

2-ey,
6



Universidad Nacional Autónoma de México

Facultad de Ingeniería

Sistema de Análisis por Computadora de la
Actividad Eléctrica Unitaria Extracelular

Tesis Profesional

Que para obtener el título de

INGENIERO EN COMPUTACION

P r e s e n t a

YOLANDA FIGUEROA MARQUEZ



Director de Tesis: Dr. Francisco Cervantes Pérez

México, D. F.

1986



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.

INDICE

CAPITULO 1	INTRODUCCION	
1.1	DESCRIPCION DEL PROBLEMA.	1-2
1.1.1	Análisis A Realizar.	1-6
1.2	OBJETIVOS DEL SISTEMA.	1-7
CAPITULO 2	ANALISIS TEORICO	
2.1	CONFIGURACION DEL SISTEMA NERVIOSO	2-1
2.2	ESTRUCTURA DE LAS NEURONAS	2-6
2.3	POTENCIAL DE MEMBRANA Y POTENCIAL DE REPOSO	2-11
2.4	POTENCIAL DE ACCION.	2-11
2.4.1	Pulso Difatorio	2-16
2.4.1.1	Descripción Funcional	2-16
2.4.1.2	Circuito Básico	2-18
2.5	ESTUDIOS SOBRE HISTOGRAMAS DE FRECUENCIA E INTERVALOS	2-22
CAPITULO 3	ENTRADA AL SISTEMA	
3.1	METODO DE REGISTRO	3-7
3.2	TIPO DE SEMAL REGISTRADA	3-8
3.3	EQUIPO ADICIONAL	3-10
3.3.1	Amplificador	3-10
3.3.2	Sonda De Alta Impedancia Y Acoplador De Impedancias	3-11
3.3.3	Grabadora	3-11
3.3.4	Bomba De Infusión	3-11
3.3.5	Regulador De Temperatura	3-11
CAPITULO 4	SISTEMA DE ANALISIS	
4.1	INTRODUCCION.	4-2
4.2	ANALISIS EN COMPUTADORA.	4-3
4.2.1	Módulo De Captura De Espigas	4-5
4.2.1.1	Proceso	4-7
4.2.1.2	Programación	4-10
4.2.2	Módulo De Análisis.	4-14
4.2.2.1	Histogramas De Frecuencia	4-14
4.2.2.2	Histogramas De Intervalo	4-14
4.2.2.3	Respuesta	4-15
4.2.2.4	Ajuste A Funciones.	4-17
4.3	IMPLEMENTACION EN UNA APPLE IIE.	4-22
4.3.1	Arquitectura Del Microprocesador 6502.	4-22
4.3.2	Registros De Propósito General	4-24
4.3.3	Registro De Estado Del Procesador	4-24
4.3.4	Stack	4-24
4.3.5	Organización De La Memoria	4-25

4.3.5.1	Espacio De Memoria Para Periféricos	4-27
4.4	INTERFAZ ANALOGICA/DIGITAL	4-30
4.4.1	Hardware	4-30
4.4.2	Lectura Y Escritura A Un Canal Del A/D 1 D/A	4-31
4.5	DISCUSION.	4-33

CAPITULO 5 RESULTADOS

5.1	GUIA DE USUARIO	5-1
5.1.1	Módulo De Captura	5-1
5.1.2	Módulo De Análisis	5-3
5.1.2.1	Generación De Histogramas	5-3
5.1.2.1.1	Impresión De Histogramas En Panel.	5-4
5.1.2.2	Ajuste A Funciones	5-6
5.2	EJEMPLOS	5-9
5.2.1	Ejemplo #1	5-10
5.2.2	Ejemplo #2	5-16
5.2.3	Ejemplo #3	5-20
5.2.4	Ejemplo #4	5-23

CAPITULO 6 CONCLUSIONES

6.1	EVALUACION DEL SISTEMA	6-1
6.1.1	Panel De La Ingeniería En Estudios Fisiológicos.	6-2
6.1.2	Sistema Desarrollado.	6-3
6.1.3	Análisis Desarrollado.	6-4
6.2	FUTURAS IMPLEMENTACIONES DEL SISTEMA	6-5
6.3	FUTURAS ETAPAS	6-5

APENDICE A PROGRAMAS PARA LA APPLE IIe

APENDICE B TABLAS AUXILIARES

APENDICE C GLOSARIO MEDICO

CAPITULO 1

INTRODUCCION

Nuestra percepción y acción ante el mundo dependen del procesamiento de información que realiza el sistema nervioso tanto del mundo externo como de los mecanismos internos que controlan el organismo humano. Una de las funciones más importantes de este procesamiento de información es el deseo de conocimiento, de adquisición de información para comprender y controlar el universo. Este problema a pesar de que ha inquietado al hombre durante toda la historia, todavía plantea problemas y misterios que aún no se pueden esclarecer y que actualmente son objeto de estudio de diferentes disciplinas como la neuroanatomía, la electrofisiología, la etología y los estudios teóricos del cerebro (Rolando Lara, 1995).

Sobre la organización cerebral, estudios clínicos en pacientes con lesiones propusieron la idea de que el cerebro no es una estructura homogénea, sino que está dividida en muchas regiones y que pacientes con lesiones en lugares específicos presentan un tipo particular de alteraciones funcionales en su conducta. De aquí nace la idea de que cada zona del cuerpo está representada por una zona cerebral. Esta teoría después fue corregida para postular que las funciones cerebrales no están localizadas en zonas específicas sino que la función está regulada por un grupo de zonas jerárquicamente organizadas y que dependiendo del nivel afectado es el grado de pérdida de la función.

El enfoque sobre el estudio de las funciones cerebrales sufrió un giro de 180 grados cuando Ramón y Cajal a finales del siglo XIX y principios del XX, demuestra que el sistema nervioso está constituido por un conjunto de células nerviosas a las que posteriormente se les llamó neuronas. A partir de este punto, surgen múltiples disciplinas que estudian la organización anatómica y funcional del sistema nervioso dando origen a lo que hoy conocemos

INTRODUCCION

como las neurociencias. Entre las más importantes se destaca la neuroetología, cuya preocupación es el explicar los mecanismos neuronales que controlan la conducta global de los organismos, en particular del hombre. Ante la aparición de las neurociencias, sobre todo de la neurofisiología, neuroetología, neuropsicología, etc., el problema de la localización de las funciones mentales ha podido ser estudiado más profundamente.

Es necesario para el estudio de la estructura y función del sistema nervioso, el estudio cuidadoso de la actividad de cada neurona en particular, denominada Actividad Unitaria. Debido a que este tipo de actividad se presenta como trenes de impulsos eléctricos, esto es, una secuencia de impulsos eléctricos semejantes a espigas, suele llamárseles también TRENES DE ESPIGAS. Una de las formas de atacar el problema del entendimiento de como funciona el sistema nervioso es, primero entendiendo la actividad de espigas de la neurona individualmente y luego, como ésta actividad se relaciona con la de otras neuronas en esta compleja red neuronal. Esto realiza la importancia de los procesos de espigas para un entendimiento de las interacciones neuronales (Glaser Edmund, 1976).

Sin embargo debe decirse que ésta es una área en la que no se ha proporcionado el suficiente soporte analítico a la biología experimental. Las dificultades matemáticas inherentes a la formulación y solución de problemas son grandes y lo mismo pasa en las técnicas prácticas para el análisis de los datos. En este estudio se presentan las ideas y técnicas utilizadas para el análisis de los datos de la actividad unitaria (Moore, Perkel y Segundo 1966).

1.1 DESCRIPCION DEL PROBLEMA.

Generada por estos problemas surgió la necesidad de cambiar la forma de análisis y registro manual que se hacía sobre estas señales, a una forma de adquisición y proceso que además de dar mayor precisión, fuera mucho más rápida y diera otras posibilidades en el análisis. Fue entonces cuando se pensó en automatizar estos procesos mediante el diseño de un sistema de análisis por computadora de la actividad eléctrica unitaria. Anteriormente para analizar esta actividad, el investigador registraba la señal y la grababa (generalmente en cinta magnética), mandándola además a un polígrafo, el cual imprimía la señal en papel, paso necesario ya que a partir de esos resultados en papel se realizaba el conteo de las espigas para la generación de los histogramas en forma manual. Este

INTRODUCCION

Proceso llevaba días la mayoría de las veces, imprimiendo además un error muy grande debido a fallas en la visualización correcta del fenómeno. Con este sistema automatizado se pretende brindar al investigador ahorro en tiempo y ganancia en precisión, así como algunos otros beneficios que se enunciaron más adelante. Antes de ello, mencionaré las características de las señales biológicas de actividad unitaria para poder formular las bases o premisas en que se fundamenta este sistema de análisis, así como las hipótesis que se pretenden demostrar con el mismo.

De una manera general, se puede decir que todos los procesos biológicos se pueden considerar como señales. Para ello primero se deben esclarecer los conceptos y propiedades de las señales con respecto a los procesos biológicos, debido a que el conocimiento de estas propiedades influencia, guía y algunas veces determina la forma de cómo deberá analizarse la señal, en la computadora.

Existen ciertos procesos biológicos en los cuales la información de mayor importancia está en la secuencia específica de puntos que denotan la presencia de eventos de la señal. En estos casos, sin embargo, no se les da importancia, o tan solo la básica, a las características morfológicas que presenta la señal. A este tipo de procesos se les da el nombre matemático de **Procesos Puntuales**. El ejemplo más importante de este tipo de procesos es el representado por la generación de potenciales de acción neural o "espigas" (1). Aquí la información transmitida por la neurona (de acuerdo con las bases actuales) se dice estar contenida sólo en la secuencia temporal de los potenciales de acción de la neurona (Capítulo 2). Consecuentemente uno ignora la detallada estructura del potencial hasta que se ve involucrado con el mecanismo de generación de espigas (registro intracelular), o al tratar de distinguir la actividad de una neurona de entre muchas.

Siguiendo esta filosofía, el presente estudio se basa sobre la siguiente premisa:

Tanto la amplitud como la duración de las espigas permanece constante en el tiempo, por lo que la morfología de la señal no es importante para el estudio de la actividad unitaria (registro extracelular).

(1) Consultar glosario de términos médicos en el apéndice C.

INTRODUCCION

Esto puede observarse en la representación del tren de espigas que presenta la figura 1.1. En ella puede observarse que tanto la amplitud como la duración de cada espiga son constantes y lo que no nos proporciona indicios de dónde se encuentra contenida la información que maneja el sistema nervioso. En la misma figura puede observarse también que la secuencia de eventos no es estacionaria, esto es, no es determinística, sino aleatoria o variante en el tiempo. Esto da la pauta para formular las hipótesis de trabajo sobre las que se basa este estudio.

- 1.- La información está contenida en la relación temporal que existe entre cada uno de los eventos (Frecuencia o Intervalos), más no en la morfología del evento en sí.
- 2.- La información se analiza como un Proceso estocástico (naturaleza Probabilística) o bien, como un Proceso matemático determinístico (tasas de cambio).

Aceptando que la duración y la amplitud de las señales es constante por lo que se puede ver que esos parámetros no nos proporcionan información sobre el manejo del sistema nervioso, se puede presuntar entonces, dónde está contenida la información que maneja éste ? La respuesta es que la información está contenida en la relación temporal que siguen los eventos, debido a que ésta varía en el tiempo (lo que se observa muy claramente en el tren de espigas de la figura 1.1), lo que hace pensar que, de ser así la información se podría estar transmitiendo a través de los cambios de frecuencia de las espigas o bien, por medio de los intervalos de separación interespigas, a las diferentes partes del organismo.

INTRODUCCION

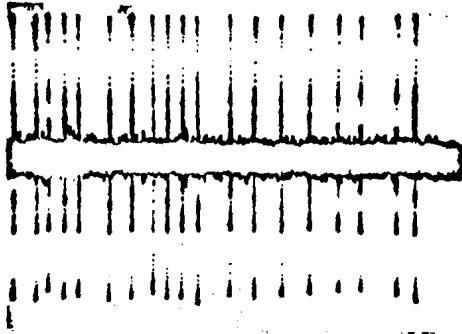


FIGURA 1.1.- Fotografía (pantalla del osciloscopio) de un tren de espigas en la descarga de una neurona del sistema nervioso.

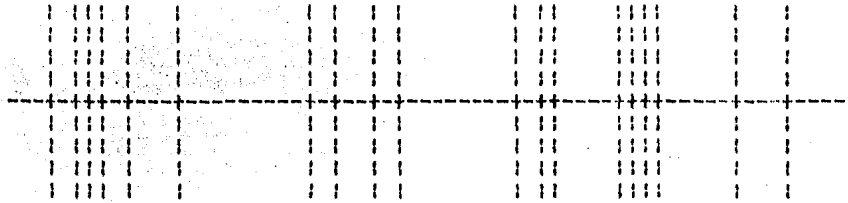


FIGURA 1.2.- Representación (a manera de modelo) de un Tren de Espigas neuronal.

Se tienen evidencias que prueban que esta actividad no es determinística y las repeticiones exactas de un experimento no dan siempre la misma respuesta eléctrica, en el sistema nervioso. Las diferencias aparecen como desviaciones en las latencias de picos característicos, cambios en las amplitudes de los picos y algunas

INTRODUCCION

veces en cambios en la forma de onda de la respuesta. Es por eso que se asocia cierta variabilidad con las respuestas de actividad unitaria. Los ejemplos principales de señales neurobiológicas no determinísticas son: la actividad de espigas de una sola neurona o de un pequeño grupo de neuronas aisladas y el Electroencefalograma (EEG). Las señales no son determinísticas porque los mecanismos responsables de ellas están sujetos a diversas influencias cuya respuesta conjunta nunca ha podido ser descritas totalmente (cada momento, cada condición).

Por esta razón las señales neurobiológicas no pueden ser descritas en forma inherente por una ecuación explícita válida para todo momento, y aunque fuera posible determinar una, no tenemos toda la información a la mano para poder hacerlo. Esto ha generado un problema de un interés biológico considerable especialmente debido a razones teóricas y prácticas. Este tipo de señales parece seguir leyes probabilísticas mas que determinísticas es por eso que al describir señales de este tipo se habla de medias, varianzas, etc., pero generalmente no se habla de sus descripciones funcionales excepto en algunos casos.

Es por esto, y debido a que las características relevantes de las señales de actividad neuronal unitaria son las de ser procesos puntuales y aleatorios, que la segunda hipótesis de trabajo de este estudio postule que estas señales se puedan analizar ya sea como procesos estocásticos (naturaleza probabilística), o bien determinísticamente en función de las tasas de cambio.

