acceso rápido asociado.

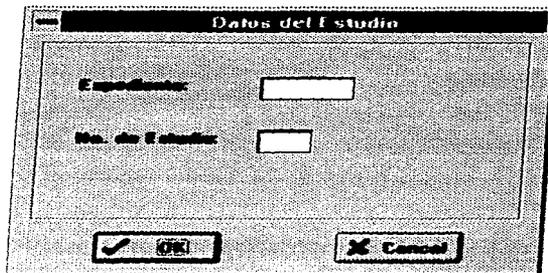


Figura 4.9. Cuadro de diálogo donde se introducen los datos del estudio.

Cuando se presiona el botón "Grabar" y se inicia el almacenamiento, este botón "Grabar" cambia su etiqueta a la de "Fin Grabar", lo cual indica que presionando este mismo botón se detendrá el proceso de grabado. Cuando se está en el proceso de almacenamiento, en la parte superior derecha de la pantalla se exhibe el número de muestra que se está visualizando y almacenando, esto con el fin de ubicar la atención en partes que sean más interesantes de analizar cuando la señal almacenada se reproduzca o se imprima.

#### *Visualización de señales EEG Almacenadas*

La pantalla que va a mostrar las señales EEG almacenadas en disco es similar a la de visualización en tiempo real, pero con diferencias en su concepto, dado que las ventajas de observar una señal almacenada son mayores principalmente porque se pueden reproducir analizar e imprimir las muestras de la señal que interesen, tantas veces como se desee.

Lo primero que se tiene que hacer es llamar a la opción "Señal Almacenada" del menú "EEG", del menú principal del sistema, y aparecerá el cuadro de diálogo de selección de archivo para apertura, que se muestra en la figura 4.10.

La figura 4.11 muestra como es la pantalla de visualización de señales almacenadas. En ésta se observa que existen también botones para iniciar o para detener el proceso de visualización, pero en el caso que el sistema termine de graficar en pantalla todas las muestras de

señal contenidas en un archivo, el proceso se detendrá automáticamente, y si se decide presionar el botón de iniciar, el archivo se leerá de nuevo desde el principio.

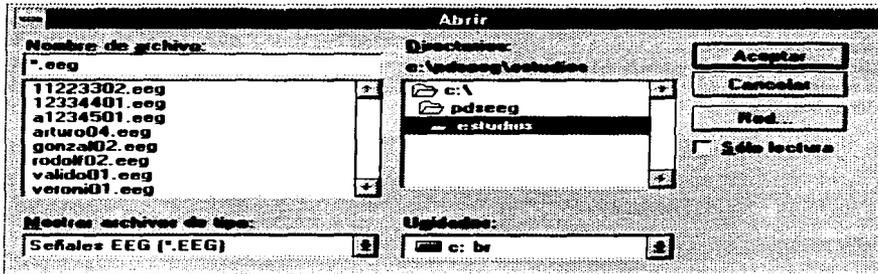


Figura 4.10. Cuadro de diálogo de apertura de archivo de EEG.

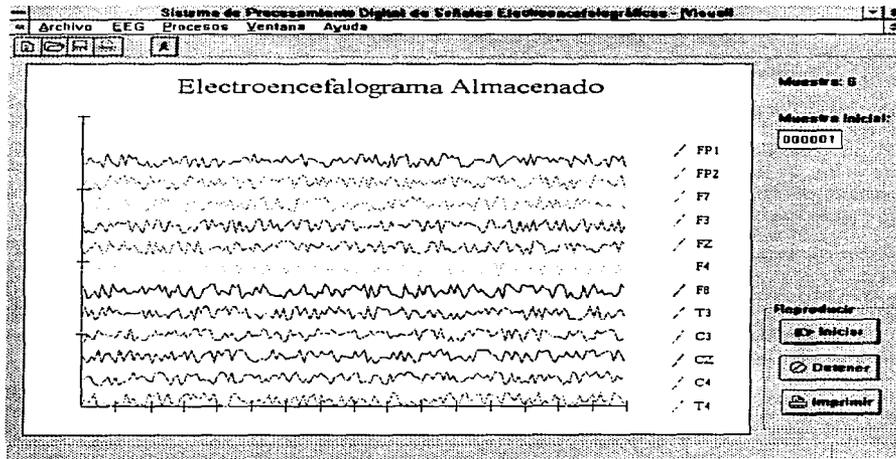


Figura 4.11. Visualización de la señal EEG almacenada.

En la pantalla de visualización de señal almacenada se cuenta con un cuadro de edición de texto en la parte superior derecha, en el que se podrá teclear la muestra en particular que se desee ver y a partir de la cual arrancar la visualización (figura 4.11).

En esta pantalla se cuenta también con un botón de “Imprimir” que al presionarlo con la visualización detenida se activará un cuadro de diálogo de impresión en Windows en donde se podrá mandar a una impresora la gráfica de la muestra que se visualice en ese momento.

### **4.3.2 Tratamiento de la señal**

Una vez que una señal EEG ha sido adquirida y almacenada, es objeto entonces de un análisis posterior más profundo objeto del cual es este apartado. El tratamiento de una señal EEG comprende su análisis en el dominio de la frecuencia (espectro de potencia), recordando del capítulo uno que es precisamente la frecuencia la característica más importante de las ondas cerebrales y criterio bajo el cual se clasifican éstas mismas. Adicionalmente, se muestra una representación cromática de la presencia de distintos ritmos cerebrales en un individuo a partir de las señales EEG adquiridas; a este proceso se le denominó mapeo electroencefalográfico.

#### *Espectro de Potencia*

El proceso denominado “Espectro de Potencia” permitirá obtener información relevante de la señal EEG, la cual no se presenta más que visualmente en su representación gráfica en el tiempo, y esto sólo en forma genérica. Es decir, para un experto puede resultar fácil identificar en una gráfica de tiempo la presencia de ciertos ritmos EEG, a simple vista, pero sin poder dar una descripción precisa de las características de ese pedazo de señal en frecuencia, ya que como se explico en el capítulo uno, los ritmos están caracterizados por un rango de frecuencias, y es imposible que visualmente un usuario, aun siendo especialista, pueda decir a qué frecuencia está correspondiendo dicha señal. Quizá diga, por ejemplo: “aquí vemos claramente una señal delta”; pero será incapaz de dar información precisa sobre la frecuencia dentro del rango de las señales delta.

Por otro lado, se debe recordar también que en un espacio de tiempo dado una señal EEG puede tener más de un ritmo presente, y por ende más de una frecuencia y más de una amplitud que interese rastrear. Toda esta información que queda en cierto modo oculta en el dominio del tiempo se vuelve claramente visible en el dominio de la frecuencia, y por lo tanto, lo que aquí se requiere, es una manera ágil de cambiar la señal discreta de espacio de representación.

En esta aplicación se ha implementado este cambio de espacio a partir de la transformada rápida de Hartley. Como se vio en el capítulo dos, la FHT es análoga a la transformada rápida de Fourier (FFT) sólo que la primera hace un mapeo de una función real en el tiempo a una función real en la frecuencia, mientras que la segunda hace este mismo mapeo, pero de una señal real en el tiempo a una señal imaginaria en la frecuencia. Es intuitivo y demostrable que el obtener la FHT requiere de menos espacio de almacenamiento y de menos recursos de memoria durante su procesamiento, simplemente por no tener que lidiar con la parte imaginaria.

En resumen, tanto la FFT como la FHT transforman la señal discreta del dominio del tiempo al dominio de la frecuencia, con lo que si se tienen 256 muestras de la señal en el tiempo discreto, se obtendrán 256 números en el dominio de la frecuencia discreta. De cualquiera de estas transformaciones se puede obtener la potencia de las frecuencias en la señal original. Para calcular los valores de la señal en la frecuencia sólo se necesitan los valores que representan a la señal en el tiempo, y el número de puntos (N).

Ahora bien, para interpretarlos se necesita algo más, es necesario saber cuánto tiempo representan los puntos en el tiempo para encontrar la cuantificación que se obtiene en el dominio de la frecuencia. Es decir, para aplicar el algoritmo de la transformación sólo se necesita el número de muestras (N) y su valor desde 0 hasta N-1. Y sin más se obtienen N puntos en el dominio de la frecuencia, pero para saber a qué frecuencias corresponden tales puntos, si se requiere saber cuánto tiempo representó la señal en el tiempo, información que se conocerá si se conoce cuál fue la frecuencia de muestreo, esto se ejemplificará más adelante.

En la aplicación aquí presentada se pueden obtener tanto la FHT como la FFT, si bien el algoritmo con que se obtiene la FFT parte de haber previamente obtenido la FHT.

Suponiendo que al adquirir una señal EEG, ésta se muestrea a una frecuencia de 100 Hz, lo cual satisface de entrada el requerimiento de tener una frecuencia de muestreo mayor al doble de la frecuencia máxima que se encuentra presente en la señal origen. Si a esta señal se le aplica el proceso de Espectro de Potencia, se tiene lo siguiente:

Si el tamaño de la muestra es de 256, entonces la señal transformada tendrá 256 números que representarán el contenido frecuencial en rangos de  $1/t$ , donde  $t$  es el tiempo que duró la señal en el tiempo representada por las N muestras. Si  $N=256$  y la frecuencia de muestreo es 100 Hz, entonces la duración de la señal es de 2.56 segundos, y entonces la razón  $1/t = 0.3906$ , da la división en el eje de las abscisas en el dominio de la frecuencia, tal que, el primer valor en la serie de 256 números en la frecuencia representa la potencia de la frecuencia de 0 a 0.3906 Hz (también representa el componente de DC de la señal), el valor siguiente representa la potencia de la frecuencia en la señal origen de 0.3906 Hz a  $2(0.3906)$  Hz, y así sucesivamente.

Si se observan los valores del espectro de potencia se ve que los números  $v$  y  $256-v$  son idénticos, es decir, la serie de números presenta una simetría par. Esto también implica de nuevo que la frecuencia máxima de la cual se pueda obtener información es menor a la mitad de la frecuencia de muestreo, que en este caso sería menor a 50 Hz, ya que  $128(0.3906 \text{ Hz}) = 50 \text{ Hz}$ . Por lo que recordando que una señal de N puntos arroja un espectro de potencia de  $(N/2)+1$  puntos, como se observa, los siguientes puntos de la serie son simétricos a los anteriores.

Para ilustrar esto se presentan un par de gráficas a continuación; en la primera (figura 4.12), se observará una señal creada a partir de la suma de dos senoidales de amplitud unitaria, la primera senoidal tiene una frecuencia de 1 Hz y la segunda de 10 Hz, se obtuvieron un total de 512 puntos ( $N=512$ ), y ya que fueron muestreadas a 100 Hz, entonces quiere decir que se tiene una señal de duración en el tiempo de 5.12 segundos.

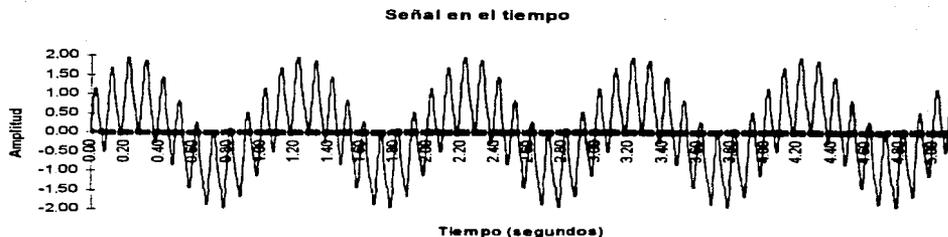


Figura 4.12.  $f(t)=\text{sen}(2\pi t)+\text{sen}(20\pi t)$ , muestreada durante 5.12 segundos a 100 Hz.

Como se esperaba, al obtener el espectro de frecuencia de esta señal en el tiempo, mostrado en la figura 4.13, este es un par de espigas que están localizadas alrededor de 1 y 10 Hz, que son las frecuencias de las senoidales presentes en la señal en el tiempo.

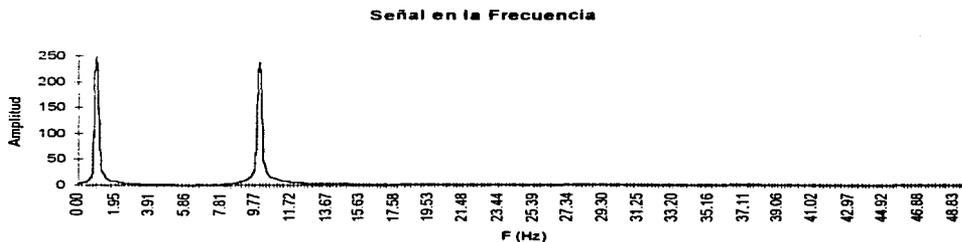


Figura 4.13.  $F(v)$ ,  $v=0, 1, 2, \dots, 512$  y la frecuencia está dada por  $v(0.1953)$ , presentando aquí solamente los primeros 256 puntos.

De este ejercicio se observa la presencia de componentes importantes en los valores de la señal en el dominio de la frecuencia para cada una de las frecuencias presentes en la señal en el dominio del tiempo. Esto será de utilidad también para el siguiente proceso realizable por la aplicación, que es el mapeo electroencefalográfico.

### Mapeo EEG

Esta opción dentro del menú Procesos permite al usuario visualizar de forma accesible los ritmos cerebrales presentes en un EEG mediante una representación cromática de la amplitud

promedio registrada para cada uno de ellos. Para ejecutar este proceso, el usuario deberá primeramente seleccionar el estudio almacenado sobre el cual desea efectuar dicho mapeo. Esto se hace seleccionando dentro del menú Archivo/Abrir el archivo .EEG que contenga el registro de las señales originales para el paciente deseado (recordar que el nombre de los archivos que contienen el registro de las señales originales está formado por seis caracteres que contienen el número del expediente del paciente más dos caracteres con el número del estudio realizado, y todos tienen la extensión .EEG).

Algo que se debe tener siempre presente al utilizar el mapeo EEG es que, antes de ejecutarlo, se debe estar seguro de haber realizado y almacenado el proceso de Análisis Espectral para todos los canales activos. Esto es debido a que el mapeo EEG basa su funcionamiento en los resultados que arroja el análisis espectral para cada canal. Si acaso el usuario olvidara analizar espectralmente alguno de los canales activos, y solicitara efectuar el mapeo EEG, el sistema presentará una ventana con un mensaje de error como la que se muestra en la figura 4.14.



Figura 4.14. Mensaje de error arrojado por el mapeo EEG.

Una vez seleccionado el estudio sobre el que se desea efectuar el mapeo y ya habiendo realizado y almacenado el análisis espectral para todos los canales activos, se desplegará una ventana como la de la figura 4.15.

Este ejemplo muestra un estudio realizado con un montaje de 8 electrodos (FP1, FP2, F7, F3, FZ, F4, F8 y CZ). Como se observa, cada ritmo se muestra en un recuadro por separado, lo que permite apreciar más de cerca su comportamiento durante el estudio. A la derecha se tiene una escala con colores predeterminados los cuales indican la amplitud promedio presentada por cada ritmo.

Para este ejemplo, se observa que durante el estudio *estuvieron* presentes los cuatro ritmos, cada uno con una amplitud predominante diferente, dependiendo de la zona estudiada. Cabe aclarar que el mapeo EEG de la figura 4.15 no corresponde a un estudio con un paciente real, sino que fue el resultado de una simulación que se hizo solamente con fines ilustrativos.

Si el usuario así lo desea, podrá imprimir el resultado del mapeo, seleccionando en el menú principal la opción Archivo/Imprimir o también haciendo clic en el botón Imprimir.

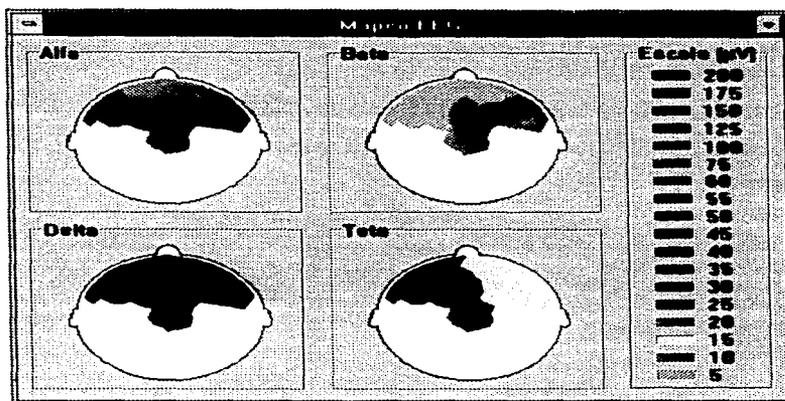


Figura 4.15. Ejemplo del mapeo EEG.

A lo largo de este capítulo se ha mostrado la aplicación de software para el sistema de procesamiento digital de señales EEG. Al programa obtenido se le denomina prototipo, ya que es funcional en toda su extensión, sin embargo, mientras no se implemente la etapa de hardware, no será posible utilizar este programa con datos adquiridos ciertamente en tiempo real. Se ha podido constatar que efectivamente, utilizar la herramienta de programación Delphi ha sido un acierto, dada la facilidad y rapidez con la que fue desarrollada la aplicación, además del aspecto profesional que presenta.

Es importante mencionar que debido a que no se contó con una tarjeta de adquisición de datos, no se pudieron realizar pruebas reales con el muestreo de señales, por lo que si se llega a implementar el sistema será necesario coordinar el muestreo con la visualización dependiendo de los tiempos reales necesarios para cada operación.

Sin embargo, si se cuenta con el software de interfaz para la tarjeta DAQ descrito en el capítulo tres, y la aplicación, si considera, todas las funciones de configuración y muestreo necesarias para llevar a cabo una adquisición de datos efectiva, en tiempo real.

## Conclusiones

El procesamiento digital de señales representa realmente un universo de posibilidades que se debe explotar para auxiliar a profesionales de distintas áreas a obtener un mayor provecho de la información que poseen. Tal ha sido el caso del presente trabajo, en el cual se ocuparon técnicas propias del PDS para presentar distintos enfoques de una misma información básica: las ondas cerebrales. Es así, que hasta aquí se ha recorrido una ruta que comenzó con la presentación del problema, atravesó por la explicación de los diversos temas involucrados en la EEG, tanto a nivel fisiológico como a nivel físico, adquiriendo y ejercitando ciertos conocimientos, y finalmente se presentó el trabajo de diseño. Es decir, se recorrió el *qué pasa, cómo pasa y para qué sirve saberlo*. Todo esto fundamentado no sólo en la búsqueda de la utilidad en la herramienta sino también en su facilidad de uso.

Ahora bien, ya tratándose del sistema diseñado, se debe de empezar por criticar sus limitantes o debilidades para poder en cierto grado elogiar sus virtudes y fortalezas.

Cabe aclarar, que las limitantes que se deben mencionar no tienen por qué ser catalogadas como defectos, simplemente son aspectos que se salieron del alcance pretendido por este trabajo. En primer lugar es preciso tener en cuenta que solamente se ocupó una pequeña parte de lo que constituye en sí toda la teoría disponible sobre el PDS, de hecho este sistema podría mejorarse en diversos aspectos de su biblioteca de herramientas de PDS. Así también, las limitaciones se presentaron en el diseño del hardware, por ejemplo, el número de canales de adquisición de la señal en tiempo real quedó limitado a dieciséis canales (por características de la tarjeta de adquisición), pero posteriormente se podría utilizar multiplexaje si se deseara ampliar el sistema hasta los diecinueve canales previstos por el sistema 10-20 de registro de señales EEG.

En lo referente a las bondades o a la utilidad del sistema, es suficiente justificación saber que un sistema de estas características no sólo ofrece ahorro al reducir los tiempos necesarios para un diagnóstico, sino que también se ahorra en los recursos físicos, simplemente como el papel. Además de que pone al alcance de expertos (y no tan expertos) las bondades del PDS aplicado a las señales EEG, permitiendo que tareas completamente manuales, tediosas y largas, se realicen ahora por procesos automáticos de computadora y no por una persona.

De esta forma se lograron implementar procesos que presentaran la señal registrada en forma de gráfica de tiempo, el espectro de potencia de las ondas cerebrales registradas para clasificación de los ritmos EEG, así como un mapeo electroencefalográfico de las componentes de las señales; todos ellos, procesos de utilidad para el médico o experto que revise los resultados de un estudio practicado.

Por otro lado, se vislumbra con agrado que las posibilidades de crecimiento que desde un principio se han comentado en este trabajo, son factibles, pues se ve la posibilidad de incrementar más adelante los alcances actuales del sistema. Con esto se concluye también que lo más importante es que se ha llegado no a una solución total en el tema de EEG, sino a un destino importante en el camino de el procesamiento digital de señales, puesto que desde aquí resulta mas claro el lograr nuevas metas en el futuro y sobre todo resulta más realizable. Y así como en su momento existió aliento y motivación por parte del trabajo de alguien más para adentrarse a la senda del desarrollo de sistemas de PDS y de esta manera lograr ponerle ingredientes propios de ingenio, de ciencia y por qué no decirlo de arte, así también existe la esperanza de que este trabajo sirva para alentar a otros a seguir trabajando en este mundo de las señales.

Evaluando la aplicación desde el punto de vista de la herramienta en la cual se desarrolló, se observa que realmente resulta útil contar con herramientas que permiten desarrollo rápido de aplicaciones, como es el caso de *Delphi* de Borland, ya que le evitan al desarrollador el gasto excesivo de tiempo en la búsqueda de proveer un ambiente gráfico amigable al usuario, además de que al ser orientado a objetos, permite más fácilmente explotar la modularidad de un diseño y por ende su crecimiento y mantenimiento. Así que resultaría fácil simplemente agregar módulos con rutinas programadas de análisis de las señales y acoplar estos al resto del menú de opciones de procesamiento.

En cuanto a las perspectivas, se debe mencionar que una de las posibilidades más atractivas de cualquier sistema hoy en día es la optimización de los recursos disponibles, y si se está trabajando con señales de larga duración, y además sabiendo que las señales durante varios instantes de tiempo se pueden presentar como repeticiones de instantes previos y futuros, se puede entonces ahorrar una cantidad importante de espacio de almacenamiento y de espacio de memoria de proceso si se obtiene una representación más compacta de la señal. Esta representación pudiera ser las primeras 16 correlaciones [LMP84] de pedazos de señal de longitud fija, por ejemplo 30 segundos, con lo que de entrada se reduce el número de elementos (valores) almacenados de tres mil a dieciséis, esto para una frecuencia de muestreo de 100 Hz. Pero todavía más, si se tiene que hay un cierto número de conjuntos de correlaciones los cuales son lo suficientemente iguales (en base a un umbral), entonces se

---

reducirá la señal a un número  $M$  de clases de correlaciones que representan los  $P$  periodos de la señal representados por cada conjunto de correlaciones, donde  $P \gg M$ , y donde  $M$  es inversamente proporcional al umbral definido. De igual manera, el sistema podría contar después con la función de filtrado digital [OpWi83], con la cual eliminar frecuencias no deseadas dentro de un periodo de señal en determinados canales. En fin, las posibilidades son ilimitadas y las herramientas están disponibles para hacer crecer este tipo de aplicaciones casi tanto como se quiera y se necesite.

Los autores de este trabajo agradecen la lectura, el estudio y la crítica de este *su* trabajo.

