

90  
2 Ejen.



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

DISEÑO E IMPLANTACION DE UN SISTEMA  
DE ANALISIS DE SEÑALES BIOLÓGICAS  
EN CRONOBIOLOGIA

T E S I S  
QUE PARA OBTENER EL TITULO DE  
INGENIERO EN COMPUTACION  
P R E S E N T A:  
ARTURO VEGA GONZALEZ

DIRECTORES DE TESIS:

DR. RAUL AGUILAR ROBLERO

DR. FRANCISCO GARCIA UGALDE



CIUDAD UNIVERSITARIA, 1994

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 está 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.

La presente tesis se llevó a cabo bajo el apoyo del proyecto de DGAPA IN202891 y el Instituto de Fisiología Celular.

Dedico la presente tesis a mis padres por el apoyo, confianza y comprensión que me han brindado en todos estos años.

GRACIAS.

# Índice

<b>I Introducción</b>	
I.1 Aspectos generales de cronobiología	1
I.2 Osciladores físicos	1
I.3 Osciladores biológicos	4
I.4 Problemas generales en el estudio de los ritmos biológicos	8
<b>II Fundamentos de Procesamiento de Señales</b>	<b>13</b>
II.1 Secuencia	14
II.2 La función de muestreo	15
II.3 Uso de la función de muestreo	17
II.4 El espectro de la señal de muestreo	18
II.5 La relación de espectros de las señales en tiempo continuo y discreto	19
II.6 Señales discretas especiales	
II.6.1 Señal impulso unitario	20
II.6.2 Secuencia escalón unitario	21
II.7 El concepto de frecuencia	
II.7.1 Señales sinusoidales en tiempo continuo	22
II.7.2 Señales sinusoidales en tiempo discreto	23
II.8 Exponenciales complejas relacionadas armónicamente	24
II.9 Probabilidad y procesos aleatorios	
II.9.1 Probabilidad básica	25
II.9.2 Variables aleatorias	27
II.10 Cuantización de secuencias	30
II.11 Procesos Aleatorios, autocorrelación y densidad espectral	31
II.12 Transformada discreta de Fourier	34
II.12.1 Espectro de potencia	37
<b>III Especificación del sistema</b>	
III.1 Conceptos básicos del diseño orientado a objetos (DOO)	39
III.1.1 Representación gráfica para el DOO	43
III.2 Definición del problema y objetivo	45
III.3 Entorno del sistema	47
III.4 Diseño del sistema propuesto	49
III.4.1 Descripción de la información	49
III.4.2 Objetos, atributos y operaciones	51
III.4.3 Descripción funcional del sistema	62
<b>IV Elementos del sistema</b>	
IV.1 Características y requerimientos	65
IV.3 Configuración del sistema	66
IV.4 Organización del sistema	67
IV.5 Despliegue de resultados	68
IV.6 Utilerías externas	69

<b>V Métodos implantados</b>	
<b>V.1 Actograma</b>	71
<b>V.2 Determinación del periodo</b>	73
V.2.1 Estimación gráfica del periodo	74
V.2.2 Estimación gráfica de la fase	75
V.2.3 Cosinor	77
V.2.4 Periodograma	79
V.2.5 Demodulación compleja	82
V.2.6 Poincaré	85
V.2.7 Densidad espectral	87
<b>V.3 Breve comparación entre los metodos implantados</b>	89
<b>Conclusiones</b>	91
<b>Bibliografía</b>	93
<b>Apéndice</b> Breve Manual de Usuario	i

# I INTRODUCCIÓN

## I.1 Aspectos Generales de Cronobiología

"Cronobiología" es el estudio de la organización temporal de los organismos. La organización temporal puede definirse como la resultante de los aspectos temporales predictivos (no aleatorios), de las funciones de los organismos, esta organización temporal los caracteriza individualmente y poblacionalmente.

No se tiene que observar demasiados organismos vivientes, para saber que varias conductas están organizadas temporalmente en patrones periódicos, de hecho si tenemos que decidir cual es la característica más conspicua de un ecosistema natural, podríamos decir que es la periodicidad diaria y estacional y la consecuente organización temporal de la estructura del nicho, la cadena alimenticia y la conducta. De hecho la conducta rítmica de organismos en sus hábitats naturales, es seguramente una de las características más obvias de cualquier ecosistema.

Por otro lado tenemos que la recurrencia de cualquier evento dentro de un sistema biológico a intervalos mas o menos regulares puede ser considerado como un ritmo biológico (Kalmus, 1935). La noción de un ritmo puede ser demasiado vaga si no la definimos en términos de física. Los ritmos pueden ser distinguidos acorde a:

- 1) una característica como la frecuencia;
- 2) el sistema biológico (por ejemplo una población), en el cual el ritmo es observado;
- 3) El tipo de proceso que genera el ritmo;
- 4) las funciones que el ritmo cubre.

## I.2 Osciladores Físicos

El lenguaje de osciladores desde el punto de vista de la física y la aplicación de estos a los ciclos biológicos no es familiar para la mayoría de la gente, por lo que es necesario una aproximación desde el punto de vista de la física antes que desde el punto de vista de la biología. Los términos y conceptos se definirán con respecto a una situación especial, la conducta de un simple oscilador físico. Los términos y conceptos de física serán generalizados para la aplicación a la biología, tal como es costumbre.

Por ejemplo, como un sistema modelo considere la conducta de un resorte, el cual por uno de sus extremos a una superficie rígida y por el otro tiene una masa, si se ignora la fricción, una vez que comience la oscilación continuará a menos que sea perturbada por una fuerza externa, en la figura 1A se pueden observar varias posiciones de la masa ocurriendo a diferentes tiempos.

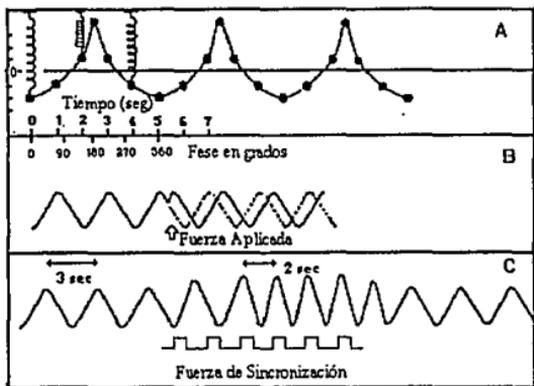


Figura 1.1 Oscilador

La posición cero, en el eje de las ordenadas, representa la posición de descanso para la masa, la cual es desplazada en el tiempo cero y liberada. La masa entonces se desplaza a una posición donde alcanza una velocidad de cero y cambia de dirección, regresando a su posición original de donde se liberó, esto completa una oscilación y entonces continua de una forma indefinida, esta oscilación recurrente comprende o forma un ritmo y la curva de un ritmo puede obtenerse graficando la posición de la masa como función del tiempo.

La unidad de repetición del ritmo es referida como ciclo y la longitud del tiempo requerido para completar un ciclo es referido como periodo, para el ejemplo mostrado el periodo del ritmo es de 5 segundos.

Un ciclo puede estar normalizado para ocurrir en  $360^\circ$ . Esto después de seleccionar un punto arbitrariamente como el grado cero, podemos referirnos a un punto en el ciclo, a  $90^\circ$  por ejemplo. La fase se refiere a un punto específico de un ciclo esto es se refiere a un estado instantáneo de la oscilación.

Seleccionando el punto inicial de cero grados  $0^\circ$  como lo muestra la figura 1.1A tenemos que el máximo desplazamiento ocurre a los  $180^\circ$ , generalmente la amplitud se refiere al máximo desplazamiento de la masa, desde el cero ó la posición media.

Los ritmos pueden desplazarse en tiempo sin cambiar las características básicas de la oscilación (ver figura 1.1B), este desplazamiento es llamado corrimiento de fase y en el ejemplo la oscilación ó ritmo ha sido recorrida  $90^\circ$  y todos los puntos posteriores han sido recorridos  $90^\circ$ .

Los corrimientos de fase pueden producirse por la aplicación de una fuerza externa a un punto particular de la fase en el ciclo, para un tiempo dado y pueden ocurrir en ambas direcciones y en cualquier cantidad, dependiendo del tipo de fuerza aplicada y en que fase del ciclo es aplicada. Obviamente un corrimiento de fase en el ciclo se realiza por un incremento temporal en la velocidad del oscilador (avance de fase) o un decremento (retraso de fase) para un cierto tiempo. Esto generalmente, toma algunos ciclos para que el oscilador vuelva a su estado estable (donde el periodo permanece constante de un ciclo al próximo), después de la aplicación de la fuerza externa. Los ciclos entre los estados estables de antes y después de la aplicación de la fuerza externa son referidos como transitorios.

Para nuestro caso donde no existe fuerza de fricción, el oscilador posee un cierto periodo natural para un desplazamiento dado en ausencia de fuerzas externas, en estas condiciones se dice que oscila libremente (oscilación espontánea) y exhibiendo un periodo en ese estado, el cual es llamado periodo de oscilación espontánea.

Si aplicamos una fuerza externa capaz de modificar la oscilación, con un periodo diferente al del oscilador, entonces el oscilador exhibirá el periodo de la fuerza externa aplicada, a este fenómeno se le llama sincronización, es decir la fuerza externa sincroniza al oscilador.

El oscilador manteniendo su periodo de oscilación espontánea lo cambiará por el periodo de la fuerza externa ó fuerza de sincronización. En adición a esto, se establece una relación fija de fase es establecida entre el oscilador y la fuerza aplicada durante la sincronización.

En la figura 1.1C, la curva inicial representa una oscilación con un periodo de oscilación espontánea de 3 segundos, después se aplica una fuerza externa con un periodo de 2 segundos, la cual se representa en la curva inferior, durante la aplicación de la fuerza, el oscilador tiende a tener el periodo de la fuerza externa, estos ciclos que ocurren hasta que esto suceda son los transitorios, una vez que alcanza el estado estable con el periodo de la fuerza externa, (note que hay una relación fija de fase entre el oscilador y la fuerza externa periódica). Cuando la fuerza se deja de aplicar el oscilador regresa a su periodo de oscilación espontánea, después de algunos ciclos, los cuales son llamados transitorios.

Estos osciladores son controlables, pero solo dentro de ciertos límites determinados por sus características físicas y las características de la fuerza aplicada.

La fuerza de sincronización puede ser aplicada con diferentes longitudes de periodo, duración y amplitudes, pero para un oscilador dado solo puede ser sincronizado dentro de un rango fijo de periodos dependiendo de los parámetros de la fuerza aplicada y el oscilador.

### I.3 Osciladores Biológicos

La ritmicidad es una propiedad fundamental de la materia viva, desde los organismos unicelulares hasta los animales complejos, innumerables estructuras y funciones exhiben cambios periódicos de sus formas e intensidades. Los ritmos biológicos se extienden sobre una amplia gama de frecuencias, desde un ciclo por segundo hasta un ciclo por varios años. Estos pueden ser observados en simples células, en redes de tejido y órganos, en el organismo completo, o en poblaciones. Los ritmos biológicos pueden ser la respuesta del organismo a una influencia periódica ambiental (Ritmo exógeno), o bien pueden ser originadas internamente por el organismo (Ritmo endógeno).

Como ya se mencionó anteriormente existen muchos ritmos en los diferentes niveles de organización en organismos multicelulares, sin embargo no todos son de interés y por lo tanto no todos han sido descritos o medidos, algunos de los más conocidos son: de actividad, de temperatura corporal, de ingesta de agua, de la tasa metabólica, de la actividad eléctrica del sistema nervioso, de concentración de varios constituyentes de la sangre, por mencionar algunos. Para un individuo, muchos ritmos están presentes simultáneamente, pero es importante notar que estos ritmos pueden tener diferentes relaciones de fase con el medio ambiente.

Otro tipo de ritmo que ha sido explorado, es el ritmo de la sensibilidad a lo largo del día de organismos intactos para varios agentes biológicos de interés (Halberg 1960). Una dosis de un insecticida tendrá diferentes efectos dependiendo de la hora de administración. Una cantidad dada de un insecticida matará alrededor de un 80% de un grupo de ratones cuando es administrado al inicio de la tarde, pero tendrá pocos efectos cuando es administrado tan solo 8 horas después. Los ritmos de sensibilidad son también conocidos en humanos para algunos agentes farmacéuticos comunes. Estos ritmos de sensibilidad o ritmos de susceptibilidad son el reflejo del hecho de que los estados fisiológicos y bioquímicos de un organismo varían a lo largo del día.

Dentro de los muchos ritmos biológicos que existen en los organismos, los más prominentes son los circádicos. Estos ritmos están estrechamente relacionados con la rotación de la tierra alrededor de su eje, es decir, el ciclo solar de luz y oscuridad. La expresión "es tan seguro que la noche siga después del día", refleja la estabilidad de ciertos ciclos en nuestro medio ambiente. La tierra gira sobre su propio eje aproximadamente cada 24 horas, a consecuencia del movimiento de rotación de la tierra, los seres vivos que habitan las regiones superficiales, (plantas y animales) están expuestos a ritmos de luz y temperatura. La disponibilidad de comida y la actividad de predadores se ven afectados por estas variaciones periódicas. Para algunos esto no es sorprendente ya que se descubrió que la conducta de muchos organismos siguen un esquema de 24 horas.

En los mamíferos, los ritmos circádicos expresan una organización temporal de procesos fisiológicos y conductuales para una óptima adaptación al medio ambiente. Esto es porque los ritmos circádicos ayudan a sincronizar al organismo con las fluctuaciones periódicas en el medio externo, y facilitan la integración del medio interno del individuo.

La explicación más obvia para los ritmos de 24 horas es que plantas y animales responden pasivamente a los ciclos de nuestro medio ambiente, sin embargo cuando un organismo es aislado de todo el medio ambiente y por supuesto de las claves de tiempo, - cuando la luz, comida, temperatura y sonido se mantienen constantes -, la mayoría de estos ritmos persiste con un periodo independiente. El ritmo expresado bajo estas condiciones se dice que es un ritmo en oscilación espontánea (*free-running*), el cual es similar al ritmo natural de nuestro oscilador físico en ausencia de fuerzas externas, el periodo que exhibe bajo estas condiciones se denomina periodo de oscilación espontánea (*free-running*) y se indica por medio de la letra  $\tau$ , el cual es cercano a las 24 horas. Este fenómeno demuestra la naturaleza endógena de los ritmos circádicos.

A pesar de la naturaleza endógena de los ritmos circádicos, los marcapasos endógenos se ajustan constantemente por el ciclo de luz-obscuridad (LO). Este proceso se llama sincronización. Las vías visuales son las principales responsables del proceso de la sincronización. Podemos decir entonces que la luz es similar a la fuerza externa de nuestro modelo. Las señales periódicas de luz son uno de los agentes sincronizadores más importantes, ya que pueden sincronizar virtualmente a todos los organismos.

Al aplicar pulsos de luz podemos cambiar la fase del ritmo, figura 1.2, sin embargo en el proceso de sincronización una característica importante es que un ritmo circádico en corrimiento espontáneo puede alargarse o acortarse su periodo en uno o varios ciclos, dependiendo de la fase en donde se presenten los pulsos de luz.

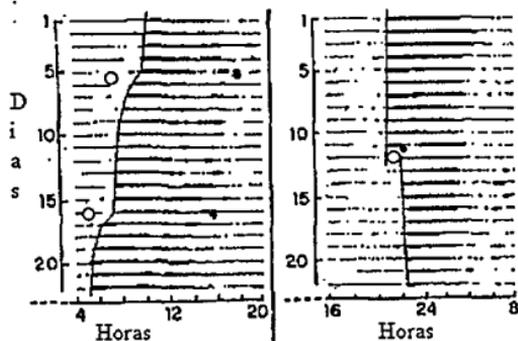


Figura 1.2 Actogramas donde se observa un avance de fase (izq.), y un retraso de fase (der.), estos ocurrieron al aplicar pulsos de luz.

La magnitud y la dirección del cambio de fase depende de la fase del ritmo en la cual se aplica el estímulo luminoso. Esta relación se gráfica en la curva de respuesta de fase, como se ilustra en la figura 1.3.

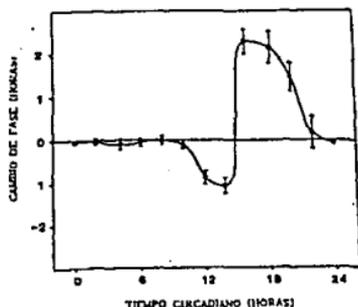


Figura 1.3 Curva de Respuesta de Fase

En hámsters, el comienzo de la actividad se usa como referencia para la fase y se define como tiempo circádico (TC) 12:00. El hámster es activo durante la noche subjetiva (de 12:00 a 24:00 TC) e inactivo durante el día subjetivo (de 0:00 a 12:00 TC).

La curva de respuesta de fase tiene tres características generales : 1) La zona muerta, cuando no ocurren cambios de fase después de que se aplican los pulsos luminosos durante el día subjetivo, 2) El retraso de fase, ocurre cuando los pulsos se aplican durante la noche subjetiva temprana y 3) el avance de fase, cuando los pulsos se aplican durante la noche subjetiva tardía. El retraso de fase causa un alargamiento del periodo y el avance de fase causa un acortamiento del periodo.

Aunque la curva de respuesta de fase se obtiene por mediciones de la función rítmica expresada, esta refleja las propiedades temporales de los osciladores responsables de esta función. En otras palabras las mediciones se obtienen del ritmo expresado por los efectores, sin embargo, estos efectores, como su nombre lo indica, no son el oscilador pero por medio de ellos podemos inferir las propiedades temporales del oscilador, que también recibe el nombre de marcapasos.

La representación clásica de una serie de tiempo en cronobiología es por medio de la gráfica llamada actograma en *double-plot*, como lo muestra la figura 1.4, donde se puede apreciar fácilmente el periodo de la actividad (alfa).

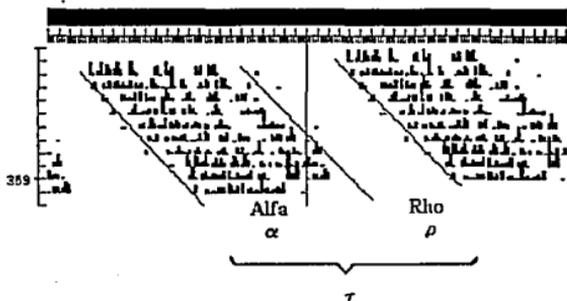


FIG. 1.4 Actograma en Double-Plot

Un ciclo conductual esta compuesto por dos fracciones, una de actividad llamada alfa ( $\alpha$ ) y una de reposo llamada rho ( $\rho$ ) en la figura 1.4 se observa un ritmo circádico de ingesta de agua en rata en corrimiento espontáneo y se pueden distinguir claramente ambas fracciones, donde para cada ciclo es evidente que  $\tau = \alpha + \rho$ .

Los ritmos circádicos se dice son el reflejo de la funcionalidad de un reloj biológico, este es llamado "biológico" porque es generado dentro del organismo y es porque mide el tiempo en una forma periódica, algunas veces con sorprendente precisión, con periodos cercanos a las 24 horas.

Por otro lado tenemos que este reloj biológico tiene relativa independencia de la temperatura , es decir la longitud del periodo de un ritmo circádico a diferentes temperaturas permanece cercano al mismo valor, entonces podemos decir que la velocidad de los relojes biológicos en los organismos estudiados no esta regulada por la temperatura.

De todo lo anterior podemos concluir que la organización temporal endógena es seguramente, mediada por diferentes órganos y procesos bioquímicos, sin embargo, algunos aspectos funcionales parecen ser universales, como ;

- 1 ) Una periodicidad espontanea cercana a las 24 horas
- 2 ) Relativa independencia de la temperatura.
- 3 ) Persistencia del ritmo aún en pequeñas unidades de función biológica (células)

4 ) Inmunidad a muchos tipos de perturbaciones químicas.

5 ) Sensibilidad a la luz visible de un color. La luz puede suprimir la oscilación en algunos organismos, dando un breve pulso de luz podemos poner en fase el ritmo, dando pulsos periódicos se puede sincronizar el ritmo, si el periodo de los pulsos no es muy diferente del periodo natural del ritmo.

## **I.4 Problemas Generales en el Estudio de los Ritmos Biológicos**

Los problemas en el estudio de los ritmos biológicos los podemos dividir en tres grandes áreas, de acuerdo al tipo de problema, estas áreas son:

- 1- Análisis fisiológico de la ritmicidad
- 2- Análisis matemático de la ritmicidad
- 3- Captura de la información ( colección de datos a analizar)

El que los diversos ritmos circádicos presenten propiedades comunes, ha sugerido que, desde el punto fisiológico, obedezcan a mecanismos también comunes. De hecho, los problemas fundamentales en el estudio fisiológico de los ritmos circádicos, puede resumirse de la siguiente manera:

- 1- Localización del oscilador biológico que rige la ritmicidad circádica, y caracterización de su organización funcional.
- 2- Análisis de los mecanismos de comunicación entre el oscilador circádico, y los efectores comandados por él, y
- 3- Análisis de los mecanismos de sincronización entre el oscilador y el medio ambiental.

El análisis matemático de los ritmos circádicos ayuda a resolver algunos problemas en el análisis fisiológico, es por esto que el conocer los análisis existentes para tal propósito así como las características de los mismos, se vuelve muy importante. El objetivo del análisis matemático de los ritmos circádicos es caracterizar a la señal rítmica proveniente de algún organismo.

La caracterización de la ritmicidad incluye:

- Estimación de la forma de la onda.
- Estimación de la amplitud de la oscilación
- Descripción de la fase en sincronización.
- Determinación de la fase del fenómeno durante corrimiento espontaneo.
- Estimación del periodo de un ritmo, en los diferentes estados que esté presenta.

La forma de la onda puede ser vista como la desviación de la forma respecto a una onda sinusoidal pura. El periodo ó longitud de ciclo es la duración de un ciclo completo, es decir el intervalo entre dos puntos de referencia idénticos, uno seguido de otro, comúnmente expresado en unidades de tiempo. La amplitud la definiremos como la diferencia entre el valor máximo y el mínimo sin embargo cuando estamos hablando de señales sinusoidales o cosenoidales como en el caso del cosinor, la definiremos como la mitad de la diferencia entre la cresta y el valle, o como la diferencia entre la media y el valor máximo, como lo indica la figura 1.5. La fase es el tiempo estimado de la ocurrencia para un punto ó evento dado. El ángulo de fase el cuál lo definiremos como la diferencia de fases entre dos señales, donde una corresponde al agente sincronizador y la otra a la variable indicadora. Si una es la fase de referencia, cuando la fase de la otra variable ocurre antes será un avance y si ocurre después será un retraso de fase.

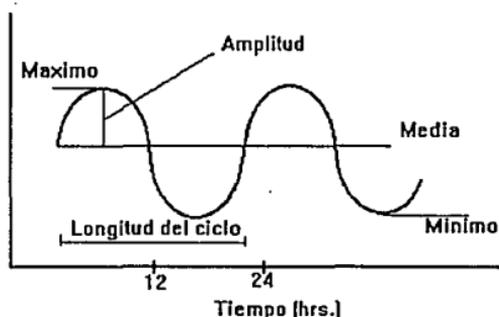


Fig. 1.5 Ritmos circádicos tienen una sola longitud de ciclo, aproximadamente 24 hrs

Para poder caracterizar al oscilador, el cambio cíclico que se observe debe ser medible. Ya que este cambio es continuo, para que lo podamos caracterizar por alguno de los métodos de análisis utilizados en el estudio de los ritmos biológicos, debemos obtener una secuencia de eventos o números, que representan el comportamiento del cambio cíclico, esto se logra muestreando las variables de interés, A la secuencia obtenida le llamaremos serie de tiempo, la cuál representaremos como:

$$X(n) = X_1, \dots, X_m$$

Al muestrear las variables de interés, debemos considerar la frecuencia de muestreo, ya que en el caso de estudios en cronobiología, una restricción en la frecuencia de muestreo puede influir considerablemente en el resultado obtenido. Considere, por ejemplo, la serie de datos hipotéticos de la figura 1.6, en donde se ilustran los mismos datos a diferentes frecuencias de muestreo, se observa que sólo los picos principales están presentes en la mayoría de los casos, y la descripción es por tanto aproximada, por lo que

se deben tomar muestras sobre un periodo de tiempo que cubra por lo menos dos ciclos completos, esto debido al teorema del muestreo (ver capítulo II).

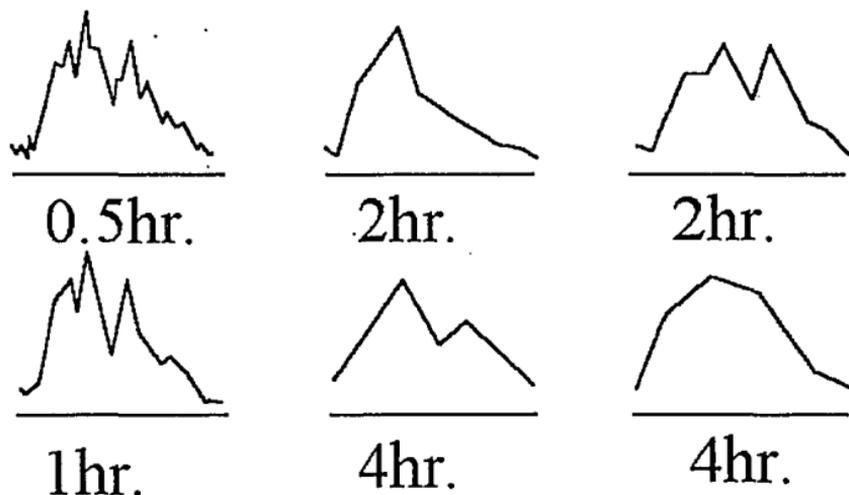


Fig. 1.6 Mismos Datos muestreados a diferentes tiempos. (Tomado de *Mathematical and Statistical Analysis of Circadian Rhythms*, D. S. Minors and J. M. Waterhouse. *Psychoneuroendocrinology* V13 No6 443-464, 1988)

También para poder realizar la caracterización de la señal se hace necesario el conocer el tipo de señal que se analizara, por supuesto tomando en cuenta de donde proviene y como se comporta generalmente, ya que esto determina cuales técnicas se utilizarán, evidentemente no todas las técnicas descritas para el procesamiento digital de señales son aplicables.

Los procesos fisiológicos por medio de los cuales la mayoría de los animales mantienen una relación de fase con un medio ambiente regularmente cíclico, es un ritmo endógeno, sincronizable, usualmente auto sostenido, en el cual una sola frecuencia ó una banda muy estrecha de frecuencias dominan la conducta y fisiología del animal. En un ritmo autosostenido, la fase es conservada por largos periodos, si no es que indefinidamente.

Estas características de los ritmos biológicos crean una muy especial y limitada clase de problemas en el análisis de los datos, para lo cual las técnicas útiles y apropiadas son comúnmente, diferentes de las diseñadas para series de tiempo provenientes de otras fuentes.

En el análisis de los ritmos circádicos, para su representación gráfica, detección, comparación y descripción, existen numerosas técnicas de análisis, como por ejemplo;

- Perfil de actividad
- Actogramas
- Histogramas de frecuencias.
- Periodograma
- Densidad espectral.
- Cosinor.

Cada una de estas técnicas ofrecen diferentes características y por ello es indispensable efectuar más de un análisis para una definición, descripción e identificación más clara y precisa, sin embargo, no sólo las características propias de alguna técnica de análisis se deben tomar en cuenta, ya que existen factores de adquisición de datos los cuales influyen en el resultado del análisis.

La forma en que se captura la información, juega un papel muy importante, si bien es cierto que para algunos de los experimentos fisiológicos carece de importancia debido a que están bien definidos los posibles resultados, también es cierto que para el análisis matemático, la captura o colección de datos cobra gran importancia.

El como capturar una señal varía de acuerdo al tipo de señal de interés, sin embargo, en el estudio de ritmos circádicos de actividad locomotora en roedores, los registros longitudinales son los más utilizados. El ciclo de actividad-reposo, es comúnmente monitoreado por medio de un registrador de eventos del tipo *Esterlin-Angus*, del cual se obtienen gráficas diarias que hay que recortar, pegar y fotografiar en double-plot, para el análisis de fase y periodo (sobre la gráfica). Este método es de intensa labor y no provee datos numéricos para un análisis cuantitativo.

Con el advenimiento de la electrónica y las computadoras, los métodos de adquisición han cambiado y ahora se pueden obtener datos numéricos, de tal forma que alguno de los análisis matemáticos se pueda realizar. Actualmente en el laboratorio de cronobiología se cuenta con un sistema de captura de datos, provenientes de la actividad locomotora ó de ingesta de agua. La figura 1.7 muestra la organización del sistema de captura.

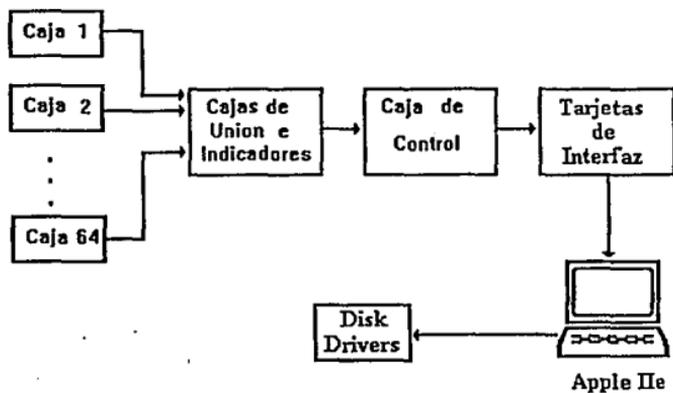


Figura 1.7 Sistema de Registro

El sistema descrito en la figura esta implantado en una computadora Apple IIe, y tiene una capacidad máxima de 64 canales.