1.1.1 Análisis A Realizar.

Las investigaciones neurobiológicas en el procesamiento de información de sistemas sensitivos están basados en un amplio campo de técnicas experimentales. La explicación de cómo el animal reconoce los estímulos del medio ambiente puede obtenerse al registrar la actividad neuronal provocada por los potenciales eléctricos, en diferentes partes del sistema nervioso. Así como por los potenciales provocados que se registran a partir de estímulos provocados al animal en condiciones de laboratorio para comparar resultados de ambos tipos de pruebas y poder formular modelos. De esta manera el método de estudio de la respuesta neuronal es a través de estudios de comportamiento y técnicas de microestimulación (a base de electrodos), para posteriormente realizar un análisis de dichas respuestas neuronales (Goff, 1978).

INTRODUCCION

Hasta el momento casi la totalidad de la literatura plantea un tipo de análisis de actividad unitaria, el cual toma en consideración únicamente la naturaleza probabilística de estas señales; para lo cual se han utilizado herramientas tales como: Autocorrelaciones, Correlaciones Cruzadas, etc., las que como resultado arrojan probabilidades de ocurrencia de eventos o espigas en un determinado tiempo. Puede observarse entonces, que la información que nos proporcionan todas estas herramientas es cualitativa, no contando entonces con información cuantitativa sobre el proceso.

Con el fin de generar información cuantitativa sobre las tasas de cambio en este tipo de señales, se hace uso de dos herramientas (aplicadas en trabajos anteriores) para hacer el análisis de una manera diferente a como lo propone la mayoría de la literatura que sobre esta materia existe. Dichas herramientas son los Histogramas de Frecuencia y los Histogramas de Intervalos (de los cuales se hablará más ampliamente en el Capítulo 2), estos presentan la información de la neurona en dos formas diferentes en cuanto su distribución temporal, lo que proporciona dos puntos de vista muy distintos y bien definidos. El tipo de análisis, realizado es el determinístico, debido a que éste nos da resultados cuantitativos sobre la respuesta de actividad unitaria, los cuáles hasta el momento de realizar este trabajo no se habían obtenido y que podrían llegar a contribuir en el planteamiento de modelos teóricos sobre la conducta de una neurona en particular. Es así como se pretende aportar al investigador una herramienta extra de análisis, para que combinada con las anteriores tenga mayores elementos de apoyo en el análisis de sus resultados y en la obtención de sus conclusiones.

1.2 OBJETIVOS DEL SISTEMA.

El objetivo de este sistema es dar una herramienta más a las ramas de la Biomedicina y la Fisiología para el análisis y cuantificación de los complicados procesos que se realizan en el sistema nervioso, a través del estudio de las respuestas neuronales de actividad unitaria por computadora, tratando de encontrar posibles interrelaciones entre el funcionamiento de diferentes zonas del sistema nervioso y que influencias se ejercen entre ellos. Además se pretende que este sistema sea completamente transparente al usuario, permitiéndole a éste un despliegue gráfico a todos los niveles (adquisición de datos y análisis), diferentes métodos de análisis, adquisición de información a varios niveles (tiempo real o pregrabada) para finalmente facilitar todo el proceso de estudio de actividad unitaria y poder llegar a la formulación de modelos de

INTRODUCCION

comportamiento en las zonas de estudio.

El sistema se desarrolló basado en dos subsistemas principales: Captura y Análisis de información, siendo este último el encargado de brindar las herramientas de análisis en distintas formas. Los siguientes capítulos de este trabajo pretenden dar una mejor descripción tanto del sistema de análisis por computadora de la actividad eléctrica unitaria extracelular (SACAEUE), como del tipo de señales que se procesan y de su generación en el cuerpo humano. El Capítulo 2 da la introducción teórica sobre el tipo de señales biológicas que se analizarán así como de la zona del cuerpo humano que se tomó como ejemplo para la aplicación de este estudio. El Capítulo 3 trata sobre el equipo de registro que se necesitó para la implantación del SACAEUE. El Capítulo 4 describe por una parte el equipo de cómputo empleado, así como la implementación del sistema en el mismo y sus características. Los Capítulos 5 y 6 presentan los resultados y conclusiones a los que se llegó por la aplicación de este sistema a un problema real.

INTRODUCCION

BIBLIOGRAFIA

- 1.- Glaser, Edmund M.
Principles of Neurobiological Signal Analysis
Academic Press, Inc., (1976).
- 2.- Goff, W.R., Allison, T. and Vaughn, H.C.
Event related potentials in Brain.
Academic Press, N.Y., (1978).
- 3.- Lara y Zavala Rolando
La Cibernética del Cerebro.
En Prensa (1985).
- 4.- Moore, Gerstein and Perkel
Neuronal Spike Trains and Stochastic Point Processes
The Single Spike Train.
Biophysical Journal, (1966).
- 5.- Moore, Perkel and Segundo
Detection of Functional Interactions among Neurons.
Biophysical Journal, (1966).

CAPITULO 2

ANALISIS TEORICO

En el organismo humano se registran señales bioeléctricas en casi todas las regiones del mismo. Sin embargo, han sido los estudios sobre el cerebro y el sistema nervioso en general los que han proporcionado mayor cantidad de información en cuanto a la interrelación entre distintas zonas del organismo humano. Es por esto, que se ha considerado para el análisis de actividad unitaria una zona del sistema nervioso: el Bulbo Olfatorio y en particular de éste, el Núcleo del Tracto Solitario para observar la interrelación entre éste y las vías sástricas.

2.1 CONFIGURACION DEL SISTEMA NERVIOSO

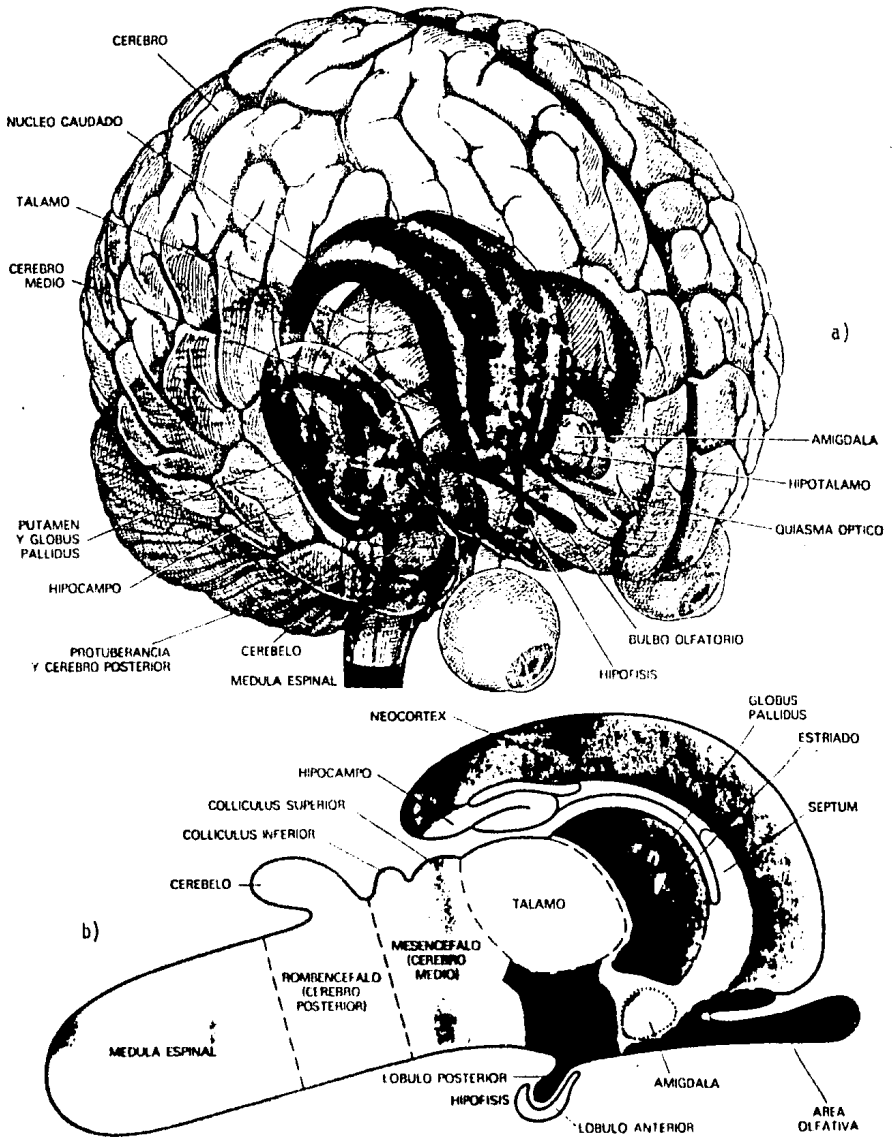
El cerebro es la estructura de mayor complejidad en el universo y está constituido por 10^{11} neuronas (aproximadamente) donde cada una de ellas recibe y manda miles de fibras de y a otras células del sistema nervioso. Ramón y Cajal observó que el sistema nervioso en general y el cerebro en particular no es una masa uniforme de tejido neural sino que está constituido por un conjunto de zonas heterogéneas. Cada una de estas zonas tiene una arquitectura distinta a la de las otras comprendiendo un diverso número de neuronas de diferente forma, tamaño, respuesta fisiológica, etc.. Gracias a las técnicas anatómicas desarrolladas hasta el presente se ha visto que estas zonas interactúan entre sí de múltiples maneras estableciendo circuitos y mallas de alta complejidad.

En forma general y con objetivos exclusivamente didácticos ya que la interacción y división de las zonas cerebrales es demasiado compleja y aún no está totalmente estudiada, el sistema nervioso se ha dividido en sistema nervioso central y sistema nervioso periférico. El sistema nervioso periférico está constituido por el sistema

ANALISIS TEORICO

nervioso simpático y el parasimpático que comunican de la periferia y hacia la periferia, la información proveniente del sistema nervioso central. El sistema nervioso central ha sido dividido en cerebro y médula espinal. El cerebro ha sido dividido en cerebro anterior (telencéfalo), cerebro medio (mesencéfalo) y cerebro posterior (rombocéfalo). El cerebro anterior está constituido por los hemisferios cerebrales, cuya superficie es la corteza cerebral, y comprende al Hipocampo, la neocorteza y los campos olfatorios. La parte inferior del cerebro anterior está constituida por el diencéfalo, formado éste por el Tálamo y el Hipotálamo. El cerebro medio está formado por el colículo superior e inferior y el cerebro posterior incluye al cerebelo. Finalmente, la médula espinal está constituida por las motoneuronas, interneuronas y fibras ascendentes y descendentes que regulan la actividad muscular y visceral del animal y que transmiten la información multisensorial. Las figuras 2.1 a,b y c nos muestran la interrelación entre las diferentes zonas del cerebro, mientras que la figura 2.2 nos muestra el diagrama general de la organización del mismo.

ANALISIS TEORICO



ANALISIS TEORICO

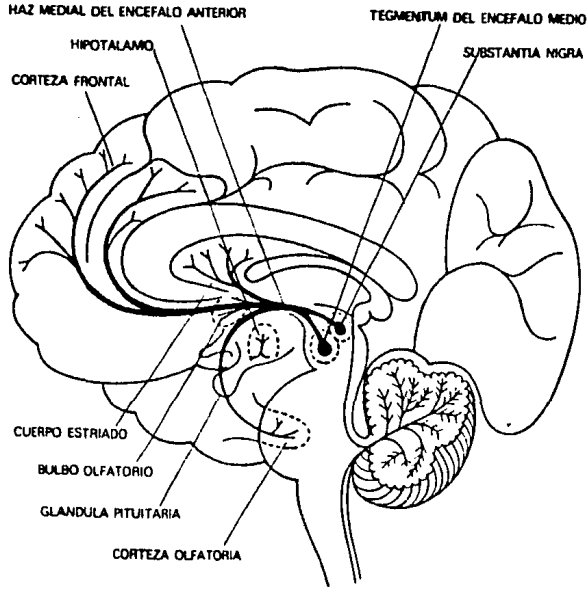


FIGURA 2.1.- El cerebro y la médula espinal (a,b) del hombre y otros mamíferos. En la parte superior se ha representado el cerebro de forma que las estructuras internas sean visibles a través de las capas externas 'transparentes'. En la parte media se ha representado un diagrama muy esquemático del cerebro de los mamíferos. En la tercera figura se muestra una configuración un poco más detallada de la relación que guardan ciertas zonas del cerebro humano.

ANALISIS TEORICO

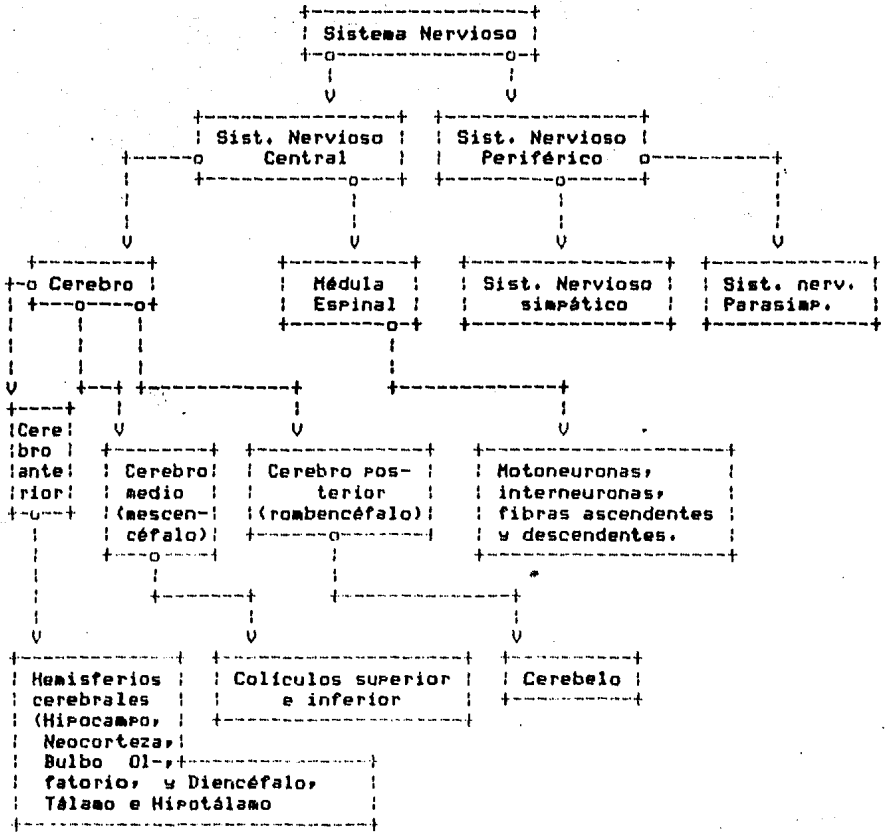


FIGURA 2.2 .- Estructura organizativa del cerebro.