## Bibliografía

- Alca88 Alcántara Silva, Rogelio. Apuntes de Procesamiento Digital de Señales. Realizados, revisados e impartidos por el Dr. Alcántara como base para la clase de PDS en la Maestría en Eléctrica de la DEPEFI, 1988
- Asto90 Aston, Richard. Principles of Biomedical Instrumentation and Measurement. Merril Publishing Company, Columbus, Ohio 1990.
- BoNa96 Boylestad, Robert and Nashelsky, Louis. Electronic Devices and Circuit Theory. Prentice Hall, U.S.A., 1996.
- BeGe90 Berlin, Howard M., Getz Jr., Franc C. Fundamentals of operational amplifiers and linear integrated circuits, Merril Publishing Company, 1990.
- Brac78 Bracewell, R., The Fourier Transform and Its Applications, McGraw-Hill, New York, 1978.
- Brac83 Bracewell, R. N., Discrete Hartley Transform, J. Opt. Soc. Amer., vol 73, pp. 1832-1835, Dec. 1983.
- Brac84 Bracewell, R. N., The fast Hartley transform, Proc. IEEE, vol. 72, pp. 1010-1018, Aug. 1984.
- Brac86 Bracewell, Ronald N., The fast Hartley transform, New York: Oxford University Press, 1986.
- Cerc92 Cechione, Robert G., Data acquisition made simpler, IEEE Spectrum, Noviembre 1992, pp. 66-71.
- CoDr93 Coghlin, Robert F. y Driscoll, Frederick F. Amplificadores Operacionales y Circuitos Integrados Lineales, Prentice Hall (México), 1993.
- CoTu65 Cooley, J. W. y J. N. Tukey, An Algorithm for the Machine Calculation of Complex Fourier Series, Math. Comput., Vol. 19, No. 2, pp. 297-301, April 1965.
- Dail89 Dailey, Denton J. Operational Amplifiers and Linear Integrated Circuits. Theory and Applications, McGraw-Hill, 1989.
- DeMi83 DeMarre, Den A., Michaels, David, Bioelectronic Measurements, Prentice Hall Inc., 1983.

- EmKi91 Embree, Paul M. and Kimble, Bruce C language algorithms for digital signal processing, Prentice Hall, 1991. pp. 109-115. TK5102. 5. E53
- EySa93 Eynde, Frank, Sansen, Willy Analog Interfaces for Digital Signal Processing Systems, Kluwer Academic Publishers, 1993.
- GaRo75 Gabel, Robert A. y Richard A. Roberts., Signals and Linear Systems, John Wiley and Sons, Inc. 1973.
- HJBu84 Heideman, M. T., D. Johnson, y C.S. Burrus, Gauss and the history of the Fast Fourier Transform, IEEE ASSP Magazine, Vol. 1, No. 4, pp. 14-21, Oct. 1984.
- Hou95 House, Richard, Choosing the right software for data acquisition, IEEE Spectrum, Mayo 1995, pp. 24-39.
- HouH87 Hou, Hsieh S., The Fast Hartley Transform Algorithm, IEEE Transactions on Computers, Vol. C-36, No. 2, pp. 147-156, February 1987.
- IWZe81 Isaksson, Anders, Arne Wennberg and Lars H. Zetterberg, Computer Analysis of EEG Signals with Parametric Models. Proceedings of the IEEE. Vol. 69. No. 4. April 1981.
- JaCo92 Jacobs, Ernest C., Collura, Thomas F., The Epilog System. Automated Long-Term EEG Monitoring for Epilepsy, IEEE Computer, September 1992, pp. 5-14.
- KaMa81 Kay, Steven M. y Marple Jr., Stanley Lawrence. Spectrum Analysis - A Modern Perspective. Proceedings of the IEEE. Vol. 69. No. 11. November 1981. pp. 1380-1419.
- Lara87 Lara y Zavala, Rolando. Cibernética del Cerebro, Compañía Editorial Continental (México), 1987.
- Line91 Lines, David. Building Power Supplies, Master Publishing (Texas), 1991.
- LMPr84 Le Roux, J., N. Moreau y J. Prado, Real-time analysis and semi-automatic classification of sleep EEG signals using a microprocessor, pp. 521-544. 1984.
- Mann96 Manning, Michelle; Delphi 2 Guía Oficial de Borland, Prentice-Hall, México, 1996.
- Mart89 Martínez R., Arturo Instrumentación Digital. Convertidores Digital-Analógico, LIMUSA-AMICEE, 1989.
- Meik90 Meiksw, Z. H. Complete Guide to Active Filter Design, Operational Amplifiers, and Pasive Components, Englewod Cliffs, N.J. Prentice Hall, 1990. TK7872 F5 M45

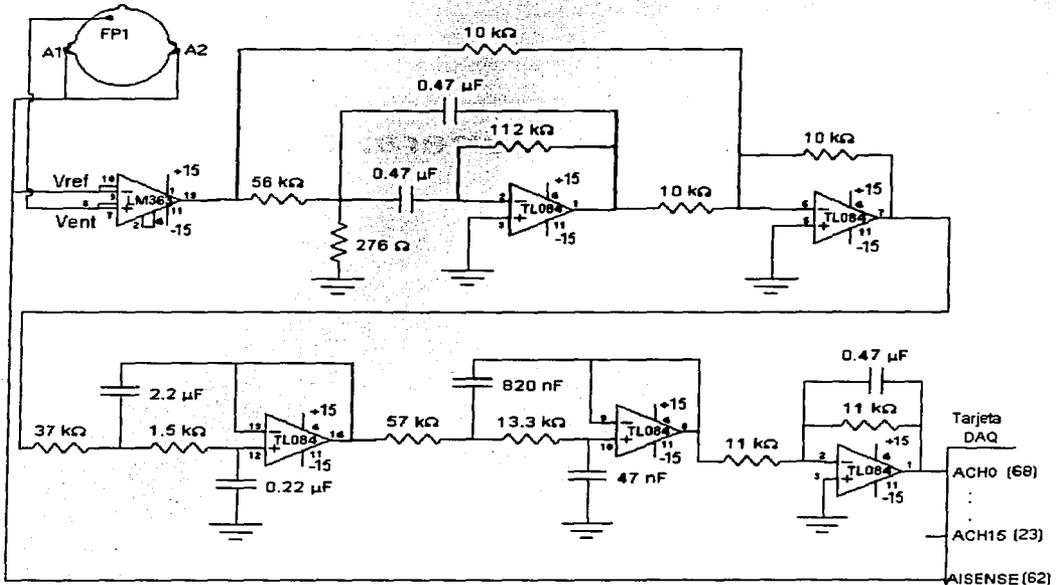
- MiHo79 Michael, D. y J. Houchin, Automatic EEG Analysis: a segmentation procedure based on the autocorrelation function, Electroencephalography and Clinical Neurophysiology, 1979, 46, 232-235
- Nati88 National Semiconductor Corporation, Linear Databook 1, 1988
- Nati96 National Instruments Corporation, NI-DAQ Function Reference Manual for PC Compatibles, 1996.
- Nati97 National Instruments, Instrumentation Reference and Catalogue, 1997.
- Natn96 National Instruments Corporation, DAQ Hardware Overview Guide, 1996.
- Nats96 National Instruments Corporation, NI-DAQ User Manual for PC Compatibles, 1996.
- O'Ne88 O'Neill, Mark A., Faster Than Fast Fourier, Byte magazine, April 1988.
- Oppe86 Oppenheim, Alan V. Advanced Topics in Signal Processing. Series Editor.
- OpWi83 Oppenheim, Alan V. and Willsky, Alan S., Señales y Sistemas. Prentice Hall, 1983. Primera edición
- OsGB96 Osier, Dan; Grobman, Steve y Batson, Steve; Aprendiendo Delphi 2 en 21 días, Prentice-Hall, México, 1996.
- Pfle91 Pfleeger, Shari Lawrence. Software engineering: the production of quality software Maxwell Macmillan International Editions. 2nd. Edition 1991. QA76. 758. P45
- RaGo75 Rabiner, L. R. and B. Gold, Theory and Application of Digital Signal Processing, Prentice-Hall, Englewood Cliffs, NJ, 1975.
- Webs92 Webster, John G. Medical Instrumentation: Application and Design, Houghton Mifflin Company, 1992.
-

## **Apéndices**

---

## Apéndice 1

# Diagrama General del Hardware

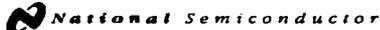


El diagrama representa las etapas de registro, amplificación, filtrado y entrada a la tarjeta de adquisición de datos. Obsérvese que solamente se muestra el circuito correspondiente a un canal, los otros quince tienen una configuración idéntica, conectándose a la tarjeta DAQ en la entrada correspondiente (para el canal "N" utilizar la entrada ACHN). Para todos los canales la referencia son los electrodos colocados en A1 y A2, los cuales se conectan a la entrada AISENSE de la tarjeta DAQ.

## **Apéndice 2**

### **Hojas de Datos de Hardware**

En este apéndice se muestran las hojas de datos de todos los componentes electrónicos utilizados en el diseño del hardware del sistema así como las especificaciones de la tarjeta de adquisición de datos elegida. Las hojas se presentan en el formato original publicado por el fabricante sin realizarles ninguna modificación a su contenido.



April 1981

## LM363 Precision Instrumentation Amplifier

### General Description

The LM363 is a monolithic true instrumentation amplifier. It requires no external parts for fixed gains of 10, 100 and 1000. High precision is attained by on-chip trimming of offset voltage and gain. Super-beta bipolar input stage drives very low input bias current and voltage noise, extremely low offset voltage drift, and high common-mode rejection ratio. A two-stage amplifier design yields an open loop gain of 10,000,000 and a gain bandwidth product of 30 MHz, yet remains stable for all closed loop gains. The LM363 operates with supply voltages from  $\pm 5V$  to  $\pm 18V$  with only 1.5 mA current drain.

The LM363's low voltage noise, low offset voltage and offset voltage drift make it ideal for amplifying low-level, low-impedance transducers. At the same time, its low bias currents and high input impedance (both common-mode and differential) provide excellent performance at high impedance levels. These features, along with its ultra-high common-mode rejection, allow the LM363 to be used in the most demanding instrumentation amplifier applications, replacing expensive hybrid, module or multi-chip designs. Because the LM363 is internally trimmed, precision external resistors and their associated errors are eliminated.

The 16-pin dual-in-line package provides pre-adjustable gains of 10, 100 or 1000, its two differential shield drivers

eliminate bandwidth loss due to cable capacitance. Compensation pins allow over-compensation to reduce bandwidth and output noise, or to provide greater stability with capacitive loads. Separate output force sense and reference pins permit gains between 10 and 1000 to be programmed using external resistors.

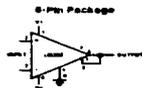
On the 8-pin metal can package, gain is internally set at 10, 100 or 500 but may be increased with external resistors. The shield driver and offset adjust pins are omitted on the 8-pin version.

The LM363 is rated for  $0^{\circ}C$  to  $70^{\circ}C$ .

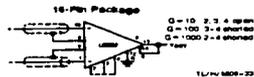
### Features

- Offset and gain pretrimmed
- 12 nV/ $\sqrt{Hz}$  input noise ( $G = 500/1000$ )
- 130 dB CMRR typical ( $G = 500/1000$ )
- 2 mA bias current typical
- No external parts required
- Dual shield drivers
- Can be used as a high performance op amp
- Low supply current (1.5 mA typ)

### Typical Connections

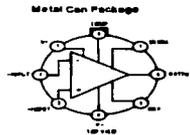


TLV9008-1

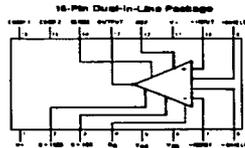


TLV9008-2

### Connection Diagrams



Order Number LM363M-10,  
LM363N-100 or LM363M-500  
See NS Package Number M08C



Order Number 363D  
See NS Package Number D16C

TLV9008-3

© 1981 National Semiconductor Corporation, TLV9008

NSD 8222-116-0 (Rev. 11-80) U.S.A.

LM363 Precision Instrumentation Amplifier

# Diseño de un Sistema de Procesamiento Digital de Señales Electroencefalográficas

<b>Absolute Maximum Ratings</b> (Note 5)					
If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.			Input Voltage	Equal to Supply Voltage	
Supply Voltage			Reference and Sense Voltage	±25V	
Differential Input Voltage			Lead Temp. (Soldering, 10 sec)	300°C	
Input Current			ESD rating to be determined		
<b>LM363 Electrical Characteristics</b> (Notes 1 and 2)					
Parameter	Conditions	LM363			Units
		Typ	Tested Limit (Note 3)	Design Limit (Note 4)	
<b>FIXED GAIN (8-PIN)</b>					
Input Offset Voltage	G = 500	30	150	400	μV
	G = 100	50	250	700	μV
	G = 10	0.5	2.5	5	mV
Input Offset Voltage Drift	G = 500	1		4	μV/°C
	G = 100	2		5	μV/°C
	G = 10	20		75	μV/°C
Gain Error (±10V Swing, 2 kΩ Load)	G = 500	0.1	0.5	0.5	%
	G = 100	0.07	0.7	0.5	%
	G = 10	0.05	0.5	0.7	%
<b>PROGRAMMABLE GAIN (16-PIN)</b>					
Input Offset Voltage	G = 1000	50	250	800	μV
	G = 100	100	450	800	μV
	G = 10	1	3.5	5	mV
Input Offset Voltage Drift	G = 1000	1		5	μV/°C
	G = 100	2		10	μV/°C
	G = 10	10		100	μV/°C
Gain Error (±10V Swing, 2 kΩ Load)	G = 1000	2.0	3.0	3.5	%
	G = 100	0.1	0.7	0.5	%
	G = 10	0.6	2.0	3.5	%
<b>FIXED GAIN AND PROGRAMMABLE</b>					
Gain Temperature Coefficient	G = 1000	40			ppm/°C
	G = 500	20			ppm/°C
	G = 100, 10	10			ppm/°C
Gain Non-Linearity (±10V Swing, 2 kΩ Load)	G = 10, 100	0.01	0.03	0.05	%
	G = 500, 1000	0.01	0.05	0.05	%

LM363 Electrical Characteristics (Continued) (Notes 1 and 2)					
Parameter	Conditions	LM363			Units
		Typ	Tested Limits (Note 3)	Design Limit (Note 4)	
Common-Mode Rejection Ratio ( $-10V \leq V_{CM} \leq 10V$ )	$G = 1000, 500$	130	114	100	dB
	$G = 100$	120	94	80	dB
	$G = 10$	105	80	60	dB
Positive Supply Rejection Ratio (5V to 15V)	$G = 1000, 500$	130	110	100	dB
	$G = 100$	120	100	80	dB
	$G = 10$	100	85	70	dB
Negative Supply Rejection Ratio ( $-5V$ to $-15V$ )	$G = 1000, 500$	120	100	80	dB
	$G = 100$	106	85	70	dB
	$G = 10$	95	70	60	dB
Input Bias Current		2	10	20	nA
Input Offset Current		1	3	8	nA
Common-Mode Input Resistance		100	8		G $\Omega$
Differential Mode Input Resistance	$G = 1000, 500$	0.2			G $\Omega$
	$G = 10$	2			G $\Omega$
Input Offset Current Change	$-11V \leq V_{CM} \leq 13V$	20	100	200	pA/V
Reference and Sense Resistance	Min	50	30	27	$\Omega$
	Max		80	80	$\Omega$
Open Loop Gain	$G_{CL} = 1000, 500$	10	1		V/ $\mu$ V
Supply Current	Positive	1.2	2.4	3.0	mA
	Negative	1.6	2.8	3.5	mA

Note 1: These conditions apply unless otherwise noted.  $V^+ = 15V$ ,  $V^- = -15V$ ,  $V_{CM} = 0V$ ,  $R_L = 2\text{ k}\Omega$ , reference pin grounded, sense pin connected to output and  $T = 25^\circ\text{C}$ .  
 Note 2: Reference limits are guaranteed over full temperature range. Operating ambient temperature range is  $0^\circ\text{C}$  to  $70^\circ\text{C}$  for the LM363.  
 Note 3: Guaranteed and 100% production tested.  
 Note 4: Guaranteed but not 100% tested. These limits are not used in determining shipping quality levels.  
 Note 5: Maximum rated junction temperature is  $100^\circ\text{C}$  for the LM363. Thermal resistance, junction to ambient, is  $150^\circ\text{C/W}$  for the TO-99C package and  $100^\circ\text{C/W}$  for the ceramic D-P (D).

**Application Hints**

The LM333 was designed to be as simple to use as possible, but several general precautions must be taken. The differential inputs are directly coupled and need a return path to ground supply common. Worst-case bias currents are only 10 nA for the LM333, but the load impedance can be as high as 100 M $\Omega$ . Ground loops between signal return and IC supply common should not be ignored. While the LM333 has excellent common-mode rejection, signals must be returned within the proper common-mode range for the specification to apply. Operating common-mode range is guaranteed from  $-10V$  to  $+10V$  with  $\pm 15V$  supplies.

The high-gain (500 or 1000) versions have large gain-bandwidth products (15 MHz or 30 MHz) so board layout is fairly critical. The differential input leads should be kept away from output force and sense leads, especially at high impedances. Only 1 pF from output to positive input at 100 k $\Omega$  source impedance can cause oscillations. The gain adjust leads on the 16-pin package should be treated as inputs and kept away from the output wiring.

**POWER SUPPLY**

The LM333 may be powered from split supplies from  $\pm 5V$  to  $\pm 18V$  (or single-ended supplies from 10V to 36V). Positive supply current is typically 1.2 mA independent of supply voltage. The negative supply current is higher than the positive by the current drawn through the voltage dividers for the reference and sense inputs (vs 800  $\mu A$  total). The LM333's excellent PSRR often masks regulated supplies unnecessarily. Actually, supply voltage can be as low as 7V total, but PSRR is severely degraded so that well-regulated supplies are recommended below 10V total. Split supplies need not be balanced; output swing and input common-mode range will simply not be symmetrical with unbalanced supplies. For example, at  $+12V$  and  $-5V$  supplies, input common-mode range is typically  $+10.5V$  to  $-3V$  and output swing is  $+11V$  to  $-4V$ .

When using ultra-low offset versions, best results are obtained at  $\pm 15V$  supplies. For example, the LM333-500's offset voltage is guaranteed within 150  $\mu V$  at  $\pm 15V$  at 25°C. Running at  $\pm 5V$  results in a worst-case negative PSRR error of 10V ( $-15V$  to  $-5V$ ) multiplied by  $3 \times 10^{-4}$  (110 dB) or 32  $\mu V$ , increasing the worst-case offset. Positive PSRR results in another 10  $\mu V$  worst-case change.

**INPUTS**

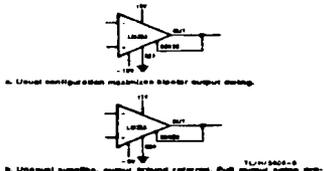
The LM333 input circuitry is depicted in the Simplified Schematic. The input stage is run relatively non ( $50 \mu A$ ) for low voltage noise and wide bandwidth, super-beta transistors and bias-current cancellation (not shown) keep bias currents low. Due to the bias-current cancellation circuitry, bias current may be either polarity at either input. Where input current noise is high relative to bias current, it is not significant until source resistance approaches 100 dB. Input common-mode range is typically from 3V above  $V^-$  to 1.5V below  $V^+$ , so that a large potential drop between the input signal and output reference can be accommodated. However, a return path for the input bias current must be provided; the differential input stage is not isolated from the supplies. Differential input swing in the linear region is equal to output swing divided by gain and typically ranges from 1.3V at  $G=10$  to 13 mV at  $G=1000$ . Clamp diodes are provided to prevent zener breakdown and resulting degradation of the input transistors. At large input

overdrives these diodes conduct, greatly increasing input currents. This behavior is illustrated in the  $I_{in}$  vs  $V_{in}$  plot in the Typical Performance Characteristics. (The graph is not symmetrical because at large input currents a portion of the current into the device flows out the  $V^-$  terminal.) The input protection resistors allow a full 10V differential input voltage without degradation even at  $G=1000$ . At input voltages more than one diode drop below  $V^-$  or two diode drops above  $V^+$ , input current increases rapidly. Diode clamps to the supplies or external resistors to limit current to 20 mA will prevent damage to the device.

**REFERENCE AND SENSE INPUTS**

The equivalent circuit is shown in the schematic diagram. Limitations for correct operation are as follows. Maximum differential swing between reference and sense pins is typically  $\pm 12V$  ( $\pm 10V$  guaranteed) if this limit is exceeded, the sense pin no longer controls the output, which then goes high or low. The negative common-mode limit is 1.5V below  $V^-$ . (This is permissible because R2 and R4 are returned to a node biased higher than  $V^-$ .) If large positive voltages are applied to the reference and sense pins, the common-mode range of the signal input begins to suffer as the drop across R13 and R16 increases. For example, at  $\pm 15V$  supplies,  $V_{ref} = V_{sense} = 0V$ , signal input range is typically  $-12V$  to  $+13.5V$ . At  $V_{ref} = V_{sense} = 15V$ , signal input range drops to  $-11V$  to  $+13.5V$ . The reference and sense pins can be as much as 10V above  $V^+$  as long as a restricted signal common-mode range ( $-10V$  min) can be tolerated.

For maximum bipolar output swing at  $\pm 15V$  supplies, the reference pin should be returned to a voltage close to ground. At lower supply voltages, the reference pin need not be halfway between the supplies for maximum output swing. For example, at  $V^+ = +12V$  and  $V^- = -5V$ , grounding the reference pin still allows a  $-11V$  to  $-4V$  swing. For single-supply systems, the reference pin can be tied to either supply if a single output polarity is all that is required. For a bipolar input and output, create a low impedance reference with an op amp and voltage divider or a regulator (e.g. LM336, LM385, LM317). This forms the reference for all succeeding signal-processing stages. Don't connect the reference terminal directly to a voltage divider; this degrades gain error! See Figure 1.



**FIGURE 1. Reference Connections**

**Application Hints (Continued)**

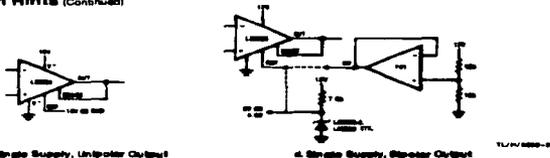


FIGURE 1. Reference Connections (Continued)

**OUTPUTS**

The LM363's output can typically swing within 1V of the supplies at light loads. When specified to drive a 2 k $\Omega$  load to  $\pm 10V$ , current limit is typically 15 mA at room temperature. The output can safely drive capacitive loads up to 400 pF. For higher load capacitance the amplifier may be overcompensated (see COMPENSATION section, following). The output may be continuously shorted to ground without damaging the device.

**OFFSET VOLTAGE**

The LM363's offset voltage is internally trimmed to a very low value. Note that data sheet values are given at  $T_c = 25^\circ C$ ,  $V_{CC} = 0V$  and  $V^- = V^+ = 15V$ . For other conditions, warm-up drift, temperature drift, common-mode rejection and power supply rejection must be taken into account. Warm-up drift (due to chip and package thermal gradients), is an effect separate from temperature drift. Typical warm-up drift is tabulated in the Electrical Characteristics; settling time is approximately 5 minutes in still air. At load currents up to 5 mA, thermal feedback effects are negative:  $\Delta V_{OS} = -2\mu V$  at  $G = 1000$ .

Care must be taken in measuring the extremely low offset voltages of the high gain amplifiers; input leads must be held isothermal to eliminate thermopile effects. Oscillations due to either heavy capacitive loading or stray capacitance from input to output, can cause erroneous readings; in either case, overcompensation will help. High frequency noise fed into the inputs may be rectified internally, and pro-

duce an offset shift. A simple low-pass RC filter will usually cure this problem (Figure 2). Use film type resistors for their low thermal EMF in highly noisy environments. LC filters can be substituted for increased RF attenuation.

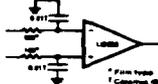


FIGURE 2. Low Pass Filter Prevents RFI Rectification

Instrumentation amplifiers have both an input offset voltage ( $V_{OS}$ ) and an output offset voltage ( $V_{OOS}$ ). The total input-referred offset voltage ( $V_{OS(amp)}$ ) is related to the instrumentation amplifier gain ( $G$ ) as follows:  $V_{OS(amp)} = V_{OS} + V_{OOS}/G$ . The offset voltage given in the LM363 specifications is the total input-referred offset. As long as only one gain is used, offset voltage can be ruled at either input or output as shown in Figures 3a and 3b. When the 16-pin device is used at multiple gain settings, both  $V_{OS}$  and  $V_{OOS}$  should be ruled to get minimum offset at all gains as shown in Figure 3c. The correct procedure is to trim  $V_{OOS}$  for zero output at  $G = 10$ , then trim  $V_{OS}$  at  $G = 1000$ .