## II Fundamentos de Procesamiento Digital de Señales

El presente capítulo, tiene como finalidad presentar los conceptos fundamentales en el procesamiento de señales, disciplina que ha cobrado gran importancia en la era de información en la que vivimos, ya que ahora con el advenimiento de la electrónica y la computación, ha sido posible poner en práctica la teoría del procesamiento digital de señales

El procesamiento digital de señales comienza con una señal cuantizada en tiempo discreto la cual aparece en la computadora como una secuencia de valores digitales. La figura 2.1 muestra un ejemplo de una operación en procesamiento digital de señales ( en lo sucesivo lo referiremos como PDS), donde hay una secuencia de entrada  $x(n)$ , el operador  $\zeta\{ \}$  y una secuencia de salida  $y(n)$ , los operadores pueden clasificarse como lineales y no lineales.

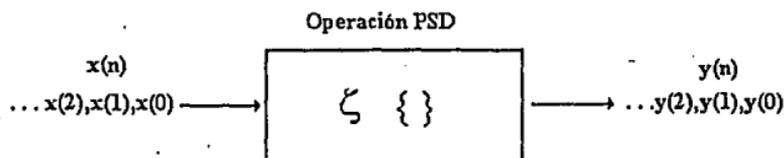


Figura 2.1 Operador para PDS

Los operadores son aplicados a las secuencias para obtener alguno de los siguientes resultados

- 1) Extraer parámetros o características de la secuencia
- 2) Producir una secuencia similar con características particulares, mejoradas o eliminadas, respecto a la original.
- 3) Restaurar la secuencia a algún estado anterior.
- 4) Codificar o comprimir una secuencia

Un sistema completo de PDS puede consistir de muchas operaciones sobre la misma secuencia así como operaciones sobre el resultado de las operaciones, un ejemplo es el mostrado en la figura 2.2

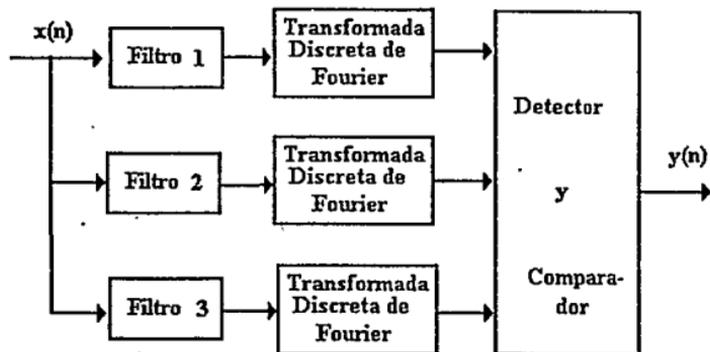


Figura 2.2 Posibles operaciones sobre una secuencia

## II.1 Secuencias

Para que la computadora pueda manejar una señal, esta debe haber sido muestreada a algún intervalo de tiempo, en la figura 2.3 se muestra un ejemplo de la función en tiempo continuo la cual ha sido muestreada a intervalos fijos de  $T$  segundos de duración, el conjunto de números resultante es llamado una secuencia. Si la función en tiempo continuo fue  $x(t)$ , entonces las muestras deben ser  $x(nT)$ , donde  $n$  es un entero extendiéndose sobre un rango finito de valores.

En la práctica es común normalizar el intervalo de muestreo a 1 y facilitar así las ecuaciones, la secuencia es entonces  $x(n)$ . Se debe tener mucho cuidado cuando se calcula la potencia o energía de la secuencia, en que el intervalo de muestreo, (incluyendo unidades de tiempo o espacio) sea reinsertado en los puntos apropiados.

Una secuencia como una representación de una señal continua tiene las siguientes características.

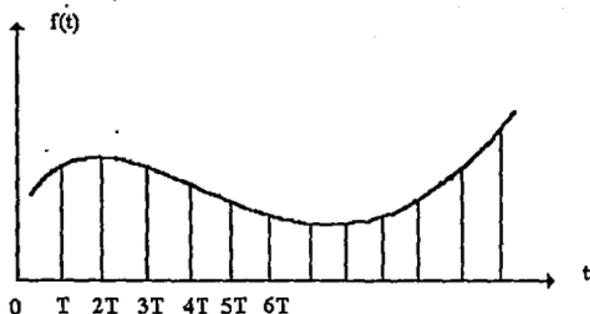


Figura 2.3 muestreo

- 1) La señal está muestreada, tiene valores finitos solo en puntos discretos en el tiempo
- 2) La señal está truncada. Tiene una longitud finita en un intervalo finito de tiempo
- 3) La señal está cuantizada. Está limitada a pasos discretos en amplitud, donde el tamaño del paso y por lo tanto la fineza ó fidelidad de la señal, depende de la precisión aritmética de la computadora.

Estas características deben ser tomadas en cuenta, para comprender los resultados que los operadores PDS producen.

## II.2 La Función de Muestreo

La función de muestreo es la llave para viajar entre los mundos de tiempo continuo y tiempo discreto y es llamada con diferentes nombres, función delta de Dirac, función singular entre otros, la cual tiene las siguientes propiedades:

propiedad 1 :

$$\int_{-\infty}^{\infty} f(t) \delta(t - \tau) dt = f(\tau) \quad 2.1$$

Propiedad 2 :

$$\int_{-\infty}^{\infty} \delta(t - \tau) dt = 1 \quad 2.2$$

Donde  $\tau$  puede ser cualquier número real

Para estudiar como esta función puede ser vista como una función de muestreo ideal, primero consideremos la función de muestreo  $\Delta(t)$  ilustrada en la figura 2.4, donde el ancho del pulso es de una unidad de tiempo y su amplitud es de una unidad de amplitud, esta claramente exhibe la propiedad 2 de la función de muestreo mencionada anteriormente sin embargo, cuando  $\Delta(t)$  es multiplicada por la función a ser muestreada, la función de muestreo  $\Delta(t)$  selecciona no solo un instante en tiempo sino un rango de menos un medio a un medio,  $(-1/2, 1/2)$ , como resultado, la propiedad 1 de la función de muestreo no se cumple, así tenemos que:

$$\int_{-\infty}^{\infty} f(t)\Delta(t-\tau)dt = \int_{\tau-1/2}^{\tau+1/2} f(t)dt \quad 2.3$$

Esto puede pensarse como un cierto tipo de artefacto del proceso de muestreo, cruzando una banda la cual esta relacionada con el ancho del pulso de  $\Delta(t)$ . Una mejor aproximación a la función de muestreo podría ser una función  $\Delta(t)$  con un ancho de pulso muy pequeño, cuanto más pequeño, la amplitud debe incrementarse. En el limite podemos ver que la función ideal de muestreo debe tener un ancho de pulso infinitamente pequeño para que muestree a un solo instante de tiempo y una amplitud infinitamente grande para que la señal muestreada contenga la misma energía finita.

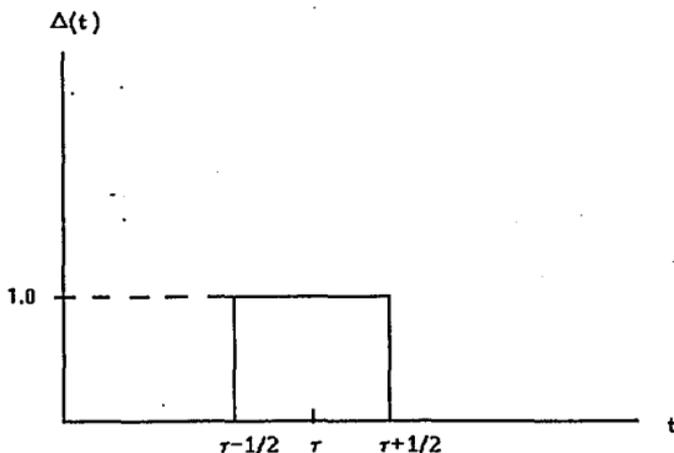


Figura 2.4 Función de muestreo realizable

## II.3 Uso de La Función de Muestreo

La figura 2.5 ilustra el proceso de muestreo ideal a intervalos de  $T$  unidades de tiempo. La forma de onda resultante en tiempo puede ser escrita como :

$$x_s(t) = \sum_{n=-\infty}^{\infty} x(t)\delta(t-nT) \quad 2.4$$

La forma de onda que resulta de este proceso es imposible de visualizar debido a la amplitud infinita y al ancho de casi cero de la función ideal de muestreo, sin embargo esto puede ser fácilmente visto en la figura 2.5 para dar una idea mas o menos de la función de muestreo ideal ( con un ancho muy pequeño y una amplitud muy grande), la cual es multiplicada por una forma de onda en el tiempo continuo.

Se debe enfatizar que  $x_s(t)$  es una forma de onda en el tiempo continuo hecha de la superposición de un conjunto infinito de señales en tiempo continuo  $x(t)\delta(t-nT)$  y también puede ser escrita como:

$$x_s(t) = \sum_{n=-\infty}^{\infty} x(nT)\delta(t-nT) \quad 2.5$$

Debido a que la función de muestreo da un multiplicador distinto de cero solo cuando  $t=nT$  en esta ultima ecuación la secuencia  $x(nT)$  toma su apariencia, esto es el conjunto de números con los cuales trabajaremos.

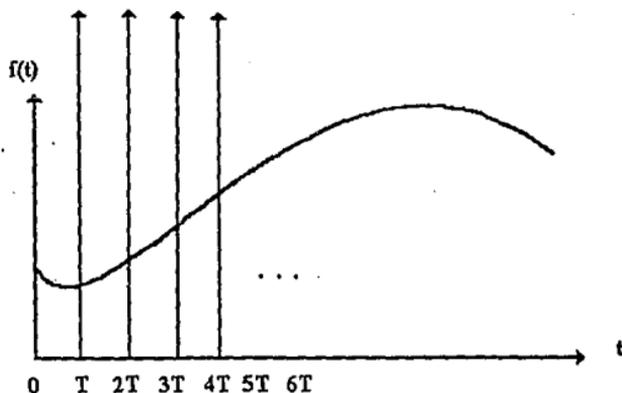


Figura 2.5 función de muestreo ideal

## II.4 El Espectro de La Señal Muestreada

Utilizando la teoría de la transformada de Fourier, el espectro de frecuencia de la forma de onda en tiempo continuo  $x(t)$  puede definirse como:

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi ft} dt \quad 2.6$$

y la función en el tiempo puede ser expresada como

$$x(t) = \int_{-\infty}^{\infty} X(f) e^{j2\pi ft} df \quad 2.7$$

Debido a que esto es verdad para cualquier función en el tiempo continuo,  $x(t)$ , también es verdad para  $x_s(t)$

$$X_s(f) = \int_{-\infty}^{\infty} x_s(t) e^{-j2\pi ft} dt \quad 2.8$$

reemplazando  $x_s(t)$  por su representación muestreada

$$X_s(f) = \int_{-\infty}^{\infty} \left[ \sum_{n=-\infty}^{\infty} x(t) \delta(t - nT) \right] e^{-j2\pi ft} dt \quad 2.9$$

El orden de la suma y la integral se puede intercambiar y aplicando la propiedad 1 de la función de muestreo tenemos

$$X_s(f) = \sum_{n=-\infty}^{\infty} x(nT) e^{-j2\pi fnT} \quad 2.10$$

Esta ecuación es la forma exacta de una representación en series de Fourier de  $X_s(f)$ , que es una función periódica de frecuencia, teniendo por periodo  $1/T$ . Los coeficientes de la serie de Fourier son  $x(nT)$  y pueden ser calculados de la siguiente integral.

$$x(nT) = T \int_{-\frac{1}{2T}}^{\frac{1}{2T}} X_s(f) e^{j2\pi fnT} df \quad 2.11$$

Las dos ecuaciones anteriores constituyen un par de series de Fourier las cuales permiten el calculo de la señal en el tiempo o su espectro en frecuencia en términos del miembro opuesto par. Note que la señal  $x_s(t)$  está eliminada y en su lugar se usa la secuencia  $x(nT)$ .

## II.5 Relación de los Espectros de las Señales en Tiempo Continuo y Discreto

Al evaluar la ecuación 2.7 en  $t=nT$  y poniendo el resultado igual al lado derecho de la ecuación 2.11, se obtiene la siguiente relación entre los dos espectros

$$x(nT) = \int_{-\infty}^{\infty} X(f) e^{j2\pi f nT} df = T \int_{-\frac{1}{2T}}^{\frac{1}{2T}} X_s(f) e^{j2\pi f nT} df \quad 2.12$$

El lado derecho de la ecuación 2.7 se puede expresar como la suma infinita de un conjunto de integrales con límites finitos

$$x(nT) = \sum_{m=-\infty}^{\infty} \int_{\frac{2m-1}{2T}}^{\frac{2m+1}{2T}} X(f) e^{j2\pi f nT} df \quad 2.13$$

haciendo el cambio de variables  $\lambda = f - m/T$  (sustituyendo  $f = \lambda + m/T$  y  $df = d\lambda$ )

$$x(nT) = \sum_{m=-\infty}^{\infty} \int_{-\frac{1}{2T}}^{\frac{1}{2T}} X(\lambda + \frac{m}{T}) e^{j2\pi \lambda nT} e^{j2\pi \frac{m}{T} nT} d\lambda \quad 2.14$$

Moviendo la suma dentro de la integral y reconociendo que  $e^{j2\pi mn}$  (para todos los enteros  $m$  y  $n$ ) es igual a 1, e igualando dentro de la integral la parte similar de la ecuación 2.11 tenemos la siguiente relación

$$X_s(f) = \sum_{m=-\infty}^{\infty} X(f + \frac{m}{T}) \quad 2.15$$

La ecuación anterior muestra que el espectro en frecuencia de la señal muestreada en tiempo es igual a, una suma infinita de replicas recorridas del espectro en frecuencia de la señal en tiempo continuo, sobrepuesta una con otra.

El corrimiento de las replicas es igual a la frecuencia de muestreo  $1/T$ , entonces es interesante examinar las condiciones bajo las cuales dos espectros son iguales uno con otro al menos para un rango limitado de frecuencias.

En el caso donde no hay componentes espectrales de frecuencia mayores que  $1/2T$  en la forma de onda original en el tiempo continuo, los dos espectros son iguales sobre el rango de frecuencias  $f=-1/2T$  a  $f=1/2T$ . Por supuesto el espectro de la señal muestreada repetirá este mismo conjunto de amplitudes periódicamente para todas las frecuencias, mientras que el espectro de la señal en el tiempo continuo es cero para todas las frecuencias fuera del rango especificado.

El criterio de muestreo de Nyquist esta basado en el planteamiento anterior y dice que cuando una señal en tiempo continuo se muestrea a una frecuencia mayor que dos veces la componente de frecuencia máxima en su espectro, entonces puede ser reconstruida completamente a partir de la forma de onda muestreada. Inversamente si una señal en tiempo continuo se muestrea a una frecuencia inferior a su componente de frecuencia máxima, ocurrirá un fenómeno llamado *aliasing*. Si una señal en tiempo continuo es reconstruida de una representación con este fenómeno, el resultado estará distorsionado y el grado de distorsión depende del grado de *aliasing*.

## II.6 Señales Discretas Especiales

### II.6.1 Señal impulso unitario ó señal delta

La señal impulso unitario es una secuencia que vale la unidad cuando su argumento es cero y vale cero para cualquier otro argumento.

$$\delta[k] = \begin{cases} 1, & k = 0 \\ 0, & k \neq 0 \end{cases} \quad 2.16$$

La utilización del impulso unitario en los sistemas lineales, nos 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 la relación

$$\delta[k-n] = \begin{cases} 1, & k = n \\ 0, & k \neq n \end{cases} \quad 2.17$$

De la igualdad anterior, se desprende que una secuencia arbitraria  $x[k]$  puede ser representada como una suma de impulsos unitarios apropiadamente desplazados y multiplicados por los valores de la secuencia  $x[k]$

$$x[k] = \sum_{n=-\infty}^{\infty} x[n] \delta[k-n] \quad 2.18$$

Al impulso unitario también se le conoce como muestra unitaria, secuencia delta o secuencia impulso.

## II.6.2 Secuencia escalón unitario.

La secuencia escalón unitario toma el valor de la unidad para todos los argumentos mayores o igual a cero y, es igual a cero para los demás valores del argumento.

$$u[k] = \begin{cases} 1, & k \geq 0 \\ 0, & k < 0 \end{cases} \quad 2.19$$

El escalón unitario esta relacionado con el impulso por:

$$u[k] = \sum_{n=-\infty}^k \delta[n] = \sum_{n=0}^{\infty} \delta[k-n] \quad 2.20$$

Además también se cumple la relación siguiente:

$$\delta[k] = u[k] - u[k-1] \quad 2.21$$

## II.7 El concepto de frecuencia

De física conocemos que la frecuencia esta relacionada a un tipo específico de movimiento periódico llamado oscilación armónica la cual está descrita por funciones sinusoidales. El concepto de frecuencia está directamente relacionado al concepto del tiempo. Tiene la dimensión del tiempo inverso, por lo tanto la naturaleza del tiempo (continuo ó discreto), debe afectar en forma acorde a la naturaleza de la frecuencia.

### II.7.1 Señales sinusoidales en tiempo continuo

Una oscilación armónica está descrita matemáticamente por :

$$x(t) = A \cos(\Omega t + \theta) \quad 2.22$$

Donde A es la amplitud  $\Omega$  es la frecuencia en radianes por segundo y  $\theta$  es la fase en radianes. Usamos F en vez de  $\Omega$ , la cual esta en ciclos por segundos o Hertz, entonces

$$\Omega = 2\pi F \quad 2.23$$

$$x(t) = A \cos(2\pi Ft + \theta) \quad 2.24$$

La señal anterior está caracterizada por las siguientes propiedades:

1) Para cada valor fijo de F,  $x(t)$  es periódica

$$\begin{aligned} x(t + T_p) &= x(t) \\ T_p &= \frac{1}{F} \quad \text{Periodo Fundamental} \end{aligned} \quad 2.25$$

2) Señales sinusoidales en tiempo continuo con distintas frecuencias son distintas

3) Un incremento en la frecuencia tiene como efecto un incremento en la razón de oscilación de la señal, en esencia quedan más periodos incluidos en un intervalo de tiempo.

Por definición, frecuencia es una cantidad física positiva inherente, esto es obvio si interpretamos frecuencia como el número de ciclos por unidad de tiempo en una señal periódica, sin embargo en muchos casos, solo por conveniencia matemática se necesita introducir frecuencias negativas.

## II.7.2 Señales sinusoidales en tiempo discreto

Una señal sinusoidal en tiempo discreto, puede expresarse como:

$$x(n) = A \cos(\omega n + \theta), \quad -\infty < n < \infty \quad 2.26$$

Donde

$n$  : Variable entera, llamada número de muestra

$A$  : Amplitud

$\omega$  : Es la frecuencia en radianes por muestra

$\theta$  : Es la fase en radianes

En vez de  $\omega$  usamos la variable de frecuencia  $f$  definida por :  $\omega = 2\pi f$

$$x(n) = A \cos(2\pi f n + \theta), \quad -\infty < n < \infty \quad 2.27$$

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

La señal anterior está caracterizada por las siguientes propiedades:

1) Una senoide en tiempo discreto es periódica solo si la frecuencia  $f$  es un número racional

$$x(n + N) = x(n) \quad N: \text{Periodo } N > 0 \quad 2.28$$

El valor más pequeño para el cual lo anterior es verdad se le llama periodo fundamental, entonces

$$\cos[2\pi f_0(N+n) + \theta] = \cos(2\pi f_0 n + \theta) \quad 2.29$$

Esta relación solo es verdad si existe un entero tal que:

$$2\pi f_0 N = 2k\pi \quad \text{ó} \quad f_0 = k/N \quad 2.30$$

Observe que un pequeño cambio en frecuencia resultará en un gran cambio en periodo.

2) Las senoideas discretas cuyas frecuencias estén separadas por un entero múltiplo de  $2\pi$  son idénticas

3) La razón más alta de oscilación en una senoide discreta es alcanzada cuando

$$\omega = \pi \quad \text{ó} \quad \omega = -\pi$$

## II.8 Exponenciales complejas relacionadas armónicamente

Las señales sinusoidales y exponenciales complejas juegan un papel muy importante en el análisis de señales y sistemas. Es interesante trabajar con funciones exponenciales complejas (o sinusoidales) relacionadas armónicamente. Estas forman conjuntos de exponenciales complejas periódicas, con frecuencias fundamentales que son múltiplos de una frecuencia positiva.

Puesto que una exponencial compleja en tiempo discreto es periódica si su frecuencia relativa es un número racional, entonces seleccionando  $f_0 = 1/N$  y definiendo así a los conjuntos de exponenciales complejos relacionados armónicamente, como:

$$S_k(n) = e^{j2\pi k f_0 n} \quad k = 0, \pm 1, \pm 2, \dots \quad 2.31$$

En contraste con el caso en tiempo continuo se puede escribir:

$$S_{k+N}(n) = e^{j2\pi n(k+N)/N} = e^{j2\pi n} S_k(n) = S_k(n) \quad 2.32$$

Esto es consistente con  $x(n+N)=x(n)$ , es decir solo hay  $N$  distintas exponenciales complejas periódicas en el conjunto descrito por la ecuación anterior. Más aún, todos los miembros del conjunto tienen un periodo común de  $N$  muestras. Podemos seleccionar cualquier conjunto de exponenciales complejas, por ejemplo  $k=n_0$  a  $k=n_0+N-1$ , para formar un conjunto armónicamente relacionado con frecuencia fundamental  $f_0 = 1/N$ , comúnmente por conveniencia se selecciona el conjunto para  $n_0 = 0$ , esto es el conjunto

$$S_k(n) = e^{j2\pi nk/N} \quad k=0,1,2,\dots,N-1 \quad 2.33$$

Como en el caso de las señales en tiempo continuo, es obvio que la combinación lineal

$$x(n) = \sum_{k=0}^{N-1} C_k S_k(n) = \sum_{k=0}^{N-1} C_k e^{j2\pi kn/N} \quad 2.34$$

resulta en una señal periódica con periodo fundamental  $N$ , como se vera más adelante esta es la representación en series de Fourier para una secuencia periódica en tiempo discreto con coeficientes de Fourier  $\{C_k\}$ . La secuencia  $S_k(n)$  es llamada la  $k$ -ésima armónica de  $x(n)$

## II.9 Probabilidad y Procesos Aleatorios

Las señales de interés en los problemas de procesamiento digital de señales están dentro de un medio ambiente de ruido e interferencia. El ruido puede deberse a señales espurias adquiridas durante la transmisión (*Interferencia*), a las características de ruido de los dispositivos electrónicos que reciben la señal. Para tratar efectivamente una señal con ruido, se debe utilizar algún modelo del ruido ó de la señal mas el ruido.

Comúnmente son usados modelos probabilísticos debido a que el ruido es por naturaleza impredecible. De tal forma que es necesario introducir los conceptos básicos de probabilidad y procesos aleatorios los cuales son básicos para el procesamiento digital de señales.

### II.9.1 Probabilidad Básica

La probabilidad comienza por definir la probabilidad de un evento "A" como  $P(A)$ . El evento A puede ser el resultado del lanzamiento de una moneda, los resultados de una carrera de caballos o cualquier, otro resultado de una actividad, el cual no es completamente predecible. Hay tres características ó atributos para esta probabilidad  $P(A)$ :

- 1)  $P(A) \geq 0$ . Esto simplemente quiere decir que cualquier resultado tiene una oportunidad positiva de ocurrencia o ninguna oportunidad de ocurrencia.
- 2)  $P(\text{todos los posibles resultados}) = 1$ . Esto indica que algún resultado entre los posibles está en la región de ocurrencia, ciertamente existe una probabilidad de 1
- 3) Para  $\{A_i\}$ , donde  $(A_i \cap A_j) = 0$ ,  $P(\cup A_i) = \sum_i P(A_i)$ . Para un conjunto de eventos,  $\{A_i\}$ , donde los eventos son mutuamente excluyentes.

Con la probabilidad definida de esta forma, la discusión puede extenderse a las probabilidades conjunta y condicionales. La probabilidad conjunta esta definida como la probabilidad de ocurrencia de un conjunto específico de dos o más eventos como resultado de una sola realización. Ejemplificando, la probabilidad de que un caballo A termine en tercer lugar y un caballo B termine en primero en una carrera de caballos particular, es una probabilidad conjunta, la cual la podemos escribir como:

$$P(A \cap B) = P(A \text{ y } B) = P(AB) \quad 2.35$$

La probabilidad condicional esta definida como la probabilidad de ocurrencia de un evento A dado que se conoce que el evento B ha ocurrido. Por ejemplo, si la actividad es conducir al trabajo, entonces dos de los posibles eventos en el conjunto de todos los posibles eventos son:

- A: Llegar a tiempo al trabajo, y  
B: Se tenga un neumático desinflado

Estos eventos no son mutuamente excluyentes, Se puede salir suficientemente temprano para reparar el neumático y llegar a tiempo al trabajo. Pero queremos asociar una probabilidad al evento A si conocemos que el evento B ha ocurrido. Entonces la probabilidad asignada al evento A esta condicionada por el evento B, esto puede ser escrito como:

$$P(A \text{ dado } B) = P(A/B). \quad 2.36$$

Si esta probabilidad condicional,  $P(A/B)$  y la probabilidad de B son conocidas, se puede calcular la probabilidad de la ocurrencia de los dos eventos (probabilidad conjunta), como:

$$P(AB) = P(A/B)P(B) \quad 2.37$$

La probabilidad condicional es multiplicada por la probabilidad de la condición (evento B), para obtener la probabilidad conjunta. Otra forma de escribir esta ecuación es:

$$P(A/B) = P(AB)/P(B) \quad 2.38$$

Esta es otra forma de definir la probabilidad condicional una vez conocida la probabilidad conjunta.

El teorema de Bayes es una ecuación comúnmente utilizada en la teoría de probabilidades y esta definido de la siguiente forma

$$P(AB) = P(BA) \quad 2.39$$

Esto se satisface debido a que ninguna de las probabilidades conjuntas se ve afectada por el orden en el cual los eventos sean presentados ó listados. El uso más común del teorema de Bayes es para la definición de la probabilidad condicional.

$$P(A/B)P(B) = P(B/A)P(A) \quad 2.40$$

$$P(A/B) = [P(B/A)P(A)]/P(B) \quad 2.41$$

La ultima ecuación es comúnmente utilizada cuando la probabilidad de los eventos individuales y una de las probabilidades condicionales es conocida y nos gustaría encontrar la otra probabilidad condicional.

## II.9.2 Variables Aleatorias

En el procesamiento de señales, la probabilidad de que una señal tome un cierto valor ó caiga en cierto rango de valores, es comúnmente deseable de conocer. La señal en este caso puede pensarse como una variable aleatoria (VA), (un elemento cuyo conjunto de posibles valores es el conjunto de resultados de la actividad). Si nuestra VA es  $X$  podemos tener el siguiente conjunto de eventos que pueden ocurrir.

Evento A :  $X$  toma el valor de 5 ( $X=5$ )

Evento B :  $X=19$

Evento C:  $X=1.66$

etc..

Esto es un conjunto útil de eventos para variables discretas las cuales solo pueden tomar ciertos valores. Un conjunto más práctico de eventos para variables continuas asocia cada evento con la variable dentro de un rango de valores. Si se define entonces una función de distribución acumulativa para una VA como sigue:

$$F(x) = P(X \leq x) \quad 2.42$$

Esta función de distribución es entonces una función monotonicamente creciente de la variable independiente  $x$ , y solo es válida para la variable aleatoria particular  $X$ , la figura 2.6 muestra un ejemplo de una función de distribución para una variable aleatoria. Si diferenciamos  $F(x)$  con respecto a  $x$  obtenemos la función de densidad de probabilidad para  $X$ , representada como sigue:

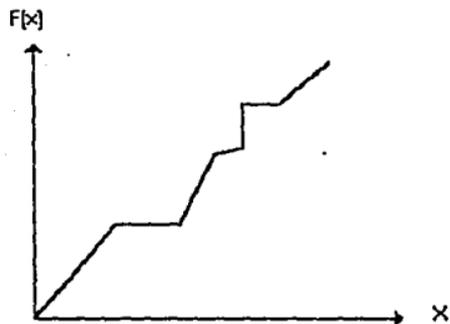
$$p(x) = \frac{dF(x)}{dx} \quad 2.43$$

Integrando  $p(x)$  obtenemos la función de distribución, como sigue:

$$F(x) = \int_{-\infty}^x p(\lambda) d\lambda \quad 2.44$$

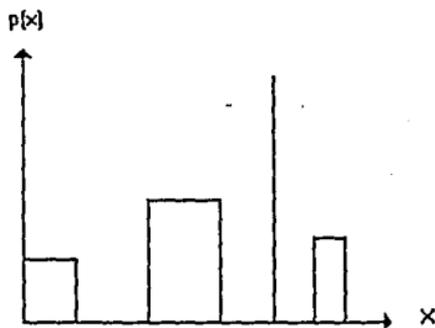
Debido a que  $F(x)$  es siempre monotonicamente creciente,  $p(x)$  debe ser siempre positiva o cero, la figura 2.5 muestra la función de distribución y su respectiva función de densidad en la figura 2.6. La utilidad de estas funciones puede ser ilustrada, determinando la probabilidad de que la VA  $X$  caiga entre  $a$  y  $b$ , utilizando la propiedad 3 de la probabilidad

$$P(X \leq b) = P(a < X \leq b) + P(X \leq a) \quad 2.45$$



Función de Distribución

Figura 2.5



Función de Densidad

Figura 2.6

Esto se verifica porque las dos condiciones del lado derecho son independientes y  $X$  debe ser conocida para una u otra. Si conocemos la condición del lado izquierdo se puede re escribir la ecuación usando la definición de la distribución.

$$P(a < X \leq b) = F(b) - F(a)$$

$$= \int_a^b p(x) dx \quad 2.46$$

En esta forma conociendo la distribución o función de densidad, podemos conocer ó calcular la probabilidad de que X caiga dentro de un rango dado.