ANÁLISIS TEÓRICO

2.2 ESTRUCTURA DE LAS NEURONAS

Desde un punto de vista morfológico la neurona es una célula diferente a las otras células del organismo ya que del cuerpo neuronal surgen prolongaciones finas que se desarrollan y ramifican a diferentes distancias. No existen dos neuronas iguales en cuanto a forma. No obstante, sus formas suelen agruparse en unas pocas categorías amplias y la mayoría de las neuronas comparten ciertas características estructurales que hacen posible distinguir tres regiones celulares (ver figura 2.3): el cuerpo celular (soma), las dendritas y el axón. El cuerpo o soma de la célula contiene el núcleo de la neurona y la maquinaria bioquímica para la síntesis de enzimas y otras moléculas esenciales para la vida de la célula. En general, la forma del cuerpo celular es variada. Las dendritas son delicadas expansiones en forma de tubo que tienden a ramificarse repetidamente formando un arbusto alrededor del cuerpo de la célula. Proporcionan la principal superficie física por la cual la neurona recibe las señales de entrada (aférentes). El axón se extiende a partir del cuerpo celular y constituye la vía por la información, una vez procesada por la neurona, puede viajar largas distancias desde el cuerpo celular a otras partes del sistema nervioso o del organismo. El axón difiere de las dendritas tanto por su estructura como por las características de su membrana. La mayoría de los axones son más largos y delgados que las dendritas y presentan un modelo de ramificación distinto: mientras las ramas de las dendritas tienden a agruparse cerca del cuerpo de la célula, las ramas de los axones tienden a nacer al final de la fibra, allí donde el axón se comunica con otras neuronas. La figura 2.3 muestra esquemáticamente la estructura de una neurona.

La información pasa de una célula a otra por puntos de contacto especializados: la sinapsis. Una neurona típica puede tener de 1000 a 10,000 sinapsis y puede recibir información de aproximadamente 1000 neuronas (2). Aunque las sinapsis se realizan con más frecuencia entre el axón de una célula y la dendrita de otra (axo-dendrítica), hay otros tipos de unión sináptica entre axón y axón (axo-axónica), entre dendrita y dendrita (dendro-dendrítica) y entre axón y cuerpo celular (axo-somática) figura 2.4a. En una sinapsis el axón suele dilatarse para formar el botón terminal, que es la parte de la unión que libera la información. El botón terminal contiene diminutas estructuras esféricas denominadas vesículas sinápticas (figura 2.4b), cada una de las cuales puede contener varios miles de moléculas de transmisor químico (neurotransmisor). La comunicación entre neuronas (la sinapsis) se lleva a cabo a través de un proceso bioquímico, a la

(2) El Cerebro (Scientific American), 1981 pag. 25

ANALISIS TEORICO

llegada de un impulso nervioso al botón terminal, algunas de las vesículas sinápticas liberan su contenido en la estrecha hendidura (espacio sináptico) que separa el botón de la membrana de otra dendrita, axón, o cuerpo celular, destinada a recibir el mensaje químico. Por tanto, la información se transmite de una neurona a otra mediante la combinación neurotransmisor con el receptor. La sinapsis se realiza en un solo sentido, es decir, desde el botón terminal axónico (presinapsis) hacia las dendritas o cuerpo celular de la otra neurona (postsinapsis). Algunas sinapsis son excitadoras por cuanto tienden a provocar la puesta en marcha; mientras que otras son inhibitorias, siendo capaces de cancelar señales que de otra manera excitarían a una neurona. Las neuronas pueden funcionar así porque sus membranas tienen propiedades especiales, por ello buena parte de la investigación reciente está enfocada sobre el estudio de las propiedades de la membrana, responsables del impulso nervioso, de la transmisión sináptica y de la comunicación célula-célula.

ANÁLISIS TEÓRICO

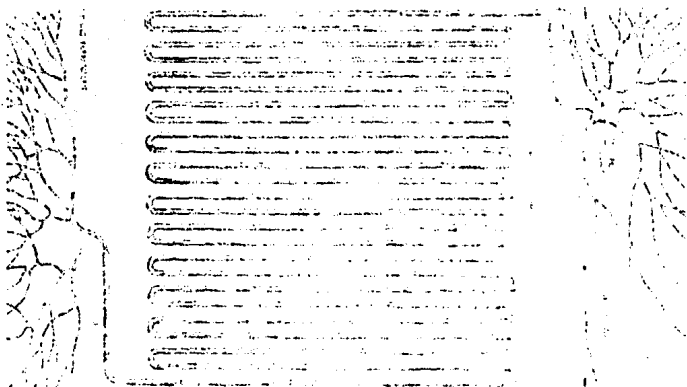


FIGURA 2.3.- Representación esquemática de una neurona típica en un animal vertebrado aumentada 250 veces. El axón, reflexado por conveniencia de representación, tendría un centímetro de largo a tamaño real. Fotografía tomada del libro el Cerebro, Scientific American página 26.

ANÁLISIS TEÓRICO

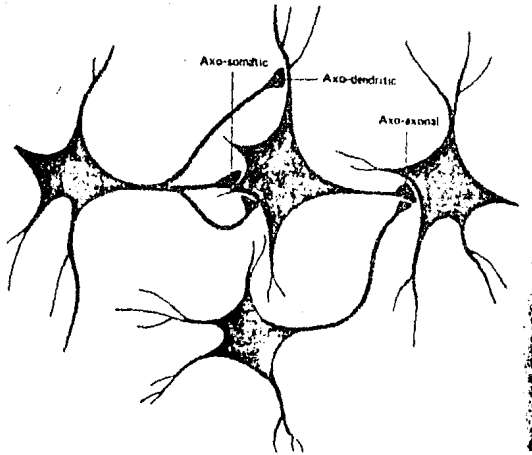


FIGURA 2.4a.- Diagrama esquemático que muestra los diferentes tipos de sinapsis de acuerdo a su localización.

ANÁLISIS TEÓRICO

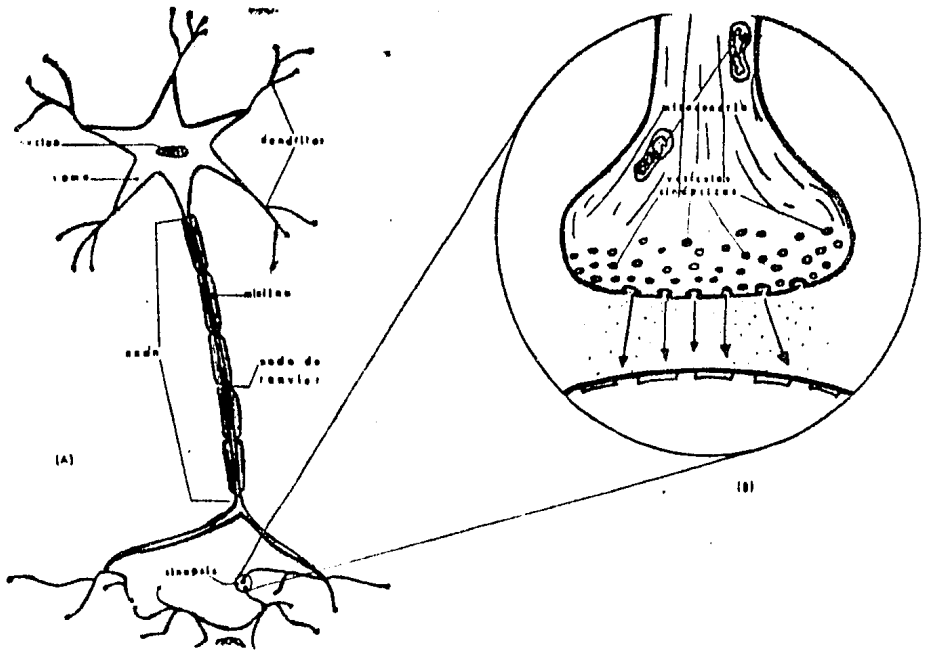


FIGURA 2.4b.- Diagrama de una neurona y detalle de una sinapsis donde se observan las vesículas sinápticas y la liberación de el neurotransmisor.

ANÁLISIS TEÓRICO

2.3 POTENCIAL DE MEMBRANA Y POTENCIAL DE REPOSO

La membrana de las neuronas, como la membrana de todas las células, tiene un espesor de unos cinco nanómetros y consta de una estructura trilaminar típica (Aidley 1978). La diferencia entre membranas celulares radica en las proteínas específicas que están asociadas con ella. Su función es mediar la comunicación entre los medios intra y extracelular. Debido a que tanto el medio extracelular como el intracelular son ricos en iones de sodio, potasio, cloro y calcio, y que éstos se encuentran en distinta concentración en el interior y el exterior de la célula, existe una diferencia de potencial a través de las mismas, siendo el interior negativo con respecto al exterior. A esta diferencia de potencial se le ha denominado potencial de reposo de la membrana y por convención se escribe con un signo menos, significando que el interior es negativo con relación al exterior. Su magnitud varía considerablemente de tejido a tejido, y va desde -9 hasta -100 mV (Ganong 1984).

2.4 POTENCIAL DE ACCIÓN.

Una de las funciones principales de las neuronas es la de transferir señales a largas distancias, de receptores periféricos al sistema nervioso central, de éste a músculos y glándulas, y entre diferentes partes del mismo. La transferencia de información entre neuronas tiene lugar en los axones mediante salvas de potenciales de acción o trenes de espigas.

En sus estudios clásicos sobre la transmisión del impulso nervioso en el axón gigante del calamar hace un cuarto de siglo, A.L. Hodgkin, A.F. Huxley y Bernard Katz (1960), demostraron que la propagación del impulso nervioso coincide con cambios repentinos en la permeabilidad de la membrana del axón respecto a los iones sodio y potasio. Cuando un impulso nervioso empieza en el origen del axón, habiendo sido disparado (en la mayoría de los casos) por el cuerpo celular en respuesta a una excitación en las sinapsis dendríticas, la diferencia de voltaje a través de la membrana del axón disminuye localmente. Inmediatamente por delante de la región alterada eléctricamente los canales de la membrana se abren, permitiendo que los iones sodio entren a raudales en el axón. El proceso se refuerza a sí mismo: el flujo de iones sodio a través de la membrana abre más canales, haciendo más fácil la entrada de otros iones posteriores. De ahí en adelante el potencial asciende rápidamente, sin embargo si el potencial no logra vencer un cierto umbral característico de la neurona (-45 mV aprox.), solo producirá un potencial local, que no será transmitido por la neurona postsináptica, si por el contrario el

ANÁLISIS TEÓRICO

potencial rebasa el umbral entonces éste asciende rápidamente alcanzando la línea isopotencial (potencial cero) y sobrepasándola aproximadamente a +35 mV esto es, se genera una inversión del potencial debido a la entrada de iones sodio, calcio y potasio. La brusca diferencia de potencial primero y negativa después, se presenta como una espiga en un osciloscopio, se conoce como el potencial de acción y es la manifestación eléctrica del impulso nervioso. Durante el potencial de acción la membrana es refractaria a otros estímulos y sin embargo durante el inicio del potencial de acción está hiperexcitable. Una vez abiertos, los canales sodio se cierran muy pronto, abriéndose otro grupo de canales que dejan salir los iones potasio. Este flujo de salida reestablece el voltaje dentro del axón a su valor de reposo de -70 mV.. Esta onda de voltaje avanza a través del axón hasta alcanzar su extremo, de manera análoga a como viaja la llama a lo largo de la mecha de un petardo, transmitiendo así el impulso nervioso o espiga. Las características esenciales de los potenciales de acción se muestran en la figura 2.5.

La esencia de este mecanismo es el de ser regenerativo; en este aspecto se asemeja a una mezcla de dinamita explosiva, algunos dispositivos electrónicos y muchos otros sistemas físicos y químicos. Esta propiedad regenerativa puede ser vista como el mecanismo necesario para la transmisión de señales. La porción de la membrana que está sufriendo el cambio del potencial en ese momento, genera una corriente que actúa para depolarizar a la porción vecina de la membrana. Esta nueva porción realiza exactamente el mismo ciclo, y así sucesivamente a lo largo del axón. El potencial de acción al final del axón es idéntico al del inicio del axón; no existe pérdida en la transmisión (este concepto es aceptado actualmente). El ciclo de generación de una secuencia de impulsos es ilustrado en la figura 2.6.

De acuerdo a todo esto, una pregunta fundamental será: cómo se transfiere información a lo largo de los axones por medio de este tipo de potenciales?. La hipótesis es, que en el axón, la información puede estarse transmitiendo a manera de códigos de frecuencia, o códigos de intervalos, reflejando los intervalos de tiempo entre impulsos sucesivos, o la longitud de los mismos. Todos los impulsos nerviosos tienen la misma amplitud, de modo que la información que llevan está representada por el número de impulsos generados por unidad de tiempo, un sistema conocido como codificación de frecuencia. Cuanto mayor es la magnitud del estímulo a transmitir, más rápido es el ritmo de activación. Es por esto que las neuronas son capaces de generar impulsos nerviosos a diferentes frecuencias, desde uno o menos por segundo a varios centenares por segundo.

ANÁLISIS TEÓRICO

Como hemos visto, la intensidad de un estímulo viene codificada por la frecuencia de los impulsos nerviosos. El descifrado en la sinapsis se lleva a cabo mediante dos procesos: suma temporal y una suma espacial. En la suma temporal, cada potencial postsináptico se añade al total acumulativo de sus predecesores para conseguir un cambio de voltaje cuya amplitud promedio refleje la frecuencia de los impulsos nerviosos entrantes. En otras palabras una neurona que está produciendo impulsos rápidamente libera más moléculas de transmisor en sus uniones terminales que una neurona que actúe con menor rapidez. Cuantas más moléculas de transmisor se liberan en un tiempo dado, más canales se abren en la membrana postsináptica y, por lo tanto, mayor es el potencial postsináptico. La suma espacial es un proceso equivalente, con la excepción de que refleja la integración de los impulsos nerviosos que llegan de todas las neuronas que pueda haber en contacto sináptico con una neurona dada. El cambio de voltaje obtenido por suma temporal y espacial está codificado bajo la forma de frecuencia o intervalos entre impulsos nerviosos para la transmisión a otras células "rio abajo" en la red nerviosa. Se ha descrito lo que comúnmente se considera como el flujo normal de información en los circuitos nerviosos, en los cuales los cambios de voltaje postsináptico están codificados bajo la forma de frecuencia del impulso nervioso o intervalos entre impulsos nerviosos y son transmitidos por el axón a otras neuronas.