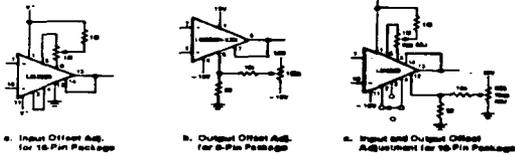


FIGURE 3. Offset Voltage Trimming

**Application Hints (Continued)**

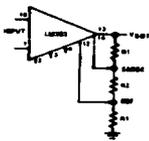
Because the LM363's offset voltage is so low to begin with, offset nulling has a negligible effect on offset temperature drift. For example, zeroing a 100- $\mu$ V offset assuming external resistor TC of 200 ppm/°C and worst-case internal resistor TC results in an additional drift component of 0.08  $\mu$ V/°C. For this reason, drift specifications are guaranteed, with or without external offset nulling.

**Gain Adjustment**

Gain may be increased by adding an external voltage divider between output force and sense and reference. The preferred connection is shown in Figure 4. Since both the sense and reference pins look like 50 k $\Omega$  (±20 k $\Omega$ ) to V<sup>-</sup>, impedances presented to both pins must be equal to avoid offset error. For example, a 100k $\Omega$  impedance can create a

worst-case output offset of 50 mV, creating an input-referred error of 5 mV at G = 10 or 50  $\mu$ V at G = 1000. Increasing gain this way increases output offset error. An LM363H-100 may have an output offset of 5 mV, resulting in input-referred offset component of 50  $\mu$ V. Raising the gain to 200 yields a 10 mV error at the output and changes input-referred error by an additional 50  $\mu$ V.

External resistors connected to the reference and sense pins can only increase the gain if ultra-low output impedance is not critical; the technique in Figure 5 can be used to trim the gain to normal value. Alternatively, the V<sub>OS</sub> adjustment terminals on the 16-pin package may be used to trim the gain (Figure 10b).



R1 and R2 should be as low as possible to limit errors due to 50 k $\Omega$  input impedance of reference and sense pins. Total resistance (R1 + 2R2) should be about 4 k $\Omega$ , however, to prevent excessive load on the LM363 output. The exact formula for calculating gain (G) is:

$$G = G_0 \left( 1 + \frac{2R1}{R2} \frac{R1}{50k} \right)$$

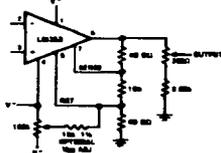
G<sub>0</sub> = preset gain

The last term may be ignored in applications where gain accuracy is not critical. The table below gives suggested values for R1 and R2 along with the calculated error due to "nominal" values. However, 1% resistor total gain error (resistor impedance contributions from LM363 G<sub>0</sub> error and resistor impedance (±1%) and worst but to approximately 2.5% in every case. Please refer to 16 pin package. The same technique can also be used with 8 pin version.

TLV5600-13

Gain Increase	1.5	2	2.5	3	4	5	6	7	8	9	10
R1	1.21k	1.21k	2k	2k	1.75k	2k	2.49k	2.94k	3.48k	3.92k	4.42k
R2	5k	2.49k	2.74k	2.05k	1.21k	1k	1k	1k	1k	1k	1k
Error (typ)	-0.6%	-0.2%	0	-0.3%	-0.6%	-0.8%	-0.5%	-0.9%	-0.4%	-0.9%	-0.7%

FIGURE 4. Increasing Gain



Please refer to 16 pin version. The same technique can also be used with 8 pin version.

FIGURE 5. Adjusting Gain, Alternate Technique

**Application Hints (Continued)**

**COMPENSATION AND OUTPUT CLAMPING**

The LM353 is internally compensated for unity feedback from output to inverting input. Increasing gain with external devices will decrease the bandwidth and increase stability margin. Without external compensation, the amplifier can stably drive capacitive loads up to 400 pF. When used as an op amp (inverting and noninverting) and feedback to inverting input, the LM353 is stable for gains of 100 or more. For greater stability, the device may be overcompensated as in Figure 6, Tables I and II to correct suggested compensation components along with the resulting changes in large and small signal bandwidth for the 8-pin and 16-pin packages, respectively.

Note that the RC network from pin 8 of the 8-pin device to ground has a large effect on power bandwidth, especially at low gains. The Miller capacitance utilized for overcompensating the 16-pin device permits higher slew rate and larger load capacitance for the same bandwidth, and is preferred when bandwidth must be greatly reduced (e.g. to reduce output noise).

Heavy Miller overcompensation on the 16-pin package can degrade AC PSRR. A large capacitor between pins 15 and 16 couples transients on the positive supply to the output buffer. Since the amplifier bandwidth is severely rolled off, it cannot keep the output at the correct state at moderate frequencies. Hence, for good PSRR, either keep the Miller capacitance under 1000 pF or use the pin 15-to-ground compensation shown in Table I.

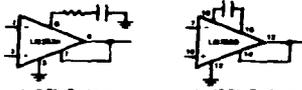


FIGURE 6. Overcompensation

TABLE I. Overcompensation on 8-Pin Package

Gain	Compensation Network (Pin 8 to Ground)*	Small Signal 3 dB Bandwidth (Hz)	Power Bandwidth ( $\pm 10V$ Swing) (Hz)	Maximum Capacitive Load (pF)
500	—	125	100k	400
	100 pF 15k	95	15k	600
	1000 pF 5k	45	1.8k	800
	0.01 $\mu$ F 500 $\Omega$	10	200	1000*
100	—	240	100k	400
	100 pF 15k	170	15k	900
	1000 pF 5k	80	1.8k	1300
	0.01 $\mu$ F 500 $\Omega$	20	300	1800*
10	—	240	100k	400
	100 pF 15k	170	15k	900
	1000 pF 5k	90	1.8k	1200
	0.01 $\mu$ F 500 $\Omega$	20	200	1800*
	0.1 $\mu$ F	2	20	2000*

\* Also stable for  $C_2 = 0.01 \mu F$   
 \* Pin 15 is ground on 16-pin package

TABLE II. Overcompensation on 16-Pin Package

Gain	Compensation Capacitor (Pin 15 to 16)	Small Signal 3 dB Bandwidth (Hz)	Power Bandwidth ( $\pm 10V$ Swing) (Hz)	Maximum Capacitive Load (pF)
1000	—	45*	45k	1000*
	10 pF	16k	16k	2000*
	100 pF	2.5k	2.5k	2500*
	1000 pF	250	250	3000*
100	—	140k	100k	800
	10 pF	50k	50k	1600
	100 pF	7.5k	7.5k	2000*
	1000 pF	750	750	2500*
10	—	180k	90k	600
	10 pF	9k	9k	1100
	100 pF	9k	9k	1600
	1000 pF	80	80	2000*
	0.01 $\mu$ F	80	80	2000*

\* Also stable for  $C_2 = 0.05 \mu F$

**Application Hints (Continued)**

Because the LM353's output voltage is approximately one diode drop below the voltage at pin 15 (pin 8 for the 8-pin device), the diode may be used to limit output swing as seen in Figure 7c. Current available from the pin is only 50  $\mu$ A, so that zeners must have a sharp breakdown to clamp accurately. Alternatively, a diode tied to a voltage source could be used as in Figure 7d.

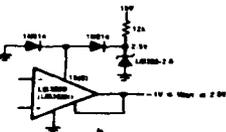


FIGURE 7. Output Clamp

**SHIELD DRIVERS**

When differential signals are sent through long cables, three problems occur. First, noise both common-mode and differential is picked up. Second, signal bandwidth is reduced by the RC low-pass filter formed by the source impedance and the cable capacitance. Finally, when these RC time constants are not identical (unbalanced source impedance and/or unbalanced capacitance), AC common-mode rejection is degraded, amplifying both induced noise and ground noise. Either filtering at the amplifier inputs or slowing down the amplifier by overcompensating will indeed reduce the noise, but the price is slower response. The LM353D's dual shield drivers can actually increase bandwidth while reducing noise.

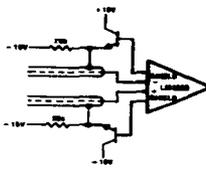
The way this is done is by bootstrapping out shield capacitance. The shield drivers follow the input signal. Since both sides of the shield capacitance swing the same amount, it is effectively out of the circuit at frequencies of interest since the input signals are not in phase and AC CMRR is not degraded (Figure 8). The LM353D shield drivers can handle capacitances (shied to center conductor) as high as 1000 pF with source resistances up to 100  $\Omega$ . For best results, identical shielded cables should be used for both signal inputs, although small mismatches in shield driver to ground capacitance (1-500 pF) do not cause problems. At certain low values of cable capacitance (50 pF-200 pF), high frequency oscillations can occur at high source resistance ( $\geq 10 \text{ k}\Omega$ ). This is alleviated by adding

50 pF to ground at both shield driver outputs. Do not use only one shield driver for a single-ended signal; oscillations can result; shield driver to input capacitance must be roughly balanced ( $\pm 20\%$ ). To further reduce noise pickup, the shielded signal lines may be enclosed together in a grounded shield. If a large amount of RF noise is the problem, the only sure cure is a filter capacitor at both inputs, otherwise the RFI may be internally rectified, producing an offset.

DC loading on the shield drivers should be minimized. The drivers can only source approximately 40  $\mu$ A, above the value the input stage base voltages change, degrading  $V_{OS}$  and CMRR. With the shield drivers can sink several mA,  $V_{OS}$  may degrade severely at loads above 100  $\mu$ A (see Shield Driver Loading Error curve in Typical Performance Characteristics). Because the shield drivers are one diode drop above the input levels, unbalanced leakage paths from shield to input can produce an input offset at high source impedances. Buffering with emitter-followers (Figure 8b) reduces the leakage current by reducing the voltage differential and eliminates any loading on the amplifier.



a. Standard Configuration



b. NPN Followers to Reduce Offset

FIGURE 8. Driving Shielded Cables

**MISCELLANEOUS TRIMMING**

The  $V_{OS}$  adjust and shield driver pins available on the 16-pin package may be used to trim the other parameters besides offset voltage, as illustrated in Figure 10. The bias-current trim relies on the fact that the voltage on the shield driver and gain setting pins is one diode drop respectively above and below the input voltage. Input bias current can be held to within 100 pA over the entire common-mode range, and input offset current always stays under 30 pA. The CMRR trim uses the shield driver pins to drive the  $V_{OS}$  adjust pins, thus maintaining the LM353's ultra-high input impedance.

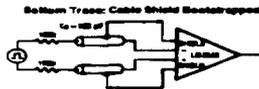
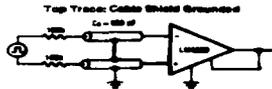
**Application Hints** (Continued)

If power supply rejection is critical, frequently only the negative PSRR need be adjusted, since the positive PSRR is more tightly specified. Any or all of the trim schemes of Figure 10 can be combined as desired. As long as the center tap of the 15kΩ potentiometer returned to a voltage 200 mV below  $V^-$ , the trim schemes shown will not greatly affect

$V_{OS}$ . Both the gain and DC CMRR trim can degrade positive PSRR; the positive PSRR can then be raised and if desired the correct order of trimming from left to right is bias current, gain, CMRR, negative PSRR, positive PSRR and  $V_{OS}$ .

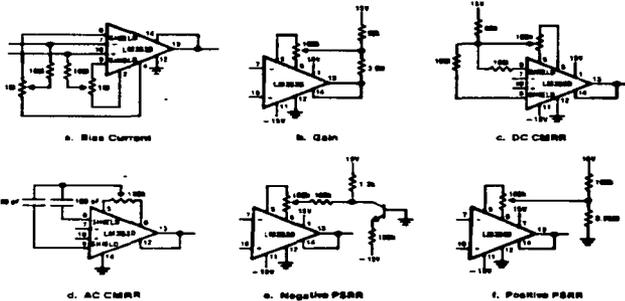


TU/15408-17



TU/15408-18

FIGURE 9. Improved Response using Shield Drivers



TU/15408-19

FIGURE 10. Other Trims for 15-Pin Package

LM363 Precision Instrumentation Amplifier

**Physical Dimensions** (inches (millimeters) (Continued))

HERMETIC DUAL-IN-LINE PACKAGE (D)  
 Order Number LM363D  
 NS Package Number D16C

**LIFE SUPPORT POLICY**

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which (a) are intended for surgical implant into the body of (b) support or sustain life, and whose failure to perform when properly used in accordance with instructions for use provided in the labeling can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

<p><b>National Semiconductor Corporation</b>                  1901 West Park Drive                  Sunnyvale, CA 95051                  Tel: (408) 737-2000                  Fax: (408) 737-1700</p>	<p><b>National Semiconductor Europe</b>                  P.O. Box 51, Park Road                  Colindale Avenue, London NW9 1EQ                  England                  Tel: (44) 0181-608330                  Telex: 9140                  Fax: (44) 0181-608340</p>	<p><b>National Semiconductor Hong Kong Ltd.</b>                  250 Queen's Road East                  Queen's Center, 5th Floor                  Hong Kong, China                  Tel: (852) 2533 6222                  Fax: (852) 2533 6220</p>	<p><b>National Semiconductor Japan Ltd.</b>                  3-1-1 Shimo-Ogino 4-Chome                  Atsugi-City, Kanagawa-Pref.                  243-0292                  Japan                  Tel: (81) 468-552311                  Fax: (81) 468-552312</p>
---	---	---	--

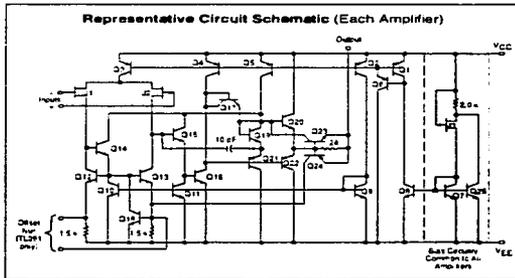


## JFET Input Operational Amplifiers

These low-cost JFET input operational amplifiers combine two state-of-the-art linear technologies on a single monolithic integrated circuit. Each internally compensated operational amplifier has well matched high voltage JFET input devices for low input offset voltage. The BIFET technology provides wide bandwidths and fast slew rates with low input bias currents, input offset currents, and supply currents.

These devices are available in single, dual and quad operational amplifiers which are pin-compatible with the industry standard MC1741, MC1458, and the MC3403/LM324 bipolar products.

- Input Offset Voltage Options of 6.0 mV and 15 mV Max
- Low Input Bias Current: 30 pA
- Low Input Offset Current: 5.0 pA
- Wide Gain Bandwidth: 4.0 MHz
- High Slew Rate: 13 V/ $\mu$ s
- Low Supply Current: 1.4 mA per Amplifier
- High Input Impedance:  $10^{12} \Omega$



### ORDERING INFORMATION

Op Amp Function	Device	Operating Temperature Range	Package
Single	TL081ACD, CD	$T_A = 0^\circ \text{ to } +70^\circ \text{C}$	SO-8
	TL081ACP, CP		Plastic DIP
Dual	TL082ACD, CD	$T_A = 0^\circ \text{ to } +70^\circ \text{C}$	SO-8
	TL082ACP, CP		Plastic DIP
Quad	TL084CN, CN	$T_A = 0^\circ \text{ to } +70^\circ \text{C}$	Plastic DIP

Order this document by TL081C/D

**TL081C,AC  
TL082C,AC  
TL084C,AC**

## JFET INPUT OPERATIONAL AMPLIFIERS

**SEMICONDUCTOR TECHNICAL DATA**

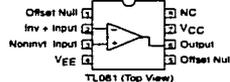


**P SUFFIX  
PLASTIC PACKAGE  
CASE 628**

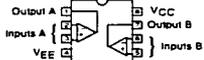


**D SUFFIX  
PLASTIC PACKAGE  
CASE 751  
(SO-8)**

### PIN CONNECTIONS



TL081 (Top View)

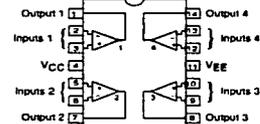


TL082 (Top View)



**N SUFFIX  
PLASTIC PACKAGE  
CASE 848**

### PIN CONNECTIONS



TL084 (Top View)

© Motorola, Inc. 1996

Rev D

**TL081C, AC TL082C, AC TL084C, AC**

**MAXIMUM RATINGS**

Rating	Symbol	Value	Unit
Supply Voltage	VCC VEE	+18 -18	V
Differential Input Voltage	V <sub>ID</sub>	±30	V
Input Voltage Range (Note 1)	V <sub>IDR</sub>	±15	V
Output Short Circuit Duration (Note 2)	I <sub>SC</sub>	Continuous	
Power Dissipation Plastic Package (N, P) Derate above T <sub>A</sub> = +47°C	P <sub>D</sub> 1/θ <sub>JA</sub>	680 10	mW mW/°C
Operating Ambient Temperature Range	T <sub>A</sub>	0 to +70	°C
Storage Temperature Range	T <sub>stg</sub>	-65 to +150	°C

NOTES: 1 The magnitude of the input voltage must not exceed the magnitude of the supply voltage or 15 V, whichever is less.  
2 The output may be shorted to ground or either supply. Temperature and/or supply voltages must be limited to ensure that power dissipation ratings are not exceeded.

**ELECTRICAL CHARACTERISTICS (V<sub>CC</sub> = +15 V, V<sub>EE</sub> = -15 V, T<sub>A</sub> = T<sub>low</sub> to T<sub>high</sub> [Note 3].)**

Characteristics	Symbol	Min	Typ	Max	Unit
Input Offset Voltage (R <sub>S</sub> ≤ 10 k, V <sub>CM</sub> = 0) TL081C, TL082C TL084C TL08_AC	V <sub>IO</sub>	—	—	20 20 7.5	mV
Input Offset Current (V <sub>CM</sub> = 0) (Note 4) TL08_C TL08_AC	I <sub>IO</sub>	—	—	5.0 3.0	nA
Input Bias Current (V <sub>CM</sub> = 0) (Note 4) TL08_C TL08_AC	I <sub>IB</sub>	—	—	10 7.0	nA
Large-Signal Voltage Gain (V <sub>O</sub> = ±10 V, R <sub>L</sub> ≥ 2.0 k) TL08_C TL08_AC	A <sub>vOL</sub>	15 25	—	—	V/mV
Output Voltage Swing (Peak-to-Peak) (R <sub>L</sub> ≥ 10 k) (R <sub>L</sub> ≥ 2.0 k)	V <sub>O</sub>	24 20	—	—	V

NOTES: 3 T<sub>low</sub> = 0°C for TL081AC, TL082AC, TL084AC; T<sub>high</sub> = +70°C for TL081AC, TL082AC, TL084AC.  
4 Input Bias currents of JFET input op amps approximately double for every 10°C rise in Junction Temperature as shown in Figure 3. To maintain junction temperature as close to ambient temperature as possible, pulse techniques must be used during testing.

Figure 1. Unity Gain Voltage Follower

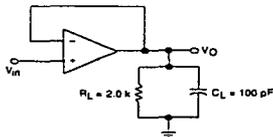
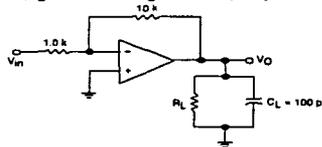


Figure 2. Inverting Gain of 10 Amplifier



## TL081C,AC TL082C,AC TL084C,AC

ELECTRICAL CHARACTERISTICS ( $V_{CC} = +15$  V,  $V_{EE} = -15$  V,  $T_A = +25^\circ\text{C}$ , unless otherwise noted.)

Characteristics	Symbol	Min	Typ	Max	Unit
Input Offset Voltage ( $R_S \leq 10$ k, $V_{CM} = 0$ ) TL081C, TL082C TL084C TL08_4C	$V_{IO}$	—	5.0 5.0 3.0	15 15 8.0	mV
Average Temperature Coefficient of Input Offset Voltage $R_S = 50$ $\Omega$ , $T_A = T_{low}$ to $T_{high}$ (Note 3)	$\Delta V_{IO}/\Delta T$	—	10	—	$\mu\text{V}/^\circ\text{C}$
Input Offset Current ( $V_{CM} = 0$ ) (Note 4) TL08_1C TL08_4C	$I_{IO}$	—	5.0 5.0	200 100	$\mu\text{A}$
Input Bias Current ( $V_{CM} = 0$ ) (Note 4) TL08_1C TL08_4C	$I_B$	—	30 30	400 200	$\mu\text{A}$
Input Resistance	$r_i$	—	$10^{12}$	—	$\Omega$
Common Mode Input Voltage Range TL08_1C TL08_4C	$V_{ICR}$	$\pm 10$ $\pm 11$	—	+15, -12 —	V
Large Signal Voltage Gain ( $V_O = \pm 10$ V, $R_L \geq 2.0$ k) TL08_1C TL08_4C	$A_{VOL}$	25 50	150 150	—	V/mV
Output Voltage Swing (Peak-to-Peak) ( $R_L = 10$ k)	$V_O$	24	28	—	V
Common Mode Rejection Ratio ( $R_S \leq 10$ k) TL08_1C TL08_4C	CMRR	70 80	100 100	—	dB
Supply Voltage Rejection Ratio ( $R_S \leq 10$ k) TL08_1C TL08_4C	PSRR	70 80	100 100	—	dB
Supply Current (Each Amplifier)	$I_D$	—	1.4	2.8	mA
Unity Gain Bandwidth	BW	—	4.0	—	MHz
Slew Rate (See Figure 1) $V_{in} = 10$ V, $R_L = 2.0$ k, $C_L = 100$ pF	SR	—	13	—	V/ $\mu\text{s}$
Rise Time (See Figure 1)	$t_r$	—	0.1	—	$\mu\text{s}$
Overshoot ( $V_{in} = 20$ mV, $R_L = 2.0$ k, $C_L = 100$ pF)	OS	—	10	—	%
Equivalent Input Noise Voltage $R_S = 100$ $\Omega$ , $f = 1000$ Hz	$e_n$	—	25	—	nV/ $\sqrt{\text{Hz}}$
Channel Separation $A_V = 100$	CS	—	120	—	dB

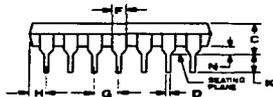
NOTES: 3  $T_{low} = 0^\circ\text{C}$  for TL081AC C, TL082AC C, TL084AC C  
 $T_{high} = +70^\circ\text{C}$  for TL081AC C, TL082AC C, TL084AC C

4 Input bias currents of JFET input op amps approximately double for every  $10^\circ\text{C}$  rise in Junction Temperature as shown in Figure 3. To maintain junction temperature as close to ambient temperature as possible, pulse techniques must be used during testing.

TL081C, AC TL082C, AC TL084C, AC

OUTLINE DIMENSIONS

N SUFFIX  
PLASTIC PACKAGE  
CASE 846-08  
ISSUE L



- NOTES
- LEADS WITHIN 0.13 (0.005) RADIUS OF TURN FROM THE 1ST AND 2ND PLANE OF SYMMETRY
  - MATERIAL CONDITION
  - DEVIATIONS TO CENTER OF LEADS WHEN FORMED PARALLEL
  - OPERATION DOES NOT INCLUDE WELD PLASMA
  - WELD CORNERS OPTIONAL

UNIT	INCHES	MILLIMETERS
A	0.115	2.93
B	0.250	6.35
C	0.250	6.35
D	0.018	0.46
E	0.018	0.46
F	0.100	2.54
G	0.100	2.54
H	0.100	2.54
I	0.100	2.54
J	0.100	2.54
K	0.100	2.54
L	0.100	2.54
M	0.100	2.54
N	0.100	2.54
O	0.100	2.54
P	0.100	2.54
Q	0.100	2.54
R	0.100	2.54
S	0.100	2.54
T	0.100	2.54
U	0.100	2.54
V	0.100	2.54
W	0.100	2.54
X	0.100	2.54
Y	0.100	2.54
Z	0.100	2.54

Motorola reserves the right to make changes without further notice to any products herein. Motorola makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Motorola assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damage. Typical parameters which may be provided in Motorola data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters including "Typicals" must be validated for each customer application by customer's technical experts. Motorola does not convey any license under its patent rights, nor the rights of others. Motorola products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body or other applications intended to support or sustain life, or for any other application in which the failure of the Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such unintended or unauthorized application, Buyer shall indemnify and hold Motorola and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Motorola was negligent regarding the design or manufacture of the part. Motorola and are registered trademarks of Motorola, Inc. Motorola, Inc. is an Equal Opportunity/Affirmative Action Employer.