Existe un operador en la teoría de variables aleatorias llamado el operador esperanza matemática usualmente escrito como  $E[x]$ . Esta expresión es llamada también "El valor esperado de x". El operador  $E[x]$  extrae de una VA el valor que toma más comúnmente. El valor esperado es llamado también la media, promedio, o primer momento de la VA y es calculado a partir de la función de densidad como sigue:

$$E[x] = \int_{-\infty}^{\infty} xp(x)dx \quad 2.47$$

El valor esperado puede ser visto como el "centro de gravedad" ó primer momento de la variable aleatoria X

La varianza de una VA se define como

$$\sigma^2 = Var\{x\} = E \left\{ (x - E[x])^2 \right\} \quad 2.48$$

La varianza es llamada también, valor medio cuadrático de X y la desviación estándar  $\sigma$  es la raíz cuadrada de la varianza. Extendiendo el uso del operador esperanza para densidades de probabilidad conjunta, una variable Y puede ser función de dos variables aleatorias, s,t tal que

$$E[Y] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \zeta \{s,t\} P(s,t) ds dt \quad 2.49$$

Entonces el valor esperado de Y será

$$E[Y] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \zeta \{s,t\} P(s,t) ds dt \quad 2.50$$

Donde  $P(s,t)$  es la densidad de probabilidad conjunta de s,t. Definiendo la correlación de dos VA como el valor esperado de sus productos, se tiene

$$E[st] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} stP(s,t) ds dt \quad 2.51$$

Existe un conjunto de VA llamadas VA Gaussianas cuyas funciones de densidad tienen características especiales que las hacen particularmente fáciles de analizar, también muchos procesos físicos se pueden aproximar a esta función de densidad. Una VA gaussiana tiene la siguiente forma:

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right] \quad 2.52$$

Donde  $\mu$  es el valor medio de  $x$ , y  $\sigma^2$  es la varianza.

## II.10 Cuantización de Secuencias

La cuantización es para el dominio de la amplitud de una señal analógica continua, lo que el muestreo es para el dominio del tiempo. Es el paso en el cual a una señal de amplitud continua se le representa en incrementos de amplitud discreta, disponibles en una computadora digital.

Para definir un cuantizador, se debe especificar un conjunto de niveles de decisión  $\{d_k, k=0,L\}$  y un conjunto de niveles de reconstrucción  $\{r_k, k=0,L-1\}$ , donde  $d_0$  y  $d_L$  son los valores mínimo y máximo de la función  $f(x)$ . La cuantización se realiza dando el nivel  $r_k$  a todos los valores de  $f(x)$  en el intervalo  $\{d_k, d_{k+1}\}$ .

Con el propósito de definir los niveles  $d_k$ , y  $r_k$  se pueden usar diferentes tipos de cuantización.

El error introducido en la representación es llamado error de cuantización, la operación se puede denotar como  $Q[x[n]]$  y  $Xq(n)$  denota la secuencia de muestras cuantizadas,  $Xq(n) = Q[x[n]]$ , entonces el error es una secuencia  $e_q(n)$ , definida como la diferencia entre el valor cuantizado y el valor actual de la muestra.  $e_q(n) = Xq(n) - X(n)$

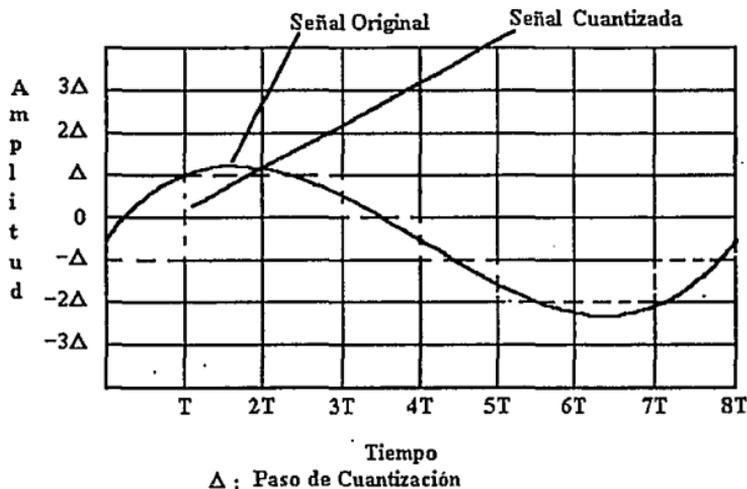


Figura 2.7 : Cuantización y muestreo de una señal analógica.

## II.11 Procesos Aleatorios, Autocorrelación y Densidad espectral

Un Proceso aleatorio  $F(A,t)$ , puede ser visto como una función de dos variables, un evento  $A$ , y un tiempo  $t$ , para cada  $t$ , el proceso puede ser considerado como una variable aleatoria. Para  $t=a$  hay una variable aleatoria la cual tiene una densidad de probabilidad, un valor esperado ó media y una varianza, como se definió anteriormente.

Uno de los aspectos importantes de un proceso aleatorio es la forma en la cual las VA a diferentes puntos en el proceso están relacionadas unas con otras. Aquí se necesita extender el concepto de distribución conjunta a las funciones de distribución y densidad. Una distribución de probabilidad conjunta se define como:

$$F(s,t) = P(S \leq s, T \leq t) \quad 2.53$$

Donde  $s, t$  son algunas constantes.

Y la correspondiente función de densidad esta definida como

$$P(s,t) = \frac{\partial^2 F(s,t)}{\partial s \partial t} \quad 2.54$$

La integral que relaciona ambas funciones es

$$F(s, t) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p(\alpha, \beta) d\alpha d\beta \quad 2.55$$

Anteriormente se vio que la correlación de dos VA es el valor esperado de sus productos. La autocorrelación de un proceso aleatorio es el valor esperado de los productos de la VA que constituyen al proceso.

El símbolo para la autocorrelación es  $R_{ff}(t_1, t_2)$  para la función  $f(t)$  y la definición es:

$$\begin{aligned} R_{ff}(t_1, t_2) &= E[f(t_1)f(t_2)] \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \alpha \beta p_f(\alpha, \beta; t_1, t_2) d\alpha d\beta \quad 2.56 \end{aligned}$$

Donde  $P_f(\alpha, \beta; t_1, t_2)$  es la densidad de probabilidad conjunta de  $f(t_1)$  y  $f(t_2)$  al incluir  $\alpha$  y  $\beta$  en los paréntesis la dependencia de  $P_f$  sobre estas variables se hace explícita.

En el caso general la autocorrelación puede tener diferentes valores para cada valor de  $t_1$  y  $t_2$ , sin embargo hay una clase especial de procesos aleatorios llamados procesos estacionarios para los cuales la forma de la autocorrelación es algo más simple. En los procesos aleatorios estacionarios, la autocorrelación es solo una función de la diferencia entre dos variables en el tiempo, para procesos estacionarios podemos escribir entonces

$$R_{ff}(t_2 - t_1) = R_{ff}(\xi) = E[f(t - \xi)f(t)] \quad 2.57$$

Por su naturaleza un proceso ruidoso no puede ser especificado como una función del tiempo en la misma forma que una señal determinística, por lo que debemos conformarnos con la función de probabilidad y el primer y segundo momentos del proceso, aunque esto es solo una caracterización parcial, una cantidad considerable de análisis puede realizarse utilizando tan solo los parámetros de momentos.

El primer momento de un proceso es simplemente su promedio o valor medio, para efectos de simplificación del álgebra y derivaciones, consideremos que los procesos tienen media igual a cero.

El segundo momento de un proceso es la autocorrelación

$$r[n, n-k] = E[u(n)u^*(n-k)] \quad \text{para } k=0, \pm 1, \pm 2, \dots \quad 2.58$$

Los procesos se consideran estacionarios en sentido amplio para el segundo momento. Esto quiere decir que la estadística de primer y segundo orden no cambian con el tiempo, esto permite representar a la autocorrelación como

$$r[n, n-k] = r[k] \quad \text{para } k=0, \pm 1, \pm 2, \dots \quad 2.59$$

Debido a que es una función de diferencias en el tiempo entre muestras y no en la variable del tiempo por sí misma. En cualquier proceso un miembro del conjunto de valores de la autocorrelación es  $r[0]$

$$r[0] = \varepsilon[u(n)u^*(n-k)] = \varepsilon[|u(n)|^2] \quad 2.60$$

El cual es el valor medio cuadrático del proceso, para un proceso de media cero esto es igual a la varianza.

$$r[0] = \text{Var}[u] \quad 2.61$$

Si representamos el proceso por un vector  $u(n)$  donde

$$u(n) = \begin{bmatrix} u(n) \\ u(n-1) \\ u(n-2) \\ \vdots \\ u(n-M+1) \end{bmatrix}$$

2.61b

Entonces la autocorrelación puede ser representada en forma de matriz

$$R = \varepsilon[u(n)u^H(n-k)] \quad 2.62$$

Donde  $u^H$  denota la matriz hermitiana  $A^H = (A^T)^* = (A^*)^T$

Los segundos momentos de un proceso son importantes porque están directamente relacionados al espectro de potencia del proceso, por la relación

$$S(f) = \sum_{k=-M+1}^{M-1} r(k) e^{-j2\pi f k} \quad 2.63$$

La cual es la transformada discreta de Fourier de la autocorrelación del proceso. Esto es la autocorrelación es la descripción en el dominio del tiempo de la estadística de segundo orden y la densidad espectral de potencia  $S(f)$  es la representación en el dominio de la frecuencia.

## II:12 La Transformada Discreta de Fourier

A lo largo de este capítulo hemos hecho referencia a la transformada de Fourier en varias ocasiones, pero no la hemos definido, entonces la transformada de Fourier para una secuencia causal se define como:

$$\mathfrak{F}\{x(n)\} = X(f) = \sum_{n=0}^{\infty} x(n) e^{-j2\pi f n} \quad 2.64$$

Donde el periodo de muestreo ha sido normalizado a uno ( $T=1$ ). Si la secuencia es de duración limitada (lo cual debe ser verdad para utilizarse con una computadora), tenemos que:

$$X(f) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi f n} \quad 2.65$$

Donde la forma de onda muestreada en el dominio del tiempo tiene una longitud de  $N$  muestras. La transformada inversa de Fourier es:

$$\mathfrak{F}^{-1}\{X(f)\} = x(n) = \int_{-1/2}^{1/2} X(f) e^{-j2\pi f n} df \quad 2.66$$

Como  $X(f)$  es periódica con periodo  $1/T=1$ , la integral puede ser tomada sobre cualquier periodo completo, por lo tanto.

$$x(n) = \int_0^1 X(f) e^{-j2\pi f n} df \quad 2.67$$

Estas representaciones de la transformada de Fourier son adecuadas pero tienen desventajas para las aplicaciones digitales - la variable de frecuencia es continua, no discreta. Para solucionar este problema debemos aproximar ambas representaciones (frecuencia y tiempo) de la señal.

Para crear una representación discreta de la transformada de Fourier debemos utilizar una versión muestreada de la forma de onda en frecuencia. Este muestreo en el dominio de la frecuencia es equivalente a la convolución en el dominio del tiempo con la siguiente forma de onda en el tiempo.

$$h_1(t) = \sum_{r=-\infty}^{\infty} \delta(t - rT) \quad 2.68$$

Esto crea duplicados de la forma de onda en el dominio del tiempo, las cuales se repiten con periodo T, T es igual al T utilizado anteriormente. Ahora bien utilizando el mismo número de muestras en el periodo de la forma de onda repetida en el dominio de la frecuencia, como un periodo en el dominio del tiempo, se obtiene un par de TDF el cual es una buena aproximación del par de transformada de Fourier para variables continuas, entonces la transformada discreta de Fourier es:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j 2\pi kn/N} \quad 2.69$$

y la transformada inversa es :

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j 2\pi kn/N} \quad 2.70$$

A continuación se presenta una tabla con las propiedades de simetría de la transformada discreta de Fourier.

### Propiedades de simetría de la TDF

Secuencia	Transformada Discreta de Fourier
$x(n)$	$X(\omega)$
$x^*(n)$	$X^*(-\omega)$
$x^*(-n)$	$X^*(\omega)$
$x_R(n)$	$x(n)$
$jx_I(n)$	$X_o(\omega) = \frac{1}{2}[X(\omega) - X^*(-\omega)]$
$x_e(\omega) = \frac{1}{2}[x(n) + x^*(-n)]$	$X_R(\omega)$
$x_o(\omega) = \frac{1}{2}[x(n) - x^*(-n)]$	$X_I(\omega)$

### Señales Reales

	$X(\omega) = X^*(-\omega)$
	$X_R(\omega) = X_R(-\omega)$
	$X_I(\omega) = X_I(-\omega)$
Cualquier Señal Real	$ X(\omega)  =  X(-\omega) $
$x(n)$	$\angle X(\omega) = -\angle X(-\omega)$

$x_e(\omega) = \frac{1}{2}[x(n) + x(-n)]$	$X_R(\omega)$
(Real y Par)	(Real y Par)
$x_o(\omega) = \frac{1}{2}[x(n) - x(-n)]$	$jX_I(\omega)$
(Real e Impar)	(Real e Impar) odd

## II.12.1 Espectro de Potencia

La TDF es comúnmente usada como una herramienta de análisis para determinar el espectro de una secuencia de entrada. Más comúnmente la amplitud de una componente de frecuencia particular en la señal de entrada. La TDF puede separarse en amplitud y fase como sigue:

$$\begin{aligned}X(f) &= X_{real}(f) + jX_{imag}(f) \\ \dot{X}(f) &= |X(f)|e^{j\theta(f)} \\ |X(f)| &= \sqrt{X_{real}^2 + X_{imag}^2} \\ \theta(f) &= \tan^{-1} \left[ \frac{X_{imag}}{X_{real}} \right]\end{aligned} \quad 2.71$$

Si estamos interesados en el espectro de potencia de la señal entonces

$$X(k) X^*(k) = |X(k)|^2 = X_{real}^2 + X_{imag}^2 \quad 2.72$$

Existen algunos problemas con la TDF como una herramienta de análisis de espectro. El problema de interés concierne al supuesto en la derivación de la TDF en que la secuencia contenga un solo periodo de una forma de onda periódicamente repetitiva, como se muestra en la figura 2.8. Para casi todas las secuencias habrá discontinuidades en la forma de onda en el tiempo, en los límites de estos seudoperiodos.

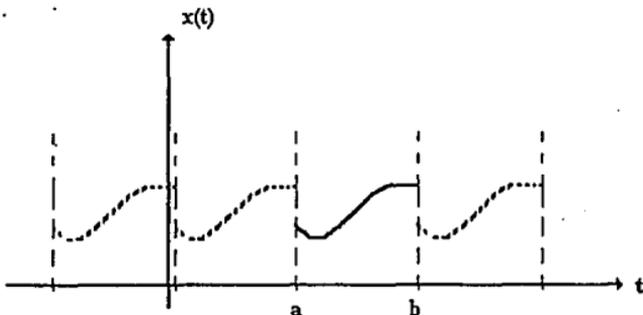


Figura 2.8 Haciendo una forma de onda periódica de un segmento

Esta discontinuidad acarrea componentes de muy alta frecuencia en la forma de onda resultante y debido a que estos componentes pueden ser mucho más altos que el límite del teorema del muestreo, se puede generar el fenómeno llamado "*aliasing*" dentro de la mitad del espectro.

La técnica usada para resolver este problema es llamada "ventaneo". El problema a resolver es la probable discontinuidad en el inicio y al final de cada periodo de la forma de onda. Debido a que el algoritmo de la TDF es de propósito general no hay forma de saber el grado de discontinuidad en los límites.

La técnica de "ventaneo" simplemente reduce la amplitud de la secuencia en los límites, esto lo hace en forma gradual y suave, por lo que no se producen nuevas discontinuidades y el resultado es una reducción substancial de las componentes de frecuencia que provocan "*aliasing*".

Esta mejora no viene sin un costo, dado que el ventaneo esta modificando la secuencia antes de que la TDF se realice debe esperarse alguna reducción en la fidelidad de la representación espectral. El resultado es la reducción en resolución de las componentes de frecuencia cercanas. Las mejores ventanas son aquellas que reducen al máximo el "*aliasing*", con el mínimo de degradación de la resolución espectral.

Existe una gran variedad de ventanas, pero todas ellas trabajan exactamente de la misma forma, atenuando los elementos de la secuencia cerca de los límites, (cerca de  $n=0$  y  $n=N-1$ ) y compensando al incrementar los valores lejanos de los límites.

## III Especificación del Sistema

### III.1 Conceptos Básicos del Diseño Orientado al Objeto

Las técnicas de ingeniería de *software* orientadas al objeto han generado un amplio interés en los últimos años. La naturaleza única del *Diseño Orientado al Objeto* (DOO), esta ligada a su habilidad para construir un *software* basándose en tres conceptos importantes del diseño: abstracción, ocultación de la información y modularidad.

Como otros métodos de diseño, para la derivación de un diseño de *software* el DOO introduce un nuevo conjunto de terminología, notación y procedimientos. Es por esto que para poder comprender el diseño del sistema es necesario comprender los conceptos subyacentes que guían todos los diseños orientados al objeto, y por lo tanto en esta sección nos concentraremos en la terminología del DOO.

La función del *software* es relacionada, cuando una estructura de datos (de distintos niveles de complejidad) se manipula mediante uno ó más procesos, de acuerdo a un procedimiento definido mediante un algoritmo estático ú órdenes dinámicas. Para crear un diseño orientado al objeto, debemos establecer los mecanismos para:

- 1) La representación de la estructura de datos;
- 2) la especificación del proceso; y
- 3) el procedimiento de llamada.

Un objeto es un componente del mundo real que se transforma en el "dominio" del *software*. En el contexto del *software*, un objeto es normalmente un procedimiento o consumidor de información, o elemento de información. Por ejemplo, objetos típicos pueden ser: máquinas, órdenes, archivos, cadenas alfanuméricas o cualquier otra persona, lugar o cosa. Cuando un objeto se transforma en una realización de *software*, consta de una estructura de datos privada y procesos, llamados operaciones ó métodos, que pueden transformar legítimamente la estructura de datos.

Las operaciones contienen el control y las construcciones que pueden ser llamadas mediante un mensaje - una petición al objeto para que ejecute una de sus operaciones -. La realización *software* de un objeto se muestra en la figura 3.1

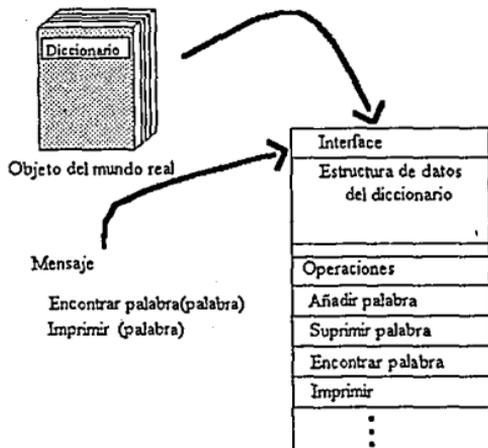


Figura 3.1 Objetos

Refiriéndonos a la figura, un objeto del mundo real (un diccionario), se transforma en una realización de *software*, dicha realización exhibe una estructura de datos privada y las operaciones relativas a ella. Las entradas del diccionario están compuestas de: palabras, una guía de pronunciación, una o más definiciones y también un dibujo o diagrama puede estar contenido dentro de una entrada. El objeto diccionario también contiene un conjunto de operaciones (p. ej., añadir palabra, encontrar palabra) que pueden procesar los elementos de la estructura de datos.

La parte privada de un objeto es la estructura de datos y el conjunto de operaciones para la estructura de datos. Un objeto tiene también una parte compartida o pública que es su interfase. Los mensajes se mueven a través de la interfase y especifican que operaciones del objeto se desean, pero no como se va a realizar la operación. El objeto que recibe el mensaje determina cómo se implementara la operación solicitada.

Definiendo un objeto con una parte privada y dando mensajes para llamar los procesamientos adecuados, conseguimos la ocultación de la información, esto es, los detalles de la implantación están ocultos a todos los elementos del programa, exteriores al objeto. Los objetos y sus operaciones dan una modularidad inherente, esto es, los elementos del *software* (datos y procesos), se agrupan junto con los mecanismos de interfaces bien definidos ( en este caso los mensajes).

De lo anterior se puede decir que un objeto contiene implícitamente encapsulamiento, por lo que para acceder a el puede definirse una estructura de datos y un grupo de procesamiento, de tal forma que los usuarios puedan acceder a ella mediante un conjunto de interfaces cuidadosamente documentadas, controladas y estandarizadas.

Estas estructuras de datos encapsulados, llamadas objetos, son equivalentes a datos activos que pueden ser solicitados para hacer cosas enviándoles mensajes.

Muchos objetos del mundo tienen unas características razonablemente similares y ejecutan operaciones razonablemente similares también. Si examinamos el suelo de una fábrica que construye equipos pesados, vemos máquinas que muelen, fresas cortadoras y taladradoras. Aunque cada uno de estos objetos es diferente, todos pertenecen a una clase más amplia, llamada herramientas de cortar metales. Todos los objetos de la clase herramientas de cortar metales tienen atributos en común (p. ej. todos utilizan motores eléctricos) y ejecutan operaciones comunes (p. ej. todos realizan corte de metales). Por lo tanto, categorizando un instrumento como un miembro de las máquinas de cortar metales, sabemos algo de sus atributos y de las operaciones que realiza, incluso sin conocer cual es su función detallada.

Las realizaciones *software* de objetos del mundo real están categorizadas de la misma forma. Todos los objetos son miembros de una clase más amplia y heredan la estructura de datos privada y las operaciones que se han definido para esa clase. Diciéndolo de otra forma, una clase es un conjunto de objetos que tienen las mismas características. Un objeto individual es por tanto una instancia de una clase más amplia.

Como ejemplo de la relación entre clases y objetos, retomemos el ejemplo del diccionario. El objeto diccionario es una instancia de la clase libros y hereda las características de datos y operaciones de libros. Por lo tanto, libros se convierte en una **abstracción de datos** que nos facilita la definición de cada instancia de una manera fácil, lo cual implica heredar los atributos de libros. Refiriéndonos a la figura 3.2, otros objetos p. ej. folletos, informes, enciclopedia, etc. pueden definirse como instancias de libros. De tal forma que solo necesitamos especificar en que difieren los nuevos objetos de los de la clase, en vez de definir todas las características del nuevo objeto.

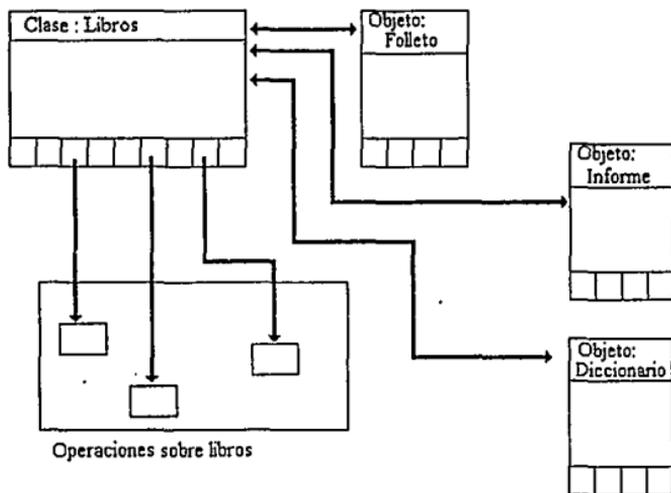


Figura 3.2 Objetos, Clases y Herencia

El uso de clases y herencia es crucial en la ingeniería moderna de *software*. La realización de componentes de programas se consigue creando objetos (instancias) que se construyen sobre los atributos y operaciones existentes que se heredan de una clase.

La herencia es útil por dos motivos. El primero es conservar el soporte de codificación, si se tiene una clase operacional que otra persona ha escrito. La herencia ayuda a reutilizar el código en la clase, no se necesita recurrir al código fuente y entender la implantación, se hacen cambios donde se considera necesario y se reutiliza el código ya existente.

El segundo motivo por el cual la herencia es útil y más sutil y eficaz, es el concepto de subclase orientada al objeto. Cuando una clase derivada hereda de una clase base, los objetos de la clase derivada retienen, todavía el número de miembros en la clase base. Mediante la derivación de muchas clases a partir de la misma clase base, se puede crear un grupo de clases que tengan una misma interface, pero diferentes implantaciones.

La utilización de interfaces idénticas con diferentes implantaciones se denomina *polimorfismo*, para ampliar un programa polimórfico basta con derivar una nueva subclase a partir de la misma clase base, que heredaron los otros objetos genéricos, entonces la nueva subclase puede manejarse por el mismo programa, ya que únicamente es un gestor para un conjunto de objetos genéricos. Por otro lado los errores se aíslan automáticamente en los mismos objetos.

### III.1.1 Representación Gráfica para el DOO

Los componentes descritos anteriormente pueden representarse gráficamente para ayudar a establecer las conexiones de la interface y dar una forma sencilla de reconocer la representación del diseño. Booch propone una notación, llamada algunas veces diagramas de Booch para los componentes, que se muestran en la figura 3.3 . En la figura un componente del programa (objeto), se representa por una caja que puede ser dividida en una parte de especificación (visible al exterior) y una parte privada (también llamada cuerpo del objeto), que permanece oculta para el exterior. La nube sin forma representa los detalles de implantación que son la parte privada o cuerpo del paquete y que aún no se han especificado.

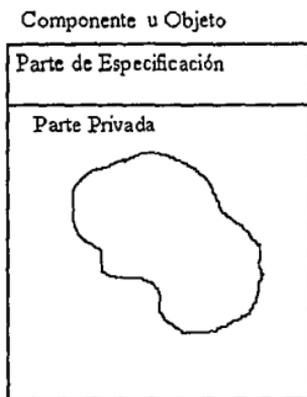


Figura 3.3 Representación gráfica para un Componente

Un paquete se representa por la notación mostrada en la figura 3.4, los objetos de datos se representan como rectángulos redondeados, mientras que las operaciones que actúan sobre los objetos se representan mediante rectángulos normales. De nuevo la forma de nube se utiliza para representar los detalles de implantación actualmente indefinidos.

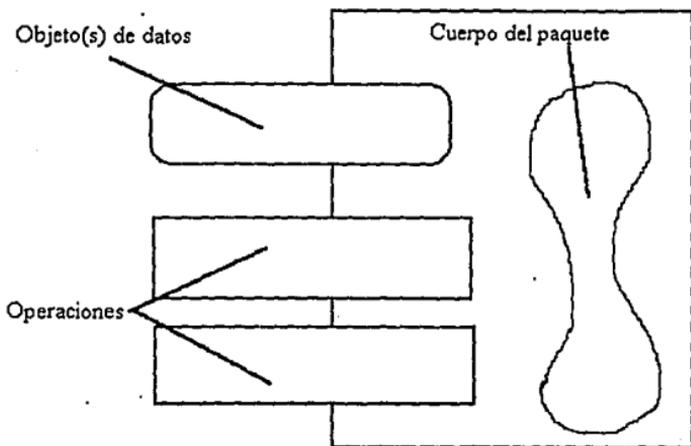


Figura 3.4 Notación de Paquete (Objeto)

La figura 3.5 ilustra el uso de los diagramas de Booch para representar las dependencias entre componentes del sistema. Las flechas de conexión indican dependencia, esto es, el paquete o componente en el origen de la flecha, depende del paquete o componente en la punta de la flecha. En la figura el componente de mas alto nivel, X, depende de los objetos y operaciones contenidos en los paquetes 1 y 2 para satisfacer sus funciones. El paquete 2 depende de los objetos y operaciones contenidos en los paquetes 3 y 4.