ANALISIS TEORICO

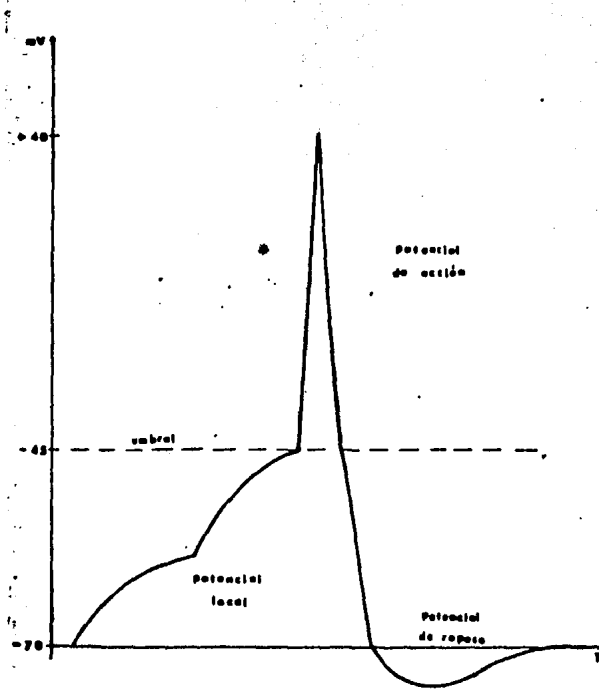


FIGURA 2.5.- Respuesta de la neurona a una excitación donde se muestra el potencial local, así como el potencial de acción cuando el estímulo alcanza el umbral.

ANALISIS TEORICO

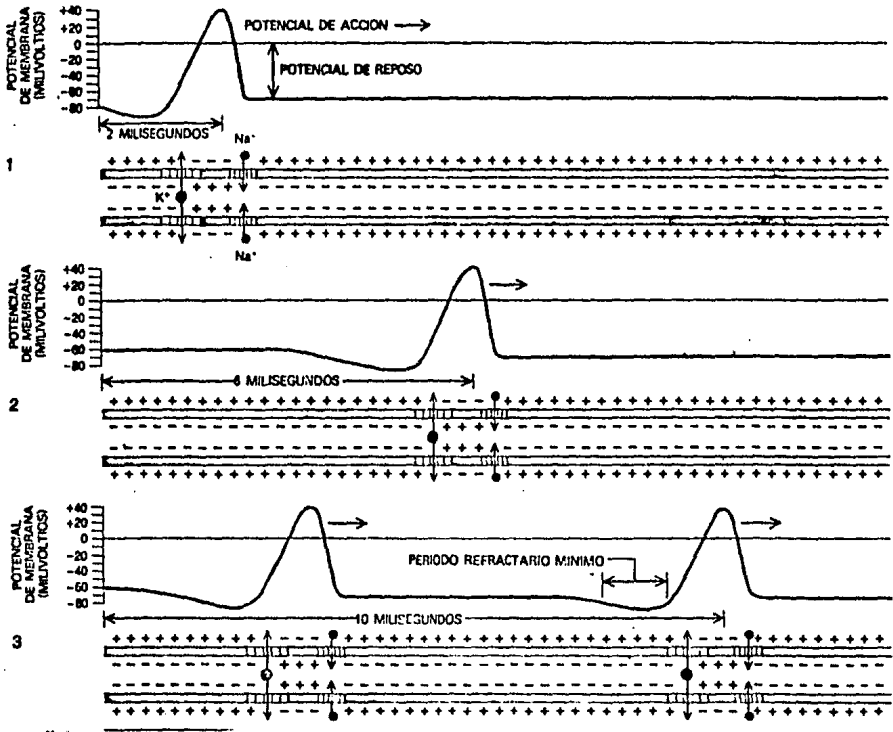


FIGURA 2.6.- Propagación del impulso nervioso a lo largo del axón, por cambios de voltaje a través de la membrana del mismo. La afluencia de iones sodio se realiza hasta que la superficie interna de la membrana es positiva localmente. La inversión de voltaje cierra el canal de sodio y abre el canal de potasio. La salida de iones potasio restablece rápidamente el potencial negativo. La inversión de voltaje, conocida como el potencial de acción, se propaga a lo largo del axón (a,b). Después de un breve periodo refractario puede seguir un segundo impulso (c).

ANÁLISIS TEÓRICO

2.4.1 Bulbo Olfatorio

Los histólogos clásicos del pasado siglo XIX tomaron un gran interés en el Bulbo Olfatorio, debido a sus laminaciones muy bien diferenciadas y sus tipos de neuronas asudamente diferenciadas también. Las deducciones a las que llegó Ramón y Cajal (1911, 1955) y sus contemporáneos sobre la organización de las neuronas en el Bulbo Olfatorio, Jusó el rol central en el desarrollo de los conceptos funcionales de la doctrina clásica de la neurona. Después de este período, sin embargo, el interés se disminuyó debido en gran parte a la casi por completo falta de progreso en el entendimiento de la naturaleza de los estímulos olfatorios. El interés en el Bulbo Olfatorio revivió por los estudios anatómicos de Le Gros Clark y colaboradores (1951, 1957, 1953) y los estudios fisiológicos de Adrian (1950, 1953). El conocimiento de la organización sináptica comenzó a principios de los setentas con investigaciones electrofisiológicas utilizando registros unitarios. Estos estudios tomaron como base el hecho de que las vías de entrada y de salida del Bulbo están completamente separadas.

2.4.1.1 Descripción Funcional -

El Bulbo Olfatorio es la interfase más importante de las vías olfatorias, ya que ahí terminan las células sensoriales de la nariz. Se encuentra constituido por las siguientes capas: la capa de nervios olfatorios, la capa de glomérulos, la capa flexiforme externa, la capa de células mitrales, la capa flexiforme interna y la capa de células granulosas. La neurona más importante del Bulbo Olfatorio es la célula mitral, que es la célula eferente de esta región. Las fibras nerviosas de los receptores olfatorios penetran al Bulbo Olfatorio, donde convergen y hacen sinapsis con las dendritas de las células mitrales. La salida del bulbo va hacia varias regiones cerebrales y a su vez varias zonas del cerebro le mandan fibras. La entrada aferente al bulbo se realiza por medio de los axones de las células sensoriales de la mucosa olfativa en la cavidad nasal. Los axones de las vías olfatorias terminan en una capa de regiones esféricas de tejido nervioso al que se le ha denominado glomérulo, que puede considerarse como un complejo grupo de interacciones sinápticas aisladas por mielina (3) que aparentemente actúan como unidad funcional para el procesamiento de información del sistema nervioso. Todas las fibras olfatorias terminan en el glomérulo. Las células mitrales envían sus dendritas hacia el glomérulo y hacia la capa

(3) Consultar glosario de términos médicos (Apéndice C)

ANALISIS TEORICO

flexiforme. Los axones de estas células se dirigen hacia el fondo del Bulbo y dan origen a la vía Olfatoria lateral. y a la cinta olfatoria, viajan hasta el tubérculo olfatorio la corteza flexiforme y el área periamigdalóidea, desde allí los mensajes olfatorios son conducidos mediante proyecciones hasta el tálamo y el hipotálamo, donde intervienen activamente en el control de la alimentación y del comportamiento reproductor. También en la cinta olfatoria hay fibras centrifugas que ejercen una influencia inhibitoria sobre el Bulbo Olfatorio. Existen varias entradas centrales al Bulbo que vienen del cerebro, algunos sitios de origen han sido indicados en el diagrama de la figura 2.7. Axones que son relativamente largos (varias micras de diámetro) pero pocos en número vienen de largas distancias, de una región en la base del cerebro llamada la banda diagonal (DB). Otros axones, finos y mucho mas numerosos, vienen de la región posterior al Bulbo, el núcleo olfatorio anterior (AON).

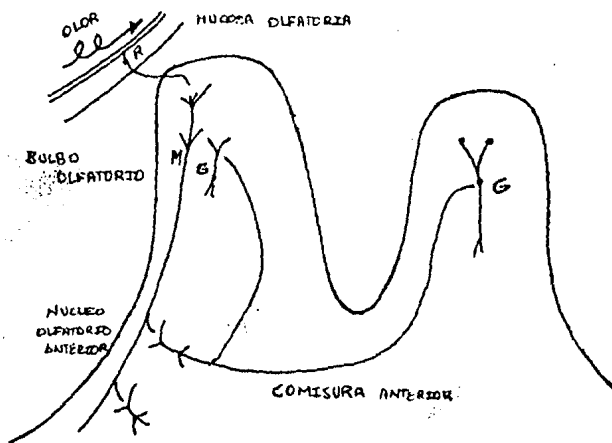


FIGURA 2.7.- Conexiones del Bulbo Olfatorio. R, células receptoras olfatorias; M, células mitrales; G, células granulosas.

ANÁLISIS TEÓRICO

2.4.1.2 Circuito Básico -

Los axones de las vías olfatorias hacen sinapsis en el glomérulo, en las dendritas de las células mitrales y en las dendritas de las células periglomerulares. En la capa plexiforme se realiza la interacción dendrodendrítica recíproca entre las dendritas de las células mitrales y las espinas dendríticas de las células granulares. Finalmente, las diversas regiones del cerebro que envían fibras al bulbo llegan a diferentes niveles de la capa granular de las células granulares. Una característica evidente del circuito básico es que hay vías verticales directas de entrada-salida y horizontales que comunican diversas zonas del bulbo olfatorio. Los dos niveles claros de procesamiento de información se localizan, primero en el glomérulo que procesa la información olfatoria, y segundo en la capa plexiforme, donde se encuentran las células mitrales que procesan la información de salida. Es importante notar que las células granulares no solo reciben aferentes de las células mitrales, sino además reciben fibras de otras regiones cerebrales, lo que sugiere que otras zonas podrían modular la acción del bulbo a través de estas conexiones que, las cuales actuarían como reguladoras de la actividad del bulbo tanto por causas internas como externas (Rolando Lara, 1985).

Debido a que la estimulación de las vías olfatorias no es sencilla, el estudio de la interacción neuronal de las células del bulbo olfatorio se ha realizado básicamente por técnicas electrofisiológicas. Y esta ha sido una de las zonas donde los estudios teóricos han podido aportar resultados concretos sobre la posible función y procesamiento de información que realiza. Por ejemplo, ha sido reportada la existencia de modificaciones en la actividad electroencefalográfica del Bulbo Olfatorio y en los potenciales evocados por estimulación hipotalámica (Aguilar Baturovi, 1976). Los potenciales evocados y las técnicas de inyección de actividad unitaria han sido usados para estudiar los efectos de la estimulación del nervio vaso en el Bulbo Olfatorio. Estos resultados sugieren la existencia de una vía Nervio Vaso-Bulbo Olfatorio, mediante la cual la actividad neuronal hipotalámica puede ser modulada por la estimulación del nervio vaso y por la distensión estomacal o gástrica (García Díaz et.al. 1984). Ambos, las líneas olfatorias y el nervio vaso, están involucrados en la regulación de la asimilación de comida debido a que se observaron proyecciones olfatorias en el hipotálamo lateral y en el área preóptica lateral usando la técnica de tinción HRP (4) las que han confirmado la estrecha interrelación entre el hipotálamo y el sistema olfatorio (Berone, et.al. 1981). Se vio también, que la distensión estomacal influenciaba la actividad de las neuronas hipotalámicas laterales que son sensibles a los estímulos olfatorios, y que la estimulación del nervio vaso provocaba la

ANÁLISIS TEÓRICO

alimentación en animales va asociado, aunque la existencia de conexiones neurales entre el hipotálamo posterior y las estructuras olfatorias no excluyen la posibilidad de influencias de otras fuentes (García Díaz et. al. 1984).

Es por esto que no se sabe aún si ésta vía es la única que interviene en el control de la alimentación. Estudios anatómicos sobre el Bulbo Olfatorio han llegado a suscribir la posibilidad de una interacción entre una nueva zona del cerebro y el control de la alimentación, así como del reconocimiento de estímulos tanto gustativos como olfatorios (García Díaz, 1984). Esta zona es el Núcleo del Tracto Solitario también conocida como Núcleo del Fascículo solitario.

Todas las fibras aferentes viscerales, esto es, fibras procedentes de las papilas gustativas, y otras procedentes de la parte posterior de la lengua y de faringe, laringe, tráquea, esófago y vísceras torácicas y abdominales, corren en el fascículo solitario, con el cual descienden a variable distancia (fig. 2.9). Terminan en una columna de neuronas que en parte rodean al fascículo y en parte están esparcidas entre sus fibras. A esta aglomeración celular se le denomina núcleo del fascículo solitario o núcleo del tracto solitario. Es un núcleo largo y delgado que aparece mejor desarrollado a nivel del origen superficial del nervio vago, donde está situado en posición lateral con respecto al núcleo motor de este nervio, y a poca distancia por debajo del piso del cuarto ventrículo. Este núcleo se extiende en dirección craneal. La porción rostral del núcleo es la parte especialmente relacionada con el sentido del gusto. Las vías aferentes secundarias del núcleo del fascículo solitario no se conocen bien. Como los impulsos gustativos despiertan sensación de sabor y los impulsos aferentes procedentes de las vísceras pueden adquirir una representación en la conciencia, debe de haber alguna vía aferente visceral secundaria, se sabe muy poco acerca de su situación (Gordon Shepherd, 1979). Las fibras nacidas del núcleo del fascículo solitario entran en la formación reticular y es probable que la mayor parte de ellas se distribuyan en los núcleos motores viscerales del Bulbo (figura 2.8) Debido a esto se tomó esta posible vía para el estudio de la interrelación entre esta zona del cerebro y el Bulbo Olfatorio (y que constituye el ejemplo de aplicación para la presentación de este trabajo) en cuanto a la actividad relacionada con la alimentación y el reconocimiento de estímulos sástricos.

En el presente estudio solo se analizan este tipo de estímulos (sástricos), por tomar un ejemplo, pero debido a que el sistema que

(4) Horseradish Peroxidase o Peroxidasa de Rábano

ANALISIS TEORICO

analiza estas interrelaciones es aplicable a cualquiera interrelación entre dos zonas del cerebro, es factible el realizar el estudio sobre otras zonas o con otro tipo de estimulación. Y ya que se ha encontrado que es posible, que algunas de las fibras del fascículo solitario, también, estén interrelacionadas con funciones tales como el control reflejo de la respiración e iniciando el reflejo de la tos y el vómito, éste podría ser el paso a seguir en el análisis de estas respuestas.

ANALISIS TEORICO

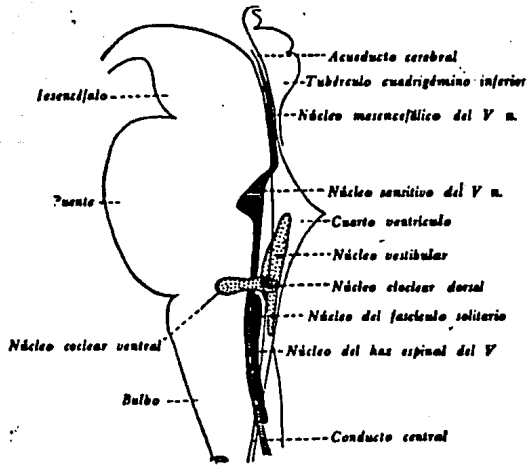


FIGURA 2.8.- Núcleos sensitivos proyectados sobre un corte sagital medio del tronco cerebral humano. En él se observan el fascículo solitario y su entrada al Bulbo Olfatorio.