How to reach us:  
USA/EUROPE/Locations Not Listed: Motorola Literature Distribution,  
P.O. Box 20812, Phoenix, Arizona 85028 1-800-441-2447 or 602-300-5454  
MFAX: RNFAX0@email.scs.mot.com TOUCHSTONE 602-244-6809  
INTERNET: http://Design-NET.com

JAPAN: Nissan Motorola Ltd. Tsuyama-SPD-JDC, 6F Seibu-Bldg, Myocho-Center,  
3-14-2 Tsuyama-Kyo-Ku, Tokyo 130, Japan 03-81-3521-8315

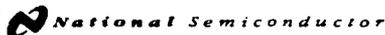
ASIA/PACIFIC: Motorola Semiconductors H.K. Ltd., 8B Ta Ping Industrial Park,  
51 Teng Kok Road, Tai Po, N.T., Hong Kong 852-26629288



0



TL081C/D



February 1985

## LM78XX Series Voltage Regulators

### General Description

The LM78XX series of three terminal regulators is available with several fixed output voltages making them useful in a wide range of applications. One of these is local on card regulation eliminating the distribution problems associated with single point regulation. The voltages available allow these regulators to be used in logic systems, instrumentation, HiFi, and other solid state electronic equipment. Although designed primarily as fixed voltage regulators these devices can be used with external components to obtain adjustable voltages and currents.

The LM78XX series is available in an aluminum TO-3 package which will allow over 1.0A load current if adequate heat sinking is provided. Current limiting is included to limit the peak output current to a safe value. Safe area protection for the output transistor is provided to limit internal power dissipation. If internal power dissipation becomes too high for the heat sinking provided, the thermal shutdown circuit takes over preventing the IC from overheating.

Considerable effort was expended to make the LM78XX series of regulators easy to use and minimize the number

of external components. It is not necessary to bypass the output, although this does improve transient response. Input bypassing is needed only if the regulator is located far from the filter capacitor of the power supply.

For output voltage other than 5V, 12V and 15V the LM117 series provides an output voltage range from 1.2V to 37V.

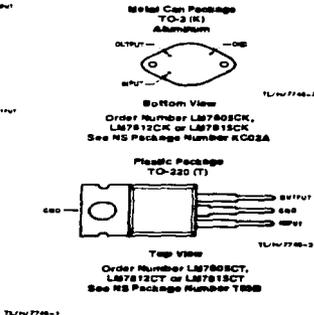
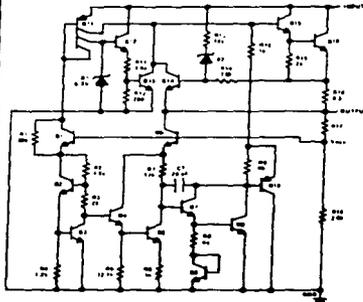
### Features

- Output current in excess of 1A
- Internal thermal overload protection
- No external components required
- Output transistor safe area protection
- Internal short circuit current limit
- Available in the aluminum TO-3 package

### Voltage Range

LM7805C	5V
LM7812C	12V
LM7815C	15V

### Schematic and Connection Diagrams



© 1985 National Semiconductor Corporation

AND 826P115 (REVISED) U.S.A.

LM78XX Series Voltage Regulators

Absolute Maximum Ratings			Maximum Junction Temperature		
If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.			(K Package) 150°C (T Package) 150°C Storage Temperature Range -65°C to +150°C Lead Temperature (Soldering, 10 sec) TO-3 Package K 300°C TO-220 Package T 230°C		
Input Voltages (V <sub>I</sub> ) = 5V, 12V and 15V 3.5V					
Internal Power Dissipation (Note 1) Internally Limited					
Operating Temperature Range (T <sub>A</sub> ) 0°C to +70°C					

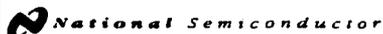
  

Electrical Characteristics LM70XXC (Note 2: 0°C ≤ T <sub>J</sub> ≤ 125°C unless otherwise noted)							
		Output Voltage		12V		15V	
Input Voltage (unless otherwise noted)		12V		15V		15V	
Symbol	Parameter	Conditions		Min	Typ	Max	Units
V <sub>O</sub>	Output Voltage	T <sub>J</sub> = 25°C, 5 mA ≤ I <sub>O</sub> ≤ 1 A		4.9	5	5.2	V
		F <sub>O</sub> = 15W, 5 mA ≤ I <sub>O</sub> ≤ 1 A		4.75	5.25	11.4	V
		V <sub>IN</sub> ≤ V <sub>IN</sub> ≤ V <sub>MAX</sub>		(7.5 ≤ V <sub>IN</sub> ≤ 20)	(14.5 ≤ V <sub>IN</sub> ≤ 27)	(17.5 ≤ V <sub>IN</sub> ≤ 30)	V
ΔV <sub>O</sub>	Line Regulation	I <sub>O</sub> = 500 mA, T <sub>J</sub> = 25°C		3	90	4	150
		ΔV <sub>IN</sub>		(7 ≤ V <sub>IN</sub> ≤ 25)	14.9	(17.5 ≤ V <sub>IN</sub> ≤ 30)	mV
		0°C ≤ T <sub>J</sub> ≤ +125°C		50	120	150	mV
		ΔV <sub>IN</sub>		(8 ≤ V <sub>IN</sub> ≤ 20)	(15 ≤ V <sub>IN</sub> ≤ 27)	(18.5 ≤ V <sub>IN</sub> ≤ 30)	mV
ΔV <sub>O</sub>	Load Regulation	T <sub>J</sub> = 25°C		50	120	150	mV
		ΔV <sub>IN</sub>		(7.5 ≤ V <sub>IN</sub> ≤ 20)	(14.6 ≤ V <sub>IN</sub> ≤ 27)	(17.7 ≤ V <sub>IN</sub> ≤ 30)	mV
		0°C ≤ T <sub>J</sub> ≤ +125°C		25	60	75	mV
		ΔV <sub>IN</sub>		(8 ≤ V <sub>IN</sub> ≤ 12)	(16 ≤ V <sub>IN</sub> ≤ 22)	(20 ≤ V <sub>IN</sub> ≤ 28)	mV
ΔV <sub>O</sub>	Load Regulation	T <sub>J</sub> = 25°C		5 mA ≤ I <sub>O</sub> ≤ 1 A	10	50	120
		250 mA ≤ I <sub>O</sub> ≤ 750 mA		10	25	60	12
I <sub>Q</sub>	Quiescent Current	5 mA ≤ I <sub>O</sub> ≤ 1 A, 0°C ≤ T <sub>J</sub> ≤ +125°C		50	120	150	mV
		I <sub>O</sub> = 1 A, T <sub>J</sub> = 25°C		8	8	8	mA
ΔI <sub>Q</sub>	Quiescent Current Change	0°C ≤ T <sub>J</sub> ≤ +125°C		8.5	8.5	8.5	mA
		5 mA ≤ I <sub>O</sub> ≤ 1 A		0.5	0.5	0.5	mA
V <sub>N</sub>	Output Noise Voltage	T <sub>A</sub> = 25°C, 10 Hz ≤ f ≤ 100 kHz		40	75	90	μV
		I <sub>O</sub> = 1 A, T <sub>J</sub> = 25°C or I <sub>O</sub> = 500 mA, 0°C ≤ T <sub>J</sub> ≤ +125°C		82	80	55	72
ΔV <sub>N</sub>	Ripple Rejection	f = 120 Hz		82	80	55	72
		V <sub>IN</sub> ≤ V <sub>IN</sub> ≤ V <sub>MAX</sub>		(8 ≤ V <sub>IN</sub> ≤ 18)	(15 ≤ V <sub>IN</sub> ≤ 23)	(18.5 ≤ V <sub>IN</sub> ≤ 28.5)	V
R <sub>O</sub>	Dropout Voltage	T <sub>J</sub> = 25°C, I <sub>OUT</sub> = 1 A		2.0	2.0	2.0	V
		f = 1 kHz		8	18	18	mΩ
		T <sub>J</sub> = 25°C		2.1	1.8	1.2	A
		Peak Output Current		2.4	2.4	2.4	A
V <sub>N</sub>	Input Voltage Required to Maintain Line Regulation	0°C ≤ T <sub>J</sub> ≤ +125°C, I <sub>O</sub> = 5 mA		0.6	1.5	1.6	mV/°C
		T <sub>J</sub> = 25°C, I <sub>O</sub> = 1 A		7.5	14.6	17.7	V

Note 1: Thermal resistance at the TO-3 package is 1°C/W; typically 0°C/W junction to case and 30°C/W case to ambient. Thermal resistance of the TO-220 package (T) is typically 0°C/W junction to case and 30°C/W case to ambient.

Note 2: All characteristics are measured with capacitors across the input of 0.22 μF, and a capacitor across the output of 0.1 μF. All characteristics except noise voltage and ripple rejection rate are measured using pulse loads (10 μs, 10 mA, duty cycle = 5%). Output voltage changes due to change in output temperature must be taken into account separately.





November 1984

## LM79XX Series 3-Terminal Negative Regulators

### General Description

The LM79XX series of 3-terminal regulators is available with fixed output voltages of  $-5V$ ,  $-8V$ ,  $-12V$ , and  $-15V$ . These devices need only one external component—a compensation capacitor at the output. The LM79XX series is packaged in the TO-220 power package and is capable of supplying 1.5A of output current.

These regulators employ internal current limiting, safe area protection and thermal shutdown for protection against virtually all overload conditions.

Low ground pin current of the LM79XX series allows output voltage to be easily boosted above the preset value with a resistor divider. The low quiescent current drain of

these devices with a specified maximum charge with line and load ensures good regulation in the voltage boosted mode.

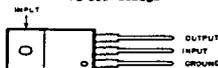
For applications requiring other voltages, see LM137 data sheet.

### Features

- Thermal, short circuit and safe area protection
- High noise rejection
- 1.5A output current
- 4% tolerance on preset output voltage

### Connection Diagrams

TO-220 Package



Front View

LM79750-118

Order Number LM7905CT, LM7912CT or LM7915CT  
See NS Package Number TO3B

### Typical Applications



LM79750-3

\*Required if regulator is separated from filter capacitor by more than 3". For value given, capacitor must be solid tantalum. 25  $\mu F$  aluminum electrolytic may be substituted.  
†Required for stability. For value given, capacitor must be solid tantalum. 25  $\mu F$  aluminum electrolytic may be substituted. Values given may be increased without limit.  
For output capacitance in excess of 100  $\mu F$ , a high current diode from input to output (1N4001, etc.) will protect the regulator from momentary input shorts.

LM79XX Series 3-Terminal Negative Regulators

Part Number		LM7805C			Units	
Output Voltage		-5V				
Input Voltage (unless otherwise specified)		-10V				
Symbol	Parameter	Conditions	Min	Typ	Max	Units
$V_O$	Output Voltage	$T_J = 25^\circ\text{C}$ $5\text{ mA} \leq I_{OUT} \leq 1\text{ A}$ $P \leq 1\text{ W}$	-4.8 -4.75	-5.0 (-20 $\leq V_{IN} \leq -7$ )	-5.2 -5.25	V V
$\Delta V_O$	Line Regulation	$T_J = 25^\circ\text{C}$ , (Note 3)		8 2 (-25 $\leq V_{IN} \leq -7$ ) (-12 $\leq V_{IN} \leq -8$ )	50 15 V	mV mV V
$\Delta V_O$	Load Regulation	$T_J = 25^\circ\text{C}$ , (Note 3) $5\text{ mA} \leq I_{OUT} \leq 1.5\text{ A}$ $250\text{ mA} \leq I_{OUT} \leq 750\text{ mA}$		15 5	100 50	mV mV
$I_Q$	Quiescent Current	$T_J = 25^\circ\text{C}$		1	2	mA
$\Delta I_Q$	Quiescent Current Change	With Load		(-25 $\leq V_{IN} \leq -7$ )	0.5	mA
$V_n$	Output Noise Voltage	$T_A = 25^\circ\text{C}$ , 10 Hz - 1 $\leq$ 100 Hz		125		$\mu\text{V}$
	Ripple Rejection	$f = 120\text{ Hz}$	54	66 (-18 $\leq V_{IN} \leq -8$ )		dB
	Dropout Voltage	$T_J = 25^\circ\text{C}$ , $I_{OUT} = 1\text{ A}$		1.1		V
$I_{OVS}$	Peak Output Current	$T_J = 25^\circ\text{C}$		2.2		A
	Average Temperature Coefficient of Output Voltage	$I_{OUT} = 5\text{ mA}$ $0^\circ\text{C} \leq T_J \leq 100^\circ\text{C}$		0.4		mV/°C

**Typical Applications** (Continued)

Variable Output

$V_{OUT} = V_{REF} \left( \frac{R_1 + R_2}{R_2} \right)$   
 Select  $R_2$  as follows:  
 LM7805C — 300 $\Omega$   
 LM7812C — 750 $\Omega$   
 LM7815C — 1k

\*Improves transient response and ripple rejection. Do not increase beyond 50  $\mu\text{F}$ .  
 TLN7340-F

Electrical Characteristics (Continued) Conditions unless otherwise noted: $I_{OUT} = 500\text{ mA}$ , $C_{IN} = 2.2\text{ }\mu\text{F}$ , $C_{OUT} = 1\text{ }\mu\text{F}$ , $0^\circ\text{C} \leq T_J \leq 125^\circ\text{C}$ , Power Dissipation = 1.5W							
Part Number		LM7918C			LM7918C		
Output Voltage		-18V			-18V		
Input Voltage (unless otherwise specified)		-18V			-23V		
Symbol	Parameter	Conditions	Min	Typ	Max	Units	
$V_O$	Output Voltage	$T_J = 25^\circ\text{C}$ $5\text{ mA} \leq I_{OUT} \leq 1\text{ A}$ $P \leq 15\text{ W}$	-11.5 -11.4	-12.0 -12.0	-12.5 -12.5	-12.4 -12.5 -15.75 -17.5	V
$\Delta V_O$	Line Regulation	$T_J = 25^\circ\text{C}$ (Note 3)	5 3	80 30	100 50	100 50	mV/V
$\Delta V_O$	Load Regulation	$T_J = 25^\circ\text{C}$ (Note 3) $5\text{ mA} \leq I_{OUT} \leq 1\text{ A}$ $250\text{ mA} \leq I_{OUT} \leq 750\text{ mA}$	15 5	200 75	200 75	200 75	mV
$I_Q$	Quiescent Current	$T_J = 25^\circ\text{C}$	1.5	3	3	3	mA
$\Delta I_Q$	Quiescent Current Change	With Line With Load $5\text{ mA} \leq I_{OUT} \leq 1\text{ A}$	0.5 0.5	0.5 0.5	0.5 0.5	0.5 0.5	mA
$V_n$	Output Noise Voltage	$T_A = 25^\circ\text{C}$ , $10\text{ Hz} \leq f \leq 100\text{ Hz}$	54	70	70	70	$\mu\text{V}$
	Reple Rejection	$f = 120\text{ Hz}$	54	70	70	70	dB
	Dropout Voltage	$T_J = 25^\circ\text{C}$ , $I_{OUT} = 1\text{ A}$	1.1	1.1	1.1	1.1	V
$I_{O(MAX)}$	Peak Output Current	$T_J = 25^\circ\text{C}$	2.2	2.2	2.2	2.2	A
	Average Temperature Coefficient of Output Voltage	$I_{OUT} = 5\text{ mA}$ $0^\circ\text{C} \leq T_J \leq 100^\circ\text{C}$	-0.8		-1.0	-1.0	mV/°C

Note 1: Absolute Maximum Ratings exceed limits beyond which damage to the device may occur. Operating Ratings establish conditions for which the device is intended to be functional, but do not guarantee specific performance levels. For guaranteed specifications and test conditions, see the Electrical Characteristics.

Note 2: Refer to Thermal Performance Characteristics and Design Considerations for Notes.

Note 3: Regulator  $\eta$  is measured at a junction temperature by pulse testing with a low duty cycle. Changes in output voltage due to heating effects may be taken into account.

### Typical Applications (Continued)

Dual Trained Supply

1L/M7905-4

### Design Considerations

The LM79XX fixed voltage regulator series has thermal overload protection from excessive power dissipation internal short circuit protection which limits the circuit's maximum current and output transistor safe area compensation for reducing the output current as the voltage across the pass transistor is increased.

Although the internal power dissipation is limited, the junction temperature must be kept below the maximum specified temperature (125°C) in order to meet data sheet specifications. To calculate the maximum junction temperature or heat sink required, the following thermal resistance values should be used:

Package	Typ #JC °C/W	Max #JC °C/W	Typ #JA °C/W	Max #JA °C/W
TO-220	3.0	5.0	50	40

$$P_{D \text{ MAX}} = \frac{T_{J \text{ MAX}} - T_A}{R_{JC}} \text{ OR } \frac{T_{J \text{ MAX}} - T_A}{R_{JA}}$$

$$R_{CA} = R_{CS} + R_{SA} \text{ (without heat sink)}$$

Solving for  $T_J$

$$T_J = T_A + P_D (R_{JC} + R_{CA}) \text{ OR}$$

$$T_J = T_A + P_D R_{JA} \text{ (without heat sink)}$$

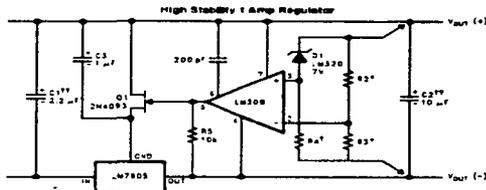
Where

- $T_J$  = Junction Temperature
- $T_A$  = Ambient Temperature
- $P_D$  = Power Dissipation
- $R_{JA}$  = Junction-to-Ambient Thermal Resistance
- $R_{JC}$  = Junction-to-Case Thermal Resistance
- $R_{CA}$  = Case-to-Ambient Thermal Resistance
- $R_{CS}$  = Case-to-Heat Sink Thermal Resistance
- $R_{SA}$  = Heat Sink-to-Ambient Thermal Resistance

### Typical Applications (Continued)

Bypass capacitors are necessary for stable operation of the LM79XX series of regulators over the input voltage and output current ranges. Output bypass capacitors will improve the transient response of the regulator.

The bypass capacitors (22  $\mu\text{F}$  on the input, 10  $\mu\text{F}$  on the output) should be ceramic or solid tantalum which have good high frequency characteristics. If aluminum electrolytics are used, their values should be 10  $\mu\text{F}$  or larger. The bypass capacitors should be mounted with the shortest leads and if possible, directly across the regulator terminals.



Load and line regulation:  $\pm 0.2\%$  temperature stability:  $\pm 0.2\%$

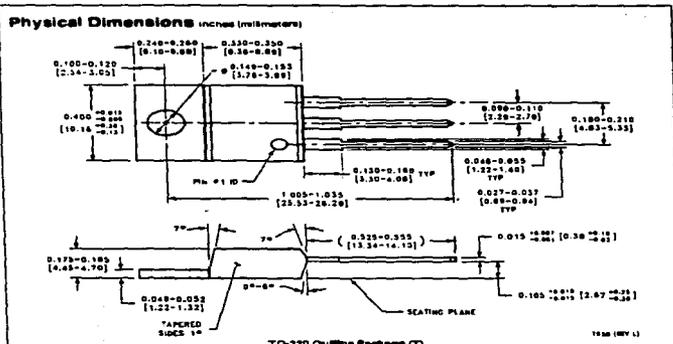
<sup>1</sup>Maximum input current

<sup>2</sup>50mA maximum

<sup>3</sup>Mount regulators to hot output voltage. 2 ppm/°C loading suggested

TUM7905-6

LM79XX Series 3-Terminal Negative Regulators



TO-220 Outline Package (T)  
 Order Number LM7901CT, LM7912CT or LM7918CT  
 NS Package Number T03B

**LIFE SUPPORT POLICY**

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

<p><b>Dallas Semiconductor Corp</b>                  Dallas, TX 75243                  Tel: (409) 777-0100                  Fax: (409) 777-7018</p>	<p><b>International Semiconductor</b>                  Europe                  P.O. Box 118, G.B. 20090 Segrate (MI) Italy                  England: Tel: (Int'l) 0362-810000                  France: Tel: (Int'l) 01-69-95-91-92                  Germany: Tel: (Int'l) 089-30-90-91-92                  Italy: Tel: (Int'l) 02-8621-10-11                  Japan: Tel: (Int'l) 03-5682-10-11</p>	<p><b>Philips Semiconductors</b>                  World Headquarters                  P.O. Box 9106, Eindhoven, The Netherlands                  Tel: (Int'l) 040-739-7261                  Telex: 9156                  Fax: (Int'l) 040-739-7262</p>	<p><b>Siemens Semiconductor Division</b>                  Munich, Germany                  P.O. Box 81, D-8520 Erding, Germany                  Tel: (Int'l) 089-947-6600                  Fax: (Int'l) 089-947-6601</p>
---	---	--	--

© 1987 National Semiconductor Corporation. All rights reserved. National Semiconductor Corporation is not responsible for errors or for consequences arising from the use of the information contained herein.