La programación orientada al objeto, es una nueva forma de pensar cómo resolver algunos de los problemas planteados en el mundo de la informática. En lugar de tratar de adaptar el problema a algún aspecto familiar a la computadora, la computadora se adapta al problema. Se ha sugerido que la programación orientada a objetos, surgió como resultado del caos que ocurrió cuando los lenguajes tradicionales trataron de enfrentarse a los grandes problemas, cuando los sistemas crecieron en complejidad. Entonces el modelo de caja negra ya no funciona debido a que los sistemas crecían, se les debía dar mantenimiento, y por lo tanto la caja negra se tenía que modificar. Los programas, realizados con lenguajes procedurales tradicionales, pueden ser difíciles de modificar y mantener. Por otro lado tenemos que, el diseño de un sistema orientado a objetos es modular, además de que su código fuente puede ser reutilizado con facilidad, mejorando los procesos de mantenimiento

Componente de más alto nivel

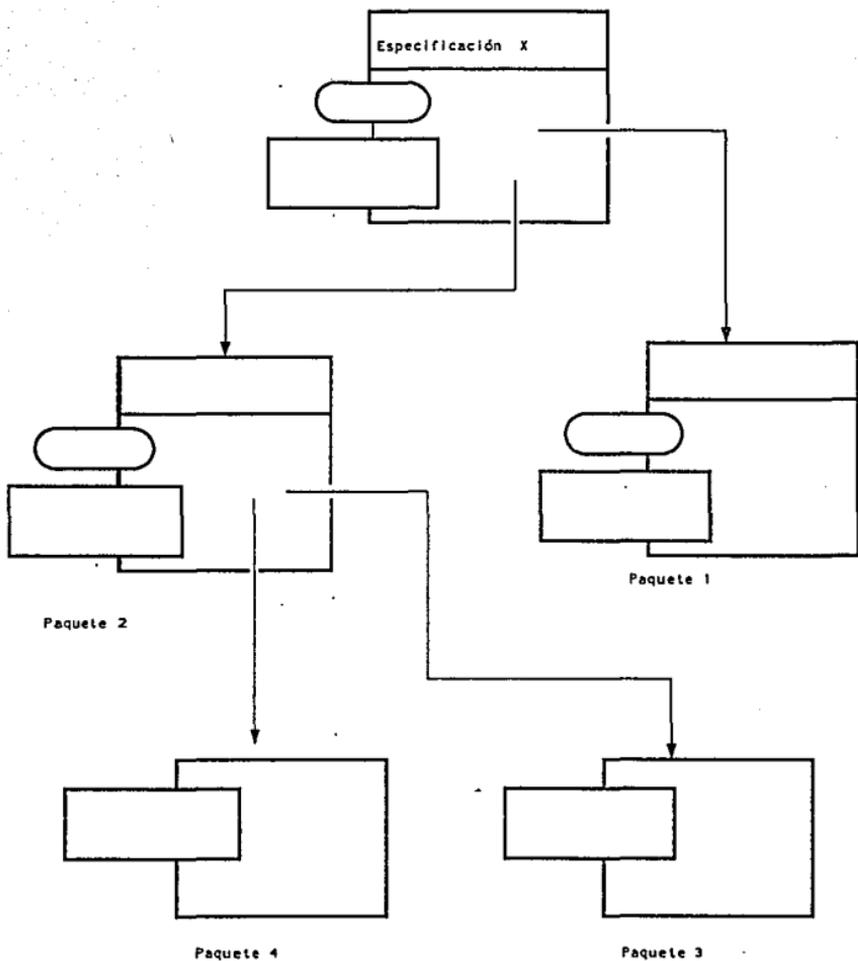


Figura 3.5 Dependencia entre componentes

### III.2 Definición del Problema y Objetivo

El estudio de los ritmos fisiológicos, así como un número considerable de disciplinas tradicionales en biología y medicina, dependen también del progreso de la ingeniería, beneficiándose de la tecnología para el monitoreo de las funciones del cuerpo humano, así como de los avances en el área de la computación. Las computadoras procesan los datos mucho más rápida y eficientemente, su contribución al estudio de los ritmos biológicos es aún más importante, precisamente por su alta velocidad de proceso y eficiencia, no sólo pueden resolver las tareas tradicionales sino también resolver nuevas tareas, por ejemplo detectar ritmos que no había sido posible anteriormente por la complejidad del análisis y el gran número de cálculos que esto requería.

La investigación en el área de cronobiología produce gran cantidad de datos, por lo que el manejo y análisis de los mismos es muy importante. Además es necesario tener más de un análisis cuyos resultados sean fáciles de interpretar, por lo que el despliegue de los mismos es de gran importancia. Los programas comerciales no están enfocados a los problemas específicos en cronobiología, por lo que el despliegue de los resultados es difícil de comprender, además de no contar con los análisis requeridos en el área, (ver tabla 3.1).

Cabe mencionar que existen en el mercado sistemas especializados fabricados en el extranjero que incluyen hardware y *software*, que realizan análisis específicos en el área, pero su precio es tan elevado que resulta inaccesible para nuestro medio, además de contar sólo con algunos de los análisis (alrededor de 3 o 4), evidentemente los más comunes, como son el actograma, periodograma, Fourier y cosinor. Por otro lado estos sistemas no tienen la posibilidad de crecer de acuerdo a las necesidades del usuario.

Algunos de los sistemas más comunes en el Procesamiento Digital de Señales para PC

Sistema	Compañía	Año	Precio US \$	Memoria Mínima en Bytes	Algoritmos y características	Generación de código
Monarch Professional , V.2.0	The Athena Group Inc.	1988, 1990	\$595	640K RAM	5 FIRs, 8 IRs, Diseño de variables de estado, polos-ceros y funciones de transferencia, 7 arquitecturas IIR, mas de 100 funciones programables, soporte extensivo de punto fijo, FFT-IFFT, salida a láser y plotter, se ejecuta directamente sobre la tarjeta seleccionada	TMS320C20, 25, DSP32, 32C DSP56000/1, HSP43220
DADISP V.2.0	DSP Development Corp.	1988, 1989	\$1695	2M	Sistema de procesamiento de señales tipo hoja de calculo, FIR, IIR, FFTs, espectro de potencia, convolución, generación de señales.	Información no disponible
MathLab	The MathWorks Inc.	1985	\$695 - \$1995	640 (PC) 2M (386)	IIR, FIR, análisis espectral 1D y 2D SSTs, correlación, convolución, Interpolación, cálculos con matrices, con gráficas 2D y 3D, opción a un modulo de toolbox de procesamiento de señales.	No Aplicable
ILS family	Signal Technology Inc.	1978, 1990	\$1000 \$2000	640K	Reconocimiento de patrones, diseño, análisis y desarrollo FFT, IIR y FIR, espectro de potencia, punto fijo y flotante, procesamiento de voz, convolución, compensación de cuantización, adquisición de datos, manipulación y transferencia de archivos	Punto Fijo (a tra vez de Momentum), punto flotante ( a tra vez de ASPI), SPOX server

Tabla 3.1 Se muestran algunos de los sistemas ó paquetes de análisis de señales (Procesamiento Digital de Señales -DSP- ), los datos fueron obtenidos de una revisión publicada en IEEE Spectrum, Noviembre de 1990.

Un aspecto importante en cronobiología es el poder caracterizar los fenómenos rítmicos satisfactoriamente, de tal forma que pueda realizarse una estimación objetiva de sus parámetros y perturbaciones del ritmo. Por lo que requiere el uso de técnicas gráficas y estadísticas específicas, las cuales como se observa en la tabla no se encuentran en los sistemas comerciales.

Por lo anterior surge la necesidad de diseñar e implementar un sistema de análisis de señales biológicas en cronobiología, el cual responda a semejantes demandas. Las ventajas de esto serían varias, entre ellas un conjunto de análisis seleccionado por el investigador, despliegue de resultados fáciles de comprender, organización y almacenamiento de datos óptimo para un mejor aprovechamiento de los recursos, y capacidad de expandirse de acuerdo a las necesidades del usuario. De tal forma que las características para dicho sistema son:

- Importación/Exportación y Organización de datos.
- Análisis matemático de series de tiempo acorde con las necesidades del usuario.
- Despliegue gráfico, fácil de comprender.
- Salida a impresora para la generación de reportes.
- Facilidad en el manejo del sistema.
- Capacidad de crecimiento acorde a las necesidades del usuario.

### III.3 Entorno del Sistema

Para el desarrollo del sistema se seleccionaron algunas técnicas para su implantación, ( seleccionadas por el usuario ), y se decidió desarrollar e implementar para una micro computadora personal del tipo IBM-PC y compatibles bajo el sistema operativo DOS y en lenguaje C++.

Las técnicas digitales han llegado ha ser el método de selección en procesamiento de señales, debido a que las computadoras han incrementado su poder de cálculo y velocidad, así como la disponibilidad de poderosos microprocesadores. Por otro lado existen dos grandes áreas de aplicaciones de *software* en procesamiento digital de señales:

- 1) Aplicaciones donde el *software* es usado para simular hardware y
- 2) Aplicaciones donde el *software* es el producto final por si mismo ( nuestro caso)

Los lenguajes C y C++ se pueden considerar como lenguajes de alto-nivel, los cuales pueden ser utilizados para ambas áreas. Tienen aspectos de lenguaje de alto-nivel que permite crear simulaciones y permite crear código eficiente el cual se aproxima al creado en lenguaje ensamblador. Para aplicaciones en tiempo real , el lenguaje C++ tiene ventajas significativas ( para aplicaciones en procesamiento digital de señales ) , sobre otros lenguajes como FORTRAN ó PASCAL.

Una importante razón es la utilidad de las estructuras de datos en C++, para las tareas de procesamiento digital de señales, también la inherente modularidad de C++, la cual es una característica importante al programar.

El procesamiento digital de señales utiliza repetidamente un conjunto de herramientas matemáticas, las cuales están bien definidas, con pequeñas variaciones en los parámetros. Los lenguajes C y C++ están contruidos para soportar el desarrollo de rutinas de librería general, las cuales pueden ser utilizadas para construir bloques en la forma requerida para el procesamiento digital de señales.

Otra razón para la utilización de este lenguaje es la popularidad y el esparcimiento en el uso de este lenguaje. Por otro lado los fabricantes de dispositivos de procesamiento digital de señales ( como Texas Instrument, AT&T, Motorola, etc. ), proveen compiladores de C para estos dispositivos (circuitos).

Se seleccionó una computadora personal del tipo IBM-PC y compatibles por ser una de las de mayor uso, y por el fácil acceso a estas. Por las mismas razones se selecciono el sistema operativo ya que es DOS el sistema que estas computadoras utilizan con mayor frecuencia. Para la Implantación del sistema se utilizo el compilador de Borland C++.

Como se mencionó anteriormente en la introducción, en el laboratorio de cronobiología se cuenta con un sistema de registro, sin embargo este sistema se encuentra implantado en una Apple IIe, por lo que para poder analizar los datos en el sistema propuesto, estos deben ser transferidos a una computadora del tipo PC, ya que de otra forma no se podría utilizar, entonces las preguntas obvias son: ¿ Porque el sistema de registro no se implemento en una PC?, ¿Porque no implementar el sistema de análisis en una Apple IIe? y consecuentemente las respuestas son: el sistema de registro se implemento desde hace tiempo y en ese entonces las Apple IIe eran las computadoras más difundidas en México, actualmente es muy difícil encontrar repuestos para estas computadoras y por lo tanto el mantenimiento y *software* de soporte es prácticamente inexistente en el mercado, por otro lado para las computadoras del tipo PC, sucede todo lo contrario, es por ello que el sistema de análisis se implemento en una computadora del tipo PC, además de haber demostrado que se han mantenido por más tiempo como las computadoras de mayor uso.

El hecho de utilizar una PC, entonces crea el problema de la lectura de los datos, ya que estos son almacenados en un formato para Apple, por lo que se desarrollaron pequeños programas, para la transferencia de la información, desde la Apple hacia la PC. Una vez teniendo los datos en la PC, su manejo para el sistema de análisis es más sencillo.

### III.4 Diseño del Sistema

La meta principal en el desarrollo del sistema de análisis aplicado a cronobiología, fue proveer un conjunto de manipulaciones estándares para señales en cronobiología. Una segunda meta fue que el sistema fuese fácil de utilizar y que fuese capaz de relacionarse con otras aplicaciones.

El sistema resultante llamado DiSPAC (de sus siglas en ingles Digital Signal Processing Applied to Chronobiology), es un sistema orientado al objeto, el cual permite al usuario manipular interactivamente la señal mediante una interface gráfica, también orientada al objeto.

Se utilizó un diseño orientado al objeto para el desarrollo de la arquitectura de alto nivel. Primero se identificaron los principales componentes del sistema y las interacciones entre ellos para después particionarlos.

#### III.4.1 Descripción de la información.

De acuerdo a lo descrito en las secciones anteriores, se observa que el sistema debe tener el siguiente flujo de información: analizar una señal para después desplegar o imprimir el resultado, o bien tan solo desplegar e imprimir la señal. De esta forma tenemos que el primer componente es la señal, sin embargo esta cae dentro del espacio del problema debido a que ya se cuenta con ella. El segundo componente es el análisis a efectuar, el cual cae dentro del espacio de la solución. Y por último tenemos el desplegado en pantalla ó impresión, este ultimo componente también cae dentro del espacio de la solución. En la figura 3.6 se pueden apreciar estos componentes así como el flujo que sigue la información.

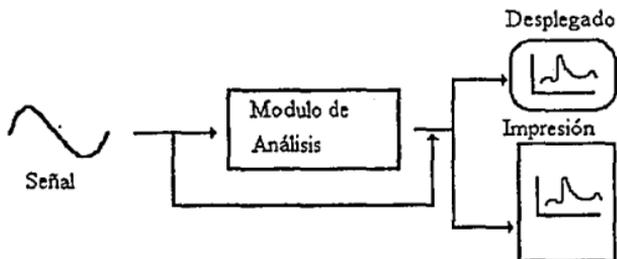


figura 3.6 Componentes iniciales del sistema de análisis, también se observa el flujo de la información

Evidentemente estos dos componentes son la parte medular del sistema y para dar una solución funcional, bastaría generar un grupo de programas que realizan las operaciones y otro que despliegue los resultados, sin embargo el sistema también debe ser fácil de usar. De tal forma que para que esto se cumpla a estos componentes los debe manejar un tercero, el cual tenga la interacción directa con el usuario. Este componente cae entonces dentro del espacio de la solución y recibe el nombre de interface del usuario.

Las interfaces del usuario han llegado a ser más importantes con el incremento en el número de usuarios y aplicaciones, también es cierto que anteriormente no se utilizaba adecuadamente el *software* por tener una interface difícil o incomoda. Sin embargo las actuales interfaces gráficas (GUI), se ha mostrado que el *software* puede ser fácil de aprender a usar y cómodo en su uso.

Como en cualquier sistema que se genera se desea tener una alta utilización, pero esta no aparece mágicamente porque nosotros la deseamos, este reside básicamente en la interface del usuario y la utilidad del sistema desarrollado. Para asegurar la utilización del sistema interactivo debemos tomarlo en cuenta durante el diseño y el desarrollo.

De lo anterior podemos entonces hacer un diagrama con una descripción más detallada del flujo de la información y de los componentes del sistema. Esto se puede apreciar en la figura 3.7.

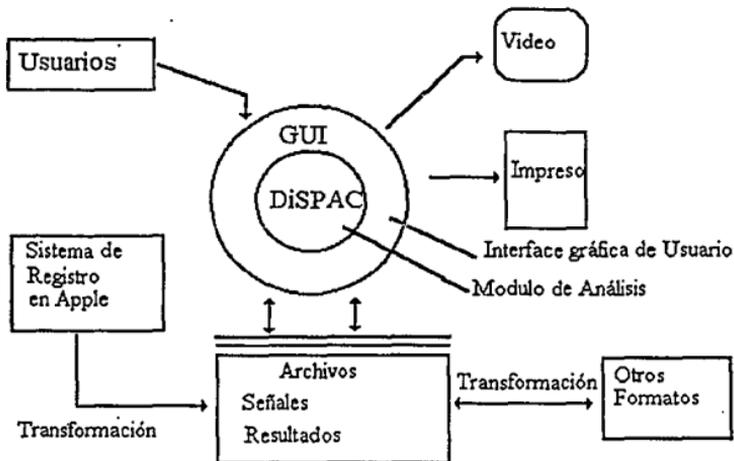


Figura 3.7 Descripción del flujo de la información y de los componentes del sistema

### III.4.2 Objetos Atributos y Operaciones

Una vez que se determinaron los principales componentes del sistema, se particionan para dar solución al problema y por ende a la arquitectura del sistema propuesto. De acuerdo a la figura 3.7 se puede apreciar que la interface gráfica es el medio por el cual el usuario se comunicara con el sistema, y consta: de ventanas, *iconos*, barras y cualquier otro objeto gráfico. Por otro lado se tiene que el módulo de análisis es el que efectuará el procesamiento de la señal, el cual contará con más de un análisis, consistiendo entonces de operaciones sobre una señal.

Otro punto importante es el manejo de los archivos ya sea de datos (señales a analizar) ó resultados generados por el sistema. Para cada operación sobre la señal se genera un resultado, este puede tener las siguientes: opciones, ser almacenado, visualizado ó impreso. Para la primera opción se turnará al manejador de archivos, mientras que para las otras dos opciones se turnará a la interface gráfica, se observa entonces que la interface gráfica no sólo consta de objetos gráficos, sino de manejo de dispositivos externos tales como: video, impresora, teclado y *mouse*, siendo los dos últimos por medio de los cuales el usuario solicitará un requerimiento.

Cuando el usuario requiere una operación sobre una señal, lo hace mediante la interface de usuario, esta a su vez comunica la petición al núcleo del sistema para que se realice la operación al terminar la operación se retorna a la interface gráfica la cual despliega el resultado, teniendo las opciones de imprimirlo ó almacenarlo en disco. Esta comunicación es completamente definida por el protocolo entre la interface del usuario y el núcleo del sistema. El protocolo es implantado por el módulo de la interface del usuario y un módulo de aplicación, el cual controla las llamadas al núcleo. Esta descripción se puede apreciar en la figura 3.8 donde se muestra la arquitectura de DiSPAC.

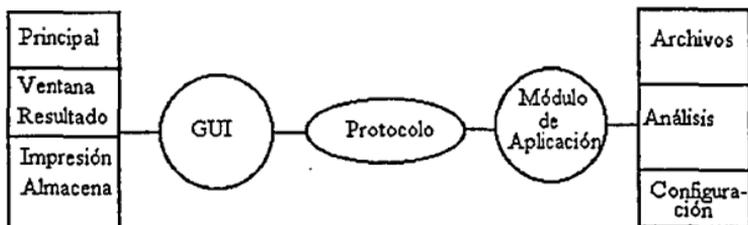


Figura 3.8 Arquitectura de DiSPAC

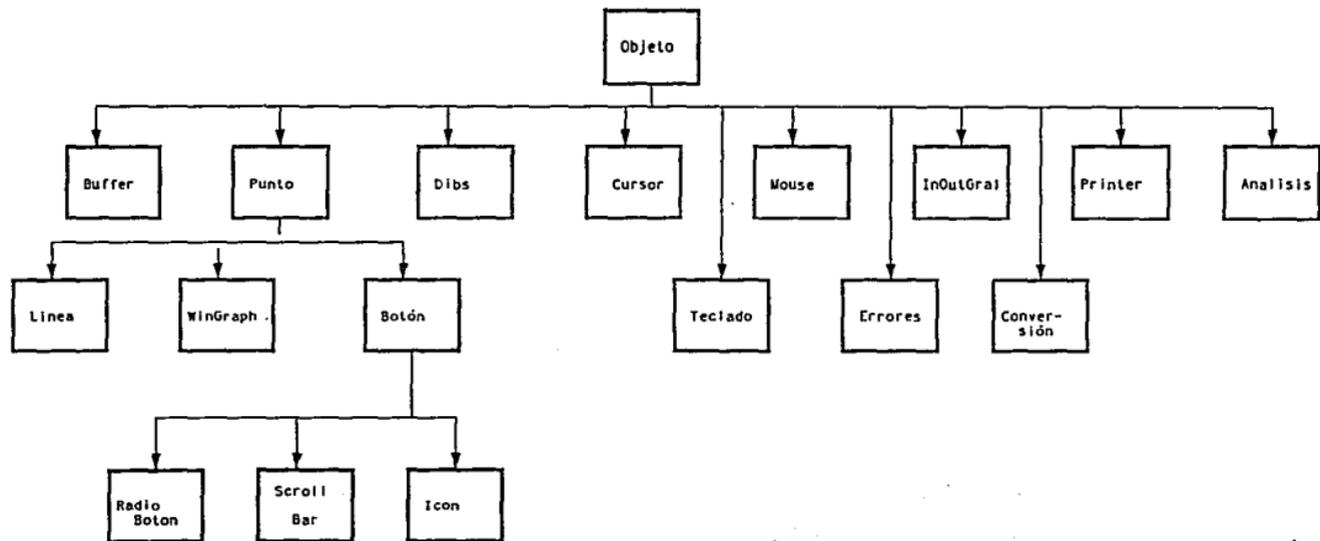


Figura 3.9 Jerarquía de objetos

A partir de la descripción anterior y basándonos en las figuras 3.7 y 3.8, se partitionaron los componentes. La interface gráfica se compone de los siguientes objetos:

Objetos Gráficos : Ventana Gráfica, Botones, Iconos, Resultado Gráfico, Lineas, y Barras

Objetos de Requerimientos : *Mouse* y Teclado

Objetos de Salida : Impresora

La lectura y almacenamiento de los archivos se realiza mediante el manejador de archivos, el cual hace uso de un grupo de objetos que son instancias de la clase InOutGral, la cual es una clase general, de esta forma aprovechamos el polimorfismo.

Para el análisis se utiliza la clase análisis, la cual procesa la información.

La jerarquía de clases se puede apreciar en la figura 3.9, nótese que se tiene una clase llamada errores, esta clase tiene el objetivo de generar todos los mensajes de errores posibles dentro del sistema y esta estrechamente ligada a la interface gráfica.

A continuación se presenta la especificación para cada una de las clases definidas en la figura 3.9, comenzando por la interface gráfica.

La interface gráfica se deriva a partir de la clase punto, la cual nos sirve como la clase abstracta, ya que en ella se especifican las principales funciones que debe cubrir un objeto gráfico, teniendo las siguientes funciones básicas:

#### Clase Punto

##### Estructura de datos:

- Coordenadas donde reside el objeto
- Color del objeto
- referencia gráfica

##### Operaciones:

- Crear
- Dibujar
- Mover
- Restaurar
- Poner color
- Posicionar
- Borrar
- Obtener color
- Obtener posición actual

La implantación del objeto así como la definición, se ven un poco modificadas, sin embargo no pierden todo el concepto establecido, ya que solo es una forma de escribirlo, para esta clase se muestra la parte de definición en lenguaje C++:

```
// se define el nombre del archivo

#ifndef CPUNTO_H
#define CPUNTO_H

// se verifica si no se han incluido los siguientes archivos
// interfaces
#ifndef GRAPHICS_H
#include <graphics.h>
#define GRAPHICS_H
#endif

// -----

// Clase punto :
// variables
// x,y : Posición en pantalla
// Color : Color del punto
// Vref : Viewport de referencia
// Funciones
// Punto : Constructor (Default)
// Mueve : Mueve el punto a las nuevas coordenadas
// Draw : Definida como virtual para las clases derivadas
//         Dibuja un punto
// Crea : Crea un objeto de tipo punto
// Restauraviewport : Restaura el Viewport de modo gráfico
// SetColor : Da nuevo Color al objeto
// SetLoc : Da nueva posición al objeto, definida como virtual
// Borra : Borra un objeto previamente dibujado, virtual
// Drag : Mueve el objeto a través de la pantalla
//         Mediante mouse o teclado
// GetColor : Obtiene el color del objeto
// GetX : Obtiene la coordenada x del objeto
// GetY : Obtiene la coordenada y del objeto
// Setdes : Define modo de operación para el destructor del objeto.
```

```
// Definición formal en lenguaje C++
```

```
class Punto
{
protected:
  int x,y,Color;
  viewporttype Vref;
public:
  char dsf;
  Punto(); //Constructores
  Punto(int ptX, int ptY, int C);
  void Mueve(int ptX, int ptY);
  virtual void Draw();
  void Crea(int ptX, int ptY, int C);
  void Restauraviewport();
  void SetColor(int C);
  virtual void SetLoc(int ptX, int ptY);
  virtual void Borra();
  virtual void Drag(int Cuanto);
  int GetColor();
  int GetX();
  int GetY();
  void Setdes(char V){ dsf=V;}
};
```

```
// Prototipos definidos en CPUNTO.CPP
```

```
// Prototipo de propósito general, no miembro
```

```
// definida en cpunto.cpp
```

```
int GetDelta(int& DeltaX, int& DeltaY);
```

```
#endif
```

Siendo la clase punto la clase abstracta de donde se derivan las demás clases se observa que no tiene un inicializador para el modo gráfico. Esto se debe a que la inicialización sólo ocurre una vez, de tal forma que se realizó mediante una rutina específica, he aquí una de las ventajas de C++, el cual permite mezclar técnicas de programación. De esta forma la inicialización utiliza la clase externa definida en graphics.h, la cual es propia del compilador de Borland.

La siguiente clase a definir es la clase Línea, la cual es una instancia de la clase punto y por lo tanto hereda la estructura de datos y las operaciones, sin embargo es necesario adicionar algunas más, entonces la especificación queda como sigue:

Clase Línea derivada de Punto

Estructura de datos:

Diferencia en X e Y

Operaciones:

Invierte una línea

Aumenta el tamaño de la línea

Crea línea con un período específico

Determina el período de una línea dados dos puntos

Obtiene diferencia en X

Obtiene diferencia en Y

La clase línea se utiliza en DiSPAC como una herramienta de análisis gráfico es por esto que cuenta con las operaciones de, calcular y obtener un período. Dentro del sistema se tienen definidos 7 objetos de esta clase.

A partir de la clase punto se deriva también la clase ventana, llamada WinGraph, la cual es la encargada de dibujar una ventana con identificador, dentro de la pantalla, de tal forma que cuando se genere un nuevo objeto gráfico, todo esté referenciado hacia la ventana. La definición de la misma se crea a partir de la clase punto teniendo por tanto las mismas operaciones, sin embargo al igual que las líneas necesita de algunas adiciones, por lo que la especificación queda como sigue:

Clase WinGraph derivada de Punto

Estructura de Datos :

Número de ventana

Coordenadas del cursor en la ventana

Ancho y alto de la ventana

Modo de operación

Objeto *buffer*

Operaciones

Activa la ventana

Desactiva la ventana

Limpia el interior de la ventana

En esta clase se cuenta con un modo de operación el cual le indica a la ventana la forma que almacena la información que se encuentra en donde será dibujada, las opciones son: no almacenarla, almacenarla en memoria y almacenarla en disco. Para el almacenamiento en memoria se cuenta con un objeto llamado *Buffer* dentro de su estructura de datos el cual se encarga de leer información de video y almacenarla en memoria con la posibilidad de escribir de nuevo en video, en la posición que se le indique y su especificación es como sigue:

Clase Buffer

Estructura de Datos

Apuntador

Operaciones

Reserva espacio en memoria

Libera el espacio reservado

Lee de video

Escribe en video

La siguiente clase a definir recibe el nombre de Botón, y se encarga de dibujar y controlar un botón, el cual servirá como elemento de selección para el usuario, y se deriva de la clase punto, teniendo la siguiente especificación:

Clase Botón derivada de Punto

Estructura de Datos

Tipo de botón

Texto del botón

Ancho y alto del botón

Operaciones

Pon texto en el botón

Define tipo de botón

Cambia apariencia

Define alto de botón

Define ancho de botón

Cambia estado del botón

Obtén estado del botón

Obtén ancho

Obtén alto

Verifica que botón este activo ó desactivo.

La clase Botón, utiliza dentro de la última operación los objetos genéricos *Mouse* y Teclado, los cuales se definirán más adelante, ya que por medio de estos determina si el botón fue seleccionado por el usuario.

A partir de la clase Botón se definen la clase *Icon*, la cual es la encargada de llevar el control sobre cada uno de los *iconos*, desplegados en la pantalla, teniendo la posibilidad de definir el dibujo del icono, mediante un programa adicional, obviamente para cada cambio de dibujo se debe recompilar la aplicación, no sin antes haber definido el nuevo dibujo, el dibujo reside dentro de un arreglo de números llamado DICON, el cual forma parte de la estructura de datos, sin embargo la modificación se realiza mediante una de sus operaciones. Su especificación queda de la siguiente forma:

#### Clase Icon Derivada de Botón

##### Estructura de datos:

DICON, arreglo donde se almacena el dibujo del Icono

Estado de *Icon*

Modo de operación

##### Operaciones:

Activa *Icon*

Desactiva *Icon*

IconActivo

El modo de operación al que se refiere es básicamente donde se detectará la activación de *icono*, ya que este puede tener un botón en la parte inferior para su activación, o bien puede ser activado seleccionando el propio *icono*. Por otro lado al igual que Botón utiliza los objetos genéricos *Mouse* y Teclado para la detección de selección.