ANALISIS TEORICO

2.5 ESTUDIOS SOBRE HISTOGRAMAS DE FRECUENCIA E INTERVALOS

El análisis de los trenes de espigas difiere de los métodos clásicos de electrofisiología debido a que no se realizan mediciones de voltajes sino de ocurrencia de eventos en el tiempo. Se toma como base de trabajo que hay una gran cantidad de información sobre la estructura y funcionamiento del sistema nervioso que puede ser obtenida del estudio cuidadoso de la aparición de espigas en el tiempo (5).

Estos análisis pueden ser usados para caracterizar los patrones de disparo de salida de las neuronas y de ahí llevar a una consecuente clasificación de las mismas; dar más información concerniente a los mecanismos neuronales y, tal vez más importante aún, para obtener interrelaciones funcionales entre células.

La información eléctricamente codificada en cada neurona se transmite en múltiples líneas formadas por sucesivas neuronas conectadas que conservan su autonomía espacial. En estas líneas la información posee significados diferentes por transcurrir y terminar en distintos lugares. Por otra parte, en cada neurona constituyente de una línea, la información se transmite utilizando códigos temporales. Las espigas constituyen la onda portadora de la información que está contenida en la magnitud de los intervalos que las separan o en las secuencias de los mismos. Existe una gradación continua de intervalos, es decir, pueden adquirir cualquier valor exceptuando los intervalos más breves que 0.5-1.0 milisegundos. Esta particularidad se debe al período refractario que sucede inmediatamente a la aparición de una espiga, durante el cual la excitabilidad disminuye considerablemente y, por tanto, la probabilidad de generación de una nueva espiga es muy baja o nula inmediatamente después de la primera.

Un método relativamente simple para apreciar las características de la descarga de una neurona es el estudio de la distribución de los intervalos interespigas de primer orden. Se denomina así al tiempo que separa una espiga de la que le sucede o antecede inmediatamente. El histograma de intervalos de primer orden indica la probabilidad de ocurrencia de los diferentes intervalos de clase, mas no da información de la evolución de los intervalos en función del tiempo.

(5) "Es en la comparación de trenes de espigas neuronales, donde está la gran promesa de estas técnicas de obtener hallazgos" Gerstein, Perkel y Moore, 1956

ANALISIS TEORICO

Mientras el histograma de intervalos de primer orden mide la distribución real de intervalos en el tren como ocurrieron, la función de densidad es una abstracción correspondiente al modelo matemático, es por esto que uno puede adaptarlo para describir este proceso.

Para construir los histogramas de intervalos de primer orden, el rango de intervalos observados se divide en bins de igual longitud (δ), si el i -ésimo intervalo interespiga observado (T), satisface la desigualdad :

$$(J - 1)\delta < T \leq (J)\delta$$

entonces el intervalo es colocado en el bin J -ésimo del histograma. Estos bins son numerados entonces : 1,2,...,J. Este procedimiento es para formar los intervalos de primer orden, sin embargo existen histogramas de intervalos de orden n , los cuales se forman como lo indica la figura 2.9. En este estudio solo se utilizaron los intervalos de primer orden debido a la razón expuesta en párrafos anteriores.

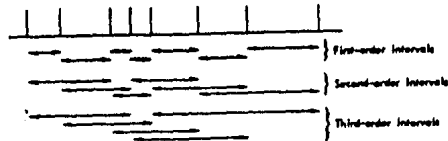


FIGURA 2.9.- Cálculo de intervalos de diversos ordenes para formar Histogramas de Intervalos.

El concepto de codificación de intervalos relativos de Sherry y Marcsynski, (Klema y Sherry, 1982) sostiene que la duración relativa de los intervalos de potenciales de acción puede ser una forma a través de la cual la información es transportada en el sistema nervioso. Estos puntos de vista generalmente no son aceptados, y la literatura se encuentra dominada por el concepto de que las espigas son procesos puntuales generados aleatoriamente, siendo la información transportada solamente a través de la frecuencia de descarga de la neurona. De aquí se generan los análisis a través de histogramas de frecuencias.

ANALISIS TEORICO

Los histogramas de Frecuencia han sido los mas documentados por la literatura existente. La forma de generar estos histogramas es la siguiente: Se escoge un intervalo de tiempo en el cual se quiere medir la cantidad de eventos presentes, a este intervalo se le denomina bin. Dicho bin se distribuye a lo largo de la línea del tiempo en forma secuencial (generando un conjunto de bins), de esta manera se cuenta con un lapso de tiempo (dividido en bins) en donde se observan las frecuencias de el evento en cada bin. Algunos autores tales como, Ferkel, Gerstein y Moore (1967), fueron los pioneros en el uso y Justificación de este tipo de histogramas.

La piedra angular de la que se parte, en el uso de los histogramas de intervalos, es un trabajo desarrollado por Wiersma y colaboradores (McGregor y Lewis, 1977) quienes mostraron que en preparaciones neuromusculares de crustáceos, cambios en el patrón de estimulación nerviosa llevaban a cambios importantes en las contracciones. El siguiente trabajo base parece ser el de Segundo y colaboradores (Segundo et. al. 1966), quienes usaron registros intracelulares en ganglios de Aplysia para demostrar que las respuestas de potenciales postsinápticos excitatorios (EPSP) y los potenciales postsinápticos inhibitorios (IPSP) a entradas sinápticas eran afectadas por el patrón de entrada de los intervalos de espigas de entrada, independientemente de las razones de disparos de entrada. Strumwasser y Wiersma (1967), también demostraron que los patrones de una neurona están con frecuencia íntimamente ligados a distintos patrones de intervalos en otras neuronas, y si las neuronas eran bilateralmente simétricas, cada una contenía el mismo patrón de intervalos relativo.

Mientras muchos investigadores creen que la frecuencia de las espigas es determinante para el entendimiento del funcionamiento neuronal, es evidente que los patrones de intervalos interespidas se ven afectados por la descarga de potenciales de acción neural en otros casos. Esto lleva a pensar que ambos patrones de frecuencia e intervalos pudieran gobernar las respuestas neuronales temporales, y por ende la codificación de información transmitida a través del sistema nervioso.

ANALISIS TEORICO

BIBLIOGRAFIA

- 1.- Aidley, D.J.
The Physiology of excitable cells.
Cambridge University Press
Arrowsmith Ltd., Bristol, England 1978.
- 2.- Cervantes Pérez Francisco, Moreno Gómez Antonio
y M. Trujillo Héctor
Tesis Profesional.
Simulación de Procesos Plásticos en la Sinapsis.
Facultad de Ingeniería de la UNAM., (1978).
- 3.- E. García Austi y M. Rufo (Jr.)
El Cerebro. "Ritmos eléctricos del cerebro e
Integración sensoriomotriz.
Libros de Investigación y Ciencia. Scientific American
Editorial Labor S.A. Barcelona, (1980).
- 4.- F. Stevens Charles
El Cerebro. "La Neurona"
Libros de Investigación y Ciencia. Scientific American
Editorial Labor S.A. Barcelona (1980).
- 5.- F.C. Barone, H.J. Wayner, S.L. Scharoun & R. Guevara-Aguilar
Afferent connections to the lateral Hypothalamus :
A horseradish Peroxidase study in the rat.
Brain Research Bulletin Vol.7 pp75-88, (1981).
- 6.- F.M.C. Crick
El Cerebro. "Reflexiones en torno al cerebro"
Libros de Investigación y Ciencia. Scientific American
Editorial Labor S.A. Barcelona, (1980).
- 7.- F. Ganong William
Fisiología Médica
Ed. El Manual Moderno S.A. de C.V., (1984).
- 8.- F. Schmidt Robert
Fundamentals of Neurophysiology
Springer-Verlag, (1970).

ANALISIS TEORICO

- 9.- Garcia Díaz David Erasmo, Rosalinda Guevara-Asuilar y Colab.
Vagus Nerve stimulation modifies the electrical activity of the
Olfactory Bulb
Brain Research Bulletin, Vol 12, (1984).
Ankho International Inc.
- 10.-Glaser, Edmund M.
Principles of Neurobiological Signal Analysis
Academic Press, Inc., (1976).
- 11.-Gordon M. Shepherd
The Synaptic organization of the Brain
Oxford University Press, (1977).
- 12.-H.U. Asuilar-Baturoni, R. Guevara-Asuilar, H. Aréchida &
Alcocer-Cuarón
Hypothalamic influences on the electrical activity of the
Olfactory Pathway
Brain Research Bulletin, Vol.1 pp263-272
Ankho International Inc., (1976).
- 13.-Klemm W.R. & C.J. Sherry.
Do Neurons Process Information by Relative Intervals in
Spike Trains ?
Neuroscience Biobehavioral Reviews, Vol 6 pp429-437
Ankho International Inc., (1982).
- 14.-Lara y Zavala Rolando
La Cibernetica del Cerebro
En prensa (1985).
- 15.-Leslie L. Iversen
El Cerebro. "Química del Cerebro"
Libros de Investigación y Ciencia. Scientific American
Editorial Labor S.A. Barcelona, (1980).
- 16.-McGregor R., & Lewis E.
Statistical Analysis of Neuronal Spike Trains
Neuronal Modeling Part III Capitulo 10. (1977).

ANALISIS TEORICO

- 17.-Peikel Donald H., Gerstein Course L. & Moore George F.
Neuronal Spike Trains - Stochastic Point Processes
The single spike train.
Biophysical Journal Vol 7. pp391-418, (1967).
- 18.-Walle J.H. Nauta y Michael Feirtes
El Cerebro. "Organización del Cerebro"
Libros de Investigación y Ciencia. Scientific American
Editorial Labor S.A. Barcelona, (1980).
- 19.-Walter Ranson Stephen & Lillard Clark Sam
Anatomía del Sistema Nervioso.
Ed. Interamericana, (1963).

CAPITULO 3

ENTRADA AL SISTEMA

El sistema de análisis por computadora de la actividad eléctrica unitaria extracelular está constituido por dos bloques esencialmente, uno de los cuales constituye la interfaz entre la preparación biológica y el SACAEUE, y el otro el software desarrollado. La figura 3.1 muestra el diagrama a bloques del sistema.

ENTRADA AL SISTEMA

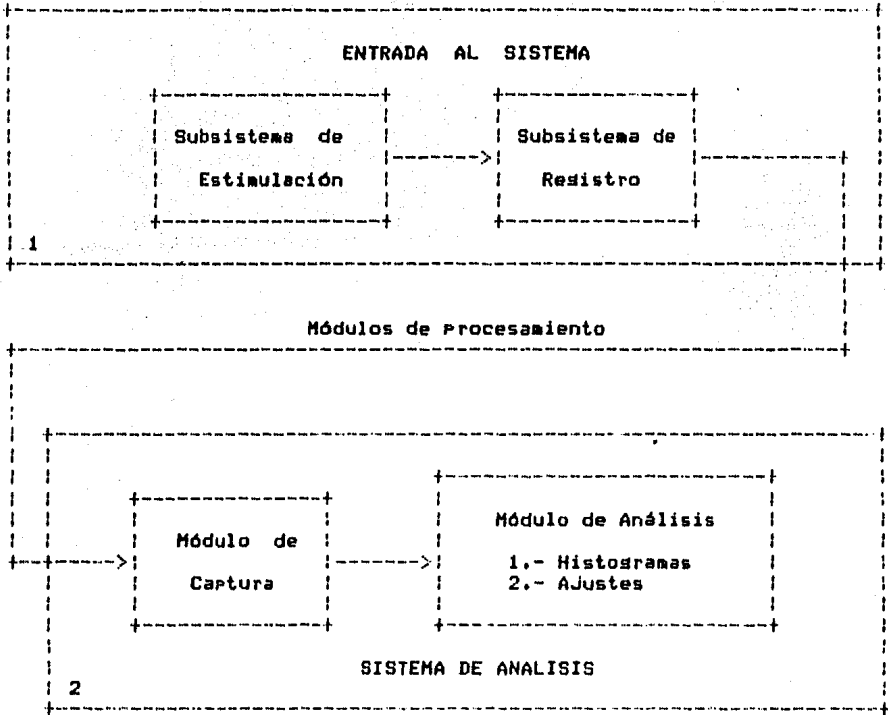


FIGURA 3.1.- Diagrama a bloques del sistema de análisis por computadora de la actividad eléctrica unitaria extracelular.

ENTRADA AL SISTEMA

La entrada al Sistema se realiza en varias etapas, durante las cuales los datos que procesará este sistema son obtenidos. Estas etapas se muestran en los diagramas de las figuras 3.2 y 3.3. El primero de ellos muestra la parte más primitiva del mismo compuesta por el subsistema de estimulación. Este subsistema está compuesto por el aparato de estimulación que se vaya a utilizar para hacer el registro de actividad evocada por estimulación, el cual puede ser eléctrico, odorífero (olores penetrantes que se le presentan al animal), gástrico (distensiones gástricas), térmico, etc. Sin embargo, cabe aclarar, que este subsistema no es indispensable, debido a que solo se requiere cuando la actividad a registrar se desea que sea provocada por algún estímulo externo. Pero como la mayoría de las neuronas del sistema nervioso no necesitan de estimulación externa para responder, esto es, tienen una actividad espontánea, sin estimulación se puede registrar esta actividad espontánea. El hecho de contar con un subsistema de estimulación o no, se presenta entonces como opcional, y depende más que nada de las necesidades del investigador al realizar su experimentación. El contar con un subsistema de estimulación permite el establecimiento de correlaciones entre la región estimulada y la conducta (respuesta) registrada en otra región.

La figura 3.2 muestra también uno de los objetivos fundamentales de este estudio, esto es, el registrar para su análisis, la propagación de cambios que se da entre dos áreas distintas del sistema nervioso, ya sea por estimulación o por actividad espontánea, esto es, tratar de definir la interrelación entre dos áreas distintas del sistema nervioso.

ENTRADA AL SISTEMA

Esa propagación de cambios equivaldría a la transmisión de información entre dos áreas distintas desde que se presenta un evento en una de ellas y hasta que se observa respuesta en la otra (área receptora). A esta transmisión de la información o propagación de cambios también se le conoce con el nombre formal de modulación de la información.