## E Series Multifunction I/O Boards for ISA

### Specifications

Typical for 25°C unless otherwise noted

#### Analog Input

##### Input Characteristics

Number of channels ..... 16 single-ended or 8 differential, software selectable

Type of ADC ..... Successive approximation

Resolution ..... 12 bits, 1 in 4,096

Maximum sampling rate ..... 100 kS/s

##### Input signal ranges

Gain (Half-Scale Equivalent Voltage)	Range (Full-Scale Equivalent Voltage)	Resolution (LSB)
0.5	±10 V	—
1	±5 V	0 to 10 V
2	±2.5 V	0 to 5 V
5	±1 V	0 to 2 V
10	±500 mV	0 to 1 V
20	±250 mV	0 to 500 mV
50	±100 mV	0 to 200 mV
100	±50 mV	0 to 100 mV

Input coupling ..... DC

##### Max working voltage

(Signal + common mode) ..... Each input should remain within ±11 V of ground

Overvoltage protection ..... ±35 V powered on, ±25 V powered off

Inputs protected ..... ACH40, 15>, AISENSE

FIFO buffer size ..... 512 samples

Data transfers ..... DMA, interrupts, programmed I/O

DMA modes ..... Single transfer, demand transfer

Configuration memory size ..... 512 words

##### Transfer Characteristics

Relative accuracy ..... ±0.2 LSB typical dithered

DNL ..... ±1.0 LSB maximum undithered

DNL ..... ±0.2 LSB typical, ±0.5 LSB maximum

No missing codes ..... 12 bits, guaranteed

##### Offset error

Pregain error after calibration ..... ±2 µV maximum

Pregain error before calibration ..... ±24 mV maximum

Postgain error after calibration ..... ±0.5 mV maximum

Postgain error before calibration ..... ±100 mV maximum

Gain error (relative to calibration reference)

After calibration (Gain = 1) ..... ±0.01% of reading maximum

Before calibration ..... ±0.20% of reading maximum

Gain = 1 with gain error

adjusted to 0 at gain = 1 ..... ±0.05% of reading maximum

##### Amplifier Characteristics

###### Input impedance

Normal powered on ..... 100 GΩ in parallel with 50 pF

Powered off ..... 3 kΩ minimum

Overload ..... 3 kΩ minimum

Input bias current ..... ±200 pA

Input offset current ..... ±100 pA

CMRR (all input ranges) ..... 90 dB, DC to 60 Hz

##### Dynamic Characteristics

###### Bandwidth

Small signal (-3 dB) ..... 200 kHz

Large signal (1% THD) ..... 300 kHz

Settling time for full-scale step ..... 10 µs maximum to ±0.5 LSB accuracy

System noise (not including quantization)

Bandwidth	Offset
0.5 to 10	0.07 LSBrms
20	0.12 LSBrms
50	0.23 LSBrms
100	0.5 LSBrms
±1000 or 400 gain	0.5 LSBrms

Crosstalk ..... -80 dB, DC to 100 kHz

##### Stability

Recommended warm-up time ..... 15 minutes

Offset temperature coefficient

Pregain ..... ±15 µV/°C

Postgain ..... ±240 µV/°C

Gain temperature coefficient ..... ±20 ppm/°C

##### Onboard calibration reference

Level ..... 5,000 V (±2.5 mV)

Temperature coefficient ..... (actual value stored in EEPROM)

Temperature coefficient ..... ±5 ppm/°C maximum

Long-term stability ..... ±15 ppm/V-1,000 h

##### Analog Output

###### Output Characteristics

Number of channels ..... 2 voltage

Resolution ..... 12 bits, 1 in 4,096

Maximum update rate ..... 100 kS/s

Type of DAC ..... Double buffered, multiplying

FIFO buffer size ..... None

Data transfers ..... DMA, interrupts, programmed I/O

DMA modes ..... Single transfer

###### Transfer Characteristics

Relative accuracy (INL)

After calibration ..... ±0.3 LSB typical, ±0.5 LSB maximum

Before calibration ..... ±4 LSB maximum

###### DNL

After calibration ..... ±0.3 LSB typical, ±1.0 LSB maximum

Before calibration ..... ±3 LSB maximum

Monotonicity ..... 12 bits, guaranteed after calibration

##### Offset error

After calibration ..... ±1.0 mV maximum

Before calibration ..... ±200 mV maximum

Gain error (relative to internal reference)

After calibration ..... ±0.01% of output maximum

Before calibration ..... ±0.5% of output maximum

Gain error

(relative to external reference) ..... 0% to -0.5% of output maximum, not adjustable

Continued on page 3-80

RANS-102-10, RANS-102-10

## E Series Multifunction I/O Boards for ISA

AT-MIO-16E-10, AT-MIO-16DE-10

### McPnpge Output

Range ..... ±10 V, 0 to 10 V, ±EXTREF 0 to  
 EXTREF (software selectable)  
 Output coupling ..... DC  
 Output impedance ..... 0.1 Ω maximum  
 Current drive ..... ±5 mA maximum  
 Protection ..... Short-circuit to ground  
 Power-on state ..... 0 V

External reference input  
 Range ..... ±11 V  
 Overvoltage protection ..... ±35 V powered on, ±25 V powered off  
 Input impedance ..... 10 kΩ  
 Bandwidth (-3 dB) ..... 300 kHz

### Dynamic Characteristics

Setting time for full-scale step ..... 10 μs to ±0.5 LSB accuracy  
 Slew rate ..... 15 V/μs  
 Noise ..... 200 μVrms, DC to 1 MHz  
 Clutch energy (at midscale transition)  
 Magnitude ..... ±100 mV  
 Duration ..... 3 μs  
 Stability

Offset temperature coefficient ..... ±50 μV/°C

Gain temperature coefficient

Internal reference ..... ±25 ppm/°C

External reference ..... 225 ppm/°C

Onboard calibration reference

Level ..... 5,000 V (±2.5 mV)

(actual value stored in EEPROM)

Temperature coefficient ..... ±5 ppm/°C maximum

Long term stability ..... ±15 ppm/V, 1,000 h

### Digital I/O

Number of channels

AT-MIO-16E-10 ..... 8 input/output

AT-MIO-16DE-10 ..... 32 input/output

Compatibility ..... TTL/CMOS

Digital logic levels

DIC00 2.5

Level	Minimum	Maximum
Input low voltage (I <sub>OL</sub> = 2.5 mA)	0 V	0.6 V
Input high voltage (I <sub>IH</sub> = 2.5 mA)	2 V	5 V
Input low current (V <sub>I</sub> = 0 V)	—	320 μA
Input high current (V <sub>I</sub> = 5 V)	—	10 μA
Output low voltage (I <sub>OL</sub> = 2.5 mA)	—	0.4 V
Output high voltage (I <sub>OH</sub> = 2.5 mA)	4.25 V	—

PA=0 7= PR=0 7= PC=0 7= (Remaining 24 lines of AT-MIO-16DE-10)

Level	Minimum	Maximum
Input low voltage (I <sub>OL</sub> = 2.5 mA)	0 V	0.6 V
Input high voltage (I <sub>IH</sub> = 2.5 mA)	2 V	5 V
Input low current (V <sub>I</sub> = 0 V)	—	40 μA
Input high current (V <sub>I</sub> = 5 V)	—	10 μA
Output low voltage (I <sub>OL</sub> = 2.5 mA)	—	0.4 V
Output high voltage (I <sub>OH</sub> = 2.5 mA)	3.0 V	—

### Handshaking

(AT-MIO-16DE-10 only) ..... 2-wire

Power-on state ..... Input (High-Z)

### Data transfers

AT-MIO-16E-10 ..... Programmed I/O

AT-MIO-16DE-10 ..... Interrupts, programmed I/O

### Timing I/O

Number of channels ..... 2 up/down counter/timers,

1 frequency scaler

### Resolution

Counter/timers ..... 24 bits

Frequency scaler ..... 4 bits

Compatibility ..... TTL/CMOS

### Base clocks available

Counter/timers ..... 20 MHz, 100 kHz

Frequency scaler ..... 10 MHz, 100 kHz

Base clock accuracy ..... ±0.01%

Maximum source frequency ..... 20 MHz

Minimum source pulse duration ..... 10 ns in edge-detect mode

Minimum gate pulse duration ..... 10 ns in edge-detect mode

Data transfers ..... DMA, interrupts, programmed I/O

DMA modes ..... Single transfer

### Triggers

#### Digital Trigger

Compatibility ..... TTL

Response ..... Rising or falling edge

Pulse width ..... 10 ns minimum

#### RTM

Trigger lines ..... 7

Bus interface ..... Slave

### Power Requirements

+5 VDC (±5%) ..... 0.7 A

Power available at I/O connector ..... +4.65 VDC to +5.25 VDC at 1 A

### Physical

#### Dimensions

not including connectors) ..... 33.8 by 9.9 cm (13.3 by 3.9 in.)

#### I/O connector

AT-MIO-16E-10 ..... 68-pin male SCSI-II type

AT-MIO-16DE-10 ..... 100-pin female 0.050 D-type

#### Environmental

Operating temperature ..... 0° to 55° C

Storage temperature ..... -55° to 150° C

Relative humidity ..... 5% to 90% noncondensing

## Apéndice 3

### Términos EEG

**Actividad:** Toda onda o secuencia de ondas.

**Actividad de fondo:** Actividad más o menos generalizada y continua, en contraste con las actividades paroxísticas o localizadas (el término no es sinónimo de ritmo alfa).

**Artefacto:** Toda diferencia de potencial de origen extracerebral registrada en el EEG.

***petit mal:*** Forma menos severa de epilepsia generalizada (afecta todo el cerebro), durante el cual fuertes ondas delta son producidas de uno a veinte segundos.

**Difuso:** Que aparece en extensas regiones, de uno solo y de los dos lados.

**Esporádico:** Que aparece con intervalos variables de tiempo.

**Fase:** Utilizado comúnmente en EEG para designar las relaciones temporales de diferentes porciones de una o de varias ondas:

- monofásico: deflexión de un lado en relación a la línea de base;
- Difásico: deflexión primeramente a un lado y luego al otro en relación a la línea de base;

- Polifásico: varias deflexiones sucesivas de sentido opuesto en relación a la línea de base.

**Foco:** Región limitada del cuero cabelludo a nivel de la cual se recoge de forma única o predominante una actividad específica.

**Husos de sueño (*spindles*):** salvas de doce a catorce ciclos por segundo, generalmente difusas, pero más amplias en las regiones centrales, que aparecen durante el sueño.

**Onda.** Todo cambio transitorio de diferencia de potencial en el EEG.

**Onda lenta:** Onda de duración superior a un octavo de segundo.

**Salva:** Grupo de ondas de aparición y desaparición brusca, distinguiéndose netamente de la actividad de fondo por su frecuencia, su morfología o su amplitud (este término no implica anomalía, no es sinónimo de paroxismo)

**Paroxismo:** Fenómeno de aparición brutal, alcanzando rápidamente su máximo y acabándose de forma súbita, diferente de la actividad de fondo, refiriéndose generalmente a las anomalías epilépticas.

## Apéndice 4

### Listado de Programas

A continuación listamos el código fuente en *Delphi* del sistema presentado en este trabajo. No se incluye la totalidad del mismo sino sólo las partes más relevantes.

```

{***Unidad Principal}
unit Main;

{$X+}

interface

uses WinTypes, WinProcs, SysUtils, Classes, Graphics, Forms, Controls, Menus,
  StdCtrls, Dialogs, Buttons, Messages, ExtCtrls, IniFiles,
  IntDAQ, { Declaraciones de interfaz con la tarjeta DAQ }
  Mapmain, { Mapeo EEG }
  Graf1, { Graficación EEG tiempo real }
  CActivos, { Indicación de canales activos modo esquemático }
  CActivNo, { Indicación de canales activos modo normal }
  Montaje, { Visualización del montaje }
  ElecDisp, { Configuración de Electroodos Disponibles }
  AuxCan, { Para rastrear el comportamiento del registro de canales }
  Repro, { Graficación EEG de una señal almacenada }
  Spectrum, { Cálculo y graficación del espectro de potencia }
  NewFile { Estudio Nuevo };

const
  MaxChannels = 16; { Número máximo de canales a visualizar }
  MaxDisponibles = 16; { Número máximo de electrodos disponibles }
  WinPath = 'C:\WINDOWS'; { Directorio donde se ejecuta Windows }

```

```
TheIniFile = 'EEGDSP.INI'; { Nombre del archivo de configuracion inicial }  
NumElectrodos = 19;
```

```
type
```

```
RegCanal = Record  
    NomReal : string[3];  
    NomInt : string[2];  
    Activo : Boolean;  
    Disponible : Boolean;  
end;
```

```
TMainForm = class(TForm)
```

```
    MainMenu: TMainMenu;  
    BottomPanel: TPanel;  
    StatusLine: TPanel;  
    Archivo: TMenuItem;  
    AbrirItem: TMenuItem;  
    RightBottomPanel: TPanel;  
    Ventana: TMenuItem;  
    Ayuda: TMenuItem;  
    N1: TMenuItem;  
    SalirItem: TMenuItem;  
    Cascadaltem: TMenuItem;  
    Mosaicoltem: TMenuItem;  
    Arreglaltem: TMenuItem;  
    Acercadaltem: TMenuItem;  
    OpenDialog: TOpenDialog;  
    GuardarItem: TMenuItem;  
    GuardarComoltem: TMenuItem;  
    Minimizaltem: TMenuItem;  
    SpeedPanel: TPanel;  
    AbrirBtn: TSpeedButton;  
    GuardarBtn: TSpeedButton;  
    SalirBtn: TSpeedButton;  
    EEG: TMenuItem;  
    CanActivos: TMenuItem;  
    VisMontaje: TMenuItem;  
    VisEEG: TMenuItem;  
    Tiempoaltem: TMenuItem;  
    Salmacenada: TMenuItem;  
    Procesos: TMenuItem;  
    AEspectral: TMenuItem;  
    MapeoEEG: TMenuItem;  
    ImprimirItem: TMenuItem;  
    N2: TMenuItem;  
    ImprimirBtn: TSpeedButton;  
    Herramientas: TMenuItem;  
    ConfiguracionItem: TMenuItem;  
    SpeedButton1: TSpeedButton;  
    Nuevoltem: TMenuItem;  
    CerrarItem: TMenuItem;  
    procedure FormCreate(Sender: TObject);  
    procedure CascadaltemClick(Sender: TObject);  
    procedure UpdateMenuItems(Sender: TObject);  
    procedure MosaicoltemClick(Sender: TObject);  
    procedure ArreglaltemClick(Sender: TObject);
```

```

procedure AbrirItemClick(Sender: TObject);
procedure SalirItemClick(Sender: TObject);
procedure GuardarItemClick(Sender: TObject);
procedure GuardarComoItemClick(Sender: TObject);
procedure MinimizarItemClick(Sender: TObject);
procedure FormDestroy(Sender: TObject);
procedure MapeoEEGClick(Sender: TObject);
procedure TiempoRealItemClick(Sender: TObject);
procedure CanActivosClick(Sender: TObject);
procedure VisMontajeClick(Sender: TObject);
procedure ConfiguracionItemClick(Sender: TObject);
procedure AESpectralClick(Sender: TObject);
procedure SalmacenadaClick(Sender: TObject);
procedure NuevoItemClick(Sender: TObject);
procedure CerrarItemClick(Sender: TObject);
private
  { Private declarations }
  procedure CreateMDIChild(const Name: string);
  procedure ShowHint(Sender: TObject);
public
  { Public declarations }
end;

var
  MainForm      : TMainForm;
  GoMapeoEEG    : TMapMainForm;
  GoGraficaTReal : TGrafica;
  GoCanalesActivos : TFCanActivosE;
  GoCanActivosNormal : TFCanActivosN;
  GoVisMontaje  : TFormaMontaje;
  GoElectrodosDisp : TFElectrodosDisp;
  MaskActivos   : Array[1..19] of RegCanal;
  NumdeActivos  : Byte;
  MontajeOK     : Boolean;
  GoAuxCanales  : TAuxCanales;
  StatusDAQ     : Integer;
  GoGraficaAlm  : TFormRepro;
  GoEspectro    : TPotencia;
  NomArchEEG    : string[8];
  GoNewFile     : TFNewFile;
  NuevoOK       : Boolean;

implementation

{$R *.DFM}

uses ChildWin,IntDAQ;

var
  K      : Byte;
  Buffpath : Word;
  ArchIni : Text;
  ArchInicio : TIniFile;

procedure TMainForm.FormCreate(Sender: TObject);
begin

```

```

Application.OnHint := ShowHint;
Screen.OnActiveFormChange := UpdateMenuItems;
end;

procedure TMainForm.ShowHint(Sender: TObject);
begin
  StatusLine.Caption := Application.Hint;
end;

procedure TMainForm.CreateMDIChild(const Name: string);
var
  Child: TMDIChild;
begin
  { create a new MDI child window }
  Child := TMDIChild.Create(Application);
  Child.Caption := Name;
end;

procedure TMainForm.AbrirItemClick(Sender: TObject);
var
  n          : integer;
  archivo, ValidoChr : string;
  PunteroHdr   : TextFile;
  activoschr   : string[2];
begin
  if OpenFileDialog.Execute then
  Begin
    sAlmacenada.Enabled := True;
    Procesos.Visible := True;
    EEG.Visible := True;
    CerrarItem.Enabled := True;

    { Actualiza la matriz con los canales activos }
    Archivo := OpenFileDialog.FileName;
    Archivo := ChangeFileExt(Archivo, '.HDR');
    AssignFile(PunteroHdr, Archivo);
    Reset(PunteroHdr);
    Readln(PunteroHdr, activoschr);
    numdeactivos:=strtoint(activoschr);
    for n := 1 to NumElectrodos do
      begin
        Readln(PunteroHdr);
        Readln(PunteroHdr, ValidoChr);
        if validochr='TRUE' then
          maskactivos[n].Activo:=true
        else
          maskactivos[n].Activo:=false;
      end;
    End
  end;
end;

procedure TMainForm.GuardarItemClick(Sender: TObject);
begin
  { save current file (ActiveMDIChild points to the window) }
end;

```

```
procedure TMainForm.GuardarComoItemClick(Sender: TObject);
begin
  { save current file under new name }
end;

procedure TMainForm.SalirItemClick(Sender: TObject);
begin
  Close;
end;

procedure TMainForm.CascadaItemClick(Sender: TObject);
begin
  Cascade;
end;

procedure TMainForm.MosaicoItemClick(Sender: TObject);
begin
  Tile;
end;

procedure TMainForm.ArreglarItemClick(Sender: TObject);
begin
  ArrangeIcons;
end;

procedure TMainForm.MinimizarItemClick(Sender: TObject);
var
  I: Integer;
begin
  { Must be done backwards through the MDIChildren array }
  for I := MDIChildCount - 1 downto 0 do
    MDIChildren[I].WindowState := wsMinimized;
end;

procedure TMainForm.UpdateMenuItems(Sender: TObject);
begin
  GuardarItem.Enabled := MDIChildCount > 0;
  GuardarComoItem.Enabled := MDIChildCount > 0;
  ImprimirItem.Enabled := MDIChildCount > 0;
  GuardarBtn.Enabled := MDIChildCount > 0;
  ImprimirBtn.Enabled := MDIChildCount > 0;
  CascadaItem.Enabled := MDIChildCount > 0;
  MosaicoItem.Enabled := MDIChildCount > 0;
  ArreglarItem.Enabled := MDIChildCount > 0;
  MinimizarItem.Enabled := MDIChildCount > 0;
end;

procedure TMainForm.FormDestroy(Sender: TObject);
begin
  Screen.OnActiveFormChange := nil;
end;

procedure TMainForm.MapeoEEGClick(Sender: TObject);
begin
  GoMapeoEEG := TMapMainForm.Create(Application);
end;
```

```

procedure TMainForm.TiempoalltemClick(Sender: TObject);
begin
  GoGraficaTReal:=TGrafica.Create(Application);
end;

procedure TMainForm.CanActivosClick(Sender: TObject);
begin
  GoCanalesActivos:=TFCanActivosE.Create(Application);
  GoCanalesActivos.Top:=30;
end;

procedure TMainForm.VisMontajeClick(Sender: TObject);
begin
  GoVisMontaje:=TFomaMontaje.Create(Application);
end;

procedure TMainForm.ConfiguracionItemClick(Sender: TObject);
begin
  GoElectrodosDisp:=TFElectrodosDisp.Create(Application);
  GoElectrodosDisp.Top:=35;
end;

procedure TMainForm.AEspectralClick(Sender: TObject);
begin
  GoEspectro:=TPotencia.Create(Application);
end;

procedure TMainForm.SalmacenadaClick(Sender: TObject);
begin
  GoGraficaAlm:=TFormRepro.Create(Application);
end;

procedure TMainForm.NuevoltemClick(Sender: TObject);
begin
  NuevoOK:=False;
  MainForm.CerrarItemClick(Application);
  GoNewFile:=TFNewFile.Create(Application);
end;

procedure TMainForm.CerrarItemClick(Sender: TObject);
begin
  EEG.Visible:=False;
  Procesos.Visible:=False;
  CerrarItem.Enabled:=False;
end;

initialization
begin
  for k:=1 to 19 Do
    MaskActivos[k].Activo:=False;
  NumDeActivos:=0;
  MaskActivos[1].NomReal := 'FP1';
  MaskActivos[1].NomInt := 'F1';
  MaskActivos[2].NomReal := 'FP2';
  MaskActivos[2].NomInt := 'F2';

```

```

MaskActivos[3].NomReal := 'F7';
MaskActivos[3].NomInt := 'F7';
MaskActivos[4].NomReal := 'F3';
MaskActivos[4].NomInt := 'F3';
MaskActivos[5].NomReal := 'F2';
MaskActivos[5].NomInt := 'F2';
MaskActivos[6].NomReal := 'F4';
MaskActivos[6].NomInt := 'F4';
MaskActivos[7].NomReal := 'F8';
MaskActivos[7].NomInt := 'F8';
MaskActivos[8].NomReal := 'T3';
MaskActivos[8].NomInt := 'T3';
MaskActivos[9].NomReal := 'C3';
MaskActivos[9].NomInt := 'C3';
MaskActivos[10].NomReal := 'C2';
MaskActivos[10].NomInt := 'C2';
MaskActivos[11].NomReal := 'C4';
MaskActivos[11].NomInt := 'C4';
MaskActivos[12].NomReal := 'T4';
MaskActivos[12].NomInt := 'T4';
MaskActivos[13].NomReal := 'T5';
MaskActivos[13].NomInt := 'T5';
MaskActivos[14].NomReal := 'P3';
MaskActivos[14].NomInt := 'P3';
MaskActivos[15].NomReal := 'P2';
MaskActivos[15].NomInt := 'P2';
MaskActivos[16].NomReal := 'P4';
MaskActivos[16].NomInt := 'P4';
MaskActivos[17].NomReal := 'T6';
MaskActivos[17].NomInt := 'T6';
MaskActivos[18].NomReal := 'O1';
MaskActivos[18].NomInt := 'O1';
MaskActivos[19].NomReal := 'O2';
MaskActivos[19].NomInt := 'O2';

```

```

if FileExists(Winpath+TheIniFile) Then
begin
  AssignFile(ArchIni,Winpath+TheIniFile);
  Reset(ArchIni);
  ArchInicio:=TIniFile.Create(TheIniFile);
  MaskActivos[1].Disponible:=ArchInicio.ReadBool('Disponibles','F1',False);
  MaskActivos[2].Disponible:=ArchInicio.ReadBool('Disponibles','F2',False);
  MaskActivos[3].Disponible:=ArchInicio.ReadBool('Disponibles','F7',False);
  MaskActivos[4].Disponible:=ArchInicio.ReadBool('Disponibles','F3',False);
  MaskActivos[5].Disponible:=ArchInicio.ReadBool('Disponibles','F2',False);
  MaskActivos[6].Disponible:=ArchInicio.ReadBool('Disponibles','F4',False);
  MaskActivos[7].Disponible:=ArchInicio.ReadBool('Disponibles','F8',False);
  MaskActivos[8].Disponible:=ArchInicio.ReadBool('Disponibles','T3',False);
  MaskActivos[9].Disponible:=ArchInicio.ReadBool('Disponibles','C3',False);
  MaskActivos[10].Disponible:=ArchInicio.ReadBool('Disponibles','C2',False);
  MaskActivos[11].Disponible:=ArchInicio.ReadBool('Disponibles','C4',False);
  MaskActivos[12].Disponible:=ArchInicio.ReadBool('Disponibles','T4',False);
  MaskActivos[13].Disponible:=ArchInicio.ReadBool('Disponibles','T5',False);
  MaskActivos[14].Disponible:=ArchInicio.ReadBool('Disponibles','P3',False);
  MaskActivos[15].Disponible:=ArchInicio.ReadBool('Disponibles','P2',False);
  MaskActivos[16].Disponible:=ArchInicio.ReadBool('Disponibles','P4',False);

```

```

MaskActivos[17].Disponible:=ArchInicio.ReadBool('Disponibles','T6',False);
MaskActivos[18].Disponible:=ArchInicio.ReadBool('Disponibles','O1',False);
MaskActivos[19].Disponible:=ArchInicio.ReadBool('Disponibles','O2',False);
ArchInicio.Free;
system.close(ArchIni);
end
else
begin
  For k:=1 to 19 Do
    MaskActivos[k].Disponible:=False;
  end;
MontajeOK:=False;
end;
end;
end. {Unit Main}

```

```

{***Unidad de manejo de Canales Activos}

```

```

unit Cactivos;

```

```

interface

```

```

uses

```

```

SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
Forms, Dialogs, StdCtrls, ExtCtrls, Menus;

```

```

type

```

```

TFCanActivosE = class(TForm)

```

```

  CabezaBMP: TImage;
  F1CBox: TCheckBox;
  F2CBox: TCheckBox;
  F7CBox: TCheckBox;
  F3CBox: TCheckBox;
  FZCBox: TCheckBox;
  F4CBox: TCheckBox;
  F8CBox: TCheckBox;
  T3CBox: TCheckBox;
  C3CBox: TCheckBox;
  CZCBox: TCheckBox;
  C4CBox: TCheckBox;
  T4CBox: TCheckBox;
  T5CBox: TCheckBox;
  P3CBox: TCheckBox;
  PZCBox: TCheckBox;
  P4CBox: TCheckBox;
  T6CBox: TCheckBox;
  O1CBox: TCheckBox;
  O2CBox: TCheckBox;
  EsquemaPop: TPopupMenu;
  Esquemaltem: TMenuItem;
  NormalItem: TMenuItem;
  procedure FormClose(Sender: TObject; var Action: TCloseAction);
  procedure NormalItemClick(Sender: TObject);
  procedure FormActivate(Sender: TObject);
  procedure F1CBoxClick(Sender: TObject);