Otra de las clases derivadas de botón es la clase Barra, llamada Scroll-Bar, la cual es útil para definir las nuevas posiciones de cualquier objeto gráfico que deseemos, para nuestro caso, es útil en el movimiento de las líneas en la pantalla de trabajo. Su especificación a partir de Botón es como sigue:

#### Clase ScrollBar derivada de Botón

##### Estructura de Datos

Posición en la barra

Tipo de barra (vertical u horizontal)

##### Operaciones

Mueve indicador

Obtén posición

Obtén dirección

Obtén por ciento dentro del espacio de la barra

Dibuja indicador

Borra indicador

Detecta movimiento

La clase ScrollBar también utiliza los objetos genéricos *Mouse* y Teclado para la detección de la selección.

La última clase que se deriva a partir de Botón, es la clase Botón circular llamada RadioBotón y la diferencia en cuanto a la clase Botón es básicamente el aspecto, ya que el botón es circular en vez de rectangular. Su definición es la siguiente:

Clase RadioBotón derivada de Botón

Estructura de datos:

Radio del botón

Operaciones

Define radio del botón

Obtén radio del botón

Se ha hablado de las clases genéricas teclado y *mouse*, las cuales son de uso global y se puede decir que son dos de las principales clases que forman el núcleo de la interfase gráfica. La clase teclado, como su nombre lo indica es la encargada de manejar el teclado de la computadora. La especificación para la clase Teclado, es la siguiente :

Clase Teclado derivada de Bios.h (externa de C)

Estructura de Datos:

Tecla leída

Objeto de tipo cursor

Operaciones :

Limpia área donde se escribe

Muestra texto leído en pantalla

Lee un número de tipo entero

Lee un número de tipo punto flotante

Lee una cadena de caracteres

Mueve cursor

Mueve hasta que sea *enter*

Observe que la clase teclado tiene en su estructura de datos un objeto del tipo cursor, el cual tiene como objetivo el simular el cursor del *mouse*, en el caso de no contar con un *mouse*. En caso de contar con un *mouse*, se tiene la clase llamada *mouse* la cual controla todos los eventos generados por el *mouse* externo. La especificación para la clase *mouse* es la siguiente:

### Clase *Mouse*

#### Estructura de datos

- Indicador para la visualización
- Indicador del evento del *mouse*
- Posición del *mouse*
- Estado del *mouse*

#### Operaciones

- Define movimiento
- Mueve a nueva posición
- Resetea *mouse*
- Obtén posición de *mouse*
- Mouse* presionado
- Mouse* liberado
- Muestra *mouse*

Otra clase importante dentro de la interface gráfica es la clase encargada de dibujar una función de dos variables dentro de un área determinada, teniendo la posibilidad de cambiar su aspecto, esta clase es llamada *Dibs*. La especificación de la clase es como sigue:

### Clase *Dibs*

#### Estructura de Datos

- Color de la gráfica
- Límites para x & y
- Tipo de gráfica
- Factores de escalamiento
- Área donde será dibujada

#### Operaciones

- Escala ejes
- Convierte coordenada de mundo real a mundo virtual
- Convierte coordenada virtual a coordenada real
- Define color
- Cambia aspecto
- Define coordenadas
- Dibuja gráfica
- Analiza un par de puntos
- Borra gráfica.

Otra clase que es importante, es la clase llamada *Printer*, la cual se encarga de controlar la impresora e imprimir las gráficas mostradas en la pantalla. Su definición es como sigue

### Clase *Printer*

#### Estructura de datos

- Opción de impresión
- Tipo de impresora
- Orientación

#### Operaciones

- Inicializa Impresora
- Lee Gráfico
- Imprime Gráfico
- Envía salto de página
- Cambia modo de impresora

La clase llamada errores, provee un conjunto de operaciones, para el manejo de los posibles errores, por medio de la interface gráfica, siendo su especificación como sigue:

### Clase Errores

#### Estructura de datos

- Coordenadas del mensaje
- Objeto ventana donde se desplegará el mensaje
- Acción

#### Operaciones

- Restaura estado
- Muestra mensaje
- Mensaje normal : aborta proceso
- Mensaje anormal : aborta programa
- Mensaje con opción : opción a continuar o abortar

Para la lectura de datos y almacenamiento en disco se utilizó una clase llamada *InOutGral*, la cual controla todos las lecturas de datos y escrituras de los mismos en otros formatos. Su especificación es la siguiente:

### Clase *InOutGral*

#### Estructura de datos

- Objeto ventana para el desplegado de información

#### Operaciones

- Obtén Datos
- Escribe Datos
- Obtén Parámetros del archivo

Completa parámetros  
Informa sobre el archivo

Aunada a esta clase y aprovechando el polimorfismo se creó una clase llamada conversión, la cual es un manejador de la clase *InOutGral*, y nos es útil para la conversión de archivos entre diferentes formatos. Su especificación es como sigue:

#### Clase Conversión

##### Estructura de datos

Estado de la conversión  
Apuntador para datos  
Dos objetos *InOutGral*

##### Operaciones

Define entrada  
Define salida  
Informa sobre archivo de entrada  
Obtén estado del proceso  
Convierte de entrada a salida

Por último tenemos la clase de análisis la cual es una clase abstracta donde lo único que cambia es el proceso que se le hace a la señal, de esta forma se aprovecha nuevamente el polimorfismo. La especificación para la clase análisis es la siguiente:

#### Clase Análisis derivada de math.h (externa de C)

##### Estructura de datos

Apuntadores para almacenamiento de información temporal

##### Operaciones

Verifica datos.  
Subproceso1  
Procesa la señal

Una vez que se tiene la especificación de cada una de las clases que intervienen en el sistema, se genera la aplicación de acuerdo a la arquitectura mostrada en la figura 3.8. Para el control de cada uno de los componentes del sistema se generan subsistemas que controlan a los diferentes objetos, estos subsistemas se comunican a través de los estados de los objetos llamados gráficos, tales como Botón, Icon y Radio Botón, los cuales forman el protocolo entre los diferentes subsistemas. Este agrupamiento en los diferentes subsistemas se puede apreciar en la figura 3.10. Por otro lado en la figura 3.11 se observa la descripción en diagramas de Booch, de algunas de las especificaciones descritas anteriormente.

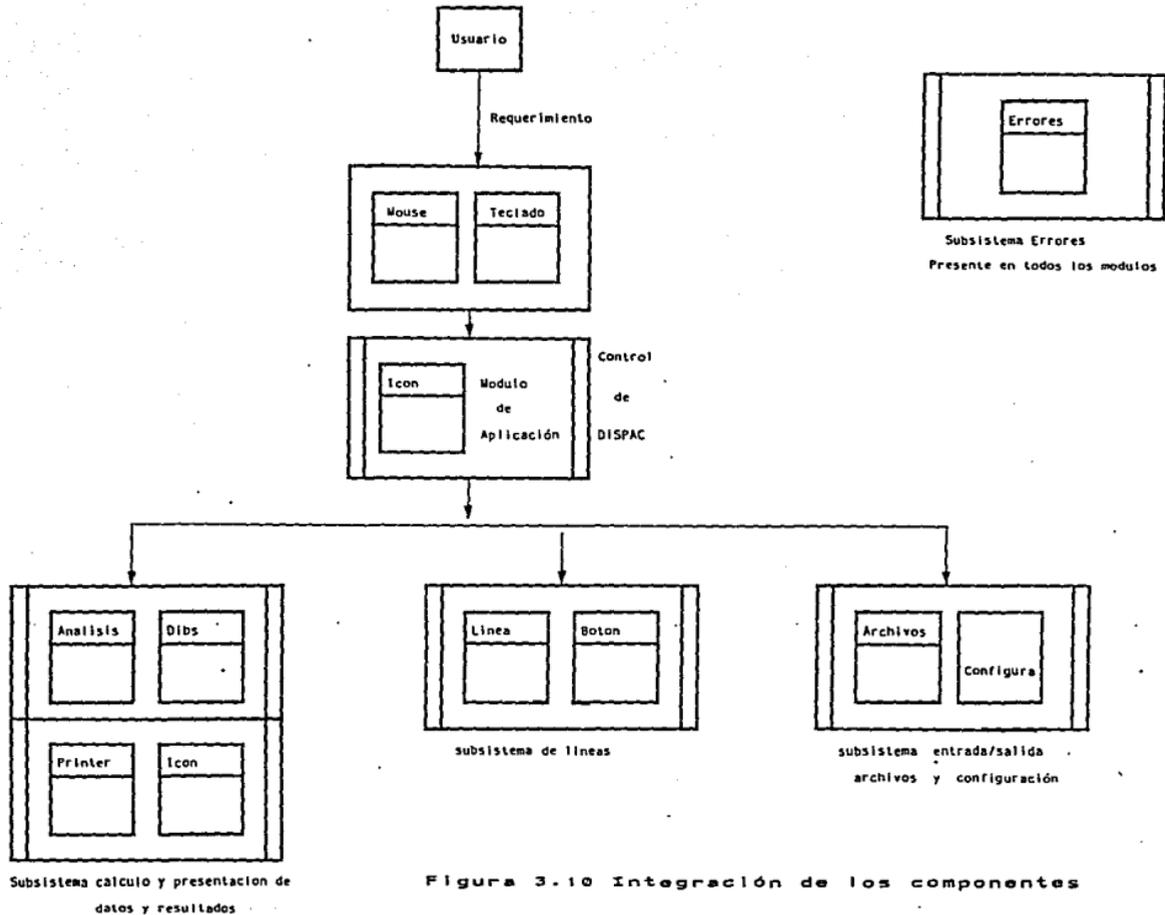


Figura 3.10 Integración de los componentes

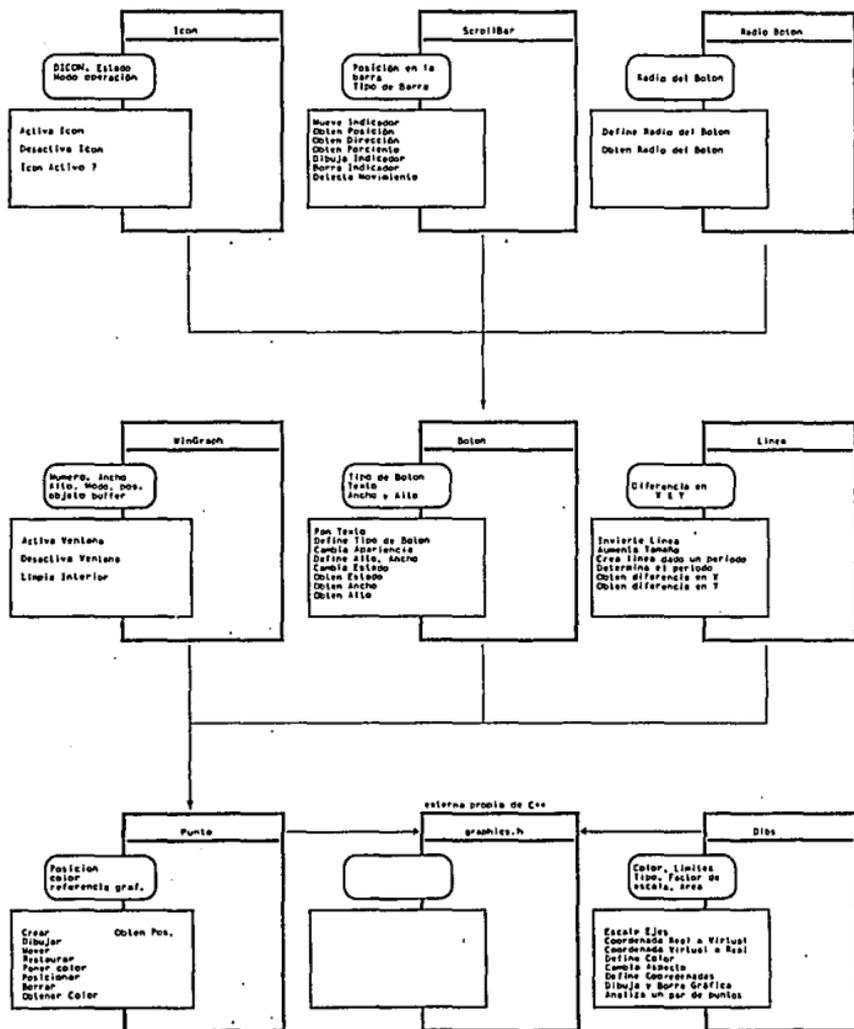


Figura 3.11 Representación en diagramas de Booch.

### III.4.3 Descripción Funcional Del Sistema

La descripción funcional del sistema es muy sencilla, una vez descrita la arquitectura del sistema y la forma en que operan los objetos que componen a cada uno de los componentes del sistema. De acuerdo a las figuras 3.7, 3.8 y 3.10 se observa que donde reside la funcionalidad es en el módulo de aplicación. Por lo que partiendo del modulo de aplicación podemos generar el diagrama de flujo que se muestra en la figura 3.12, donde podemos observar que el primer paso es la inicialización de la interface gráfica y la lectura de la configuración, la cual le indica al sistema en que tipo de formato estarán los datos a leer y los archivos que se utilizarán en la representación del actograma ( archivo de paletas y umbrales).

Una vez inicializado el sistema espera que el usuario genere un requerimiento, el cual debe consistir en especificar la lectura de datos, ya que sin datos no se podrá realizar ninguna operación. Este requerimiento como ya lo hemos mencionado es detectado mediante el teclado y el *mouse*, al seleccionar alguna opción ejecuta el proceso indicado, para cada opción se tiene un proceso el cual puede tener un submenú, las selección entonces se realiza sobre los botones circulares ó radio botones.

Cuando se selecciona un análisis, al terminar de calcularlo se muestra su resultado en forma gráfica, solo algunos casos lo muestran en forma tabular, esto se despliega dentro de una ventana, la cual contendrá no solo el resultado gráfico, sino que además las opciones para imprimir, cambiar aspecto ó analizar un punto dentro de la gráfica.

La apariencia final del sistema se muestra en las figuras 3.13 y 3.14, donde se pueden observar el aspecto que el sistema tiene, así como las opciones con las que cuenta. Una descripción más detallada sobre el uso del sistema se puede consultar en el manual del usuario.

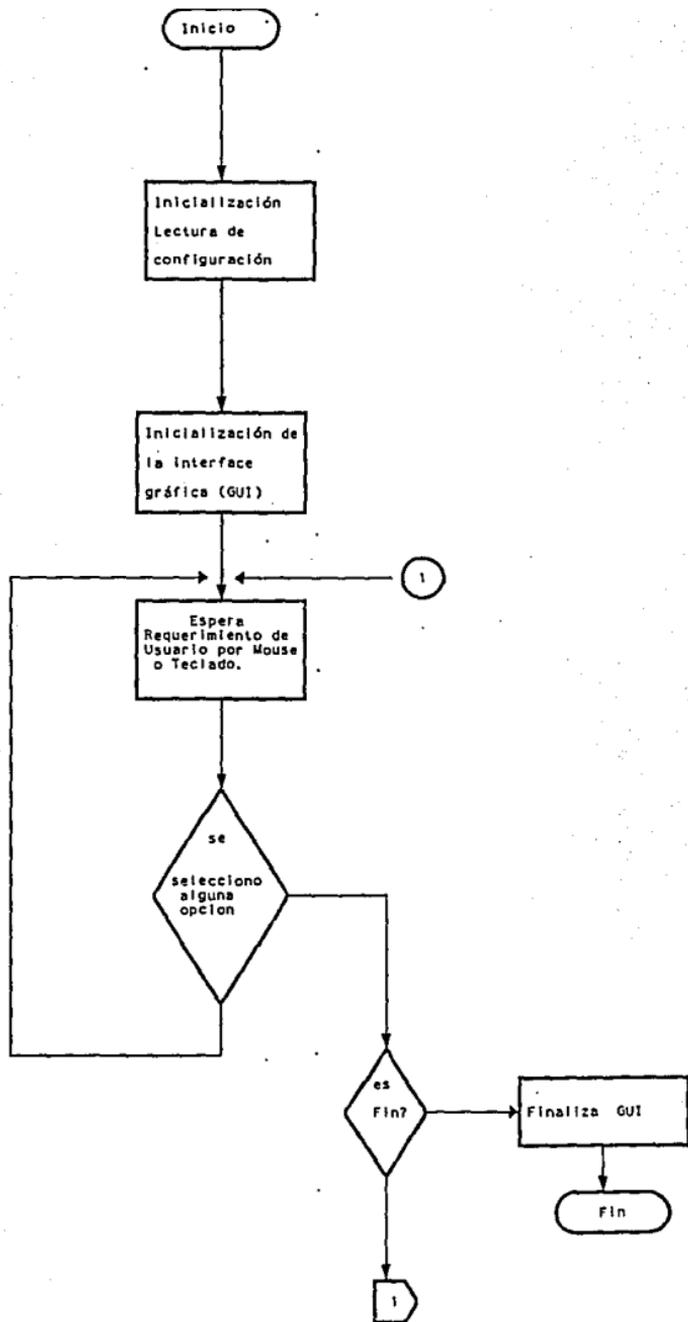


Figura 3.12 Diagrama de Flujo

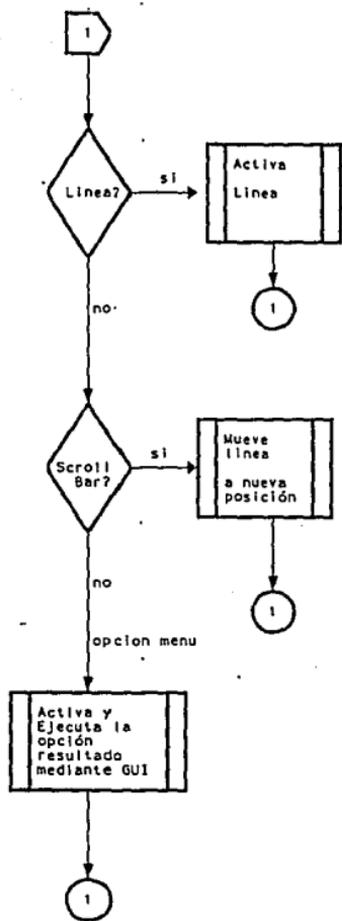


Figura 3.12: Continuación

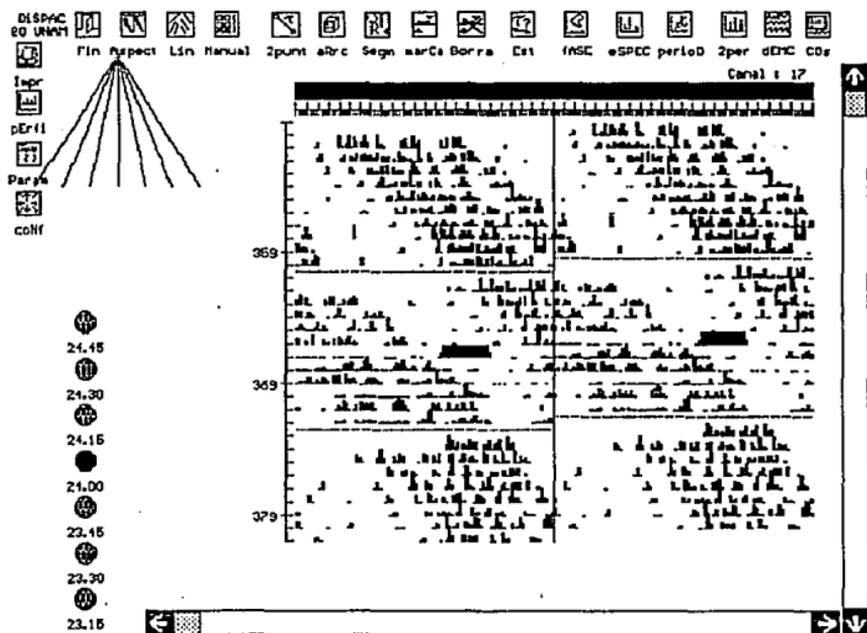


Figura 3.13 Pantalla de trabajo de DiSPAC, conteniendo un conjunto de datos ya leídos y desplegados en la representación de actograma en Double-Plot

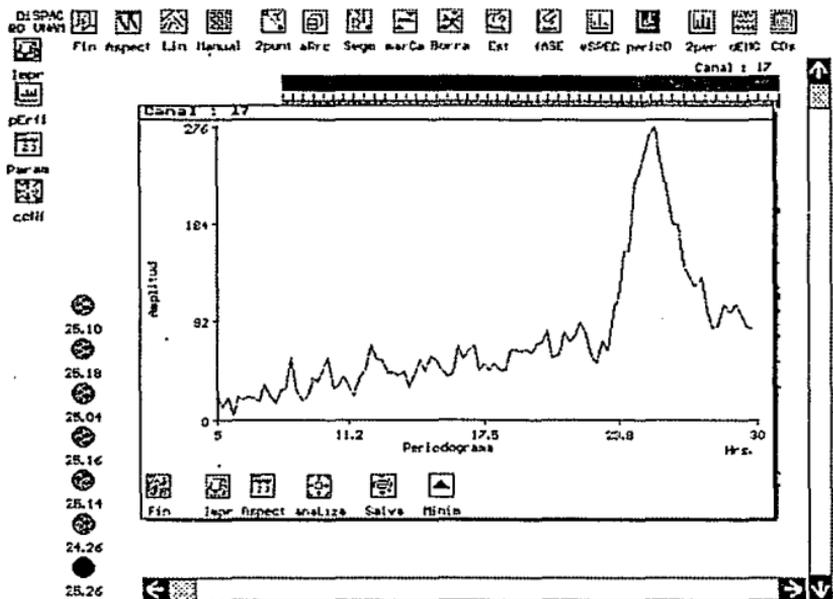


Figura 3.14 Pantalla donde se muestra la ventana de resultado de un análisis, obsérvese las posibles opciones que se tienen.

## IV Elementos del Sistema

### IV.1 Características y Requerimientos

DiSPAC es un sistema de análisis en cronobiología, el cual opera bajo un ambiente gráfico interactivo el cual se desarrollo exclusivamente para DiSPAC. Cuenta además con utilerías externas que le ayudan a ser más flexible en su uso, así como el poder comunicarse con otras aplicaciones, definiendo por medio de ellas los directorios de trabajo, tipos y modo de impresión, formato de datos, definición y creación de archivos a utilizar para umbrales y paletas de colores.

Los análisis que incluye fueron seleccionados por el usuario de tal forma que como en un principio se pretendía dar un conjunto de herramientas estándares, se seleccionaron los siguientes análisis:

- Actograma
- Perfil de actividad
- Histograma de actividad
- Periodograma
- Cosinor
- Densidad espectral

Los métodos anteriores se propusieron como estándares, sin embargo se consideró adecuado incluir otros métodos como:

- Demodulación compleja
- Autocorrelación
- Histograma con desviación estándar
- Estimación gráfica de periodo y fase
- Estimación gráfica de la arquitectura del ritmo
- Plano fase
- Diferentes estilos para el despliegado de datos y resultados

Cabe mencionar que dentro de lo que es la representación de datos se tuvieron innovaciones tales como los actogramas espectrales y sombreados con color.

DiSPAC fue diseñado para ejecutarse en una computadora del tipo IBM-PC ó compatible, bajo el sistema operativo MS-DOS, por lo que para utilizar DiSPAC se recomienda la siguiente configuración para una computadora IBM-PC ó compatible:

Memoria RAM de 640 Kb o más

Si se tiene memoria extendida ó expandida, DiSPAC puede utilizarla

Disco duro con almenos 2Mb de espacio disponible-

Adaptador de video de alta resolución VGA,EGA,CGA,HGC

Se recomienda VGA

Monitor a color (se recomienda)

Mouse

Sistema operativo 3.0 o superior

Impresora compatible con impresoras Epson ó Hp Láser

Microprocesador 80x86 Intel

Coprocesador Matemático 80x87 (se recomienda)

## IV.2 Configuración del sistema

DiSPAC cuenta con archivos externos tanto de ambiente como de utilería los de ambiente los podemos diferenciar por su extensión, y por que generalmente los crea DiSPAC, así tenemos que los archivos de ambiente son

Extensión	Descripción
CFG	Archivo de configuración
GRA	Indica donde se encuentra la interface gráfica
PAL	Indica paleta de colores para VGA en 256 colores
SHD	Indica paleta de colores y achurado
UMB	Indica umbrales
LIN	Archivo de almacenamiento de ambiente
LIS	Listado de resultados en modo texto (ASCII)
PRN	Listado de resultados en modo texto (ASCII)
MRK	Archivo de marcas realizadas sobre el Actograma
MNL	Manual
INI	Archivo de fecha inicial del experimento

Se tienen dos extensiones para el formato ASCII, con el fin de indicar cuando es de entrada (PRN) y cuando de salida (LIS)

Los archivos externos que conforman al sistema DiSPAC son los siguientes:

Nombre	Descripción
DiSPAC.exe	Programa principal
DiSPAC.cfg	Archivo de configuración
Umbrales.exe	Define umbrales (UMB)
Paletshd.exe	Define colores y achurados (SHD)
Paletvga.exe	Define paleta de colores (PAL)
	Nota: es el único al que hay que pasar el nombre del archivo al ejecutarlo, ejemplo : a>paletvga paleta.pal
SetUp.exe	Para configurar DiSPAC
Manual.exe	Despliega un breve manual en línea
DiSPAC.mnl	Archivo de texto que contiene el manual
Convert.exe	Para la conversión de formatos
Mdate.exe	Utilería para crear el archivo (INI)
Install.bat	Instala DiSPAC en disco duro

Los archivos con extensión BGI y CHR también son necesarios, estos archivos los provee Borland y son necesarios para la operación del sistema, ya que son la parte externa de la interface gráfica, utilizados para la inicialización de la interface.

### IV.3 Organización del sistema

En el capítulo anterior se describió la organización interna de DiSPAC el cual es un sistema orientado al objeto sin embargo, para dar una idea de su organización ante el usuario, (organización externa) es necesario verlo desde el punto de vista del usuario, de tal forma que solo se observan iconos los cuales pueden o no desplegar un menú.. Esta organización indica no sólo que opción realiza que cosa, sino también cuales son las posibles operaciones dentro de DiSPAC. La organización por opciones se muestra en la figura 4.1

La navegación entra cada opción se realiza mediante el *Mouse*, o bien tecleando la letra que es diferente de tamaño, al resto de las que conforman la etiqueta de cada icono. Se puede observar que se tienen pocos submenús, esto permite al usuario no solo tener la mayor parte de las opciones presentes, sino que además al contar con una descripción gráfica de la opción, se infiere de que se trata.

Para una mayor descripción de su uso y la especificación de cada opción, se puede consultar el manual en el apéndice A.

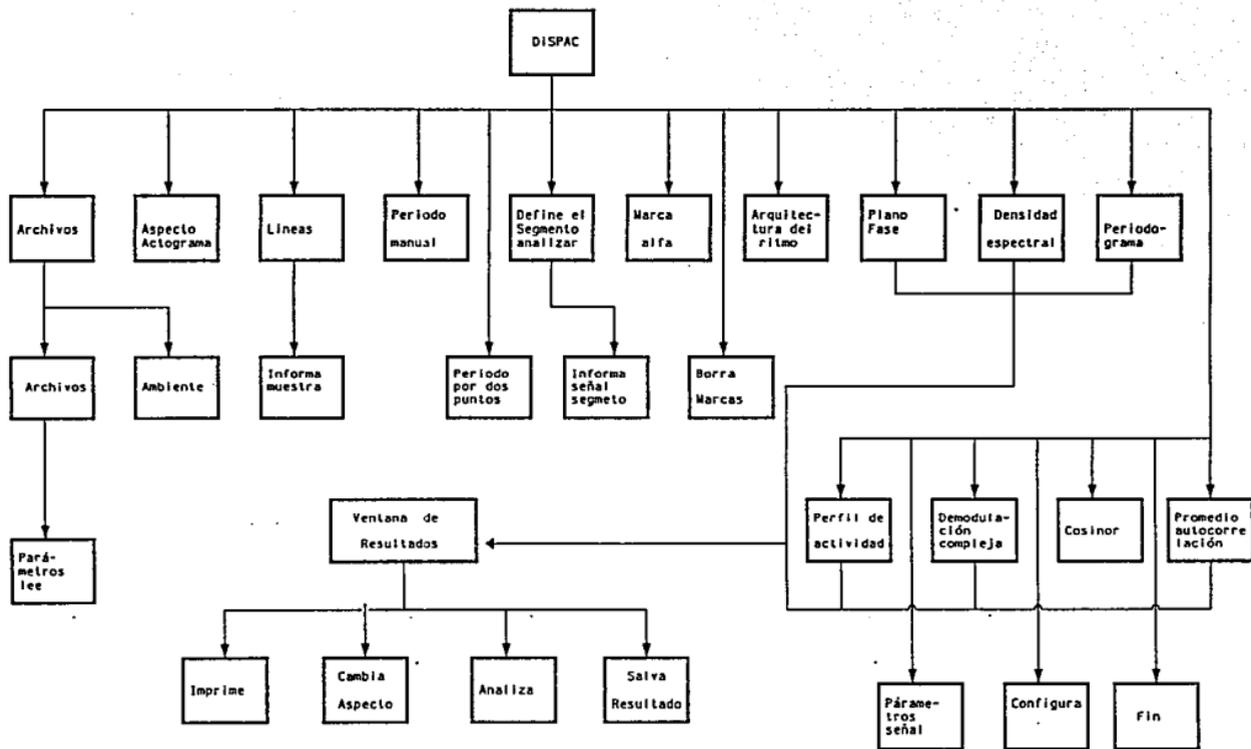


Figura 4.1 Organización de DISPAC por opciones (menu)

## IV.4 Despliegue de Resultados

El despliegue de resultados en DiSPAC como se ha mencionado anteriormente y como se puede apreciar en la figura 4.2, es mediante una ventana, la cual llamaremos ventana de resultado, (de la figura 4.1). La ventana de resultado tiene como características, que no solo muestra el resultado gráficamente, sino que además se cuenta con opciones como; imprimir la gráfica, cambiar el aspecto de la gráfica analizar y salvar los datos graficados.