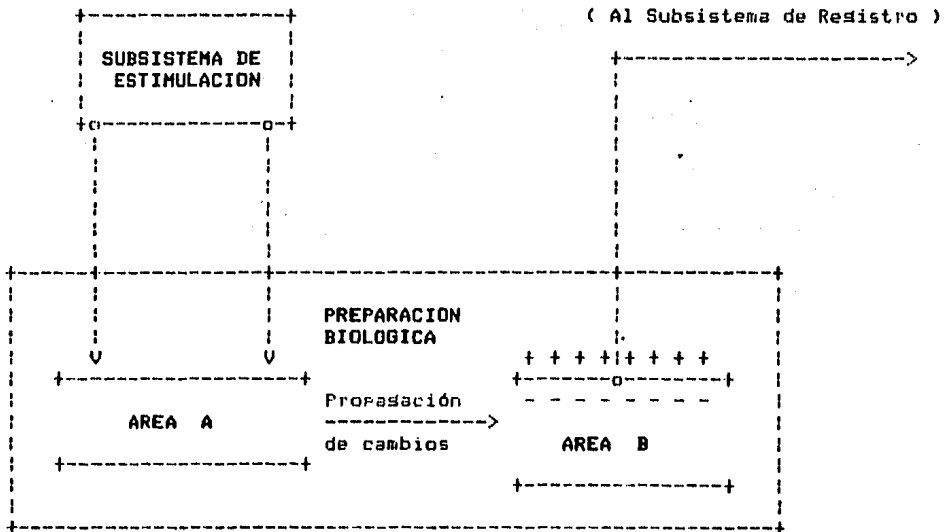


FIGURA 3.2.- Entrada al Sistema de Análisis Interfaz con Subsistema de Registro.

ENTRADA AL SISTEMA

El registro de esta propagación de cambios se hace a través de un electrodo de registro (micropipeta de vidrio especialmente diseñada con ese propósito), el cual a su vez constituye la entrada al subsistema de registro. La figura 3.3 muestra un diagrama a bloques de este subsistema, en la que se pueden observar cuatro módulos; el primero es un acoplador de impedancias, encargado de realizar el acople entre los instrumentos utilizados para monto de la preparación biológica y el registro con el electrodo. El segundo módulo es el sistema de amplificación y filtrado necesario debido a que las señales que se registran en el sistema nervioso son muy pequeñas en amplitud, (del orden de milivolts) de tal manera que tanto para su almacenamiento como para su análisis en computadora es necesario amplificarlas en 100 y hasta 1000 veces. Además algunas veces es deseable realizar un filtrado de frecuencias bajas o altas, dependiendo del parámetro que se requiera analizar.

Como se puede observar en esta figura tanto el módulo 3 como el 4 son opcionales, ya que después de amplificada la señal, ésta puede pasar directamente al sistema de captura (procesamiento en tiempo real). Sin embargo, si se opta por tenerlos, el módulo 3 constituido por el sistema de monitoreo, permite llevar un control sobre el registro, ya que permite un monitoreo de la señal durante todo el tiempo que este dure. Esto es útil debido a que este módulo de monitoreo proporciona un medio visual de la información que se está obteniendo de la preparación biológica, teniéndose así un mayor control en la precisión del registro. Si se contara con estos tres módulos únicamente, la información pasaría a ser capturada también, en tiempo real. Ya con el cuarto módulo, que es el sistema de almacenamiento analógico, la información puede almacenarse en tiempo real, para ser procesada posteriormente, esto es útil debido principalmente a que en ocasiones se desea tener registros continuos tan grandes que no es posible realizar su análisis en tiempo real debido principalmente a las restricciones en cuanto a capacidad de memoria del sistema de cómputo empleado (sólo 64K-bytes para la Apple IIe).

Una vez obtenida la información del registro, ésta pasa al sistema de análisis para ser procesada y analizada a través de los módulos de Software que constituyen la etapa de análisis por computadora, los cuáles se describen en el siguiente capítulo.

ENTRADA AL SISTEMA

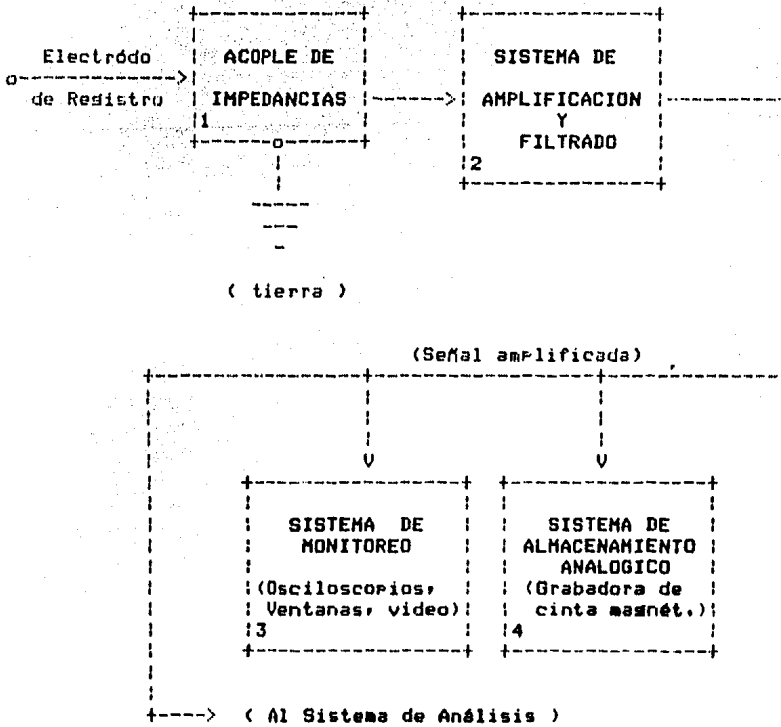


FIGURA 3.3.- Diagrama a bloques del Subsistema de Registro.

ENTRADA AL SISTEMA

Para obtener los registros de actividad unitaria, se pueden seguir varios métodos y aplicarse sobre diferentes preparaciones biológicas. El sistema diseñado en este Proyecto se utilizó para analizar registros fisiológicos obtenidos en ratas, y el método que se empleó para el registro de las espigas se explica en la siguiente sección.

3.1 METODO DE REGISTRO

Se hicieron registros extracelulares, lo que permitió una mayor fidelidad para el registro de las espigas (ver NOTA 3-1).

NOTA 3-1

El registro extracelular muestra las características impulsivas de la actividad eléctrica unitaria o potencial de acción neural, y se realiza en la superficie externa de la neurona (membrana extracelular). Mientras que el registro intracelular se realiza en la membrana intracelular y muestra la forma de onda del potencial de acción desde su génesis.

Los experimentos se realizaron en ratas Wistar, machos, de 200 a 250 gramos de peso corporal, anestesiadas con Hidrato de Cloral a dosis de 400 mg/Kg. de peso. Bajo anestesia profunda se realizó traqueostomía y se colocó una cánula endotraqueal; a través del esófago se introdujo una sonda en el estómago. Posteriormente, la cabeza del animal fué sujeta en un aparato estereotáxico y se procedió a realizar un legrado perióstico (6) dejando al descubierto las suturas craneanas. El cráneo colocado horizontalmente, fué gradado; se retiraron las meninges para evitar al máximo el dolor a la rata y se expuso la corteza cerebelosa. Enseguida, se introdujo el electrodo de registro hasta una profundidad de 8 milímetros desde la superficie. El electrodo de registro consistió en una microspuela de vidrio de 1.5 micras de diámetro en la punta, que se llenó con acetato de sodio y azul de pontamina. La resistencia del electrodo que se midió fué de alrededor de 7 a 10 Megaohms. Este electrodo se conectó mediante un alambre de plata clorada a una sonda de alta impedancia; posteriormente a un acoplador de impedancias y enseguida a un amplificador. La señal amplificada se observaba en un osciloscopio de rayos catódicos y se almacenaba en cintas magnéticas mediante una grabadora Hewlett Packard modelo 3764A.

(6) Apéndice C, Glosario de términos médicos.

ENTRADA AL SISTEMA

La actividad eléctrica unitaria extracelular registrada de esta manera, fué explorada descendiendo el electródo verticalmente en el núcleo del Tracto solitario derecho. Se utilizó un sistema de estimulación, a través de infusiones de agua para realizar una distensión gástrica. Se probaron varias neuronas y en cada neurona registrada se probó el efecto de la distensión gástrica, haciendo infusiones de distintos volúmenes de agua destilada a temperatura ambiente en la sonda intragástrica colocada previamente. El llenado gástrico se controló mediante una bomba de infusión, manteniendo el flujo constante a un mililitro por cada 5 segundos. Sólo cuando el cambio en la actividad unitaria se repetía al menos en tres ocasiones consecutivas debido a la estimulación, el efecto se consideraba positivo. Continuamente fué monitoreada la frecuencia cardíaca y la temperatura rectal de la rata la cual se mantuvo constante a 37 ± 1 grados centígrados.

Al finalizar cada experimento, se depositó azul de pontamina en el sitio registrado, por aplicación de corriente negativa (7 microamperes) durante 10 minutos. Enseguida el animal fué sacrificado con una sobredosis de anestésico, y perfundido por vía intracardiaca con 250 ml. de solución salina y formol al 10%. Se obtuvieron cortes histológicos de 75 micras, teñidos con rojo neutro, a fin de confirmar el sitio registrado.

3.2 TIPO DE SENAL REGISTRADA

La actividad unitaria extracelular registrada a partir del método descrito observó las siguientes características : (Ver figura 3.4)

Amplitudes del orden de 35 milivolts pico, duración de la espiga aproximadamente de 2 milisegundos (valor típico en registros de actividad unitaria). La relación señal-ruido, que está en función de la precisión al momento del registro, así como de factores externos (medio ambiente) se logró obtener de aproximadamente 4:1 :

$$SNR = \frac{V_{señal}}{V_{ruido}} = \frac{35mV}{0.75mV} = 4$$

ENTRADA AL SISTEMA

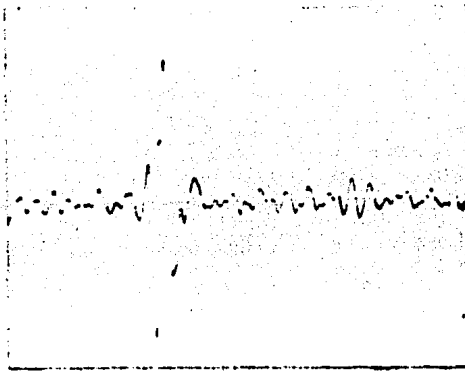


FIGURA 3.4.- Características de una espina real registrada en el núcleo del tracto solitario. Fotografía tomada de la pantalla del osciloscopio. Amplitud = 1 Volt/div. y duración = 5 μ seg/div.

Y por último, la característica más importante en relación a este tipo de señales es la aleatoriedad o la no-estacionaridad notoria en la figura 3.5 que presenta un tren de espigas registrado a partir del núcleo del tracto solitario y con el método de registro descrito previamente. En los capítulos posteriores se describe como esta información es analizada por medio del sistema desarrollado en este trabajo.

ENTRADA AL SISTEMA

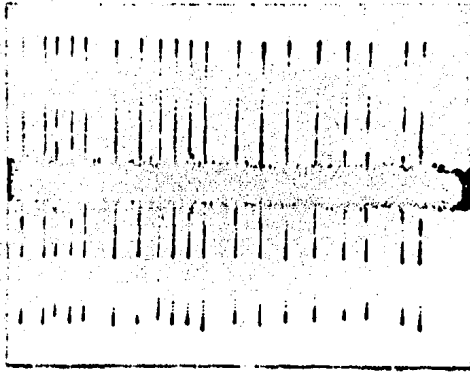


FIGURA 3.5.- Tren de espigas en el núcleo del tracto solitario. Fotografía tomada de la pantalla del osciloscopio.

3.3 EQUIPO ADICIONAL

A continuación se hace una breve descripción del equipo adicional utilizado en la etapa de entrada al sistema.

3.3.1 Amplificador

El Amplificador utilizado fué el P511 de Grass compatible con grabadoras de cinta magnética, computadoras, osciloscopios, etc. Su función es la de amplificar la señal, la mayoría de las veces hasta en 200,000 veces, debido a que, el P511 con una ganancia de 200,000, amplifica 15 microvolts a 3 Volts, que es el tipo de señal requerida por la mayoría de las grabadoras y todas las microcomputadoras.

ENTRADA AL SISTEMA

3.3.2 Sonda De Alta Impedancia Y Acoplador De Impedancias

Se utilizó una sonda de alta impedancia Grass modelo HIP16A y un acoplador de impedancias Grass modelo P16D, para realizar el acople y la selección de la señal en registro al sistema de amplificación.

3.3.3 Grabadora

Se utilizó una grabadora Hewlett Packard modelo 3964A, para obtener la grabación de las señales de actividad unitaria. Esta grabadora cuenta con 4 canales de grabación independientes, de los cuales tres fueron utilizados, uno de ellos para la grabación de la señal registrada, otro para la grabación de voz (experimentos controlados) y el último para grabar un pulso de sincronía que por convención fué de 3 Volts.

3.3.4 Bomba De Infusión

Se utilizó una bomba de infusión Harvard modelo 975, para introducir a la rata el estímulo gástrico, que consistió en una infusión de agua destilada como ya se explicó anteriormente en este mismo capítulo.

3.3.5 Regulador De Temperatura

Se utilizó un regulador de temperatura "Tele-thermometer" de "Yellow Springs Instruments", durante el experimento, para controlar la temperatura corporal de la rata. Esto es necesario debido a que los cortes en la superficie craneana de la rata, provocan que la temperatura corporal de la misma descienda, exponiéndola a ésta a la muerte.

CAPITULO 4

SISTEMA DE ANALISIS

Este capítulo se desarrolla en cinco partes, la primera de las cuales presenta una introducción sobre las características de diseño del sistema, mientras que la segunda presenta a detalle el diseño e implementación del mismo. Una tercera parte se dedica a la descripción de la arquitectura del equipo de cómputo empleado para su implementación actual. En la cuarta se describe la interfaz analógica-digital empleada, y por último, en la quinta parte se discute y evalúa el sistema desarrollado.

SISTEMA DE ANALISIS

4.1 INTRODUCCION.

Debido a las características de diseño del sistema de análisis por computadora de la actividad eléctrica unitaria extracelular (SACAEUE), éste puede implementarse en cualquier sistema de cómputo que cuente con las siguientes facilidades :

1. Al menos un puerto de entrada/salida (serie o paralelo).
2. Una unidad de disco duro, flexible o cinta.
3. Un monitor que permita realizar despliegues gráficos.
4. Un dispositivo analógico para monitoreo a la hora de captura (osciloscopio, detector de picos, etc.). [opcional]
5. Dependiendo del tiempo de captura, memoria RAM para el almacenamiento de la información capturada en línea (para 5' de captura a una frecuencia de 15 hz. alrededor de 13K-bytes de RAM).
6. Un graficador para poder obtener resultados en papel. [opcional]
7. Software para acceder puertos de entrada/salida.
8. Una Grabadora analógica para almacenamiento de la información registrada. [opcional]

Cualquier sistema de cómputo que cuente con esas características algunas de las cuales se ponen como opcionales (dispositivo analógico para monitoreo, y graficador), puede correr este sistema. El diseño se realizó en base a módulos, dos esencialmente : 1) Captura y 2) Análisis, de tal forma que sea transportable y fácil de mantener. Las características esenciales de estos módulos pueden ser observadas en la figura 4.1. El lenguaje en que se implementa tampoco pone restricción, porque en la programación se hace siguiendo los algoritmos que presenta el diseño utilizando programación estructurada.