```

```
procedure F2CBoxClick(Sender: TObject);
procedure FZCBoxClick(Sender: TObject);
procedure F3CBoxClick(Sender: TObject);
procedure F4CBoxClick(Sender: TObject);
procedure F7CBoxClick(Sender: TObject);
procedure F8CBoxClick(Sender: TObject);
procedure CZCBoxClick(Sender: TObject);
procedure C3CBoxClick(Sender: TObject);
procedure C4CBoxClick(Sender: TObject);
procedure T3CBoxClick(Sender: TObject);
procedure T4CBoxClick(Sender: TObject);
procedure PZCBoxClick(Sender: TObject);
procedure P3CBoxClick(Sender: TObject);
procedure P4CBoxClick(Sender: TObject);
procedure T5CBoxClick(Sender: TObject);
procedure T6CBoxClick(Sender: TObject);
procedure O1CBoxClick(Sender: TObject);
procedure O2CBoxClick(Sender: TObject);
procedure UpdateNormalForm;
procedure FormResize(Sender: TObject);
procedure FormCreate(Sender: TObject);
```

```
private
{ Private declarations }
public
{ Public declarations }
  Minimizar : Boolean;
end;
```

```
var
  NormalActivo : Boolean;
```

```
procedure UpdateMask(Check : TCheckBox; Index : Byte; var Contador : Byte);
```

implementation

```
($R *.DFM)
```

```
uses Main,CActivNo;
```

```
procedure UpdateMask(Check : TCheckBox; Index : Byte; var Contador : Byte);
begin
  if Check.State = cbChecked Then
  begin
    MaskActivos[Index].Activo := TRUE;
    Contador:=Contador+1;
    if Contador > MaxChannels Then
    begin
      Application.messageBox('No puede haber más de 16 canales activos'+
        ' al mismo tiempo.','Error', MB_OK);
      Check.Checked:=False;
      exit;
    end;
  end
  end
  else
  begin
```

```
MaskActivos[Index].Activo :=False;
Contador:=Contador-1
end;
(sagc)
if Contador > 0 then
begin
MainForm.VisMontaje.Enabled := True and sNuevo;
MainForm.VisEeg.Enabled := True
end
else
begin
MainForm.VisMontaje.Enabled := False;
MainForm.VisEeg.Enabled := False
end

end;

procedure TFCanActivosE.UpdateNormalForm;
begin
GoCanActivosNormal.F1CBox.State:=F1CBox.State;
GoCanActivosNormal.F2CBox.State:=F2CBox.State;
GoCanActivosNormal.F7CBox.State:=F7CBox.State;
GoCanActivosNormal.F3CBox.State:=F3CBox.State;
GoCanActivosNormal.F2CBox.State:=F2CBox.State;
GoCanActivosNormal.F4CBox.State:=F4CBox.State;
GoCanActivosNormal.F8CBox.State:=F8CBox.State;
GoCanActivosNormal.T3CBox.State:=T3CBox.State;
GoCanActivosNormal.C3CBox.State:=C3CBox.State;
GoCanActivosNormal.C2CBox.State:=C2CBox.State;
GoCanActivosNormal.C4CBox.State:=C4CBox.State;
GoCanActivosNormal.T4CBox.State:=T4CBox.State;
GoCanActivosNormal.T5CBox.State:=T5CBox.State;
GoCanActivosNormal.P3CBox.State:=P3CBox.State;
GoCanActivosNormal.P2CBox.State:=P2CBox.State;
GoCanActivosNormal.P4CBox.State:=P4CBox.State;
GoCanActivosNormal.T6CBox.State:=T6CBox.State;
GoCanActivosNormal.O1CBox.State:=O1CBox.State;
GoCanActivosNormal.O2CBox.State:=O2CBox.State;
end;

procedure TFCanActivosE.FormClose(Sender: TObject;
var Action: TCloseAction);
begin
if Not(Minimizar) then
Action:=caFree;
if NormalActivo Then
Action:=caNone;
Minimizar:=False;
end;

procedure TFCanActivosE.NormalItemClick(Sender: TObject);
begin
Minimizar:=True;
Close;
GoCanActivosNormal:=TFCanActivosN.Create(Application);
NormalActivo:=True;
```

**UpdateNormalForm;**  
**end;**

**procedure TFCanActivosE.FormActivate(Sender: TObject);**  
**begin**  
    **WindowState:=wsNormal;**  
**end;**

**procedure TFCanActivosE.F1CBoxClick(Sender: TObject);**  
**begin**  
    **UpdateMask(F1CBox,1,NumDeActivos);**  
**end;**

**procedure TFCanActivosE.F2CBoxClick(Sender: TObject);**  
**begin**  
    **UpdateMask(F2CBox,2,NumDeActivos);**  
**end;**

**procedure TFCanActivosE.F7CBoxClick(Sender: TObject);**  
**begin**  
    **UpdateMask(F7CBox,3,NumDeActivos);**  
**end;**

**procedure TFCanActivosE.F3CBoxClick(Sender: TObject);**  
**begin**  
    **UpdateMask(F3CBox,4,NumDeActivos);**  
**end;**

**procedure TFCanActivosE.FZCBoxClick(Sender: TObject);**  
**begin**  
    **UpdateMask(FZCBox,5,NumDeActivos);**  
**end;**

**procedure TFCanActivosE.F4CBoxClick(Sender: TObject);**  
**begin**  
    **UpdateMask(F4CBox,6,NumDeActivos);**  
**end;**

**procedure TFCanActivosE.F8CBoxClick(Sender: TObject);**  
**begin**  
    **UpdateMask(F8CBox,7,NumDeActivos);**  
**end;**

**procedure TFCanActivosE.T3CBoxClick(Sender: TObject);**  
**begin**  
    **UpdateMask(T3CBox,8,NumDeActivos);**  
**end;**

**procedure TFCanActivosE.C3CBoxClick(Sender: TObject);**  
**begin**  
    **UpdateMask(C3CBox,9,NumDeActivos);**  
**end;**

**procedure TFCanActivosE.CZCBoxClick(Sender: TObject);**  
**begin**  
    **UpdateMask(CZCBox,10,NumDeActivos);**  
**end;**

---

```

end;

procedure TFCanActivosE.C4CBoxClick(Sender: TObject);
begin
    UpdateMask(C4CBox, 11, NumDeActivos);
end;

procedure TFCanActivosE.T4CBoxClick(Sender: TObject);
begin
    UpdateMask(T4CBox, 12, NumDeActivos);
end;

procedure TFCanActivosE.T5CBoxClick(Sender: TObject);
begin
    UpdateMask(T5CBox, 13, NumDeActivos);
end;

procedure TFCanActivosE.P3CBoxClick(Sender: TObject);
begin
    UpdateMask(P3CBox, 14, NumDeActivos);
end;

procedure TFCanActivosE.PZCBoxClick(Sender: TObject);
begin
    UpdateMask(PZCBox, 15, NumDeActivos);
end;

procedure TFCanActivosE.P4CBoxClick(Sender: TObject);
begin
    UpdateMask(P4CBox, 16, NumDeActivos);
end;

procedure TFCanActivosE.T6CBoxClick(Sender: TObject);
begin
    UpdateMask(T6CBox, 17, NumDeActivos);
end;

procedure TFCanActivosE.O1CBoxClick(Sender: TObject);
begin
    UpdateMask(O1CBox, 18, NumDeActivos);
end;

procedure TFCanActivosE.O2CBoxClick(Sender: TObject);
begin
    UpdateMask(O2CBox, 19, NumDeActivos);
end;

procedure TFCanActivosE.FormResize(Sender: TObject);
begin
    if NormalActivo Then
        WindowState:=wsMinimized;
end;

procedure TFCanActivosE.FormCreate(Sender: TObject);
begin
    (sagc)

```

```

F1CBox.Enabled:=MaskActivos[1].Disponible and sNuevo;
F2CBox.Enabled:=MaskActivos[2].Disponible and sNuevo;
F7CBox.Enabled:=MaskActivos[3].Disponible and sNuevo;
F3CBox.Enabled:=MaskActivos[4].Disponible and sNuevo;
F2CBox.Enabled:=MaskActivos[5].Disponible and sNuevo;
F4CBox.Enabled:=MaskActivos[6].Disponible and sNuevo;
F8CBox.Enabled:=MaskActivos[7].Disponible and sNuevo;
T3CBox.Enabled:=MaskActivos[8].Disponible and sNuevo;
C3CBox.Enabled:=MaskActivos[9].Disponible and sNuevo;
C2CBox.Enabled:=MaskActivos[10].Disponible and sNuevo;
C4CBox.Enabled:=MaskActivos[11].Disponible and sNuevo;
T4CBox.Enabled:=MaskActivos[12].Disponible and sNuevo;
T5CBox.Enabled:=MaskActivos[13].Disponible and sNuevo;
P3CBox.Enabled:=MaskActivos[14].Disponible and sNuevo;
P2CBox.Enabled:=MaskActivos[15].Disponible and sNuevo;
P4CBox.Enabled:=MaskActivos[16].Disponible and sNuevo;
T6CBox.Enabled:=MaskActivos[17].Disponible and sNuevo;
O1CBox.Enabled:=MaskActivos[18].Disponible and sNuevo;
O2CBox.Enabled:=MaskActivos[19].Disponible and sNuevo;

```

```
NumDeActivos:=0;
```

```

F1CBox.Checked:=MaskActivos[1].Activo;
F2CBox.Checked:=MaskActivos[2].Activo;
F7CBox.Checked:=MaskActivos[3].Activo;
F3CBox.Checked:=MaskActivos[4].Activo;
F2CBox.Checked:=MaskActivos[5].Activo;
F4CBox.Checked:=MaskActivos[6].Activo;
F8CBox.Checked:=MaskActivos[7].Activo;
T3CBox.Checked:=MaskActivos[8].Activo;
C3CBox.Checked:=MaskActivos[9].Activo;
C2CBox.Checked:=MaskActivos[10].Activo;
C4CBox.Checked:=MaskActivos[11].Activo;
T4CBox.Checked:=MaskActivos[12].Activo;
T5CBox.Checked:=MaskActivos[13].Activo;
P3CBox.Checked:=MaskActivos[14].Activo;
P2CBox.Checked:=MaskActivos[15].Activo;
P4CBox.Checked:=MaskActivos[16].Activo;
T6CBox.Checked:=MaskActivos[17].Activo;
O1CBox.Checked:=MaskActivos[18].Activo;
O2CBox.Checked:=MaskActivos[19].Activo;
end;

```

```
initialization
```

```

begin
  NormalActivo:=False;
end;

```

```
end.(unit Cactivos)
```

```
{**Unidad de Graficación de la señal EEG: Tiempo Real y Almacenada}
```

```

unit Graf1;
{$R+}
{$M 32000,8192}
interface

```

uses

SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,  
Forms, Dialogs, VBXCtrl, Graph, StdCtrls, ExtCtrls, Buttons, Mask;

type

```
TGrafica = class(TForm)
  Timer1: TTimer;
  GroupBox1: TGroupBox;
  BitBtn1: TBitBtn;
  BitBtn2: TBitBtn;
  botongrabar: TBitBtn;
  mensaje: TLabel;
  mensaje2: TLabel;
  Graph1: TGraph;
  procedure Timer1Timer(Sender: TObject);
  procedure Button1Click(Sender: TObject);
  procedure Button2Click(Sender: TObject);
  procedure FormClose(Sender: TObject; var Action: TCloseAction);
  procedure FormCreate(Sender: TObject);
  procedure BitBtn1Click(Sender: TObject);
  procedure BitBtn2Click(Sender: TObject);
  procedure FormActivate(Sender: TObject);
  procedure botongrabarClick(Sender: TObject);
```

```
  { Private declarations }
```

```
  public
```

```
  { Public declarations }
```

```
end;
```

var

```
  Grafica: TGrafica;
```

implementation

```
{SR *.DFM}
```

```
uses main,IntDAQ;
```

```
var
```

```
  graba: boolean;
```

```
  archeeg: file;
```

```
  archhdr: textfile;
```

```
  cuenta: longint;
```

```
  ARR1:ARRAY[1..MaxChannels,1..NumDeMuestras] OF byte;
```

```
  ARR2:ARRAY[1..MaxChannels,1..NumDeMuestras] OF byte;
```

```
  Sampling : Boolean;
```

```
procedure TGrafica.Timer1Timer(Sender: TObject);
```

```
  var
```

```
    inicial,final : real;
```

```
    K,J,I,I1,J1:INTEGER;
```

```
    escritos: word;
```

```
    arr3:array[1..MaxChannels] of byte;
```

```
    ElBuffer : PInteger;
```

```
begin
```

```
  Inicial:=time;
```

```
  (*for I:=graph1.numpoints downto 1 do
```

```
    begin
```

```
      graph1.thispoint := I;
```

```

for J:=1 to graph1.numsets do
begin
graph1.thisset := J;
graph1.graphdata := arr1[J,1]+(15*J)-15;
arr2[J,1]:=random(10)

end;
end;
graph1.drawmode := 3;
("comienza segundo ciclo de llenado de array")
StatusDAQ:=Transferir(NumDisp,@BufferDAQ[1]);
if statusDAQ <> 0 Then
begin
DisplayDAQError(StatusDAQ);
Timer1.Enabled:=False;
exit;
end;
ElBuffer:=@BufferDAQ[1];
for I1:=graph1.numpoints downto 1 do
begin
graph1.thispoint := I1;
for J1:= 1 to graph1.numsets do
begin
graph1.thisset := J1;
arr2[J1,I1]:=lo(ElBuffer^);
Inc(ElBuffer);
arr3[J1]:=arr2[J1,I1];
graph1.graphdata := arr2[J1,I1]+(15*(graph1.numsets-J1+1))-15;
end;
if graba then
blockwrite(archeeg,arr3,graph1.numsets,escritos);
end;
final:=(time-inicial)*86400;
{mensaje2.caption:='tiempo usado: '+chr(13)+floattostr(final);}
if graba = true then
begin
cuenta := cuenta+1;
mensaje.caption:='muestra: '+inttostr(cuenta);
end;
graph1.drawmode := 3;
end;
procedure TGrafica.Button1Click(Sender: TObject);
begin
Timer1.Enabled:=True;
end;

procedure TGrafica.Button2Click(Sender: TObject);
begin
Timer1.Enabled:=False;
end;

procedure TGrafica.FormClose(Sender: TObject; var Action: TCloseAction);
begin
if Sampling Then
begin
StatusDAQ:=DetenerDAQ(NumDisp);

```

```
    if statusDAQ <> 0 Then
        DisplayDAQError(StatusDAQ);
    end;
    Action:=caFree;
end;

procedure TGrafica.FormCreate(Sender: TObject);
var
    k : word;
begin
    if numdeactivos <> 0 then
        begin
            graph1.numpoints := NumDeMuestras;
            graph1.numsets := numdeactivos;
            for k:= 1 to 19 do
                begin
                    if maskactivos[k].activo then
                        graph1.legendtext:=maskactivos[k].nomreal;
                    end;
                end
            end
        else
            begin
                showmessage("Seleccione los Canales Activos primero");
                Close
            end;
        end;
        Sampling:=False;
    end;

    procedure TGrafica.BitBtn1 Click(Sender: TObject);
    begin
        StatusDAQ:=ConfigEntAnalogica(NumDisp);
        if statusDAQ <> 0 Then
            begin
                DisplayDAQError(StatusDAQ);
                exit;
            end;
        StatusDAQ:=ConfigTipoMuestreo(NumDisp);
        if statusDAQ <> 0 Then
            begin
                DisplayDAQError(StatusDAQ);
                exit;
            end;
        StatusDAQ:=InicioDAQ(NumDisp, VelMuestreo);
        if statusDAQ <> 0 Then
            begin
                DisplayDAQError(StatusDAQ);
                exit;
            end;
        Sampling :=True;
        Timer1.Enabled:=True;
        botongrabar.visible:=true;
    end;

    procedure TGrafica.BitBtn2Click(Sender: TObject);
    begin
        if botongrabar.caption='Grabar' then
```

```

Begin
Timer1.Enabled:=False;
StatusDAQ:=DetenerDAQ(NumDisp);
if statusDAQ <> 0 Then
  DisplayDAQError(StatusDAQ);
  botongrabar.visible:=false;
end;
end;

procedure TGrafica.FormActivate(Sender: TObject);
begin
  WindowState:=wsMaximized;
end;

procedure TGrafica.botongrabarClick(Sender: TObject);
var
  Numestudio: string[8];
  ClickedOK: Boolean;
  k: byte;
begin
if BotonGrabar.caption='FinGrabar' then
  begin
  BotonGrabar.caption:='Grabar';
  closefile(archeeg);
  closefile(archhdr);
  graba := false;
  MainForm.Procesos.Visible := True;
  end
else
  begin
  Numestudio := NomArchEEG;
  BotonGrabar.caption:='FinGrabar';
  assignfile(archeeg,numestudio+'.eeg');
  assignfile(archhdr,numestudio+'.hdr');
  rewrite(archeeg,1);
  rewrite(archhdr);
  writeln(archhdr,NumdeActivos);
  For k := 1 to 19 do
  begin
  writeln(archhdr,maskactivos[k].nomreal);
  writeln(archhdr,MaskActivos[k].Activo);
  end;
  graba:=true;
  cuenta:=0
  end;
end;

end.(unit graph1)

```

{\*\*\*Unidad de verificación de correcto montaje de los electrodos}

unit Montaje;

interface

uses

SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,  
Forms, Dialogs, StdCtrls, ExtCtrls, Menus, Tabs;

type

```
TFormaMontaje = class(TForm)
  CabezaBMP: TImage;
  F1Led: TImage;
  F2Led: TImage;
  F7Led: TImage;
  F3Led: TImage;
  F2Led: TImage;
  F4Led: TImage;
  F8Led: TImage;
  T3Led: TImage;
  C3Led: TImage;
  CZLed: TImage;
  C4Led: TImage;
  T4Led: TImage;
  T5Led: TImage;
  P3Led: TImage;
  PZLed: TImage;
  P4Led: TImage;
  T6Led: TImage;
  O1Led: TImage;
  O2Led: TImage;
  LedYellow: TImage;
  LedGreen: TImage;
  LedOff: TImage;
  Verifica: TTimer;
  procedure FormClose(Sender: TObject; var Action: TCloseAction);
  procedure VerificaTimer(Sender: TObject);
  procedure FormCreate(Sender: TObject);
private
  { Private declarations }
  procedure ActualizaLed(Led : TImage; Index, Posi : Byte);
public
  { Public declarations }
  Minimizar : Boolean;
end;
```

implementation

{SR \*.DFM}

uses Main,IntDAQ;

Var

```
Voltajes : Array[1..MaxChannels] of Double;
BlinkIt : Array[1..MaxChannels] of Boolean;
k : Byte;
```

procedure TFormaMontaje.ActualizaLED(Led : TImage; Index, Posi : Byte);

```
begin
  if MaskActivos[Posi].Activo Then
    begin
      if (Voltajes[Index] >= Umbral) Then
```

```

    Led.Picture:=LedGreen.Picture
  else
    Led.Picture:=LedYellow.Picture;
  end
else
  begin
    if (Voltajes[Index] >= Umbral) Then
      begin
        if BlinkIt[Index] Then
          Led.Picture:=LedYellow.Picture
        else
          Led.Picture:=LedOff.Picture;
          BlinkIt[Index]:=Not(BlinkIt[Index]);
        end
      end
    else
      Led.Picture:=LedOff.Picture;
    end;
  end;
end;

procedure TFormaMontaje.FormClose(Sender: TObject;
var Action: TCloseAction);
begin
  Action:=caFree;
end;

procedure TFormaMontaje.VerificaTimer(Sender: TObject);
begin
  StatusDAQ:=ConfigEntAnalogica(NumDisp);
  if statusDAQ <> 0 Then
    begin
      Verifica.Enabled:=False;
      DisplayDAQError(StatusDAQ);
      exit;
    end;
  StatusDAQ:=LeeValor(NumDisp,@Voltajes[1]);
  if statusDAQ <> 0 Then
    begin
      DisplayDAQError(StatusDAQ);
      Verifica.Enabled:=False;
      exit;
    end;

  k:=1;
  if MaskActivos[1].Disponible Then
    begin
      ActualizaLED(F1Led,K,1);
      Inc(k);
    end;
  if MaskActivos[2].Disponible Then
    begin
      ActualizaLED(F2Led,K,2);
      Inc(k);
    end;
  if MaskActivos[3].Disponible Then
    begin

```

```
ActualizaLED(F7Led,K,3);
Inc(k);
end;
if MaskActivos[4].Disponible Then
begin
  ActualizaLED(F3Led,K,4);
  Inc(k);
end;
if MaskActivos[5].Disponible Then
begin
  ActualizaLED(FZLed,K,5);
  Inc(k);
end;
if MaskActivos[6].Disponible Then
begin
  ActualizaLED(F4Led,K,6);
  Inc(k);
end;
if MaskActivos[7].Disponible Then
begin
  ActualizaLED(F8Led,K,7);
  Inc(k);
end;
if MaskActivos[8].Disponible Then
begin
  ActualizaLED(T3Led,K,8);
  Inc(k);
end;
if MaskActivos[9].Disponible Then
begin
  ActualizaLED(C3Led,K,9);
  Inc(k);
end;
if MaskActivos[10].Disponible Then
begin
  ActualizaLED(CZLed,K,10);
  Inc(k);
end;
if MaskActivos[11].Disponible Then
begin
  ActualizaLED(C4Led,K,11);
  Inc(k);
end;
if MaskActivos[12].Disponible Then
begin
  ActualizaLED(T4Led,K,12);
  Inc(k);
end;
if MaskActivos[13].Disponible Then
begin
  ActualizaLED(T5Led,K,13);
  Inc(k);
end;
if MaskActivos[14].Disponible Then
begin
  ActualizaLED(P3Led,K,14);
```

```

    Inc(k);
end;
if MaskActivos[15].Disponible Then
begin
    ActualizaLED(PZLed,K,15);
    Inc(k);
end;
if MaskActivos[16].Disponible Then
begin
    ActualizaLED(P4Led,K,16);
    Inc(k);
end;
if MaskActivos[17].Disponible Then
begin
    ActualizaLED(T6Led,K,17);
    Inc(k);
end;
if MaskActivos[18].Disponible Then
begin
    ActualizaLED(O1Led,K,18);
    Inc(k);
end;
if MaskActivos[19].Disponible Then
begin
    ActualizaLED(O2Led,K,19);
    Inc(k);
end;
end;

procedure TFormaMontaje.FormCreate(Sender: TObject);
begin
    For k:=1 to MaxChannels Do
        BlinkIt[k]:=False;
    { For k:=1 to MaxChannels Do
        begin
            if k MOD 2 = 0 Then
                Voltajes[k]:=0.5
            else
                Voltajes[k]:=0.00001;
            end;}
            { Esta porci%dn de c%digo solo es para probar }
end;

end;

initialization
begin

end;

end.{unit Montaje}