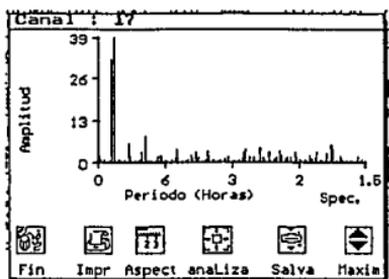


Figura 4.2 Ventana de Resultado

La opción de cambiar el aspecto de la gráfica incluye, cambiar el número de marcas para cada eje, cambiar el máximo y mínimo en cada uno de los ejes, funcionando como un "zoom", cambiar la forma en que se gráfica; puntos, línea, barras y recortar los datos por un cierto margen, de tal forma que no se grafiquen ó tomen otro valor.

La opción de analizar se refiere básicamente, a que si el usuario esta interesado en conocer el valor de un punto en la gráfica lo puede hacer mediante esta opción. El usuario marcar dos puntos con el *Mouse* y se desplegará su valor para cada uno, así como la distancia entre los mismos.

La opción de salvar almacena los datos desplegados en un archivo especificado por el usuario con la extensión LIS, para que de esta forma el usuario pueda generar la gráfica por medio de otra aplicación, o bien re analizar los datos.

La opción de impresión, como su nombre lo indica imprime la gráfica en la impresora definida por el usuario, en la orientación y tamaño también definidos por el usuario.

## IV.5 Utilerías Externas

DiSPAC cuenta con 7 utilerías externas, las cuales lo hacen más flexible en su uso y permiten la comunicación con otras aplicaciones. Las utilerías son:

Umbrales.exe	Define umbrales (UMB)
Paletshd.exe	Define colores y achurados (SHD)
Paletvga.exe	Define paleta de colores (PAL)
SetUp.exe	Para configurar DiSPAC
Manual.exe	Despliega un breve manual en línea
Convert.exe	Para la conversión de formatos
Mdate.exe	Utilería para crear el archivo (INI)

Umbrales se utiliza para crear los archivos de niveles ó umbrales, con el que operara DiSPAC, en la generación de los octogramas espectrales, donde para cada color se le asigna un nivel mínimo y un nivel máximo, estos niveles están almacenados en un archivo con la extensión UMB, de tal modo que se pueden tener n archivos UMB, y para crearlos lo hacemos mediante la utilería llamada Umbrales.

Paletshd y PaletVGA, son la parte complementaria de Umbrales, ya que por medio de estas utilerías podemos crear los archivos SHD y PAL respectivamente, los cuales contienen la definición de los colores y achurados que corresponden a cada nivel. El archivo con extensión PAL, solo es útil para monitores VGA y por lo tanto paletvga solo debe utilizarse en una computadora con monitor VGA.

SetUp es una utilería que nos ayuda a configurar DiSPAC, definiendo los directorios de trabajo, el tipo de formato de los datos, el tipo y modo de impresora a utilizar, los archivos de umbrales y paletas a utilizar. Al definir una configuración esta puede ser salvada con un nombre diferente al de *dispac.cfg*, de esta forma dentro de DiSPAC se selecciona la opción configura dando el nuevo nombre de archivo, lo cual funcionara como si personalizáremos el uso de DiSPAC para diferentes usuarios.

Manual es un pequeño manual en línea que nos ayuda a resolver algunas dudas que se tengan con respecto a DiSPAC. Para la creación de esta utilería se utilizaron técnicas de *hipertexto*. El *hipertexto* interconecta piezas relacionadas de información en una computadora de tal forma que el usuario pueda moverse a las nuevas localidades en el espacio de la información con tan solo seguir las ligas que conectan cada localidad.

La información es normalmente dividida dentro de unidades, las cuales comúnmente son desplegadas en ventanas separadas en la pantalla, estas unidades son llamadas nodos, porque un espacio completo de hipertexto forma una estructura de *gráfo*, como la que se muestra en la figura 4.3

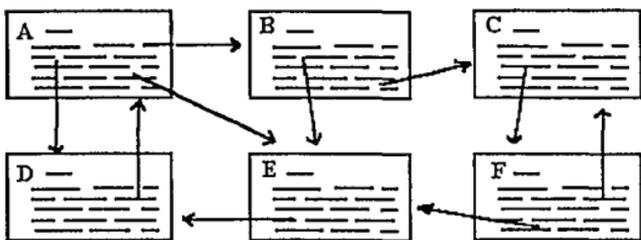


Figura 4.3 Estructura de *gráfico* en hipertexto.

Convert es la utilidad que nos permite comunicar con otras aplicaciones, ya que es la encargada de realizar la conversión de formatos, sin embargo no sólo tiene esas opciones, sino que también puede compactar nuestros archivos, y por supuesto encriptarlos, así como verificar la integridad de nuestros archivos de datos.

Mdate es útil para crear el archivo DiSPAC.INI el cual se utiliza exclusivamente para datos provenientes del laboratorio de cronobiología, ya que estos no contienen información sobre el inicio del experimento.

## V ANÁLISIS IMPLANTADOS

Antes de caracterizar a la serie de tiempo, por cualquiera de los diversos métodos, es conveniente obtener una representación gráfica de la misma ya que así podremos observar su comportamiento a lo largo del tiempo.

La representación gráfica de la serie de tiempo tiene una gran importancia en la descripción de la señal. La forma más fácil de representarla es mediante una gráfica xy, como se muestra en la figura 5.1, donde se gráfica tiempo .vs. amplitud.

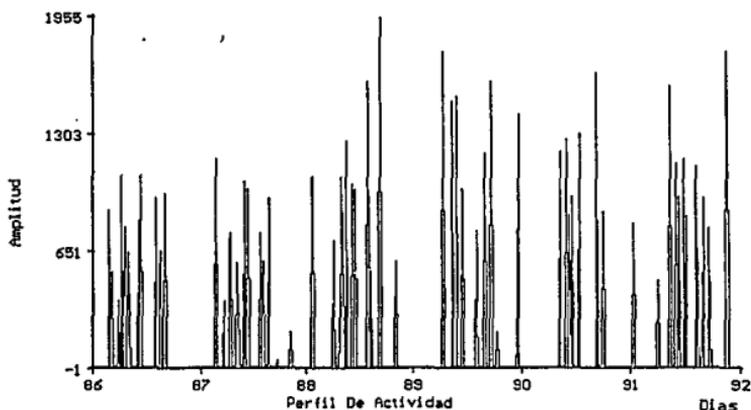


Fig. 5.1. Perfil de actividad, registro de ingesta de agua en oscuridad continua

### V.1 Actograma

Una de las formas más comunes de representar un registro de actividad (la serie de tiempo) es el actograma. El actograma como el nombre sugiere es típicamente usado para examinar datos de actividad, sin embargo también puede ser utilizado para examinar datos que no sean de actividad, ya que se puede observar el periodo y la fase de la señal.

El actograma consiste en el ordenamiento consecutivo de gráficas de ocurrencia de eventos, como lo muestra la figura 5.2.

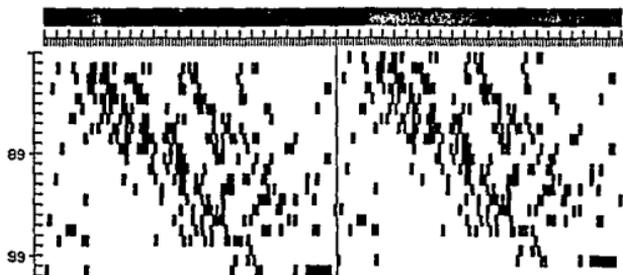


Fig. 5.2: Actograma en Double-Plot (gráfica de ocurrencia de eventos).

Las gráficas en Double-Plot son una técnica que parte del actograma y consiste en que a partir de una captura de datos, efectuada durante un periodo  $P$  a intervalos constantes, se agrupan los datos en  $n$  subintervalos y se escalan para construir una gráfica de  $n$  barras, donde  $n$  es el número de intervalos definidos en  $P$ . Además para observar mejor la ritmicidad, se repite al lado izquierdo la misma gráfica, pero recorrida un periodo, sobre la escala de las ordenadas, de ésta forma se observa que si la ritmicidad corresponde al periodo  $P$  seleccionado, se obtendrá una gráfica sin defasamiento, o bien si la ritmicidad es mayor que el periodo propuesto, la gráfica presentará una pendiente negativa, mientras que si es menor presentará una pendiente positiva, de ésta forma es posible apreciar el periodo y fase del fenómeno en estudio.

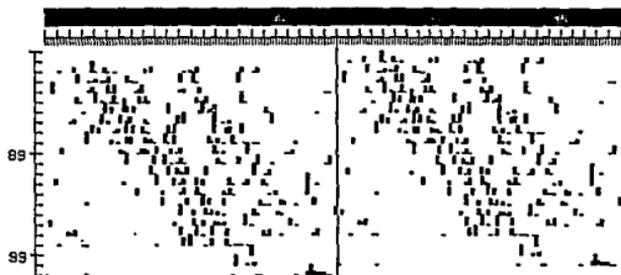


Fig. 5.3: Actograma en Double-Plot, histogramas de actividad (escalado).

La intensidad de la conducta no se puede evaluar a través de estas gráficas, sin embargo en ciertos experimentos, la evaluación de ésta es tan importante como el determinar el periodo y la fase. El uso de histogramas de frecuencia para sustituir los registros de ocurrencia de eventos ha dado algunos resultados al respecto, pero no son completamente satisfactorios. Por lo que el uso de un espectro cromático, que codifica la intensidad de la conducta en estudio, para sustituir las gráficas de ocurrencia de eventos ó

los histogramas de frecuencia, es otra opción más para la representación de los datos. El despliegue de la gráfica se realiza en un monitor poli cromático, para imprimir la gráfica en papel, debe sustituirse la asignación de colores por una de tonos de gris.

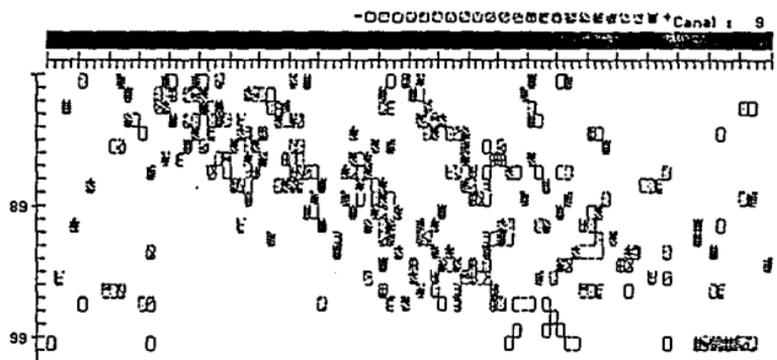


Fig. 5.4: Actogramas codificados por color (representación en sombreados)

Una vez representada la serie de tiempo, podemos comenzar a caracterizarla, es decir determinar: su periodo, fase, amplitud y forma de la onda.

## V.2 Determinación del Periodo

El análisis del periodo puede ser realizado de varias formas. Si hay un punto de referencia identificable, como el que se muestra en la figura 5.5, se puede utilizar la estimación gráfica del periodo y la fase. Si no existe un punto de referencia entonces se pueden utilizar los siguientes análisis matemáticos:

- Cosinor
- Periodograma
- Demodulación Compleja
- Densidad Espectral



Fig. 5.5: Registro de actividad en Double-Plot, se observa que se puede marcar un punto de referencia fácilmente, como es el caso del inicio de alfa como se muestra por la línea.

### V.2.1 Estimación gráfica del periodo.

La estimación gráfica del periodo y la fase es aplicable si se tiene un punto de referencia disponible, como se muestra en la figura anterior, donde este lo podemos seleccionar como el inicio de  $\alpha$ , de acuerdo con la figura tenemos que la escala horizontal está en horas<sup>1</sup> y la vertical está en días, la hora inicial es la hora en que se comienza a gráficar.

El periodo lo definimos como el intervalo entre dos puntos de referencia sucesivos.

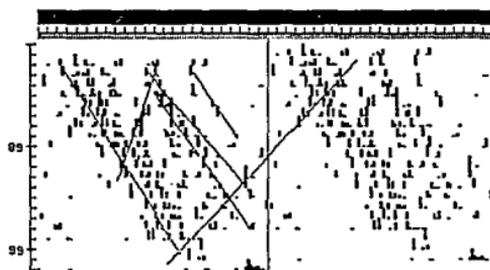


Fig. 5.6: La figura muestra líneas sobre el actograma, las cuales representan periodos propuestos y calculados, los cuáles fueron determinados sobre la gráfica. Nótese que la fase de referencia no siempre es el inicio de  $\alpha$ .

<sup>1</sup> Si el actograma se gráfica con  $\tau$  modulado, la escala horizontal estará en horas circádicas.

De esta forma, trabajando sobre la gráfica, se realizan las siguientes operaciones :

- identificar el punto de referencia
- marcar varios puntos de referencia continuos
- contar el tiempo transcurrido desde un punto al próximo
- repetir el inciso c) para todos los casos o puntos marcados
- obtener el promedio de estas cuentas

El promedio resultante de esta operación es el periodo buscado, sin embargo si se tiene un conjunto de puntos se puede realizar una regresión lineal sobre éstos puntos y la pendiente resultante (normalizada) será el periodo. Otra forma de obtenerlo es seleccionando dos puntos sobre la gráfica y obteniendo la pendiente (normalizada), la cuál será el periodo buscado.

### V.2.2 Estimación gráfica de la fase

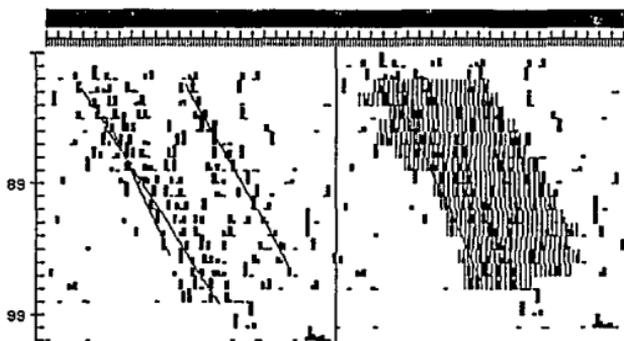


Fig. 5.7: Al marcar puntos de referencia (alfa) se pueden determinar periodos y fases.

Para la determinación de la fase basta con identificar el punto de interés y obtener la hora en que este ocurrió, esto se logra proyectando una línea hacia la escala horizontal, sin embargo en ocasiones es necesario obtener el ángulo de fase, el cuál lo definimos como el tiempo que ocurre desde el punto de interés hasta un punto de referencia. Para determinar el ángulo de fase debemos de seleccionar el punto de referencia, de tal forma que si el punto de interés ocurre antes, será un avance de fase y si ocurre después será un retraso de fase.

Por medio de DiSPAC podemos obtener el periodo y la fase en forma gráfica, con una descripción más detallada de los puntos y periodos de interés.

En las siguientes tablas se muestra el resultado obtenido de las marcas realizadas y los periodos para cada una de las líneas representadas en la figura 5.7..

Datos de marcas realizadas

Alfa	Rho	Tau	Fase_Alfa	Fase_Rho
9.45	13.00	22.45	23.15	9.00
11.30	15.00	26.30	22.00	9.30
9.30	13.45	23.15	0.30	10.00
11.15	12.30	23.45	23.45	11.00
12.45	12.45	25.30	23.30	12.15
1.00	13.45	24.45	1.00	12.00
11.15	15.15	26.30	1.45	13.00
8.45	15.15	24.00	4.15	13.00
9.45	14.45	24.30	4.15	14.00
9.30	15.00	24.30	4.45	14.15
9.30	14.30	24.00	5.15	14.45
11.00	14.30	25.30	5.15	16.15
8.30	15.00	23.30	6.45	15.15
10.15	14.15	24.30	6.15	16.30
8.45	15.15	24.00	6.45	15.30

Tabla 5.1 Datos obtenidos a partir de las marcas realizadas sobre el actograma de la fig. 5.7 ( tabla obtenida automáticamente por medio de DiSPAC)

	Alfa	Rho	Tau	Fase_Alfa	Fase_Rho
Media	10.120	14.180	24.300	9.170	13.050
S. D.	1.129	0.565	1.061	8.510	2.256
Var.	1.285	0.531	1.129	78.192	5.534

Tabla 5.2 Resultado de procesar los datos mostrados en la tabla 1

Inicio		Final		Periodo
Día	Hora	Día	Hora	
81	22.30	98	10.15	24.42
81	7.30	95	16.15	24.37
84	1.15	94	6.00	24.18
83	1.00	97	8.15	24.30

Tabla 5.3. Se describe cada una de las líneas hechas sobre el actograma, especificando día y hora de inicio y final, así como el periodo de la línea.

### V.2.3 Cosinor

El cosinor es un método de ajuste de una curva por mínimos cuadrados, el cual sirve para determinar la función de mejor aproximación cosenoidal a los datos. Para tal propósito se utiliza el siguiente modelo:

$$y(t_i) = M + A \cos(\omega t_i + \phi)$$

donde  $t_i$  tiempo,  $\phi$  Acrofase,  $A$  Amplitud,  $\omega$  Frecuencia angular ( $\omega=2\pi/\tau$ ,  $\tau$  periodo,  $1/\tau$  frecuencia),  $M$  mesor.

La función coseno fue seleccionada debido a que el coseno de  $0 = 1$  por lo que es una fase de referencia cómoda y provee una representación como el inicio del reloj cuando una gráfica polar es utilizada para representar  $\phi$  y  $A$  estimados con sus límites de confianza. La estimación de  $\tau$  con sus límites de confianza también puede ser calculada por este método.

El método del Cosinor puede ser usado iterativamente con diferentes periodos de prueba, de tal forma que se pueden estimar los parámetros de la ecuación y determinar así las armónicas. Por otro lado tenemos que de acuerdo al tipo de datos es el método a utilizar: Cosinor sencillo ó Cosinor medio. El Cosinor medio es el procedimiento original aplicable para la estimación de los parámetros ( $A$ ,  $\phi$ ), cuando tratamos con 3 ó más series biológicas este método es el más común, utilizado para evaluar las características rítmicas de un grupo ó población. El Cosinor sencillo es aplicable a una sola serie de tiempo compuesta de datos de un sujeto (datos serialmente dependientes) ó grupo de individuos (datos serialmente independientes). Cuando se trata de datos provenientes de un grupo de sujetos individuales, las series de tiempo se colocan una al final de la otra, con la opción de ajustar los datos dentro de los primeros valores relativos. Para datos serialmente dependientes ambos, Cosinor sencillo y medio pueden ser utilizados, y para datos serialmente independientes solo el Cosinor sencillo es aplicable.

Cuando utilizamos el método del Cosinor este asume que los datos son normalmente distribuidos alrededor de cada uno de los tiempos de muestreo, esto puede ser probado examinando los valores residuales resultantes del ajuste de la función de aproximación.

Una de las principales ventajas del cosinor es que permite una prueba objetiva de la hipótesis de que la amplitud del ritmo difiere de cero. En otras palabras, que el ritmo es validado para un periodo  $\tau$  considerado. Usualmente la evidencia para la detección del ritmo es aceptada cuando la probabilidad de que  $A$  sea cero, sea menor de 0.05. ( $P \leq 0.05$ )

El método también provee una aproximación de calidad de ajuste, comúnmente llamada porcentaje de ritmo (PR), muchos de los ritmos biológicos no se parecen a una función cosenoidal por lo tanto es importante determinar el PR de los datos incluidos en el 95% de los límites de confianza, para la función de mejor ajuste.

Por otro lado una ventaja importante sobre otros métodos, es que los datos de la serie de tiempo no necesitan estar igualmente espaciados y tolera bastante bien el caso de los datos perdidos o faltantes.

Los valores estimados de los parámetros con sus respectivos límites de confianza son expresados comúnmente por tablas o gráficas polares para su visualización, ver figura 5.7B

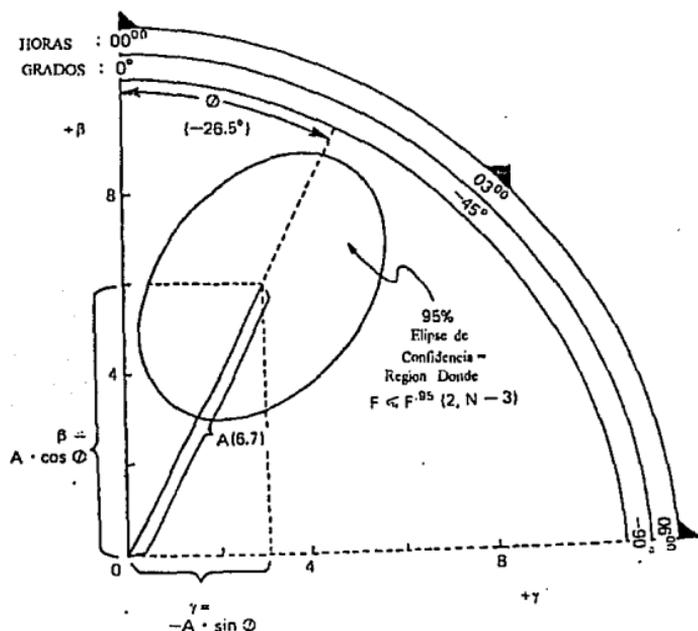


Fig. 5.7B Representación rectangular ( $\beta, \gamma$ ) y polar ( $A, \phi$ ) de los parámetros estimados mediante un ajuste por mínimos cuadrados. (Tomado de Halberg et al. Glossary of Chronobiology. Chronobiology 4 Supl. 1, 1-1891977)

## V.2.4 Periodograma

El Periodograma es una técnica matemática que nos permite estimar el periodo de los componentes responsables de la periodicidad de un fenómeno.

### Periodograma según Enright

Este análisis consiste en tomar, a intervalos regulares una cantidad razonable de mediciones, digamos una serie de tiempo de al menos 3 ciclos, entonces si existe un componente armónico principal de periodo  $P$  (sujeto a modificaciones aleatorias, provenientes de una distribución Normal). El valor de la desviación estándar  $A$  de las observaciones hechas al tiempo  $h$  agrupadas en  $p$  columnas es un buen estimador del valor de la función periódica durante este tiempo.

El valor del estimador para un periodo  $p$  es proporcional al componente armónico de periodo  $P$  que es el mayor entero menor o igual que  $p$ . Si se construye una gráfica de estimadores contra los periodos propuestos, el estimador de mayor valor para un periodo dado, indicará que ese periodo es el componente principal de la función. La gráfica que se obtiene de esta manera se conoce como Periodograma.

Notación empleada para la definición del método.

$N$  : Número total de muestras.

$X_i$  :  $i$ -ésima muestra.

$p$  : Periodo de prueba, para el cuál se calcula la estadística.

$P$  : Número de columnas en un arreglo de horas/muestras.  $P \leq p$

$h$  : Índice de una columna de las  $X_i$ 's teniendo  $P$  columnas  $1 < h \leq P$ .

$k$  : Número de renglones en el arreglo de  $P$  columnas

$X_h$  : Media de las  $X_i$ 's en la columna  $h$  del arreglo

$\bar{X}_{h,p}$  : Valor promedio para el tiempo  $h$  en el periodo  $p$  considerado, como se define a continuación:

$$\bar{X}_{h,p} = \frac{1}{k} \sum_{i=0}^{k-1} X_{(h+ip)}$$

Una vez conocido el valor de  $X_{p,h}$ , se calcula el valor de la desviación estándar  $A_p$ , de la siguiente forma:

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

$$A_p^2 = \frac{1}{P} \sum_{h=1}^P (\bar{X}_{h,p} - \bar{X}_p)^2$$

Donde

$$\bar{X}_p = \frac{1}{P} \sum_{h=1}^P \bar{X}_{h,p}$$

$A_p^2$  es un buen estimador de la frecuencia de aparición del periodo P propuesto, ya que refleja el grado en el que los valores individuales de  $X_{p,h}$  se alejan del valor promedio.

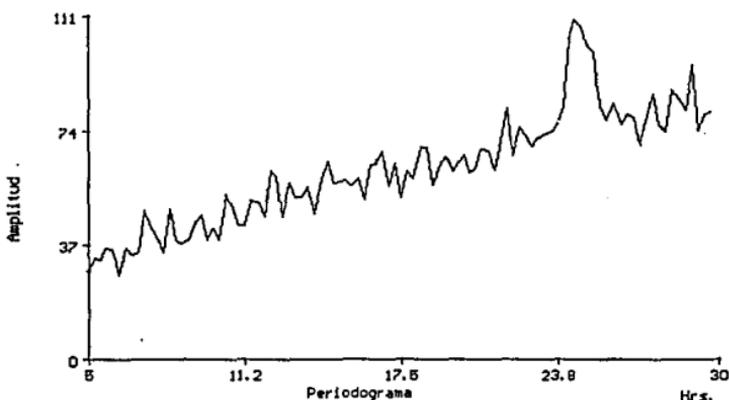


Fig. 5.8: Gráfica del Periodograma según Enright

### Chi Cuadrada ( $\chi^2$ )

$A_p^2$  es un estimador de la varianza de las medidas de P ( $\bar{X}_1, \dots, \bar{X}_p$ ) de las medias de k valores de una población de N valores, si estos valores son normalmente distribuidos con una varianza  $\sigma_x^2$ , entonces la varianza de las medias será:

$$\sigma_{\bar{x}}^2 = \frac{1}{k} \sigma_x^2$$

y la razón

$$Q_p = \frac{P A_p^2}{\sigma_x^2}$$

Debe conformar aproximadamente a la distribución chi-cuadrada con  $(p-1)$  grados de libertad.

Una estadística  $X^2$  equivalente ha sido propuesta para procesos puntuales estocásticos del tipo Poisson en el cual la serie de varianzas  $\sigma_x^2$  en la ecuación anterior es remplazada por la serie medias  $X$  para dar:

$$Q_p = \frac{k \sum_{h=1}^p (\bar{X}_h - \bar{X})^2}{\bar{X}}$$

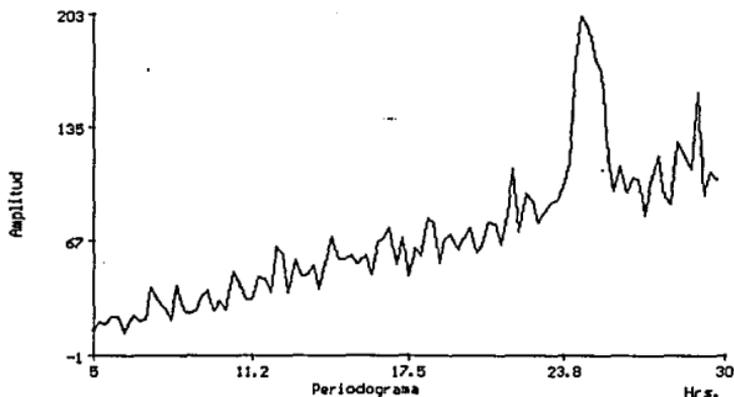


Fig. 5.9: Periodograma  $X^2$  calculado para los mismos datos que el anterior

El Periodograma aún cuando está diseñado para series de tiempo grandes estima bien el periodo para datos relativamente cortos (2,3 ciclos). Por otro lado es uno de los métodos más fáciles de implementar en una computadora. Este análisis trabaja en el dominio del tiempo y también cuenta con la estimación de los valores de significancia, lo cual es muy importante para decidir si los valores estimados son válidos o no. Por otro lado tenemos que con algunas modificaciones al método, se pueden calcular periodos no enteros, lo cual aumenta su resolución temporal.

## V.2.5 Demodulación compleja

Una serie de tiempo puede ser descrita como una función que consta de tres componentes.

$$x(t) = c(t) + d(t) + e(t)$$

Donde :

$c(t)$  : es el componente de tendencia lineal (trend)

$d(t)$  : es el componente rítmico.

$e(t)$  : es el componente aleatorio ó error.

De tal forma que como nuestro interés esta en el componente rítmico debemos eliminar los dos componentes restantes, obviamente la eliminación del componente de error es común removerlo por técnicas de promediación, sin embargo por efectos del método se reduce significativamente. Por otro lado la mayoría de las técnicas para extraer la tendencia caen dentro de las siguientes tres categorías;

- Técnicas de regresión.
- Técnicas de suavizado en el dominio del tiempo.
- Técnicas de suavizado en el dominio de la frecuencia.

La demodulación compleja cae dentro de la tercer categoría. Desde un punto de vista descriptivo la demodulación compleja es análoga al procedimiento de ajuste por mínimos cuadrados, en la interpolación entre los puntos de datos. Pero difiere de esta porque utiliza una suma de funciones trigonométricas como el modelo para "el mejor ajuste".