SISTEMA DE ANALISIS

4.2 ANALISIS EN COMPUTADORA.

El sistema completo se implementó en una microcomputadora Apple IIe, aunque el principio fundamental del que parte y la forma como fué diseñado (modular) lo hacen transportable a cualquier otro sistema de computación. Esto último indispensable, debido a la heterogeneidad de equipo con que cuenta tanto la Universidad como la mayoría de los centros de Investigación dedicados a este tipo de estudios. La implementación de este sistema se basó en dos módulos esencialmente: a) Captura de espigas y b) Análisis.

Un diagrama general de la interacción entre estos módulos se presenta en la figura 4.1, las características generales de los mismos se presentan en la tabla 4.1.

SISTEMA DE ANALISIS

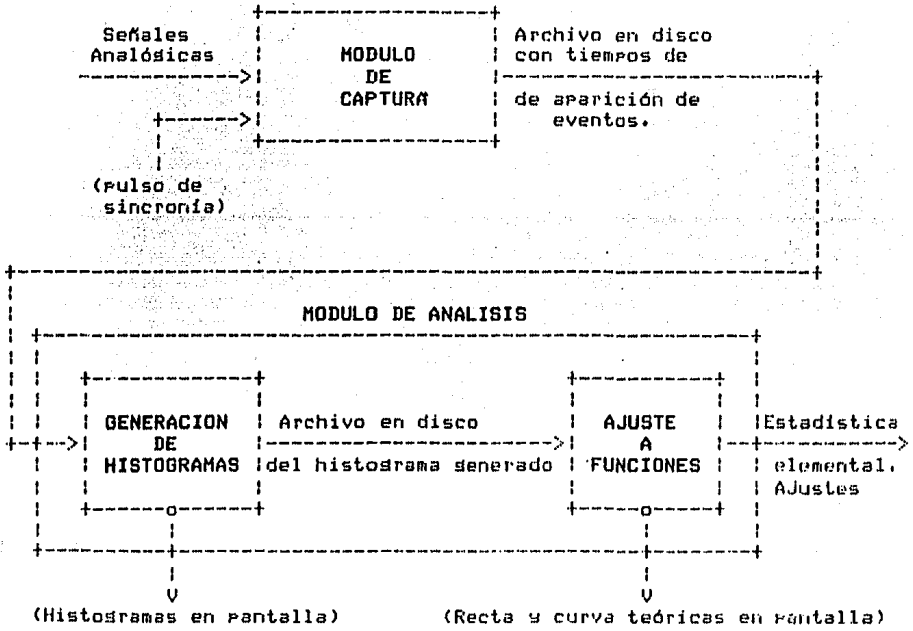


FIGURA 4.1.- Software desarrollado para el Sistema

SISTEMA DE ANALISIS

MÓDULO	CAPTURA	ANÁLISIS
CARACT.	1) Tiempo de muestreo de 41 microseg. 2) Capacidad de almacenamiento de datos = 9Kbytes. (con una frec. de 11Hz, hasta 5') 3) # de canales de entrada = 1	1) Obtiene Histogramas de Frecuencia y de Intervalos (de recalibraciones y permite ampliaciones en zonas del mismo). 2) Obtiene estadística elemental en alguna zona del histograma. 3) Ajusta la recta teórica por el método de mínimos cuadrados. 4) Ajusta la mejor función (curva).

TABLA 4.1.- Características esenciales del Software desarrollado para el Sistema.

4.2.1 Módulo De Captura De Espigas

Dentro de un tren de espigas, la forma de onda de cada una de ellas es, como ya se definió, un evento unitario que lleva información contenida en su tiempo de ocurrencia. Cualquier punto en la forma de onda de las espigas puede ser tomado para la medición del intervalo entre espigas contiguas (Glasser, 1976).

Ahora bien, los voltajes de señal que se registran directamente de la preparación biológica (como ya se vió) son del orden de milivolts, es por esto que para poder ser analizados en la computadora la señal debe ser previamente amplificada. Sin embargo, la señal registrada no es pura, o sea que no está libre de ruido, por lo que al amplificar la señal también se amplifica el ruido (Davenport, 1973). Esta característica permite tener un parámetro suya en cuanto a la frecuencia de aparición de las espigas. Si cada espiga dura alrededor de 5 milisegundos, o sea que en ese lapso no se puede presentar otra espiga, esto significa que la frecuencia máxima a la que éstos pueden ocurrir será: $1/2(5)$ ms. o lo que es lo mismo 1Khz. En un ambiente o situación libre de ruido esto no presenta problema. Empezan las dificultades, sin embargo, cuando aparece el ruido. El problema de ruido se puede evitar de dos formas: 1) Tomando en la computadora

SISTEMA DE ANALISIS

sólo los picos mayores de la señal, aunque esto supone el conocer el tamaño aproximado de la amplitud máxima de la espiga, 2) Contar con un dispositivo que permita, después de la amplificación, la selección cuidadosa de la señal deseada para el análisis. En nuestro caso se utilizó la segunda opción, debido principalmente a que proporciona una mayor precisión en la toma de los datos (evita al máximo el ruido, ver NOTA 4-1).

NOTA 4-1

En ambientes donde no se cuente con este tipo de dispositivos, la señal puede entrar directamente al sistema de conversión, siguiendo los pasos que indica la opción uno antes mencionada.

El dispositivo utilizado fué un Analizador de amplitudes (o ventana), el cual funciona de la siguiente manera :

A la ventana entra una señal ya amplificada que bien puede venir de un medio de almacenamiento análogo (por ej. una cinta magnética) o directamente de la preparación biológica a través de los electrodos de registro (procesamiento en línea). Esta señal pasa a través de un detector de nivel, en donde se seleccionan sólo aquellos eventos que rebasen el umbral establecido de antemano. De esta manera a la salida de la ventana se cuenta con una señal tipo "todo o nada" digital libre de ruido (ver NOTA 4-2). La figura 4.2 muestra el diagrama a bloques para el módulo de captura.

NOTA 4-2

El nivel de umbral para la ventana se eligió de tal manera que evadiera al máximo el ruido de señal, por lo que se tomaron los valores pico de cada evento unitario o espiga.

SISTEMA DE ANALISIS

La ventana que se utilizó cuenta con las siguientes características :

CARACTERISTICAS	VALOR
Rango de voltajes de entrada	1 a 10 Volts (bipolares)
Rango de frecuencias	De 10 Hz. a 1 KHz.
Tipos de salidas :	
Salida tipo Pulsos	Pulsos de 45 Volts TTL y 100 microsegundos de duración.
Salida Multiplex:	Salida combinada de la entrada mezclada con los niveles de ventana.

4.2.1.1 Proceso -

El Proceso para la captura es el siguiente :

1. La información proveniente de un medio de almacenamiento analógico o de un registro en línea se alimenta a la entrada de una ventana para tener a la salida una señal tipo digital.
2. De la ventana se toman dos salidas, una de las cuales va a un sistema de monitoreo (en nuestro caso un osciloscopio) para observar la señal que se desea analizar y además para fijar el nivel de ventana necesario para el corte único de las espigas, esta salida es la salida multiplex. La otra es la salida tipo Pulsos de donde se toman los datos (ocurrencia de espigas).
3. Esta última salida junto con un pulso de sincronía constituyen las entradas a un conversor analógico-digital. El pulso de sincronía por convención, es un voltaje que se lee a través de otro canal del conversor, distinto al de datos de entrada, y que puede provenir ya sea de la misma cinta magnética, de un estimulador eléctrico (aparato usado en fisiología), de una fuente de voltaje, etc., ya que la única restricción es que sea positivo.

SISTEMA DE ANALISIS

4. Finalmente, el conversor analógico/digital es el encargado de pasar la información a la microcomputadora.

Cabe aclarar que cuando se cuente con un analizador de amplitudes con una salida tipo digital el sistema de conversión analógica-digital no es estrictamente necesario. En un ambiente donde no se cuente con este tipo de instrumentos, un sistema de conversión analógico-digital es indispensable, y como la finalidad de este sistema es ser lo mas general posible, es por esta razón que el conversor se incluye como parte de la configuración general para este módulo. De esta forma es el sistema de conversión el encargado de pasar la información a la microcomputadora, así como al sistema de monitoreo para verificar que toda la información esté siendo tomada correctamente por la microcomputadora, para su procesamiento y posterior almacenamiento digital en discos flexibles.

SISTEMA DE ANALISIS

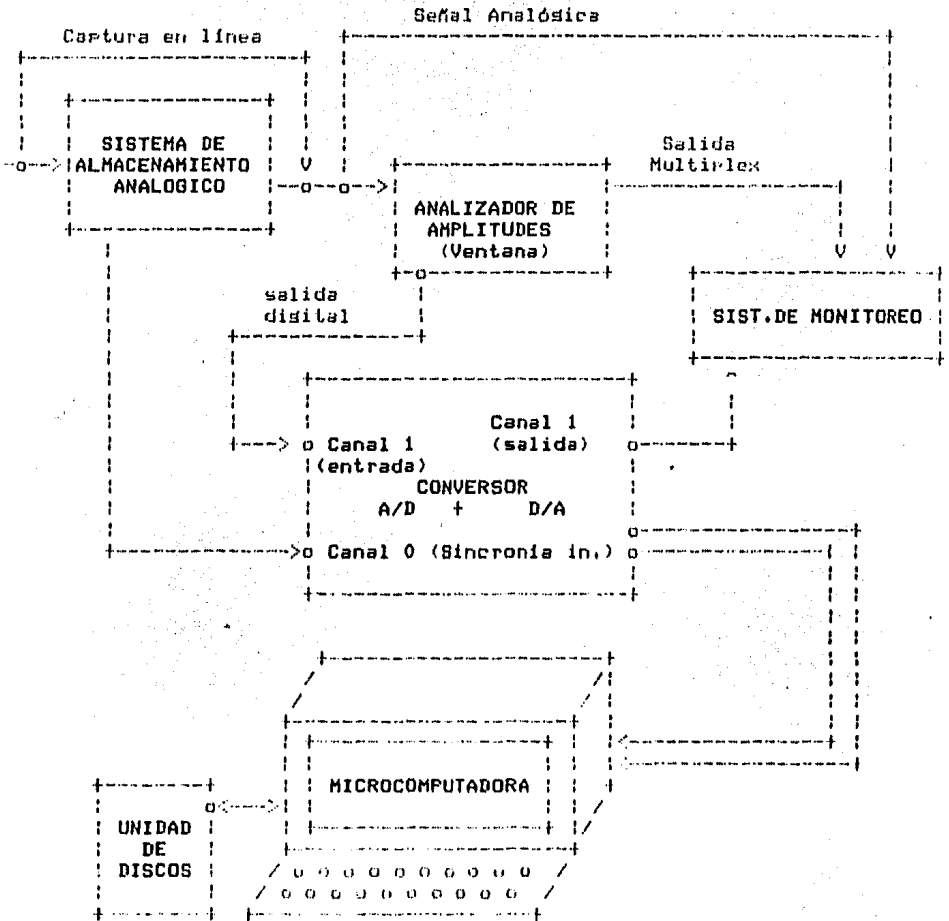


FIGURA 4.2.- Módulo de captura de espigas.

SISTEMA DE ANALISIS

4.2.1.2 Programación -

El programa encargado de la captura se realizó en Pascal (programa principal) y en ensamblador del 6502 (rutina principal de captura), ya que para la toma de las espigas se requería de un tiempo de adquisición más rápido que el que proporciona Pascal, y además porque el acceso al conversor no se puede realizar desde Pascal. La entrada de información sobre el medio ambiente se realiza primero en el programa en Pascal, en donde se pide el número de canales de toma de datos analógicos y el tiempo que durará la captura (ver NOTA 4-3).

NOTA 4-3

Hasta el momento se encuentra implementada la entrada de datos por un solo canal del conversor, sin embargo debido a las características de éste, la entrada se puede ampliar hasta en 15 canales mas haciendo un total de 16 entradas de información.

Estos datos se pasan a la rutina de captura de espigas como parámetros de entrada una vez que han sido ajustados por el programa principal; el ajuste se realiza sobre el tiempo de captura y consiste en particionar este tiempo de tal forma de dejarlo en equivalentes de segundos y minutos para cargarlo en dos bytes de información a la rutina en lenguaje ensamblador. El byte de milisegundos se encuentra cargado en la rutina de captura con un valor fijo de \$FF, 255 ciclos por 41 microses, cada ciclo, dando un tiempo de captura mínimo de 10.455 mses. (ver NOTA 4-4). El ajuste, así como la fórmula utilizada para realizarlo se muestran en la tabla 4.2.

NOTA 4-4

Para el tiempo de duración de la captura se dejó como restricción tiempos menores o iguales a seis minutos de captura continua, debido a restricciones en capacidad de memoria del sistema en que se implementó.

La rutina de captura se encarga de leer el canal de entrada del conversor y si hay espiga (voltaje positivo) se almacena en localidades secuenciales en la página cero a partir de la dirección dada por Pascal al vector de datos de entrada, el que actúa como parámetro de entrada-salida. En las dos primeras localidades de este vector se cargan los dos bytes de la cuenta del tiempo (segundos y minutos) de captura. La lógica que sigue esta rutina, se muestra en la figura 4.3. La velocidad de muestreo fué de 123 microsegundos en

SISTEMA DE ANALISIS

consideración a la frecuencia más alta de aparición de las espigas que es de 1Khz y a que las duraciones de la espiga son muy reducidas (100 microsegundos máximo), por lo que para muestrear estas señales se requiere de un muestreo de 4 o 5 veces la velocidad máxima de aparición (Glasser, 1976). En nuestro caso se muestreó 8 veces la velocidad máxima por lo que:

$$\text{Tiempo de muestreo} = 1/8(1\text{Khz}) = .123 \times 10^{-3} \text{ ses.}$$

Adicionalmente, el microprocesador 6502 cuenta con la técnica de Pipeline, esto significa que si trabaja a una velocidad de reloj de 1.023 Mhz., su velocidad real de procesamiento es equivalente a 2.5 Mhz, por lo que el tiempo real de captura se redujo hasta 41 microsegundos.

$$T_m = T_m/2.5 = 0.04 \text{ mses.}$$

SISTEMA DE ANALISIS

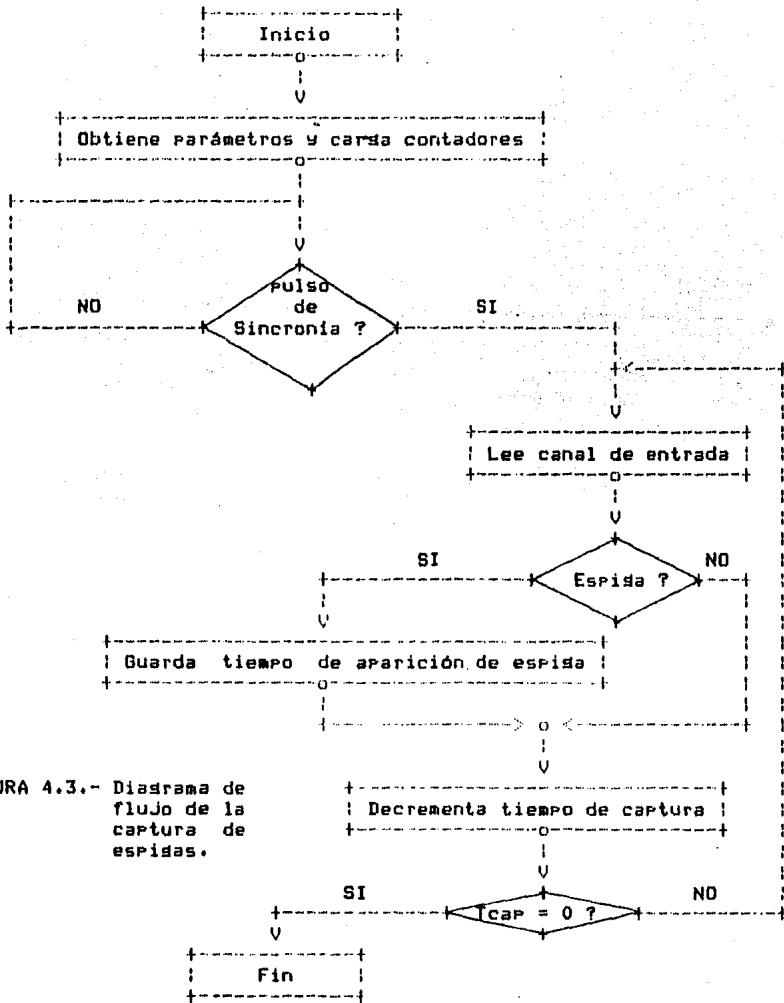


FIGURA 4.3.- Diagrama de flujo de la captura de espigas.