```

\*\*\*Unidad de Reproducción de la Señal EEG

unit Repro;

interface

uses

SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,  
Forms, Dialogs, VBXCtrl, Graph, StdCtrls, Buttons, ExtCtrls, Mask, printers;

type

```
Tformrepro = class(TForm)
  Graphgra: TGraph;
  GroupBox2: TGroupBox;
  BitBtn1: TBitBtn;
  BitBtn2: TBitBtn;
  Botonimp: TBitBtn;
  Timer2: TTimer;
  mensaje3: TLabel;
  Label1: TLabel;
  MaskEdit1: TMaskEdit;
  procedure FormActivate(Sender: TObject);
  procedure FormClose(Sender: TObject; var Action: TCloseAction);
  procedure FormCreate(Sender: TObject);
  procedure BitBtn1Click(Sender: TObject);
  procedure BitBtn2Click(Sender: TObject);
  procedure Timer2Timer(Sender: TObject);
  procedure BotonimpClick(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
```

var

formrepro: Tformrepro;  
implementation

(\$R \*.DFM)

uses main,intdaq;

var

archeegname : file;  
muestra : word;

```
procedure Tformrepro.FormActivate(Sender: TObject);
begin
  WindowState:=wsMaximized;
end;
```

```
procedure Tformrepro.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  Action:=caFree;
end;
```

```
procedure Tformrepro.FormCreate(Sender: TObject);
var
  k : word;
begin
  assignfile(archeegname,mainform.opendialog,filename);
  reset(archeegname,1);
  graphgra.numpoints := NumdeMuestras;
  graphgra.numsets := numdeactivos;
end;
```

```

procedure Tformrepro.BitBtn1Click(Sender: TObject);
var
  muest : word;
  codig : word;
begin
  val(maskedit1.text,muest,codig);
  seek(archeegname,muest);
  muestra:=muest;
  timer2.enabled:=true;
end;

procedure Tformrepro.BitBtn2Click(Sender: TObject);
begin
  timer2.enabled:=false;
end;

procedure Tformrepro.Timer2Timer(Sender: TObject);
var
  inicial,final : real;
  I1,J1:INTEGER;
  arr3:array[1..16] of byte;
begin
  Inicial:=time;

  for I1:=graphgra.numpoints downto 1 do
    begin
      graphgra.thispoint := I1;
      blockread(archeegname, arr3, graphgra.numsets);
      for J1:=1 to graphgra.numsets do
        begin
          graphgra.thisset := J1;
          graphgra.graphdata := arr3[J1]+(15*(graphgra.numsets-J1+1))-15;
        end;
      end;
      final:=(time-inicial)*86400;
      mensaje3.caption:='Muestra: '+inttostr(muestra);
      muestra := muestra+1;
      graphgra.drawmode := 3;
      if eof(archeegname) then
        timer2.enabled:=false;
    end;

procedure Tformrepro.BotonimpClick(Sender: TObject);
begin
  groupbox2.visible:=false;
  printer.orientation:=portlandscape;
  printscale:=poprinttofit;
  print;
  groupbox2.visible:=true;
end;

end.(unit Repro)

```

```
{***Unidad para el cálculo del espectro de potencia de la señal EEG}
```

```
unit Spectrum;
interface
```

```
uses
```

```
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, StdCtrls, Grids;
```

```
const
```

```
  DATASIZE = 256;
  POWERINDEX = 8;
```

```
type
```

```
  TPotencia = class(TForm)
  ButtonFHT: TButton;
  procedure ButtonFHTClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
```

```
  directionType = (directa, inversa);
  dataArrayType = array[1..DATASIZE] of real;
  dataArrayType2 = array[1..DATASIZE div 2 + 1] of real;
```

```
{ FHT - Fast Hartley Transform }
{ transformDirection define si la transformada es directa o inversa }
{ syze: Numero de elementos en el arreglo de datos de entrada }
procedure fht (var dataArray: dataArrayType;
  size,
  powerIndex : integer;
  transformDirection : directionType);
procedure get_FFT(var data : dataArrayType;
  var parteR : dataArrayType;
  var parteI : dataArrayType;
  size: integer);
procedure getMod_FFT(var data : dataArrayType;
  var fft : dataArrayType;
  size: integer);
procedure get_powerSpectrum(var data: dataArrayType;
  var powSpec : dataArrayType2;
  size: integer);
```

```
implementation
```

```
{SR *.DFM}
```

```
Uses Main;
```

```
procedure fht (var dataArray: dataArrayType;
  size,
  powerIndex : integer;
  transformDirection : directionType);
var
```

```

i, j, k,
trgInd, trgInc,
power,
t_a,
f_a,
iTemp,
section,
sStart,
sEnd      : integer;
sno, cse  : array[1..DATASIZE] of real;
accum     : array[1..2, 1..DATASIZE] of real;

```

```

{Rutina de permutación}
{Reordena los datos antes de que la transformaci%
n mariposa sea llamada}
function permuta(index: integer) : integer;

```

```

var
  i, j, s  : integer;
begin
  j := 0;
  index := index-1;
  for i:=1 to powerIndex do
  begin
    s := index div 2;
    j := j + j + index - s - s;
    index := s
  end;
  permuta := j + 1
end; {permuta}

```

```

{Calcula las funciones trigonometricas requeridas por la FHT y almacena sus}
{valores. Para una transformada de N puntos, las funciones trigonometricas}
{seran calculadas a intervalos de N}
procedure trigoTabla(nPtos : integer);
const

```

```

  PI = 3.14159265;
var
  i      : integer;
  angulo,
  omega  : real;
begin
  angulo := 0;
  omega := 2 * PI / nPtos;
  for i:= 1 to nPtos do
  begin
    sno[i] := sin(angulo);
    cse[i] := cos(angulo);
    angulo := angulo + omega
  end
end; {trigoTabla}

```

```

{calcula la direccion del indice retrogrado para el termino seno para el algoritmo}
{de localizacion dual, si se requiere}
function modifica(power, sStart, sEnd, index: integer) : integer;
begin

```

```

  if (sStart = index) or (power < 3) then
    modifica := index

```

```

else
  modifica := sStart + sEnd - index + 1
end; {modifica}

procedure mariposa(trigInd, i1, i2, i3: integer);
begin
  accum[t_a, i1] := accum[f_a, i1] +
    accum[f_a, i2] * cse(trigInd) +
    accum[f_a, i3] * sno(trigInd);
  trigInd := trigInd + size div 2;
  accum[t_a, i2] := accum[f_a, i1] +
    accum[f_a, i2] * cse(trigInd) +
    accum[f_a, i3] * sno(trigInd)
end; {mariposa}

{procedimiento principal FHT}
begin
  power := 1;
  f_a := 1;
  t_a := 2;
  trigoTabla(size);
  for i := 1 to size do
    accum[f_a, permuta(i)] := dataArray[i];

  {inicia la transformacion mariposa}
  for i := 1 to powerIndex do
    begin
      j := 1;
      section := 1;
      trglnc := size div (power + power);
      repeat
        trglnd := 1;
        sStart := section * power + 1;
        sEnd := (section + 1) * power;
        for k := 1 to power do
          begin
            mariposa(trglnd, j, j + power,
              modifica(power, sStart, sEnd, j + power));
            trglnd := trglnd + trglnc;
            j := j + 1
          end;
          j := j + power;
          section := section + 2;
        until j > size;
        power := power + power;
        iTemp := t_a;
        t_a := f_a;
        f_a := iTemp
      end;
    end;
  {fin de la mariposa de Hartley. El resultado es escalado si se necesita y luego}
  {regresado en el arreglo de datos}

  case transformDirection of
    directa : for i:=1 to size do
              dataArray[i] := accum[f_a,i]/size;
    inversa : for i:= 1 to size do

```

```

                                dataArray[i] := accum[f_a,i]
        end
    end;{FHT}

    {Calculo de la transformada de Fourier a partir del algoritmo FHT}
    procedure get_FFT(var data : dataArrayType;
                    var parteR : dataArrayType;
                    var parteI : dataArrayType;
                    size: integer);

    var
        i : integer;
    begin
        i := 2;
        parteR[1] := data[1];
        parteI[1] := 0;
        while i <= size do
            begin
                parteR[i] := (data[i] + data[size-i+2])/2;
                parteI[i] := (data[i] - data[size-i+2])/2;
                i := i + 1
            end
        end
    end;{get_FFT}

    {Calculo de la transformada de Fourier a partir del algoritmo FHT}
    procedure getMod_FFT(var data : dataArrayType;
                        var fft : dataArrayType;
                        size: integer);

    var
        i : integer;
    begin
        i := 1;
        while i <= size do
            begin
                fft[i] := sqrt(sqr(data[i] + data[size-i+1]) + sqr(data[i] - data[size-i+1]));
                i := i + 1
            end
        end
    end;{getMod_FFT}

    {Calculo del espectro de potencia a partir del algoritmo FHT}
    procedure get_powerSpectrum(var data: dataArrayType;
                                var powSpec : dataArrayType2;
                                size: integer);

    var
        i : integer;
    begin
        i := 1;
        while i <= size DIV 2 + 1 do
            begin
                powSpec[i] := (sqr(data[i]) + sqr(data[size - i + 1])) / 2;
                i := i + 1
            end
        end
    end;{get_powerSpectrum}

    procedure TPotencia.ButtonFHTClick(Sender: TObject);
    begin
        Espectro
    end

```

```

end;

{Espectro: Calcula el espectro de potencia para todos los canales activos}
Procedure Espectro;
var i,k :shortint;
    indice : Array[1..19] of shortint;
    j,dumb : Byte;
    dataArray : dataArrayType;
    powerSp : dataArrayType2;
    nameFile : string[11];
    dFile : File of Byte;
    ansFile : File of Real;

begin
    numDeActivos:= 0;
    for i:=1 to 19 do
    begin
        {obtenemos cuÑles y cuÑntos son los canales activos}
        if maskActivos[i].Activo then
            begin
                numDeActivos:= numDeActivos + 1;
                indice[numDeActivos] := i
            end;if)
    end;{for}
    for i:=1 to numDeActivos do
    begin
        AssignFile(dFile, NomArchEEG+'.eeg');
        Reset(dFile);
        for j:=0 to DATASIZE-1 do
            begin
                read(dFile, dumb);
                dataArray[j+1] := dumb;
                for k:=2 to numDeActivos do
                    read(dFile, dumb);
                end;{for j}
                fht (dataArray, DATASIZE, POWERINDEX, directa);
                get_PowerSpectrum(dataArray, powerSp, DATASIZE);
                nameFile := nomArchEEG+'.P'+MaskActivos[indice[i]].Nomint;
                AssignFile( ansFile, nameFile );
                rewrite(ansFile);
                for j:= 1 to DATASIZE div 2 + 1 do
                    begin
                        write(ansFile, powerSp[j])
                    end{for j}
                end{for i}
    end;{Espectro}

end.{Unit Spectrum}

```

```

{***Unidad que genera el mapero electroencefalográfico}

```

```

unit Mapmain;

```

```

interface

```

```

uses

```

**SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,  
Forms, Dialogs, ExtCtrls, StdCtrls, ColorGrd, Menus, Buttons;**

**type**

```
TMapMainForm = class(TForm)  
  MapColorBack: TGroupBox;  
  MapColorTxt18: TLabel;  
  MapColorTxt17: TLabel;  
  MapColorTxt16: TLabel;  
  MapColorTxt15: TLabel;  
  MapColorTxt14: TLabel;  
  MapColorTxt13: TLabel;  
  MapColorTxt12: TLabel;  
  MapColorTxt11: TLabel;  
  MapColorTxt10: TLabel;  
  MapColorTxt9: TLabel;  
  MapColorTxt8: TLabel;  
  MapColorTxt7: TLabel;  
  MapColorTxt6: TLabel;  
  MapColorTxt5: TLabel;  
  MapColorTxt4: TLabel;  
  MapColorTxt3: TLabel;  
  MapColorTxt2: TLabel;  
  MapColorTxt1: TLabel;  
  MapColor18: TPanel;  
  MapColor17: TPanel;  
  MapColor16: TPanel;  
  MapColor15: TPanel;  
  MapColor14: TPanel;  
  MapColor13: TPanel;  
  MapColor12: TPanel;  
  MapColor11: TPanel;  
  MapColor10: TPanel;  
  MapColor9: TPanel;  
  MapColor8: TPanel;  
  MapColor7: TPanel;  
  MapColor6: TPanel;  
  MapColor5: TPanel;  
  MapColor4: TPanel;  
  MapColor3: TPanel;  
  MapColor2: TPanel;  
  MapColor1: TPanel;  
  MapCajaBeta: TGroupBox;  
  MapCabezaBeta: TImage;  
  MapCajaTeta: TGroupBox;  
  MapCabezaTeta: TImage;  
  MapCajaAlfa: TGroupBox;  
  MapCabezaAlfa: TImage;  
  MapCajaDelta: TGroupBox;  
  MapCabezaDelta: TImage;
```

```
procedure FormCreate(Sender: TObject);  
procedure FormClose(Sender: TObject; var Action: TCloseAction);
```

```
private  
  { Private declarations }
```

```
procedure EstableceZonaPorCanal;
procedure EstableceEscalaColores;
procedure InicializaTabla;
procedure LlenaCabezaAlfa(Color : TColor; Canal : Integer);
procedure LlenaCabezaBeta(Color : TColor; Canal : Integer);
procedure LlenaCabezaDelta(Color : TColor; Canal : Integer);
procedure LlenaCabezaTeta(Color : TColor; Canal : Integer);
public
{ Public declarations }
end;

var
MapMainForm: TMapMainForm;

implementation

{$R *.DFM}

uses
  main;
const
  MaxPuntos = 17;
  MaxColores = 18;
  MaxRitmos = 4;
type
  ColorAmplitudType = Record
    Color : TColor;
    Amplitud : Real;
  end;
var
  MapMatrizPuntos : Array[1..NumElectrodos, 1..MaxPuntos] of TPoint;
  MapColores : Array[1..MaxColores] of ColorAmplitudType;
  MapTabla : Array[1..MaxRitmos, 1..4] of Real;

procedure TMapMainForm.EstableceZonaPorCanal;
begin
  { Establece zona constante asociada al canal F1 }
  MapMatrizPuntos[1,1] := Point(67,15);
  MapMatrizPuntos[1,2] := Point(63,24);
  MapMatrizPuntos[1,3] := Point(69,29);
  MapMatrizPuntos[1,4] := Point(64,33);
  MapMatrizPuntos[1,5] := Point(60,39);
  MapMatrizPuntos[1,6] := Point(52,35);
  MapMatrizPuntos[1,7] := Point(46,36);
  MapMatrizPuntos[1,8] := Point(43,29);
  MapMatrizPuntos[1,9] := Point(36,25);
  MapMatrizPuntos[1,10] := Point(46,21);
  MapMatrizPuntos[1,11] := Point(54,18);
  MapMatrizPuntos[1,12] := Point(60,16);
  MapMatrizPuntos[1,13] := Point(67,15);
  MapMatrizPuntos[1,14] := Point(67,15);
  MapMatrizPuntos[1,15] := Point(67,15);
  MapMatrizPuntos[1,16] := Point(67,15);
  MapMatrizPuntos[1,17] := Point(67,15);

  { Establece zona constante asociada al canal F2 }
```

```
MapMatrizPuntos[2,1] := Point(67,15);
MapMatrizPuntos[2,2] := Point(63,24);
MapMatrizPuntos[2,3] := Point(69,29);
MapMatrizPuntos[2,4] := Point(76,39);
MapMatrizPuntos[2,5] := Point(83,35);
MapMatrizPuntos[2,6] := Point(91,35);
MapMatrizPuntos[2,7] := Point(93,30);
MapMatrizPuntos[2,8] := Point(102,28);
MapMatrizPuntos[2,9] := Point(97,24);
MapMatrizPuntos[2,10] := Point(91,21);
MapMatrizPuntos[2,11] := Point(86,19);
MapMatrizPuntos[2,12] := Point(79,17);
MapMatrizPuntos[2,13] := Point(67,15);
MapMatrizPuntos[2,14] := Point(67,15);
MapMatrizPuntos[2,15] := Point(67,15);
MapMatrizPuntos[2,16] := Point(67,15);
MapMatrizPuntos[2,17] := Point(67,15);
```

```
{ Establece zona constante asociada al canal F7 }
```

```
MapMatrizPuntos[3,1] := Point(37,26);
MapMatrizPuntos[3,2] := Point(43,29);
MapMatrizPuntos[3,3] := Point(45,36);
MapMatrizPuntos[3,4] := Point(42,41);
MapMatrizPuntos[3,5] := Point(39,50);
MapMatrizPuntos[3,6] := Point(33,54);
MapMatrizPuntos[3,7] := Point(33,58);
MapMatrizPuntos[3,8] := Point(26,52);
MapMatrizPuntos[3,9] := Point(22,58);
MapMatrizPuntos[3,10] := Point(17,53);
MapMatrizPuntos[3,11] := Point(19,47);
MapMatrizPuntos[3,12] := Point(22,42);
MapMatrizPuntos[3,13] := Point(26,36);
MapMatrizPuntos[3,14] := Point(31,30);
MapMatrizPuntos[3,15] := Point(37,26);
MapMatrizPuntos[3,16] := Point(37,26);
MapMatrizPuntos[3,17] := Point(37,26);
```

```
{ Establece zona constante asociada al canal F3 }
```

```
MapMatrizPuntos[4,1] := Point(45,36);
MapMatrizPuntos[4,2] := Point(42,41);
MapMatrizPuntos[4,3] := Point(39,50);
MapMatrizPuntos[4,4] := Point(33,54);
MapMatrizPuntos[4,5] := Point(33,58);
MapMatrizPuntos[4,6] := Point(40,58);
MapMatrizPuntos[4,7] := Point(44,55);
MapMatrizPuntos[4,8] := Point(51,60);
MapMatrizPuntos[4,9] := Point(58,62);
MapMatrizPuntos[4,10] := Point(60,57);
MapMatrizPuntos[4,11] := Point(58,49);
MapMatrizPuntos[4,12] := Point(60,39);
MapMatrizPuntos[4,13] := Point(52,35);
MapMatrizPuntos[4,14] := Point(45,36);
MapMatrizPuntos[4,15] := Point(45,36);
MapMatrizPuntos[4,16] := Point(45,36);
MapMatrizPuntos[4,17] := Point(45,36);
```

{ Establece zona constante asociada al canal FZ }

MapMatrizPuntos[5,1] := Point(69,29);  
MapMatrizPuntos[5,2] := Point(64,33);  
MapMatrizPuntos[5,3] := Point(60,39);  
MapMatrizPuntos[5,4] := Point(58,49);  
MapMatrizPuntos[5,5] := Point(60,57);  
MapMatrizPuntos[5,6] := Point(67,61);  
MapMatrizPuntos[5,7] := Point(70,59);  
MapMatrizPuntos[5,8] := Point(77,57);  
MapMatrizPuntos[5,9] := Point(74,52);  
MapMatrizPuntos[5,10] := Point(78,47);  
MapMatrizPuntos[5,11] := Point(76,39);  
MapMatrizPuntos[5,12] := Point(69,29);  
MapMatrizPuntos[5,13] := Point(69,29);  
MapMatrizPuntos[5,14] := Point(69,29);  
MapMatrizPuntos[5,15] := Point(69,29);  
MapMatrizPuntos[5,16] := Point(69,29);  
MapMatrizPuntos[5,17] := Point(69,29);

{ Establece zona constante asociada al canal F4 }

MapMatrizPuntos[6,1] := Point(76,39);  
MapMatrizPuntos[6,2] := Point(78,47);  
MapMatrizPuntos[6,3] := Point(74,52);  
MapMatrizPuntos[6,4] := Point(77,57);  
MapMatrizPuntos[6,5] := Point(83,61);  
MapMatrizPuntos[6,6] := Point(88,56);  
MapMatrizPuntos[6,7] := Point(101,59);  
MapMatrizPuntos[6,8] := Point(97,56);  
MapMatrizPuntos[6,9] := Point(98,45);  
MapMatrizPuntos[6,10] := Point(95,40);  
MapMatrizPuntos[6,11] := Point(91,35);  
MapMatrizPuntos[6,12] := Point(83,35);  
MapMatrizPuntos[6,13] := Point(76,39);  
MapMatrizPuntos[6,14] := Point(76,39);  
MapMatrizPuntos[6,15] := Point(76,39);  
MapMatrizPuntos[6,16] := Point(76,39);  
MapMatrizPuntos[6,17] := Point(76,39);

{ Establece zona constante asociada al canal F8 }

MapMatrizPuntos[7,1] := Point(119,51);  
MapMatrizPuntos[7,2] := Point(114,62);  
MapMatrizPuntos[7,3] := Point(101,59);  
MapMatrizPuntos[7,4] := Point(97,56);  
MapMatrizPuntos[7,5] := Point(98,45);  
MapMatrizPuntos[7,6] := Point(95,40);  
MapMatrizPuntos[7,7] := Point(91,35);  
MapMatrizPuntos[7,8] := Point(93,30);  
MapMatrizPuntos[7,9] := Point(102,28);  
MapMatrizPuntos[7,10] := Point(108,33);  
MapMatrizPuntos[7,11] := Point(113,39);  
MapMatrizPuntos[7,12] := Point(116,44);  
MapMatrizPuntos[7,13] := Point(119,51);  
MapMatrizPuntos[7,14] := Point(119,51);  
MapMatrizPuntos[7,15] := Point(119,51);  
MapMatrizPuntos[7,16] := Point(119,51);  
MapMatrizPuntos[7,17] := Point(119,51);

{ Establece zona constante asociada al canal T3 }

MapMatrizPuntos[8,1] := Point(16,53);  
MapMatrizPuntos[8,2] := Point(22,58);  
MapMatrizPuntos[8,3] := Point(26,52);  
MapMatrizPuntos[8,4] := Point(33,58);  
MapMatrizPuntos[8,5] := Point(35,68);  
MapMatrizPuntos[8,6] := Point(33,72);  
MapMatrizPuntos[8,7] := Point(36,78);  
MapMatrizPuntos[8,8] := Point(30,82);  
MapMatrizPuntos[8,9] := Point(26,88);  
MapMatrizPuntos[8,10] := Point(21,84);  
MapMatrizPuntos[8,11] := Point(17,90);  
MapMatrizPuntos[8,12] := Point(15,82);  
MapMatrizPuntos[8,13] := Point(14,74);  
MapMatrizPuntos[8,14] := Point(14,64);  
MapMatrizPuntos[8,15] := Point(16,53);  
MapMatrizPuntos[8,16] := Point(16,53);  
MapMatrizPuntos[8,17] := Point(16,53);

{ Establece zona constante asociada al canal C3 }

MapMatrizPuntos[9,1] := Point(33,58);  
MapMatrizPuntos[9,2] := Point(35,68);  
MapMatrizPuntos[9,3] := Point(33,72);  
MapMatrizPuntos[9,4] := Point(36,78);  
MapMatrizPuntos[9,5] := Point(43,81);  
MapMatrizPuntos[9,6] := Point(48,79);  
MapMatrizPuntos[9,7] := Point(53,82);  
MapMatrizPuntos[9,8] := Point(59,77);  
MapMatrizPuntos[9,9] := Point(55,71);  
MapMatrizPuntos[9,10] := Point(58,62);  
MapMatrizPuntos[9,11] := Point(51,60);  
MapMatrizPuntos[9,12] := Point(44,55);  
MapMatrizPuntos[9,13] := Point(40,58);  
MapMatrizPuntos[9,14] := Point(33,58);  
MapMatrizPuntos[9,15] := Point(33,58);  
MapMatrizPuntos[9,16] := Point(33,58);  
MapMatrizPuntos[9,17] := Point(33,58);

{ Establece zona constante asociada al canal CZ }

MapMatrizPuntos[10,1] := Point(60,57);  
MapMatrizPuntos[10,2] := Point(58,62);  
MapMatrizPuntos[10,3] := Point(55,71);  
MapMatrizPuntos[10,4] := Point(59,77);  
MapMatrizPuntos[10,5] := Point(65,79);  
MapMatrizPuntos[10,6] := Point(67,83);  
MapMatrizPuntos[10,7] := Point(73,81);  
MapMatrizPuntos[10,8] := Point(81,78);  
MapMatrizPuntos[10,9] := Point(82,72);  
MapMatrizPuntos[10,10] := Point(81,67);  
MapMatrizPuntos[10,11] := Point(83,61);  
MapMatrizPuntos[10,12] := Point(77,57);  
MapMatrizPuntos[10,13] := Point(70,59);  
MapMatrizPuntos[10,14] := Point(67,61);  
MapMatrizPuntos[10,15] := Point(60,57);  
MapMatrizPuntos[10,16] := Point(60,57);

```

MapMatrizPuntos[10,17] := Point(60,57);
{ Establece zona constante asociada al canal C4 }
MapMatrizPuntos[11,1] := Point(83,61);
MapMatrizPuntos[11,2] := Point(81,67);
MapMatrizPuntos[11,3] := Point(82,72);
MapMatrizPuntos[11,4] := Point(81,78);
MapMatrizPuntos[11,5] := Point(87,82);
MapMatrizPuntos[11,6] := Point(93,86);
MapMatrizPuntos[11,7] := Point(102,84);
MapMatrizPuntos[11,8] := Point(105,78);
MapMatrizPuntos[11,9] := Point(101,73);
MapMatrizPuntos[11,10] := Point(105,65);
MapMatrizPuntos[11,11] := Point(101,59);
MapMatrizPuntos[11,12] := Point(88,56);
MapMatrizPuntos[11,13] := Point(83,61);
MapMatrizPuntos[11,14] := Point(83,61);
MapMatrizPuntos[11,15] := Point(83,61);
MapMatrizPuntos[11,16] := Point(83,61);
MapMatrizPuntos[11,17] := Point(83,61);