En el método de demodulación compleja el valor de los datos  $a_1..a_n$  es transformado al dominio complejo multiplicando cada dato, por las funciones seno y coseno teniendo la frecuencia a ser examinada, de tal forma que resultan dos conjuntos de datos:

$$Z_i (\text{Re}) = a_i \cos 2\pi f_m t/s \quad \text{y} \quad Z_i (\text{Im}) = a_i \sin 2\pi f_m t/s$$

$f_m$  : m-ésima frecuencia para demodulación.

$i = 1..n$ ,  $t = i-1$ ,  $n$  : número de épocas en el conjunto de datos.

$s$  : número de épocas en el periodo deseado.

Implícito en cada valor de los datos,  $a_i$ , esta la representación compleja de todas las frecuencias presentes en toda la serie de tiempo. Sin embargo la multiplicación por las

funciones seno y coseno generan productos que son sumas y diferencias de senos y cosenos, resultando en frecuencias deseables y no deseables. Por lo que hay que remover las frecuencias no deseables, esto se logra con un filtro paso bajas exponencial.

Una vez realizado lo anterior se pueden obtener los parámetros locales para cada época de los conjuntos de datos filtrados,  $Z_i$ 's, como son potencia, amplitud, fase y redemulación, como sigue:

$$\text{Potencia : } P_i = 2.0[Z_i^{*2}(\text{Re}) + Z_i^2(\text{Im})]$$

$$\text{Amplitud : } A_i^2 = P_i$$

$$\text{Fase : } \phi_i = \arctan [ Z_i^*(\text{Im})/Z_i^*(\text{Re}) ]$$

$$\text{Redemulación : } R_i = 2.0 [ Z_i^*(\text{Re}) \cos 2\pi f_m t/s + Z_i^*(\text{Im}) \sin 2\pi f_m t/s ]$$

Al obtener nuevamente los datos mediante la redemulación si éstos contenían componentes periódicos de la frecuencia estimada, se podrán observar directamente por medio de la inspección visual. Si se desea determinar más precisamente a partir de éstos datos se puede realizar otro tipo de análisis como periodograma, densidad espectral etc.. Por otro lado se puede realizar una serie de demodulaciones a diferentes frecuencias y de los valores resultantes (redemulación) se calcula la varianza, una vez hecho esto se comparan. De tal forma que si la varianza obtenida de una serie originada a una frecuencia dada es mayor que las otras, se considera entonces que el inverso de dicha frecuencia es el periodo de la oscilación analizada.

Realizando una breve comparación con otros métodos encontramos que;

a) La mayoría de las técnicas basadas en regresión que ajustan los datos a una onda cosenoidal (p.e. Cosinor), dan como resultado un sólo valor promediado de amplitud y un sólo valor promediado de la fase, para la frecuencia seleccionada.

b) La demodulación compleja provee  $n$  valores instantáneos de amplitud y  $n$  valores instantáneos de fase y además los  $n$  valores se encuentran en el dominio del tiempo para una frecuencia seleccionada. Mientras que el análisis de Fourier da como resultado un solo valor promediado de amplitud y uno de fase para cada una de las frecuencias  $n/2$  ( donde  $n$  es una potencia de 2), es decir la resolución que nos ofrece este último es poca aún cuando abarca un rango grande de frecuencias.

c) Múltiples demodulaciones proveen los mismos  $n$  valores para cada una de las  $m$  frecuencias armónicamente relacionadas (donde  $m$  es un número determinado por la frecuencia de muestreo, longitud de los datos y el principal periodo de interés, con un límite mayor de  $n/2$ ). Por lo que puede usarse en forma iterativa para diferentes periodos propuestos.

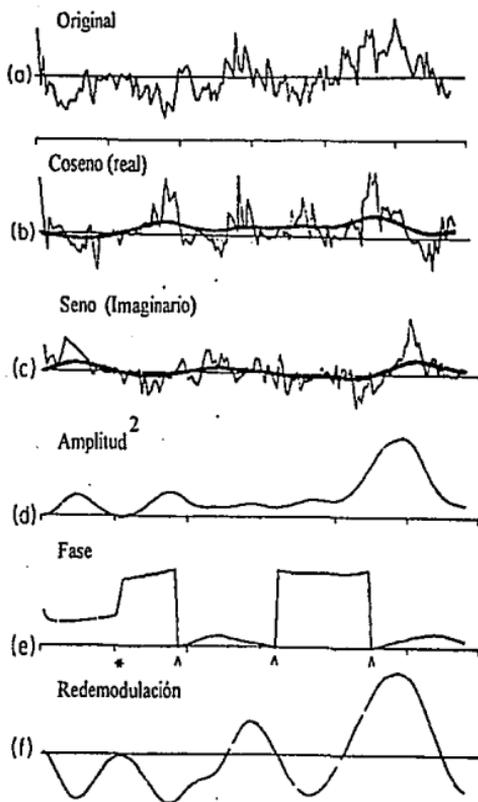


Fig. 5.10 Descripción de cada uno de los diferentes estados de la señal al analizarla mediante la demodulación compleja. (tomada de "Trend and Rhythm Analysis of Time-Series Data Using Complex Demodulation" Helen C Sing, David R Thorne, and Frederick W Hegge. *Behav. Research Met, Instruments, & Computers* 1985, 17 (6), 623-629)

## V.2.6 Poincaré

La Dinámica no lineal provee conceptos apropiados para entidades complejas, las secciones de Poincaré relacionan cada punto de la trayectoria con el siguiente, es decir el efecto de cada evento con eventos sucesivos, esto nos da información acerca de la trayectoria, tendencia, desarrollo y propiedades relevantes del sistema (ej. tipo de atractor, dimensionalidad, periodicidad, caos, etc.).

Para formar una sección de Poincaré ó mapa de Poincaré, tomamos  $E$  como el periodo en horas de nuestros datos (p.e. 24hrs.), ó podemos normalizarlo a uno, es decir ajustado el periodo a un círculo de perímetro uno. A cada uno de los valores del conjunto de datos le corresponde un tiempo de ocurrencia ó de evento, el cuál lo definiremos como fase,  $\phi$ . Entonces El mapa de Poincaré se obtiene gráficamente una fase dada,  $\phi_i$ , contra la(s) siguientes,  $\phi_i+n$ , como se muestra en la figura 5.11, donde  $n$  es un entero mayor o igual a uno y además nos indica el orden del mapa, el cuál es el intervalo entre las fases gráficas, es decir para el mapa de orden uno se gráfica la fase uno contra la fase dos, en el mapa de orden 2 se gráfica la fase uno contra la fase tres, etc..

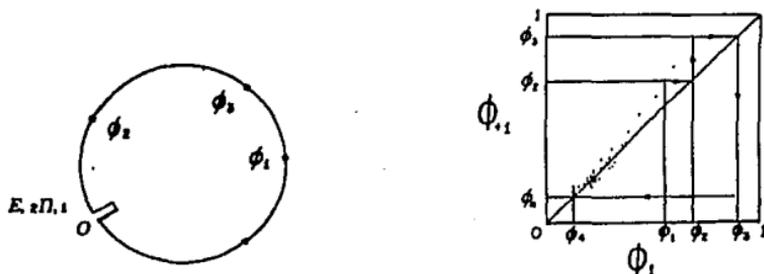


Fig. 5.11: Esquematación de como se realiza un mapa de Poincaré. (tomada de "Locking, Intermittency, and Bifurcations in a Periodically Driven Pacemaker Neuron: Poincaré Maps and Biological Implications", O. Diez Martínez, P. Pérez, R. Budelli, and J.P Segundo. Biol. Cybern. 60, 49-58(1988) ).

La interpretación se realiza entonces en términos de la teoría de sistemas dinámicos:

Para cualquier mapa ;

- La primer fase  $\phi_1$  es arbitraria.
- Diferentes secuencias de fases son reflejadas por mapas de orden sucesivo.

- c) Si los puntos en la diagonal son fijos, implican repeticiones estables de una fase a otra. Puntos fijos en el mapa de orden  $n$  representan la fijación en un periodo incluyendo  $n$  eventos.
- d) Si el mapa de orden  $n$  tiene puntos fijos, todos los mapas de orden  $kn$  los tienen. donde  $k$  es un entero.
- e) Puntos arriba o abajo de la diagonal respectivamente, y que son seguidos por puntos a su derecha ó izquierda implican secuencias corto-largo ó largo-corto.
- f) Fases de eventos agrupadas por sucesivos arrastres, se refiere a que cada evento excede el previo al menos por un intervalo..
- g) Proximidad a la diagonal, abundancia y puntos de alta densidad, caracterizan secuencias de muchas fases que cambian poco.

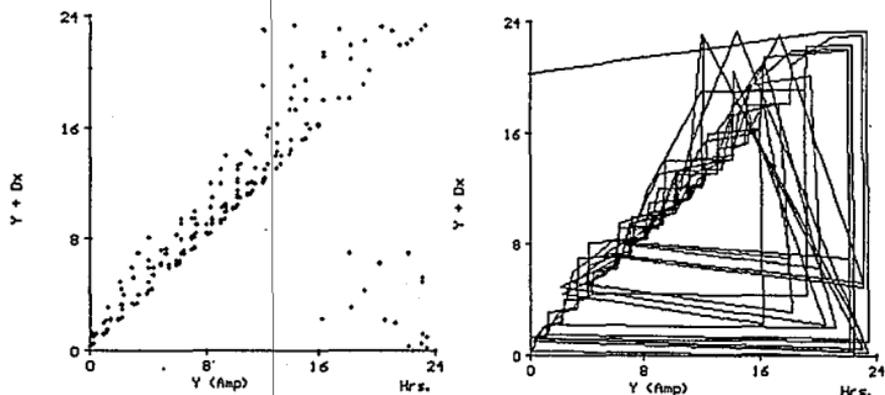


Fig. 5.12: Resultado de aplicar los procedimientos anteriores, mapa de orden 1.

Por medio de éstos mapas podemos observar los ciclos límites los cuales están directamente relacionados con el rango de sincronización y descripción de osciladores relajados, por otro lado tenemos que este tipo de mapas es una representación de los cambios que exhibe el oscilador y por lo tanto perturbaciones al oscilador se reflejarán en cambios en el desarrollo del mapa, y si después de esta perturbación regresa a su estado anterior, entonces podemos decir que el estado anterior es el estado estable del oscilador.

## V.2.7 Densidad Espectral

La Densidad Espectral es una medida que resulta de gran utilidad para el análisis directo y cuantitativo de fenómenos periódicos y se presta para llevar estadísticas con mucha información concentrada, como es el caso de los arreglos espectrales, los cuáles son conjuntos de gráficas donde se pueden observar las variaciones de los componentes principales a lo largo del tiempo

La Densidad Espectral ó espectro de potencia  $\Gamma_{xx}(F)$  se calcula a partir de la función de autocorrelación de la serie de tiempo  $x(t)$ , obteniendo la transformada de Fourier de dicha función. La  $\Gamma_{xx}(F)$  obtenida mide la distribución de la potencia de  $x(t)$  con respecto a la frecuencia. Debemos recordar que la transformada de Fourier es solo un paso del dominio del tiempo al dominio de la frecuencia.

Si pensamos que una serie de tiempo se puede describir como una suma de funciones seno y coseno teniendo diferentes frecuencias, entonces la Densidad Espectral nos daría información acerca de la contribución de cada una de esas frecuencias. De tal forma que si se obtiene un espectro con un componente principal a una frecuencia dada, ésta frecuencia es la que contribuye de manera más importante en la forma de onda del fenómeno. La figura 5.13 nos muestra que para una serie de tiempo  $x(t)$  (vista en figuras anteriores 5.1, 5.2, 5.3, 5.4, 5.5 etc.) el componente principal que contribuye a dicha serie es cercano a 24hrs. Aquí se puede observar que uno de los problemas de la Densidad Espectral es la resolución en frecuencia que ésta ofrece.

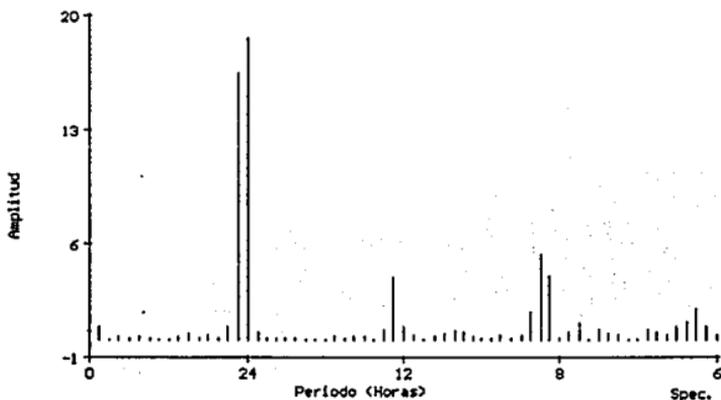


Fig. 5.13 Densidad Espectral de los datos mostrados anteriormente.

Debido al gran número de cálculos que se tienen que realizar para el cálculo de la densidad espectral surge la transformada rápida de Fourier FFT, con la cual se reduce el número de operaciones, estos métodos de FFT están bien descritos en la literatura y existe un gran número de *software* para computadoras que realizan este proceso, inclusive existen *chips* diseñados para este tipo de operaciones, los llamados DSP-CHIPS ó DSP-Micros, los cuáles están orientados a resolver problemas específicos de procesamiento de señales, generalmente en tiempo real.

Por otro lado debido a las características de la transformada de Fourier es necesario realizar un preprocesamiento de la señal para evitar artefactos en el resultado obtenido, es decir se debe pasar por una ventana para disminuir el efecto de "*aliasing*".

La Densidad Espectral es un método ampliamente utilizado en el procesamiento digital de señales, sin embargo por el tipo de implantación (mediante la FFT) debemos tener en cuenta que la cantidad de muestras que se requieren para el proceso es  $m = 2^n$ , donde  $n$  es un entero positivo mayor o igual que dos.

Este procedimiento describe muy bien a la señal y dependiendo del rango de frecuencias de nuestro interés, los datos se pueden agrupar, para dar diferentes frecuencias, ya que la resolución en frecuencia esta estrechamente ligada a la frecuencia de muestreo.

### V.3 Breve comparación entre los Métodos implantados

Las diferencias en la naturaleza de los datos y los procesos involucrados en su obtención (muestreo), tienen una profunda influencia en la utilidad de los diferentes métodos y sus respectivas interpretaciones.

Por otro lado tenemos que la inspección directa de los datos, mediante el uso de autograma, es la forma más común y rápida de caracterizar a la señal, en donde no hay duda del período en los datos, por lo que los métodos de estimación gráfica cobran gran importancia.

El análisis de periodograma puede estimar aceptablemente el período en registros relativamente cortos de datos. Sin embargo es un procedimiento lineal (aditivo) y fue diseñado para tratar series de tiempo en donde la fase se conserva.

El análisis de densidad espectral, basado en los componentes de Fourier, no está restringido a series de tiempo en donde la fase se conserva, ya que es un método no lineal (multiplicativo), además también se puede aplicar a procesos con conservación de fase tal como los ritmos circádicos, sin embargo para una buena estimación se requieren registros relativamente largos o medianos de datos, y por otro lado la estimación de la frecuencia fundamental en la región de interés es de baja resolución.

El análisis de cosinor supone que la serie de tiempo es de forma sinusoidal y debido a esto trata de ajustar los datos a la mejor función cosenoidal, entonces la pregunta es ¿Que tan válidos son los resultados obtenidos cuando la forma de la onda difiere de la sinusoidal?, para esto se deben calcular estimadores que nos indiquen la validez de los mismos.

La demodulación compleja se recomienda cuando se conoce el período ó la frecuencia principal, cuando existe un interés particular en la relación de amplitud y fase con los diferentes componentes, ya sean sobre el ritmo manifestado o sobre estímulos experimentales, y cuando la serie de tiempo contiene o es vista como una fuente identificable de ruido obscureciendo otros efectos de interés.

En la siguiente tabla se trata de resumir las características principales de algunos de los métodos más usuales, considerando; a) la naturaleza de los datos cuantitativos .vs. cualitativos y b) la cantidad de datos registrados registros cortos y registros largos (corto .vs. largo), y especificando cuando es aplicable. Así como, cuando se requiere o se recomienda un preprocesamiento.

## Métodos

Tipo de datos		Cosinor	Enright	X <sup>2</sup>	Densidad espectral	Análisis armónico	Análisis de varianza
Cualitativos	Cortos	△○ □	○	○ □	○□	○	▽
	Largos	△○ □	△○ □	△ □	△○	○	▽
Cuantitativos	Cortos	△	△+	△ +	△○□ +	△○	△
	Largos	△□	△+	△ +	△○+	△○	△

Notación :

△ : Aplicable

□ : No Recomendado

▽ : No Aplicable

+ : Requiere preprocesamiento.

○ : Aplican Reglas

Nota : Perfil, Actograma, y Estimación gráfica de periodo y fase, son aplicables en todos los casos.

# Conclusiones

## Comparación con otros sistemas de análisis

Al realizar una comparación entre DiSPAC y algunos de los programas comerciales como; TAU y DATAQUEST ambos de MiniMiter Co. Inc., USA, y CronoBioPC de PanLab España. Encontramos que todos ofrecen el mismo número de análisis, los cuales son estándares en el área de cronobiología, sin embargo ninguno de ellos posee una interface gráfica que permita al usuario interactuar fácilmente. Los comandos se dan por medio del teclado, especificado los bloques o días que se desean analizar, por lo que el usuario debe visualizar sus datos y determinar manualmente el segmento a analizar, salir de la opción y analizar el segmento de interés o en su defecto analizar todo el archivo de datos.

Los parámetros que utiliza cada uno de los diferentes sistemas, como información para cada archivo, es muy semejante. Todos ellos pueden convertir archivos ASCII a su formato, sin embargo sólo uno puede exportar sus datos a archivos ASCII.

DataQuest es el único que incluye hardware para la captura de la información. mientras que Tau ofrece el proceso por lotes, es decir procesar varios archivos realizando los mismos análisis en todos, sin la intervención del usuario, característica con la que no cuenta DiSPAC.

El uso de DiSPAC por su facilidad, no solo se aplica a investigación, sino también a la enseñanza ya que por la variedad de análisis que ofrece, puede servir para que el estudiante observe las diferentes características que ofrece cada uno de ellos. En la siguiente tabla se realiza una comparación de los análisis soportados entre dichos sistemas.

Análisis soportados	DISPAC	DATAQUEST	TAU	CRONOBIO-PC
Incluye Hardware para captura		<input checked="" type="checkbox"/>		
Actograma en single plot y double plot	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Actograma Modulado	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Actograma Espectral	<input checked="" type="checkbox"/> 22 escalas 256/16 colores, sombreados			<input checked="" type="checkbox"/> Limitado a B/N 4 escalas
Periodograma Enright	<input checked="" type="checkbox"/>			
Periodograma Sokolove y Bushell	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Cosinor	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>
Densidad Espectral	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>
Histograma	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Histograma promedio	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Histograma promedio con desviación estándar	<input checked="" type="checkbox"/>			
Histograma promedio con error estándar	<input checked="" type="checkbox"/>			
Histograma sobrepuesto	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	
Prueba de T		<input checked="" type="checkbox"/>		
Análisis de Varianza		<input checked="" type="checkbox"/>		
Estadística descriptiva		<input checked="" type="checkbox"/>		
Demodulación compleja	<input checked="" type="checkbox"/>			
Plano Fase	<input checked="" type="checkbox"/>			
Arquitectura del ritmo	<input checked="" type="checkbox"/>			
Estimación gráfica de periodo	<input checked="" type="checkbox"/>			
Estimación gráfica de fase	<input checked="" type="checkbox"/>			
Auto correlación	<input checked="" type="checkbox"/>			
Opción a cambiar el aspecto y analizar la gráfica	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	

## BIBLIOGRAFÍA

- Aschoff J., *Handbook of behavioral Neurobiology, Vol 4 Biological Rhythms*. Plenum press New York 1984.
- Booch Grady . *Object Oriented Design with Applications*. The Benjamin Cummings 1991.
- Borland International Inc. *Borland C++ Programmer's Guide*, 1991.
- Borland International Inc. *Turbo Assembler 3.0 User's Guide*, 1991.
- Diez Martinez O. , Perez P. , Budelli R. , and Segundo J. P. . *Locking, Intermittency, and Bifurcations in a Periodically Driven Pacemaker Neuron: Poincaré Maps and Biological Implications*. Biol. Cybern. 60, 49-58 (1988).
- Digglet Peter J., Fisher Nicholas I. . *Nonparametric Comparison of Cumulative Periodograms*. Appl. Statis. 1991 40 No 3 423-434.
- Enright J. T. *The Search for Rhythmicity in Biological Time-Series*. J. Theort. Biol. (1965) 8, 426-468.
- Halberg, F., *Chronobiology*, Ann. Rev. of Physiol. 31: 675-725, 1969.
- Menaker Michael, *Biological Cloks*, Bioscience 19(8) 1969 681-689,692.
- Minors D. S. and Waterhouse J. M. *Mathematical and Statistical Analysis of Circadian Rhythms*. Psychoneuroendocrinology 13(6) 443-464, 1988.
- Norton Peter , Wilton Richard, *The New Peter Norton Programmer's Guide to the IBM PC & PS/2*, Microsoft press, 1988.
- Pittendrigh Colin S. and Daan Serge , *A Funcional Analysis of Circadian Pacemakers in Nocturnal Rodents I The satability and lability of spontaneos frecueny*, Journal of Comparative Physiology, 106 223-252 1976.
- Pressman Roger S. , *Ingeniería de Software un Enfoque Practico*. segunda edición McGraw-Hill 1989.
- Proakis John G. , Manolakis Dimitris G. , *Digital Signal Processing*, Maxwell Macmillan International Editons, 1992.
- Vega G Arturo, *Análisis Estadístico en Cronobiología*, Psiquis México, Vol. 2 Num. 6 1993, 139-149.

Sing Helen C. , Thorne David R , and Hegge Frederick W. . *Trend and Rhythm Analysis of Time-Series Data Using Complex Demodulation*. Behavior Research Methods, Instruments, & Computers. 1985, 17 (6), 623-629.

Sokolove Phillip G. and Bushell Wayne N. . *The Chi Square Periodogram: Its Utility Analysis of Circadian Rhythms*. J. Theor. Biol. 1978, 72, 131-160.

Stearns Samuel D. , Hush Don R. , *Digital Signal Analysis*, Prentice-Hall 1990.

Walter D.E. and Curtis R.J. . *The Combination of Results from Fourier Analyses in the Investigation of Biological Rhythms*. International Journal of Chronobiology, 3, 263-276, 1976.

Walter Nelson, Y. Liang Tong, Jung-Kuen Lee, Franz Halberg. *Methods for Cosinor-Rhythmometry*. Chronobiologia 1979 6 305-323.

Wilton Richard . *Programmer's Guide to PC & PS/2 Video System*. Microsoft Press 1987.

## Apéndice Breve Manual de Usuario

### DiSPAC

Digital Signal Processing Applied to Chronobiology  
Procesamiento Digital de Señales Aplicado en Cronobiología

Universidad Nacional Autónoma de México (UNAM)  
Instituto de Fisiología Celular (IFC)  
Departamento de Neurociencias  
Laboratorio de Cronobiología  
C.P. 04510 Apdo. Post. 70-253  
tel. 6225650 fax. 6225607  
e-mail avega@pacific.depfi.unam.mx  
raguilar@ifcsun1.ifsiol.unam.mx

(c) 1993 1994 Arturo Vega González UNAM-IFC-205ote.

Ayudado por el proyecto  
DGAPA IN202891

# ÍNDICE

Antes de Comenzar	iii
Como Usar el Manual	iv
Convenciones Utilizadas	iv
Acerca del Programa DiSPAC	v
Archivos que Conforman al Sistema	v
Instalación	vii
Configurando DiSPAC	viii
Trabajando con el Sistema	ix
Ejecutando DiSPAC	x
Salir de DiSPAC	xii
Opciones en DiSPAC	xii
Utilerias Externas	xvii

## ANTES DE COMENZAR

Antes de empezar a utilizar el sistema DiSPAC, deberá disponer de lo siguiente:

Una computadora personal del tipo IBM-PC ó compatible que soporte el sistema operativo MS-DOS versión 3.0 o posterior.

Memoria RAM de 640 Kb o más

Si se tiene memoria extendida ó expandida, DiSPAC puede utilizarla

Disco duro con al menos 2Mb de espacio disponible

Adaptador de Video de alta resolución VGA, EGA, CGA, HGC

Se recomienda VGA

Monitor a color (se recomienda)

Mouse (se recomienda)

Sistema operativo 3.0 o superior

Impresora compatible con impresoras Epson ó HP Láser

Microprocesador 80x86 Intel

Coprocador Matemático 80x87 (se recomienda)

Tener algunos conocimientos básicos de MS-DOS

El disco que contiene los archivos necesarios para instalar el sistema DiSPAC.

La computadora debe tener la variable de ambiente DSPTEMP ó TEMP, las cuales indican el directorio de archivos temporales.

# COMO USAR EL MANUAL

El presente manual tiene como finalidad presentarle el sistema DiSPAC, y enseñarle cómo utilizar diversas características del sistema. El manual está organizado de tal forma que pueda buscar con facilidad aquello que necesite saber, sin embargo, es necesario que revise las siguientes secciones: *Antes de comenzar*, *Como usar el manual*, *Convenciones utilizadas*.

## CONVENCIONES UTILIZADAS

A lo largo de este manual se utilizan las siguientes convenciones tipográficas para distinguir los elementos que aparecen en el texto:

**Negritas** Utilizada para comandos u opciones, y porciones literales que deben aparecer exactamente como se muestran.

subrayado Utilizada para remarcar algo que es muy importante y no debe pasarse por alto.

### Nombres de las teclas

En este manual los nombres de las teclas aparecen tal y como lo hacen en el teclado de su computadora, y en ocasiones de una forma abreviada, y en letras mayúsculas. En su teclado las caras de las teclas a veces tienen los nombres abreviados o representados en alguna forma diferente.

La siguiente tabla muestra sus equivalentes con los teclados más comunes en inglés para computadoras IBM y compatibles.

nomenclatura usada en este manual	teclado IBM y compatibles
ALT	Alt
CONTROL, ^	Control, Ctrl
ESC	Esc
DEL	Del, Supr
AV PAG	PgDn, Av Pág
RE PAG	PgUp, Re Pág
SHIFT	Shift, î
BACK	Back, BackSpace
MAYUS	Caps Lock
INICIO	Home, HOME, Inicio
ENTER	Intro, Return, Enter, RET
INS	Ins, Insert

# ACERCA DEL PROGRAMA DiSPAC

Un aspecto importante en cronobiología es el poder caracterizar los fenómenos rítmicos satisfactoriamente, de tal forma que pueda realizarse una estimación objetiva de sus parámetros y perturbaciones del ritmo. por lo que requiere el uso de técnicas gráficas y estadísticas específicas, las cuales no se encuentran en los sistemas comerciales.

Por lo anterior surge la necesidad de diseñar e implementar un sistema de análisis de señales biológicas en cronobiología, el cual responda a semejantes demandas. Las ventajas de esto serían varias, entre ellas un conjunto de análisis seleccionado por el investigador, despliegue de resultados fáciles de comprender, organización y almacenamiento de datos optimo para un mejor aprovechamiento de recursos, capacidad de expandirse de acuerdo a las necesidades del usuario. De tal forma que las características para dicho sistema son:

- Importación/Exportación y Organización de datos.
- Análisis matemático de series de tiempo acorde con las necesidades del usuario.
- Despliegue gráfico, fácil de comprender.
- Salida a impresora para la generación de reportes.
- Facilidad en el manejo del sistema.

De esta forma surge DiSPAC, un sistema para cubrir semejantes demandas, el cual es un sistema interactivo bajo un ambiente gráfico, además de contar con utilerías externas que le ayudan a ser más flexible en su uso, así como el poder comunicarse con otras aplicaciones, definiendo por medio de ellas los directorios de trabajo, tipos y modo de impresión, formato de datos, definición y creación de archivos a utilizar para umbrales y paletas de colores.

## ARCHIVOS QUE CONFORMAN EL SISTEMA

Los Archivos que forman al sistema DiSPAC son los siguientes:

Nombre	Descripción
DiSPAC.exe	Programa principal
DiSPAC.cfg	Archivo de configuración
DiSPAC.pal	Archivo de paleta para 256 colores
DiSPAC.shd	Archivo de paleta para 16 colores con asurado
DiSPAC.umb	Archivo de umbrales
DiSPAC.gra	Indica donde se encuentra los archivos de interface gráfica

DiSPAC.mnl	Archivo de texto que contiene el manual
Umbrales.exe	Define Umbrales (UMB)
Paletshd.exe	Define colores y asurados (SHD)
Paletvga.exe	Define paleta de colores (PAL) Nota: es el único al que hay que pasar el nombre del archivo al ejecutarlo, ejemplo : a>paletvga paleta.pal
SetUp.exe	Para configurar DiSPAC
Manual.exe	Despliega un breve manual en línea
DiSPAC.mnl	Archivo de texto que contiene el manual
Convert.exe	Para la conversión de formatos
Mdate.exe	Utilería para crear el archivo (INI)
Instala.bat	Instala DiSPAC en disco duro

En el disco de instalación sólo aparecen los siguientes:

Instala.bat	Instala DiSPAC en su disco duro
Dsp.zip	Contiene los archivos descritos anteriormente
readme.	Instrucciones para instalar
Pkunuzip.exe	Programa que sirve para desempacar los archivos

Los archivos con extensión BGI y CHR también son necesarios, estos archivos los provee Borland y son necesarios para la operación del sistema, ya que son la parte externa de la interface gráfica, utilizados para la inicialización de la interface. Estos archivos también están incluidos dentro del archivo Dsp.zip

Por otro lado al utilizar DiSPAC, se pueden generar archivos con diferentes extensiones, estos archivos los denominaremos archivos de ambiente, aunque algunos ya están creados y listos para utilizarse, así tenemos que los archivos de ambiente son

Extensión	Descripción
CFG	Archivo de configuración
GRA	Indica donde se encuentra la interface gráfica
PAL	Indica Paleta de colores para VGA en 256 colores
SHD	Indica paleta de colores y asurado
UMB	Indica Umbrales
LIN	Archivo de almacenamiento de ambiente
LIS	Listado de resultados en modo texto, salida de DiSPAC (ASCII)
PRN	Listado de resultados en modo texto, entrada a DiSPAC (ASCII)
MRK	Archivo de marcas realizadas sobre el actograma
MNL	Manual
INI	Archivo de fecha inicial de experimento

# INSTALACIÓN

La instalación del sistema DiSPAC en el disco duro de su computadora, es muy sencillo, y para realizar dicho proceso deberá seguir los siguientes pasos, una vez que ha verificado que cumple con los requerimientos necesarios (ver sección antes de comenzar):

1) Encender su computadora.

2) Verificar que existan al menos 2Mb de espacio libre en disco duro.

3) Insertar el disco flexible que contiene los programas del sistema DiSPAC, descritos anteriormente (Disco de instalación).

4) Desde el floppy donde se encuentra el disco de instalación teclee **instala a: c:** y teclee ENTER. La letra **a:** indica el drive donde se encuentra el disco de instalación y la letra **c:** indica donde se desea instalar el sistema. En la pantalla aparecerá el siguiente mensaje: Creando directorio DiSPAC ... Descomprimiendo archivos ...

Si la instalación se completo enviara el mensaje "Instalación completada" si fue exitosa entonces aparecerá el mensaje "Instalación Exitosa", en otro caso aparecerá el mensaje "Instalación no exitosa", verifique el espacio en disco duro.

Si la instalación se realizo con éxito aparecerá el manual en línea, donde la navegación se realiza por medio de la flechas y para activar la opción teclee ENTER, para salir teclee ESC. Al salir de el manual en línea aparecerá la pantalla de configuración, por favor configure DiSPAC adecuadamente, para que no tenga problemas en su uso, (ver sección configurando DiSPAC).

Una vez que ha instalado exitosamente DiSPAC en su computadora puede o no dar de alta la variable de ambiente DSPTEMP ó TEMP, probablemente ya la tenga si su computadora opera bajo MS-DOS versión 5.0 o superior.

Esta variable de ambiente le indica a DiSPAC el directorio para archivos temporales, para hacer más rápido al sistema puede direccionar el directorio a un disco en RAM. Si no define la variable DiSPAC tomara el directorio de trabajo para crear los archivos temporales.

## CONFIGURANDO DiSPAC

Después de instalar DiSPAC deberá ejecutar el setup de DiSPAC, esto con el objeto de configurar el ambiente de DiSPAC, de tal forma que los directorios donde fue instalado correspondan, así como los archivos gráfica, paletas, sombreado y umbrales correspondan con los deseados. Así como el tipo de impresora con que se cuenta.

El setup o configuración se ejecuta al instalar DiSPAC, sin embargo puede cambiar la configuración tantas veces como quiera, para hacerlo ejecute el programa llamado SetUp.EXE, el cual se encuentra en el directorio BIN.

La configuración es muy sencilla, la pantalla principal se muestra en la figura A.1, La navegación entre las opciones es por medio de las flechas, para activar alguna opción presione ENTER, la indicación sobre que opción es la seleccionada es indicada por medio de un mayor que del lado izquierdo de la opción y en diferente color. Observe que el estado actual del archivo de configuración se muestra en el lado derecho, así como el nombre del archivo de configuración.

DISPAC SETUP	
Universidad Nacional Autónoma de México IFC - CRONOBIOLOGIA (c) 1993 by UNAM-IFC-295ote	
<b>Menu</b>	<b>Status</b>
>Archivo CFG Lee Salva Directorios Impresora Formato Archivos Fin	Archivo : DISPAC.CFG  Dir DISPAC : C:\DISPAC\BIN Dir Datos : C:\DISPAC\DSP Dir BGI : C:\DISPAC\BGI  Impresora : HP LaserJet                    Puerto LPT1 Portrait Modo 75 dpi Formato Lectura : DISPAC  Archivo .PAL : DISPAC.PAL Archivo .SHD : DISPAC.SHD Archivo .UMB : DISPAC.UMB

Figura A.1 Pantalla principal de configuración

La lectura y salvado del archivo de configuración, se ejecutan sobre el directorio definido para DiSPAC, si no lo hay, toma el actual.

Por medio de la opción directorios, se definen los siguientes directorios de trabajo: para DiSPAC donde reside el programa, para datos, donde residen los datos y donde DiSPAC leerá los archivos de datos, para los archivos BGI, donde residen los archivos BGI y CHR que provee Borland.

DiSPAC soporta dos tipos de impresora, de matriz de puntos ó Láser, las cuales deben ser compatibles con impresoras EPSON, para las de matriz de puntos y para las láser compatibles con impresoras HP Láser Jet, se cuenta además con diferentes modos de impresión para cada una de ellas, los cuales básicamente son diferentes resoluciones, también se cuenta con la opción definir la orientación de la gráfica, horizontal ó vertical, Landscape ó Portrait, así como el puerto paralelo donde esta conectada la impresora.

La definición de los formatos actualmente esta restringida a dos, el formato del laboratorio de cronobiología y el formato DiSPAC, el cual se diseño exclusivamente para el programa, sin embargo se puede utilizar la utileria convert para la conversión de otros formatos a el formato DiSPAC.

Los archivos de paletas y umbrales, que se proporcionan son los de default, sin embargo si el usuario desea utilizar otros aquí es donde los tiene que definir, sólo basta poner el nombre del archivo. Estos archivos pueden crearse con las utilerias; paletvga, paletshd, y umbrales.

## TRABAJANDO CON EL SISTEMA

Las siguientes secciones están dedicadas al uso del sistema DiSPAC y están ordenadas en forma que la consulta de alguna duda sea de la forma más rápida y fácil. Para un mejor entendimiento de los ejemplos, notas, y comandos es necesario leer la sección "Convenciones Utilizadas".

La navegación entre las opciones se realiza mediante el mouse, la selección de alguna de ellas se realiza presionando el botón izquierdo del mouse, para las marcas sobre las gráficas se presiona el botón derecho.

## EJECUTANDO DiSPAC

La ejecución del sistema requiere que este bien configurado. Una vez hecho esto, hay que cargar el manejador del mouse (mouse.com), acto seguido, pasarse al directorio de DiSPAC y teclear DiSPAC -n

ejemplo

```
c:\>mouse<ENTER>
```

```
c:\>cd\DiSPAC\BIN<ENTER>
```

```
c:\DiSPAC\BIN>DiSPAC -n <ENTER>
```

La opción -n, indica a DiSPAC que se desea utilizar la memoria convencional, en el caso que se tenga memoria extendida o expandida, las opciones son -x ó -e, respectivamente, de esta forma se tiene más espacio en memoria convencional para datos.

El modo de despliegue por default es a colores si desea un modo en blanco y negro, debe pasar la opción -m en la línea de comando, para indicarle al programa que desea modo monocromatico.

Ejemplo: Se tiene memoria extendida y se desea el modo monocromatico, entonces se debe teclear como sigue;

```
C:\DiSPAC\BIN>DiSPAC -x -m <ENTER>
```

El orden de las opciones no importa, lo importante es que se debe dejar al menos un espacio entre ambas y se debe de dar alguna de las opciones de memoria.

Si el sistema se ejecuta correctamente se aparecerá una pantalla mostrando los créditos del programa figura A.2, para entrar a sesión debe presionar el botón que dice DiSPAC, para hacerlo seleccione con el mouse donde dice DiSPAC, debajo del actograma, y presione el botón izquierdo del mouse, o bien tecleé la letra "i" desde el teclado de su computadora. después de esto aparecerá la pantalla de trabajo, la cual se muestra en la figura A.3



Procesamiento Digital de Señales Aplicado a Cronobiología  
 Digital Signal Processing Applied to Chronobiology

Universidad Nacional Autónoma de México  
 Instituto de Fisiología Celular Laboratorio de Cronobiología  
 Arturo Vega González  
 Apoyado por Proyecto DGAPA IN202851 (c) 1993 UNAM-IFC

Figura A.2 Pantalla de inicio en DiSPAC

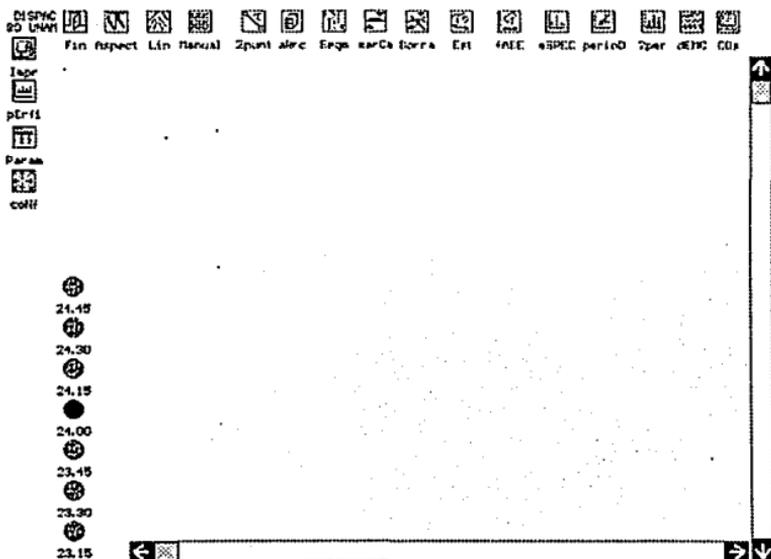


Figura A.3 Pantalla de trabajo en DiSPAC

# SALIR DE DiSPAC



Fin

Para salir de DiSPAC basta con seleccionar el Icono Fin y activarlo presionando el botón Izquierdo del mouse.

## OPCIONES EN DiSPAC

En la siguiente sección se describirá cada una de las opciones dentro de DiSPAC, La organización del sistema se puede apreciar en la figura A.4. y la pantalla de trabajo en la figura A.3



Aspect

La opción Aspect, es útil para cambiar el aspecto del actograma, por medio de ella podemos cambiar el ancho y alto del bin que representa un dato sobre el actograma, estos valores son en píxeles, también se puede seleccionar el tipo de actograma como; binario, escalado, asurado ó espectral en 256 colores, este ultimo sólo funciona para monitores VGA. Podemos definir la barra de luz-obscuridad sobre el actograma, definiendo hora de apagado y hora de encendido. Cuando no se colectaron datos estos están indicados por un punto, si no se desea que aparezca el punto se realiza mediante la opción Dper, dentro de Aspect. También se puede recortar los datos haciéndolos pasar por una ventana, la cual se define mediante la opción Clip.



Lin

Lin permite al usuario ocultar o mostrar uno ó todas las líneas de trabajo, si se selecciona una actura sobre la línea actual de trabajo, la cual se puede seleccionar mediante los botones circulares. Mediante la opción línea subopción informa, nos muestra una tabla con las posiciones de cada una de las líneas así como sus respectivos periodos, si los resultados son negativos indican que las líneas no están sobre el actograma. Por otro lado si se tiene un monitor monocromático no se podrá distinguir entre cada una de las líneas es por eso que se incluye la opción de modo, donde de acuerdo a un número se dibujará una línea punteada, punto-rama, etc.. También es útil para aumentar o disminuir el tamaño de una línea, el tamaño se define en días.



Manual

Por medio de esta opción podemos definir el periodo de la línea actual, el periodo se define en horas, si se tecléa 23.30, representa 23 horas con 30 minutos. una vez definida la línea con el periodo se podrá mover dentro del actograma mediante las barras de movimiento horizontal y vertical.



2punto

La Opción dos puntos, es la parte complementaria de Manual, y nos permite marcar dos puntos sobre el actograma, y como resultado cambia la línea actual a la línea formada por los dos puntos marcados, indicando además el periodo en horas de la misma

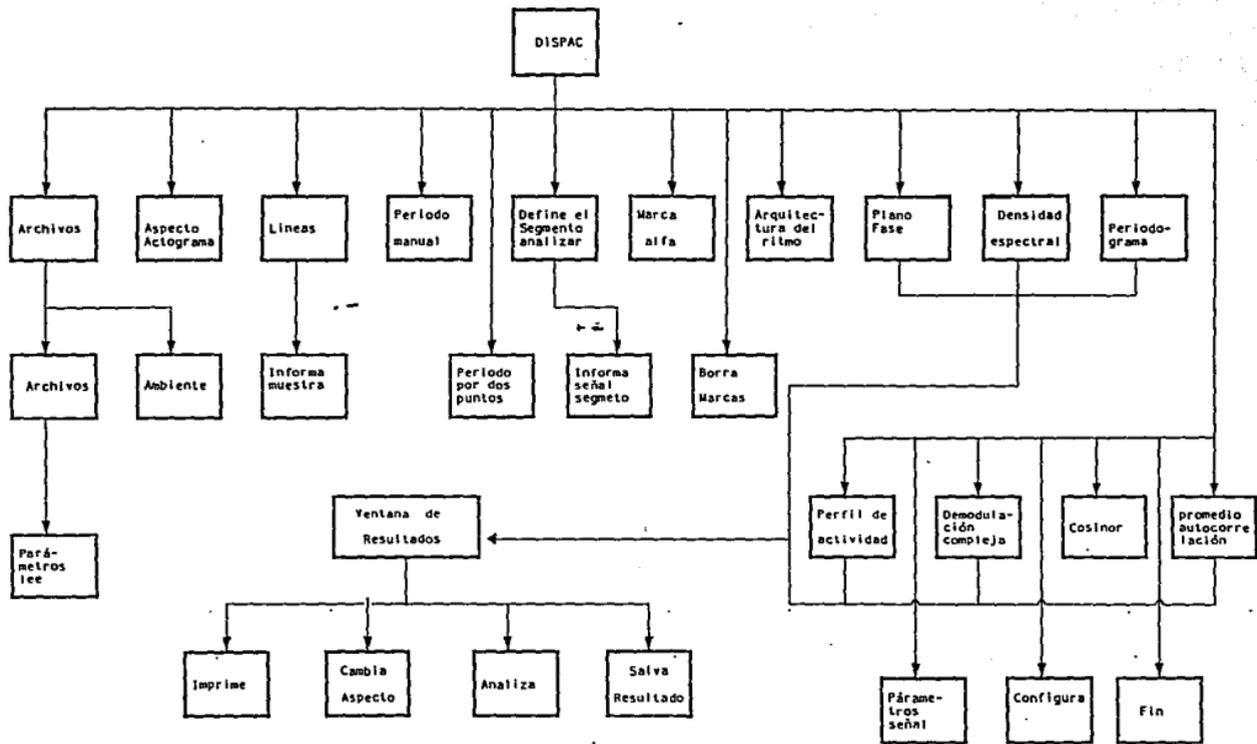


Figura A.4 Organización de DISPAC por opciones (menú)

 La opción aRc, abreviación de archivos, nos es útil para definir el archivo a analizar o bien los bloques a analizar, así como almacenar todos los parámetros y estado en el que se encuentra el sistema en ese momento, de tal forma que en otra ocasión que deseemos continuar lo podemos realizar cargando este archivo de ambiente, de esta forma no es necesario empezar de cero cuando ya se tenía algo.

 Segm, abreviación de Segmento sirve para definir el segmento a analizar, esto se hace marcando el inicio y final del segmento sobre el actograma, de esta forma podemos seleccionar exactamente el segmento de datos que deseamos almacenar. También nos es útil para saber que parte del registro corresponde al segmento y por su puesto datos acerca del registro que se muestra en el actograma, como fecha de inicio, identificación etc.

 marCa nos es útil para realizar las marcas de alfa sobre el actograma, de esta forma el usuario define las porciones que día a día corresponden a alfa, para terminar de marcar basta con indicar puntos que se encuentren fuera del actograma. La visualización de las marcas se realiza en un color diferente para que puedan ser distinguidas.

 Borra, por medio de esta opción se puede borrar una ó todas las marcas de alfa que se realizaron mediante la opción marca, si solo se desea borrar una marca se debe posicionar sobre la marca a borrar y oprimir el botón izquierdo del mouse. También es útil si deseamos ocultar las marcas realizadas de tal forma que no se observen sobre el actograma.

 Est, La opción Est, abreviación de Estabilidad, calcula apartar de las marcas realizadas, la estadística descriptiva, de alfa, rho tau, fi de alfa y fi de rho, teniendo la opción de almacenar toda esta información así como los cálculos en un archivo de texto, para referencias futuras.

 fASE es la opción por medio de la cual podemos obtener un plano fase de las marcas realizadas sobre el actograma, de alfa, rho tau, fi de alfa y fi de rho. Este plano fase se muestra en una ventana de resultado, la cual se describe más adelante.

 eSPEC, abreviación de espectro, al activar esta opción se calcula la densidad espectral, no sin antes tener la opción de pasar los datos por alguna de las siguientes ventanas; Haming, Hanning, Trinagular, Blackman y Harris. El

resultado ya normalizado, para mayor entendimiento del usuario se presenta en una ventana de resultado.



perioD

perioD, abreviación de Periodograma, realiza el calculo del periodograma teniendo dos opciones periodograma según Enright ó Chi cuadrada, el resultado también se presenta en una ventana de resultado.



2per

2per, abreviación de segundo perfil, nos da la oportunidad de realizar una promedio sobre el segmento a analizar, pudiendo observarlo en cuatro modalidades; sobrepuesto, solo el promedio, promedio con desviación estándar y promedio con error estándar.



dEMC

dEMC, abreviación de demodulación compleja, calcula la demodulación compleja para el numero de ciclos definidos por el usuario, mostrando la amplitud, fase y potencia en una ventana de resultado.



COs

COs, abreviación de cosinor, mediante esta opción se calcula el método de cosinor, dando como parámetros un periodo inicial y uno final en horas. El resultado a diferencia de los otros es mostrado en forma tabular, el cual se puede imprimir con tan solo presionar la tecla P desde el teclado de la computadora.



Impr  
trabajo.

Por medio de esta opción se define el tipo de impresora, el modo de impresión ó resolución, la orientación y el puerto en donde esta conectada la impresora. También es útil para imprimir el actograma desplegado sobre la pantalla de trabajo.



pEr fi

pEr fi, abreviación de perfil, con esta opción se puede observar el segmento seleccionado en una representación xy, también se puede obtener un plano fase del segmento donde la fase se toma como la hora en que ocurrió cada evento.

Por otro lado dentro de esta opción se puede calcular la autocorrelación. Cada una de estas opciones es mostrado en una ventana de resultado.



Par am

Param, abreviación de parámetros. Los parámetros que se pueden definir son: hora inicial, la cual se refiere a la hora en que comenzó el registro, y Fase de referencia, la cual indica si se debe tomar alguna fase especifica, esto es útil si se desea saber si existen avances o retrasos con respecto a una fase, en otro caso se tomara la hora en que ocurrió el evento. Para definir una fase se debe dar en horas, si no se desea se debe dar un valor negativo.



CONF

CONF, abreviación de configuración, por medio de esta opción se puede cambiar el directorio donde se leerán los datos, así como cambiar toda la configuración del sistema, configuración definida mediante la utilidad setup, con tan solo indicarle el nombre del archivo de configuración.

La ventana de resultado, figura A.5, de la que se ha hablado, tiene como características, que no solo muestra el resultado gráficamente, sino que además se cuenta con opciones como; imprimir la gráfica, cambiar el aspecto de la gráfica, y analizar.

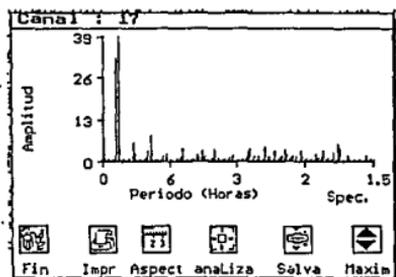


Figura A.5 Ventana de Resultado

La opción de cambiar el aspecto de la gráfica incluye, cambiar el número de marcas para cada eje, cambiar el máximo y mínimo en cada uno de los ejes, funcionando como un zoom, cambiar la forma en que se gráfica; puntos, línea, barras, y recortar los datos por un cierto margen de tal forma no se grafiquen ó tomen otro valor.

La opción de analizar se refiere básicamente, a que si el usuario esta interesado en conocer el valor de un punto en la gráfica lo puede hacer mediante esta opción. El usuario marcará dos puntos sobre la gráfica mediante el Mouse y se desplegará su valor para cada uno así como la distancia entre los mismos.

La opción de salvar almacena los datos desplegados en un archivo especificado por el usuario con la extensión LIS, para que de esta forma el usuario pueda generar la gráfica por medio de otra aplicación, o bien reanalizar los datos.

La opción de impresión, como su nombre lo indica imprime la gráfica en la impresora definida por el usuario, en la orientación y tamaño también definidos por el usuario.

Para la realización de cualquier de las opciones es necesario contar con datos en la memoria, desplegados en forma de actograma y tener definido un segmento de datos a analizar, esto lo puede hacer mediante las opciones aRRC y Segm.

## UTILERIAS EXTERNAS

DiSPAC cuenta con 7 utilerías externas, las cuales lo hacen más flexible en su uso y permiten la comunicación con otras aplicaciones. Las utilerías son:

Umbrales.exe	Define Umbrales (UMB)
Paletshd.exe	Define colores y achurados (SHD)
Paletvga.exe	Define paleta de colores (PAL)
SetUp.exe	Para configurar DiSPAC
Manual.exe	Despliega un breve manual en línea
Convert.exe	Para la conversión de formatos
Mdate.exe	Utilería para crear el archivo (INI)

Umbrales nos es útil para crear los archivos de niveles ó umbrales, con el que operara DiSPAC, en la generación de los actogramas espectrales, donde para cada color se le asigna un nivel mínimo y un nivel máximo, estos niveles están almacenados en un archivo con la extensión UMB, de tal modo que se pueden tener n archivos UMB, y para crearlos lo hacemos mediante la utilería llamada Umbrales. La pantalla del programa umbrales de muestra en la figura A.6

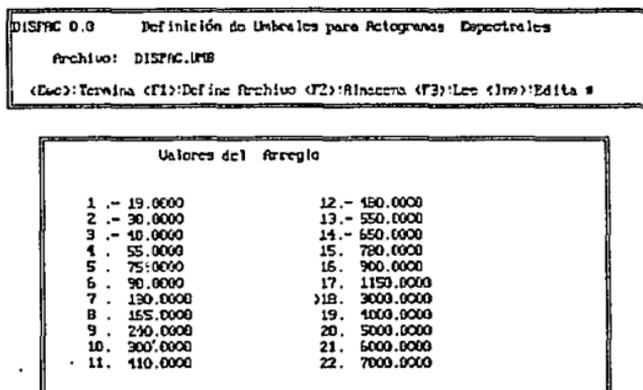


Figura A.6 Pantalla del programa umbrales

Mediante esta utilería no solo se puede crear un archivo UMB, sino que también se puede modificar uno ya existente. Esto permite al usuario definir sus propios archivos de umbrales, los cuales se pueden para que los utilice DiSPAC, es necesario definirlos mediante la configuración.

La navegación se realiza mediante las flechas, para insertar un número se debe presionar la tecla INS, al terminar de insertar o modificar, se debe presionar ENTER, el número a modificar es indicado por medio de un "mayor que" del lado derecho del número. Los intervalos se encuentran de menor a mayor, de tal forma que el primer intervalo es de cero al primer número definido en la pantalla, y así sucesivamente.

Por medio de la tecla F1, definimos el nombre del archivo a leer ó crear, con la tecla F2 almacenamos el archivo en disco, y con F3 leemos un archivo ya existente en disco el cual ha sido definido por medio de la tecla F1. Para salir del programa basta teclear ESC.

Paletshd y PaletVGA, son la parte complementaria de Umbrales, ya que por medio de estas utilerías podemos crear los archivos SHD y PAL respectivamente, los cuales contienen la definición de los colores y achurados que corresponden a cada nivel. El archivo con extensión PAL, solo es útil para monitores VGA y por lo tanto paletvga solo debe utilizarse en una computadora con monitor VGA.

Para paletshd es necesario al igual que DiSPAC, pasar las opciones de memoria y color en la línea de comando, siendo requisito pasar la opción de memoria. La navegación se realiza mediante el mouse, la línea superior muestra los posibles colores, la línea intermedia los posibles asurados, y la línea inferior muestra la combinación de ambas siendo esta la que se utilizara. Para cambiar alguno de los colores o asurados solo basta seleccionarlo con el mouse, la indicación de cual es el seleccionado actualmente es mediante un rectángulo que lo rodea. Para salvar el archivo es necesario dar la opción actualiza antes de salvar, ya que de otra forma no se actualiza la información vista con la almacenada en memoria.

Paletvga, solo funciona para computadoras que tengan un adaptador de video VGA preferentemente con monitor a color. La navegación en este programa es mediante las flechas, para la selección de un color es mediante la tecla ENTER, el color a seleccionar se encuentra marcado por un rectángulo. La paleta se muestra de lado derecho mientras que los posibles colores se muestran del lado izquierdo. para salir se debe presionar ESC, al salir automáticamente se almacena el archivo. La definición del nombre del archivo se hace por medio de la línea de comandos, ejemplo si el archivo a crear o modificar es ejemplo.pal entonces se debe ejecutar como sigue:

```
paletvga ejemplo.pal<ENTER>
```

La extensión para este archive debe ser PAL, ya que de otra forma DiSPAC no lo reconocerá.

SetUp es una utileria que nos ayuda a configurar DiSPAC, (ver configurando DiSPAC), definiendo los directorios de trabajo, el tipo de formato de los datos, el tipo y modo de impresora a utilizar, los archivos de umbrales y paletas a utilizar. Al definir una configuración esta puede ser salvada con un nombre diferente al de dispac.cfg, de esta forma dentro de DiSPAC se selecciona la opción configura dando el nuevo nombre de archivo, lo cual funcionara como si personalizaremos el uso de DiSPAC para diferentes usuarios.

Manual es un pequeño manual en línea que nos ayuda a resolver algunas dudas que se tengan con respecto a DiSPAC. Para la creación de esta utileria se utilizaron técnicas de hipertexto. La navegación se realiza mediante las flecha y la activación de la opción se realiza mediante la tecla ENTER, para salir del manual basta con teclear ESC. Manual utiliza el archivo DiSPAC.MNL, el cual se debe encontrar en el mismo directorio de Manual. Si lo desea puede imprimir el archivo DiSPAC.MNL ya que es un archivo de texto.

Convert es la utileria que nos permite comunicar con otras aplicaciones, ya que es la encargada de realizar la conversión de formatos, sin embargo no sólo tiene esas opción también puede compactar nuestros archivos, y por supuesto encriptarlos, así como verificar la integridad de nuestros archivos de datos.

La navegación se realiza con las flecha y la activación de la opción se realiza con ENTER. dependiendo de cual es el formato de origen CONVERT, pedirá algunos parámetros entre los cuales se encuentran; tamaño del día en intervalos, fecha de inicio del registro, identificación para el registro, numero de canal, y la hora inicial ó la hora en que comienza el registro. En la figura A.7 se observa la pantalla de Convert con los parámetros que se deben completar

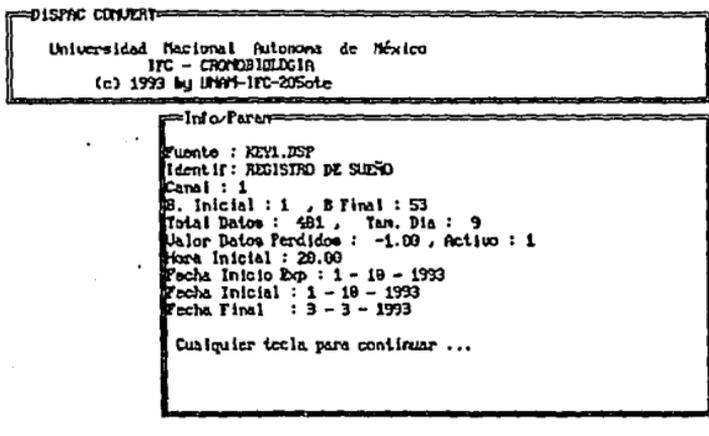


Figura A.7 pantalla de Convert, muestra parámetros de un archivo de datos.

Mdate es útil para crear el archivo DiSPAC.INI el cual se utiliza exclusivamente para datos provenientes del laboratorio de cronobiología, ya que estos no contienen información sobre el inicio del experimento. al ejecutarlo pedirá datos de la fecha día, mes y año, así como el directorio donde se desea almacenar el archivo DiSPAC.INI. Si los datos de la fecha son incorrectos los volverá a pedir hasta que estos sean validos.