```

```

{ Establece zona constante asociada al canal T4 }
MapMatrizPuntos[12,1] := Point(120,88);
MapMatrizPuntos[12,2] := Point(114,83);
MapMatrizPuntos[12,3] := Point(102,84);
MapMatrizPuntos[12,4] := Point(105,78);
MapMatrizPuntos[12,5] := Point(101,73);
MapMatrizPuntos[12,6] := Point(105,65);
MapMatrizPuntos[12,7] := Point(101,59);
MapMatrizPuntos[12,8] := Point(114,62);
MapMatrizPuntos[12,9] := Point(119,51);
MapMatrizPuntos[12,10] := Point(122,60);
MapMatrizPuntos[12,11] := Point(123,65);
MapMatrizPuntos[12,12] := Point(123,78);
MapMatrizPuntos[12,13] := Point(120,88);
MapMatrizPuntos[12,14] := Point(120,88);
MapMatrizPuntos[12,15] := Point(120,88);
MapMatrizPuntos[12,16] := Point(120,88);
MapMatrizPuntos[12,17] := Point(120,88);

```

```

{ Establece zona constante asociada al canal T5 }
MapMatrizPuntos[13,1] := Point(17,90);
MapMatrizPuntos[13,2] := Point(21,84);
MapMatrizPuntos[13,3] := Point(26,86);
MapMatrizPuntos[13,4] := Point(30,82);
MapMatrizPuntos[13,5] := Point(36,78);
MapMatrizPuntos[13,6] := Point(39,86);
MapMatrizPuntos[13,7] := Point(37,91);
MapMatrizPuntos[13,8] := Point(41,95);
MapMatrizPuntos[13,9] := Point(41,103);
MapMatrizPuntos[13,10] := Point(38,107);
MapMatrizPuntos[13,11] := Point(40,113);
MapMatrizPuntos[13,12] := Point(37,114);
MapMatrizPuntos[13,13] := Point(32,110);
MapMatrizPuntos[13,14] := Point(28,106);
MapMatrizPuntos[13,15] := Point(24,101);

```

```
MapMatrizPuntos[13,16] := Point(20,95);  
MapMatrizPuntos[13,17] := Point(17,90);  
{ Establece zona constante asociada al canal P3 }  
MapMatrizPuntos[14,1] := Point(36,78);  
MapMatrizPuntos[14,2] := Point(39,88);  
MapMatrizPuntos[14,3] := Point(37,91);  
MapMatrizPuntos[14,4] := Point(41,95);  
MapMatrizPuntos[14,5] := Point(41,103);  
MapMatrizPuntos[14,6] := Point(46,106);  
MapMatrizPuntos[14,7] := Point(56,106);  
MapMatrizPuntos[14,8] := Point(60,103);  
MapMatrizPuntos[14,9] := Point(58,96);  
MapMatrizPuntos[14,10] := Point(56,90);  
MapMatrizPuntos[14,11] := Point(59,77);  
MapMatrizPuntos[14,12] := Point(53,82);  
MapMatrizPuntos[14,13] := Point(48,79);  
MapMatrizPuntos[14,14] := Point(43,81);  
MapMatrizPuntos[14,15] := Point(36,78);  
MapMatrizPuntos[14,16] := Point(36,78);  
MapMatrizPuntos[14,17] := Point(36,78);
```

```
{ Establece zona constante asociada al canal PZ }  
MapMatrizPuntos[15,1] := Point(59,77);  
MapMatrizPuntos[15,2] := Point(56,90);  
MapMatrizPuntos[15,3] := Point(58,96);  
MapMatrizPuntos[15,4] := Point(60,103);  
MapMatrizPuntos[15,5] := Point(66,107);  
MapMatrizPuntos[15,6] := Point(69,108);  
MapMatrizPuntos[15,7] := Point(76,106);  
MapMatrizPuntos[15,8] := Point(78,100);  
MapMatrizPuntos[15,9] := Point(60,93);  
MapMatrizPuntos[15,10] := Point(76,88);  
MapMatrizPuntos[15,11] := Point(81,78);  
MapMatrizPuntos[15,12] := Point(73,81);  
MapMatrizPuntos[15,13] := Point(67,83);  
MapMatrizPuntos[15,14] := Point(65,79);  
MapMatrizPuntos[15,15] := Point(59,77);  
MapMatrizPuntos[15,16] := Point(59,77);  
MapMatrizPuntos[15,17] := Point(59,77);
```

```
{ Establece zona constante asociada al canal P4 }  
MapMatrizPuntos[16,1] := Point(81,78);  
MapMatrizPuntos[16,2] := Point(76,88);  
MapMatrizPuntos[16,3] := Point(60,93);  
MapMatrizPuntos[16,4] := Point(78,100);  
MapMatrizPuntos[16,5] := Point(66,104);  
MapMatrizPuntos[16,6] := Point(90,105);  
MapMatrizPuntos[16,7] := Point(95,101);  
MapMatrizPuntos[16,8] := Point(99,94);  
MapMatrizPuntos[16,9] := Point(105,88);  
MapMatrizPuntos[16,10] := Point(102,84);  
MapMatrizPuntos[16,11] := Point(93,86);  
MapMatrizPuntos[16,12] := Point(87,82);  
MapMatrizPuntos[16,13] := Point(81,78);  
MapMatrizPuntos[16,14] := Point(81,78);
```

MapMatrizPuntos[16,15] := Point(81,78);  
 MapMatrizPuntos[16,16] := Point(81,78);  
 MapMatrizPuntos[16,17] := Point(81,78);

{ Establece zona constante asociada al canal T6 }

MapMatrizPuntos[17,1] := Point(102,112);  
 MapMatrizPuntos[17,2] := Point(95,101);  
 MapMatrizPuntos[17,3] := Point(99,94);  
 MapMatrizPuntos[17,4] := Point(105,86);  
 MapMatrizPuntos[17,5] := Point(102,84);  
 MapMatrizPuntos[17,6] := Point(114,83);  
 MapMatrizPuntos[17,7] := Point(120,86);  
 MapMatrizPuntos[17,8] := Point(117,95);  
 MapMatrizPuntos[17,9] := Point(113,101);  
 MapMatrizPuntos[17,10] := Point(108,107);  
 MapMatrizPuntos[17,11] := Point(102,112);  
 MapMatrizPuntos[17,12] := Point(102,112);  
 MapMatrizPuntos[17,13] := Point(102,112);  
 MapMatrizPuntos[17,14] := Point(102,112);  
 MapMatrizPuntos[17,15] := Point(102,112);  
 MapMatrizPuntos[17,16] := Point(102,112);  
 MapMatrizPuntos[17,17] := Point(102,112);

{ Establece zona constante asociada al canal O1 }

MapMatrizPuntos[18,1] := Point(37,114);  
 MapMatrizPuntos[18,2] := Point(40,113);  
 MapMatrizPuntos[18,3] := Point(38,107);  
 MapMatrizPuntos[18,4] := Point(41,103);  
 MapMatrizPuntos[18,5] := Point(48,106);  
 MapMatrizPuntos[18,6] := Point(56,106);  
 MapMatrizPuntos[18,7] := Point(60,103);  
 MapMatrizPuntos[18,8] := Point(66,107);  
 MapMatrizPuntos[18,9] := Point(69,108);  
 MapMatrizPuntos[18,10] := Point(69,116);  
 MapMatrizPuntos[18,11] := Point(73,124);  
 MapMatrizPuntos[18,12] := Point(63,124);  
 MapMatrizPuntos[18,13] := Point(54,122);  
 MapMatrizPuntos[18,14] := Point(46,119);  
 MapMatrizPuntos[18,15] := Point(37,114);  
 MapMatrizPuntos[18,16] := Point(37,114);  
 MapMatrizPuntos[18,17] := Point(37,114);

{ Establece zona constante asociada al canal O2 }

MapMatrizPuntos[19,1] := Point(73,124);  
 MapMatrizPuntos[19,2] := Point(69,116);  
 MapMatrizPuntos[19,3] := Point(69,108);  
 MapMatrizPuntos[19,4] := Point(76,106);  
 MapMatrizPuntos[19,5] := Point(78,100);  
 MapMatrizPuntos[19,6] := Point(86,104);  
 MapMatrizPuntos[19,7] := Point(90,105);  
 MapMatrizPuntos[19,8] := Point(95,101);  
 MapMatrizPuntos[19,9] := Point(102,112);  
 MapMatrizPuntos[19,10] := Point(97,116);  
 MapMatrizPuntos[19,11] := Point(90,120);  
 MapMatrizPuntos[19,12] := Point(83,122);  
 MapMatrizPuntos[19,13] := Point(73,124);

```

MapMatrizPuntos[19,14] := Point(73,124);
MapMatrizPuntos[19,15] := Point(73,124);
MapMatrizPuntos[19,16] := Point(73,124);
MapMatrizPuntos[19,17] := Point(73,124);
end;

```

```

procedure TMapMainForm.EstableceEscalaColores;
begin

```

```

{ Establece escala con colores vs amplitudes }

```

```

MapColores[1].Color := clAqua;
MapColores[1].Amplitud := 5;
MapColores[2].Color := clLime;
MapColores[2].Amplitud := 10;
MapColores[3].Color := clYellow;
MapColores[3].Amplitud := 15;
MapColores[4].Color := $000080FF;
MapColores[4].Amplitud := 20;
MapColores[5].Color := clRed;
MapColores[5].Amplitud := 25;
MapColores[6].Color := clFuchsia;
MapColores[6].Amplitud := 30;
MapColores[7].Color := $008000FF;
MapColores[7].Amplitud := 35;
MapColores[8].Color := clBlue;
MapColores[8].Amplitud := 40;
MapColores[9].Color := $00FF0080;
MapColores[9].Amplitud := 45;
MapColores[10].Color := $00004080;
MapColores[10].Amplitud := 50;
MapColores[11].Color := clGreen;
MapColores[11].Amplitud := 55;
MapColores[12].Color := $00408080;
MapColores[12].Amplitud := 60;
MapColores[13].Color := $00808040;
MapColores[13].Amplitud := 75;
MapColores[14].Color := $00828282;
MapColores[14].Amplitud := 100;
MapColores[15].Color := clMaroon;
MapColores[15].Amplitud := 125;
MapColores[16].Color := clPurple;
MapColores[16].Amplitud := 150;
MapColores[17].Color := clNavy;
MapColores[17].Amplitud := 175;
MapColores[18].Color := clBlack;
MapColores[18].Amplitud := 200;

```

```

end;

```

```

procedure TMapMainForm.InicializaTabla;
begin

```

```

{ Inicializa tabla formada por:

```

	Frecuencia inferior	Frecuencia superior	Suma de Amplitudes	N-mero de ocurrencias
(ALFA) 1	8	13	0	0
(BETA) 2	13	30	0	0
(DELTA) 3	0.5	4	0	0
(TETA) 4	4	8	0	0

```
MapTabla[1,1] := 8;
MapTabla[1,2] := 13;
MapTabla[1,3] := 0;
MapTabla[1,4] := 0;
MapTabla[2,1] := 13;
MapTabla[2,2] := 30;
MapTabla[2,3] := 0;
MapTabla[2,4] := 0;
MapTabla[3,1] := 0.5;
MapTabla[3,2] := 4;
MapTabla[3,3] := 0;
MapTabla[3,4] := 0;
MapTabla[4,1] := 4;
MapTabla[4,2] := 8;
MapTabla[4,3] := 0;
MapTabla[4,4] := 0;
end;

procedure TMapMainForm.LlenaCabezaAlfa(Color : TColor; Canal : Integer);
begin
  { Colorea un área específica para el ritmo Alfa }
  with MapCabezaAlfa.Canvas do
    begin
      Pen.Color := Color;
      Brush.Style := bsSolid;
      Brush.Color := Color;
      Polygon(MapMatrizPuntos[Canal]);
    end;
  end;

procedure TMapMainForm.LlenaCabezaBeta(Color : TColor; Canal : Integer);
begin
  { Colorea un área específica para el ritmo Beta }
  with MapCabezaBeta.Canvas do
    begin
      Pen.Color := Color;
      Brush.Style := bsSolid;
      Brush.Color := Color;
      Polygon(MapMatrizPuntos[Canal]);
    end;
  end;

procedure TMapMainForm.LlenaCabezaDelta(Color : TColor; Canal : Integer);
begin
  { Colorea un área específica para el ritmo Delta }
  with MapCabezaDelta.Canvas do
    begin
      Pen.Color := Color;
      Brush.Style := bsSolid;
      Brush.Color := Color;
      Polygon(MapMatrizPuntos[Canal]);
    end;
  end;

procedure TMapMainForm.LlenaCabezaTeta(Color : TColor; Canal : Integer);
begin
```

```

{ Colorea un área específica para el ritmo Teta }
with MapCabezaTeta.Canvas do
begin
    Pen.Color := Color;
    Brush.Style := bsSolid;
    Brush.Color := Color;
    Polygon(MapMatrizPuntos[Canal]);
end;
end;

procedure TMapMainForm.FormCreate(Sender: TObject);
var
    n, i, k : Integer;
    PunteroReales : File of Real;
    NumReal, FrecuenciaHz, IncrementoHz, AmplitudProm : Real;
    Archivo : String;
    Extension : String[4];
    RegActual, TamArchivo : Longint;
begin
    EstableceZonaPorCanal;
    EstableceEscalaColores;

    { Obtiene el no. de estudio que se desea visualizar }
    Archivo := MainForm.OpenDialog.FileName;

    { Revisa si ya se realiz% análisis espectral para todos los
    canales activos y de no ser así, envYa error }
    for n := 1 to NumElectrodos do
    begin
        if (MaskActivos[n].Activo = True) then
        begin
            Extension := '.P' + MaskActivos[n].NomInt;
            Archivo := ChangeFileExt(Archivo, Extension);
            if not FileExists(Archivo) then
            begin
                Application.MessageBox('Para realizar el mapeo EEG '+'
                'debe haber realizado y alma'+
                'cenado antes el análisis esp'+
                'ectral para todos los canales '+'
                'activos.', 'Error', MB_OK);
            end;
        end;
    end;
end;

{ Barre matriz y aplica mapeo EEG s%lo con electrodos activos }
for n := 1 to NumElectrodos do
begin
    if (MaskActivos[n].Activo = True) then
    begin
        InicializaTabla;
        FrecuenciaHz := 0;
        RegActual := 0;
        { Lee archivo con análisis espectral para el canal n }
        Extension := '.P' + MaskActivos[n].NomInt;
    end;
end;

```

```

Archivo := ChangeFileExt(Archivo, Extension);
AssignFile(PunteroReales, Archivo);
Reset(PunteroReales);
{ Determina el incremento en Hz entre punto y punto }
TamArchivo := FileSize(PunteroReales);
IncrementoHz := 100 / TamArchivo;
while not Eof(PunteroReales) do
begin
  Read(PunteroReales, NumReal);
  FrecuenciaHz := FrecuenciaHz + IncrementoHz;
  { Clasifica los ritmos que va leyendo }
  for i := 1 to MaxRitmos do
    if (FrecuenciaHz >= MapTabla[i,1]) and
      (FrecuenciaHz < MapTabla[i,2]) then
      begin
        MapTabla[i,3] := MapTabla[i,3] + NumReal;
        MapTabla[i,4] := MapTabla[i,4] + 1;
        break;
      end;
    RegActual := RegActual + 1;
    Seek(PunteroReales, RegActual);
  end;
  CloseFile(PunteroReales);
  { Calcula el promedio de amplitud para cada ritmo y en base
  a Uste le asigna un color a cada ritmo }
  for i := 1 to MaxRitmos do
  begin
    if (MapTabla[i,4] <> 0) then
      AmplitudProm := MapTabla[i,3] / MapTabla[i,4]
    else
      continue;
    for k := 1 to MaxColores do
      if (AmplitudProm <= MapColores[k].Amplitud) then
      begin
        case i of
          1 : LlenaCabezaAlfa(MapColores[k].Color, n);
          2 : LlenaCabezaBeta(MapColores[k].Color, n);
          3 : LlenaCabezaDelta(MapColores[k].Color, n);
          4 : LlenaCabezaTeta(MapColores[k].Color, n);
        end;
        break;
      end;
    end;
  end;
end;
end;
end;

procedure TMapMainForm.FormClose(Sender: TObject;
var Action: TCloseAction);
begin
  Action:=caFree;
end;

end.{Unit MapMain}

```

```
{***Manejo de las funciones asociadas a la adquisición de la señal}
```

```
unit InTdaq;
```

```
interface
```

```
Uses Main;
```

```
const
```

```
VMinimo = 0.0024414; { Escal%ñ mYnimo de voltaje (2.4414 mV) }
Umbral = 0.003; { Arriba de este valor se considera que hay se%ñal }
NumDisp = 1; { N-mero de dispositivo asignado a la tarjeta DAQ }
ModoEntrada = 2; { Indica que la se%ñal tiene su propia referencia }
RangoEntrada = 10; { DE 0 a 10 Volts para se%ñal unipolar }
Polaridad = 1; { Indica se%ñal de tipo unipolar }
TipoDisparo = 0; { Disparo vYa software }
RelojExt = 0; { Fuente de pulsos de reloj = interna (el de la tarjeta) }
ModoDB = 1; { Modo de adquisici%ñ de datos en doble buffer habilitado }
Unidades = 0; { Indica muestras / segundo }
Ganancia = 50; { Ganancia de voltaje aplicada por la tarjeta DAQ }
NumDeMuestras = 100; { N-mero de muestras a tomar en cada adquisici%ñ }
VelMuestreo = 100; { Frecuencia de muestreo en muestras/s }
```

```
type
```

```
 PDouble = ^Double;
 PInteger = ^Integer;
 PLongInt = ^LongInt;
 PByte = ^PByte;
```

```
var
```

```
 BufferDAQ : Array[1..MaxChannels*NumDeMuestras*2] of Integer;
```

```
function ConfigEntAnalogica(xDevice : Integer) :integer;
```

```
function ConfigTipoMuestreo(xDevice : Integer) : Integer;
```

```
function InicioDAQ(xDevice : Integer; xRate : Double) : integer;
```

```
function Transferir(xDevice : Integer; xPBuffer : PInteger) : Integer;
```

```
function DetenerDAQ(xDevice : Integer) : Integer;
```

```
function LeeValor(xDevice : Integer; voltaje : PDouble) : integer;
```

```
Procedure DisplayDAQError(NumError : Integer);
```

```
implementation
```

```
Uses SysUtils,Forms,WinTypes;
```

```
{SI wdaq_tp.inc}
```

```
function ConfigEntAnalogica(xDevice : Integer) :integer;
```

```
begin
```

```
 ConfigEntAnalogica:=AI_Configure(xDevice,-1,ModoEntrada,RangoEntrada,
 Polaridad,0);
```

```

end;

function ConfigTipoMuestreo(xDevice : Integer) : Integer;
var
  status : Integer;
begin
  Status:=DAQ_Config(xDevice,TipoDisparo,RelojExt);
  if Status = 0 Then
    Status:=DAQ_DB_Config(xDevice,ModoDB);
  ConfigTipoMuestreo:=Status;
end;

function InicioDAQ(xDevice : Integer; xRate : Double) : Integer;
var
  Status      : Integer;
  xTimeBase   : Integer;
  xSampleInterval : Word;
  xNumChans   : Integer;
  xChanVector : Array[0..MaxChannels-1] of Integer;
  xgainVector : Array[0..MaxChannels-1] of Integer;
  k           : Integer;
  NumConverts : Longint;
begin
  Status:=DAQ_Rate(xRate,Unidades,xTimeBase,xSampleInterval);
  if status = 0 Then
    begin
      xNumChans:=0;
      for k:=0 to MaxChannels-1 Do
        begin
          xChanVector[k]:=0;
          xGainVector[k]:=0;
        end;
      For k:=1 To NumElectrodos Do
        begin
          if MaskActivos[k].Disponibile and MaskActivos[k].Activo Then
            begin
              xNumChans:=xNumChans+1;
              xChanVector[k-1]:=k-1;
              xGainVector[k-1]:=Ganancia;
            end;
          end;
        Status:=SCAN_Setup(xDevice,xNumChans,@xChanVector[0],@xGainVector[0]);
        if Status = 0 Then
          begin
            NumConverts:=xNumChans*NumDeMuestras;
            Status:=SCAN_Start(xDevice,@bufferDAQ[1],NumConverts,
              xTimeBase,xSampleInterval,0,0);
          end;
        end;
      InicioDAQ:=Status;
    end;

function Transferir(xDevice : Integer; xPBuffer : PInteger) : Integer;
  Var

```

```

Status,
DatosOK,
xDAQStopped : Integer;
K,
PtsTfr      : LongInt;

begin
Repeat
Status:=DAQ_DB_HalfReady(xDevice, DatosOK, xDAQStopped);
if (xDAQStopped = 1) or (Status <= 0) Then
break;
Until DatosOK = 1;
if xDAQStopped = 0 Then
Transferir:=DAQ_DB_Transfer(xDevice,@BufferDAQ[1],PtsTfr,xDAQStopped)
else
Transferir:=Status;
For k:=1 to PtsTfr Do
BufferDAQ[k]:=BufferDAQ[k] shr 4;
end;

function DetenerDAQ(xDevice : Integer) : Integer;
begin
DetenerDAQ:=DAQ_Clear(xDevice);
end;

function LeeValor(xDevice : Integer; Voltaje : PDouble) : integer;
Var
status : Integer;
xChanVector : Array[0..MaxChannels-1] of Integer;
xgainVector : Array[0..MaxChannels-1] of Integer;
k : Integer;

begin
For k:=0 to MaxChannels-1 Do
begin
xChanVector[k]:=k;
xgainVector[k]:=Ganancia;
end;
Status:=SCAN_Setup(xDevice,MaxChannels,@xChanVector[0],@xGainVector[0]);
if Status = 0 Then
Status:=AI_VRead_Scan(xDevice,Voltaje);
LeeValor:=Status;
end;

Procedure DisplayDAQError(NumError : Integer);
var
stringError : Array[0..255] of Char;

begin
strPCopy(StringError,'Error de interfaz DAQ No. '+intostr(NumError)+' ');
Application.MessageBox(stringError,'Error DAQ',MB_OK);
end;

end.{Unit IntDaq}

```

---

**(\*\*\*Declaraciones de las funciones DAQ usadas en el sistema)**


---

```
{
*
*   NI-DAQ para Windows - Prototipos de Funciones
*
*.....}
```

```
function AI_Configure (
    device,
    chan,
    inputMode,
    inputRange,
    polarity,
    driveAIS : Integer) :
integer; far; external 'nidaq';

function AI_VRead_Scan (
    device : Integer;
    reading : PDouble) :
integer; far; external 'nidaq';

function DAQ_Clear (
    device : Integer) :
integer; far; external 'nidaq';

function DAQ_Config (
    device,
    startTrig,
    extConv : Integer) :
integer; far; external 'nidaq';

function DAQ_DB_Config (
    device,
    dbMode : Integer) :
integer; far; external 'nidaq';

function DAQ_DB_HalfReady (
    device : Integer;
    var halfReady,
    status : Integer) :
integer; far; external 'nidaq';

function DAQ_DB_Transfer (
    device : Integer;
    hbuffer : Pointer;
    var ptsTfr : Longint;
    var status : Integer) :
integer; far; external 'nidaq';

function DAQ_Rate (
    rate : Double;
    units : Integer;
    var timebase : Integer;
    var sampleInt : Word) :
integer; far; external 'nidaq';
```

```
function SCAN_Setup(  
    device,  
    num_chans : Integer;  
    chans,  
    gains : PInteger) :  
integer; far; external 'nidaq';  
  
function SCAN_Start (  
    device : Integer;  
    buffer : Pointer;  
    cnt : Longint;  
    tb1 : Integer;  
    si1 : Word;  
    tb2 : Integer;  
    si2 : Word) :  
integer; far; external 'nidaq';  
{Fin de declaraciones de prototipos de funciones DAQ}
```