SISTEMA DE ANALISIS

Carga minutos (m)	Carga segundos (s)	Tiempo [ses] real usuario	
0	0	0.01	
0	1	0.029	
0	2	0.030	
0	.	.	Fijo :
0	.	.	
0	15	0.167	Byte de milisegundos = \$FF
0	16	0.177	(10.455 mses.)
0	.	.	
0	.	.	Fórmula para segundos :
0	239 ---> (\$EF)	2.498	(segundos)(1000)
0	.	.	s = -----
0	.	.	255 (.041)
0	255 ---> (\$FF)	2.66	
1	0	5.373	s = round(s) - 1
2	0	7.86	
3	0	10.747	Fórmula para minutos :
.	0	.	(minutos)(1000)
.	0	.	m = -----
16	0	45.677	65535(.041)
17	0	48.364	
.	0	.	
.	0	.	m = round(m) - 1
223 ---> (\$DF)	0	601.873	
.	0	.	
.	0	.	
239 ---> (\$EF)	0	644.864	El 0.041 es la velocidad
.	0	.	real de captura del 6502
.	0	.	o sea .123/3 = .041 [mses]
255 ---> (\$FF)	0	687.855	
.	0	.	
.	0	.	
255 ---> (\$FF)	ff	690.531 ==> 11.5 min.	

TABLA 4.2.- Ajustes al tiempo de captura, para la carga de los dos bytes a la rutina de ensamblador del 6502.

SISTEMA DE ANALISIS

4.2.2 Módulo De Análisis.

Este módulo lo conforman dos Programas que se corren separadamente. El primero de ellos es el encargado de la generación de los histogramas y de su almacenamiento en disco. Su entrada es un archivo en disco conteniendo los tiempos de aparición de los espigas a partir de los cuales se construyen dos tipos de histogramas (a selección por el usuario). Dichos histogramas son: Frecuencia e Intervalos. El segundo Programa es el encargado de realizar ajustes a funciones sobre ciertas zonas del histograma, basados en la experiencia del usuario.

4.2.2.1 Histogramas De Frecuencia -

El histograma de frecuencia se construye a partir de un bin que elige el usuario (7). Una vez elegido el bin, se recorre el archivo secuencialmente, incluyendo en el bin o múltiplos de este, los eventos que caigan dentro del rango de tiempo que abarca el mismo (barra del histograma). La información que presentan entonces estos histogramas es número de espigas aparecidas Vs. tiempo (en mseg.).

4.2.2.2 Histogramas De Intervalo -

La generación de estos histogramas se da a partir de un intervalo base que el usuario elige previamente. Y para generarlos se recorre el archivo secuencialmente, incluyendo dentro del intervalo base o múltiplos de este, aquellos intervalos entre espigas (interespigas) que caigan dentro del rango especificado por el intervalo base. Así la información que presentan estos histogramas será número de intervalos Vs. tamaño del intervalo, esto es, nos indica cuantos intervalos del tamaño base (proporcionado por el usuario) o múltiplos de este ocurrieron durante el tren de espigas analizado (Cramer, 1946).

(7) bin = lapso de tiempo en el que se cuantificará el número de espigas aparecidas.

SISTEMA DE ANALISIS

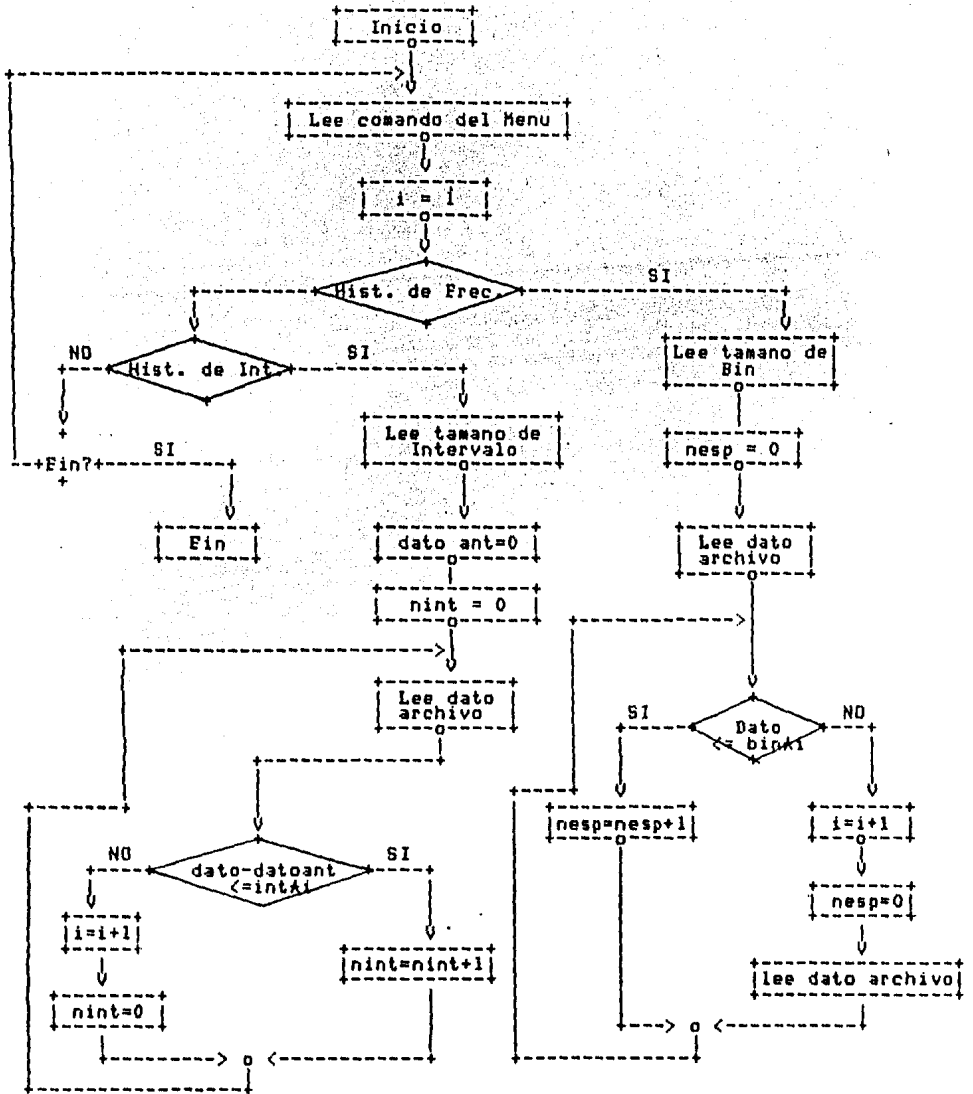
4.2.2.3 Despliegue -

Una vez generados los histogramas, éstos son desplegados en pantalla (monitor) y sobre ellos se puede ampliar alguna zona, presentar en pantalla con barras simples o ahuradas y escalar a algún valor base (ver NOTA 4-5). El histograma mostrado en pantalla se puede almacenar en disco si se desea analizarlo posteriormente (a través de ajustarlo a una función) o si se desea pasarlo a papel utilizando un programa adicional para esto, como complemento al mismo. El diagrama de la figura 4.4 muestra la lógica que sigue este programa. En el capítulo 5 que muestra los resultados del sistema, se muestran diversos histogramas de ambos tipos así como las opciones mencionadas anteriormente.

NOTA 4-5

Restricciones en cuanto a tamaño de la pantalla de graficación, llevó a necesidad de utilizar notación Científica en las divisiones numéricas del eje X que presentan los histogramas.

SISTEMA DE ANALISIS



SISTEMA DE ANALISIS

4.2.2.4 Ajuste A Funciones. -

La información que proveen los histogramas de Frecuencia o Intervalos de espigas proporciona una herramienta útil para el análisis de actividad unitaria pero debe ser complementada, hecho que se logra al utilizar una herramienta adicional en el descifrado de éste fenómeno. Dicha herramienta la constituye por ejemplo, el análisis de las tasas de variación de estos procesos a lo largo de trenes de espigas. El análisis descrito en este capítulo va dirigido a generar evidencia para apoyar el rechazo o aceptación de las hipótesis propuestas en este trabajo (Capítulo 1).

El programa funciona de la siguiente manera :

La entrada la constituye un histograma almacenado en disco el cual, puede ser de cualquiera de los dos tipos generados por el programa anterior (frecuencia o intervalos). El usuario fija los límites del análisis sobre el histograma y sobre ellos se obtienen las siguientes estadísticas básicas : a) Media aritmética, b) Varianza, c) Coeficiente de Variación y d) Moda. Estas estadísticas proporcionan el primer resultado del análisis que se presenta en pantalla a manera de tabla. Por otro lado, éstos son los únicos parámetros que se obtienen basados en la naturaleza aleatoria de estas señales. Y se le proporcionan al usuario para, adicionalmente saber en forma estadística (análisis somero) de que manera variaron los eventos a lo largo del tren de espigas y así apoyar a los siguientes resultados en el ajuste. A partir de aquí se presenta al usuario un menú con 5 diferentes opciones para ajuste en los límites preestablecidos. Los ajustes se harán a criterio del usuario, teniendo como opción los siguientes tipos de ajuste :

- | | |
|--|--|
| 1.- Recta teórica | $Y = mX + b$ |
| 2.- Función Potencial | $Y = aX^m$ |
| 3.- Función Exponencial | $Y = se^{mX}$ |
| 4.- Función Exponencial Saturada
o Berthalanffy | $Y = Y_{max} \left[1 - e^{-m(X-X_0)} \right]$ |
| 5.- Función Logarítmica | $Y = A / [1 + Be^{-AmX}]$ |

SISTEMA DE ANALISIS

donde : $A = N(X_n) =$ Ultimo valor de X

$b =$

$B = a$

$m =$ pendiente

$b =$ ordenada al origen

Durante la generación de las curvas, los parámetros m y b son obtenidos por el método de mínimos cuadrados. El usuario elige un tipo de ajuste a curvas, para el cual se obtienen la recta y en su caso, la curva teórica. Para cada una de ellas se hace un despliegue en pantalla. La recta se obtiene por el método de mínimos cuadrados de donde se obtienen además, los parámetros de pendiente que representa la rapidez de cambio o tasa de variación sobre el área de estudio, la ordenada al origen (b) de la recta y el factor constante " a " de la función ajustada, que servirán posteriormente para poder ajustar la curva teórica que el usuario elija. El procedimiento analítico que se siguió para el ajuste a las funciones propuestas se muestra en las siguientes páginas. Se puede seguir el mismo procedimiento para más de un tren de espigas, esto es, se pueden sumar trenes de espigas y hacer el análisis (ajuste) para esta suma de trenes. El diagrama de la figura 4.5 muestra la lógica que sigue esta parte del módulo para la realización de dicho ajuste. Las siguientes páginas presentan el método de ajuste seguido para cada una de las funciones.

SISTEMA DE ANALISIS

2.- FUNCION POTENCIAL

$$y = ax^m$$

Desarrollo :

$$\text{Ln}y = \text{Ln}a + \text{Ln}x^m$$

$$\text{Ln}y = \text{Ln}a + \text{Ln}x (m)$$

$$\text{Ln}y = m\text{Ln}x + \text{Ln}a$$

$$y = mx + b$$

3.- FUNCION EXPONENCIAL

$$y = ae^{mx}$$

Desarrollo:

$$\text{Ln}y = \text{Ln}a + \text{Ln}e^{mx}$$

$$\text{Ln}y = \text{Ln}a + mx$$

$$\text{Ln}y = mx + \text{Ln}a$$

$$y = mx + b$$

4.- FUNCION EXPONENCIAL SATURADA O BERTHALANFFY

$$y = y_{\max}[1 - Be^{-k(x-x_0)}]$$

Si $x_0 = 0$:

$$y = y_{\max}[1 - Be^{-kx}]$$

Desarrollo:

$$y = y_{\max} - y_{\max}Be^{-kx}$$

$$\text{Ln}[y_{\max}Be^{-kx}] = \text{Ln}y_{\max} - \text{Ln}y$$

$$\text{Ln}[y_{\max}Be^{-kx}] = \text{Ln}[y_{\max} - y]$$

$$\text{Ln}y_{\max} + \text{Ln}Be^{-kx} = \text{Ln}[y_{\max} - y]$$

$$\text{Ln}Be^{-kx} = \text{Ln}[y_{\max} - y] - \text{Ln}y_{\max}$$

$$\text{Ln}B + \text{Ln}e^{-kx} = \text{Ln}[(y_{\max} - y)/y_{\max}]$$

$$\text{Ln}[(y_{\max} - y)/y_{\max}] = \text{Ln}B - kx$$

$$y = b + mx$$

SISTEMA DE ANALISIS

5.- FUNCION LOGARITMICA

$$y = [A/(1 + Be^{-Akx})]$$

Desarrollo:

$$y + yBe^{-Akx} = A$$

$$\text{Ln}y + \text{Ln}[yBe^{-Akx}] = \text{Ln}A$$

$$\text{Ln}[yBe^{-Akx}] = \text{Ln}A - \text{Ln}y$$

$$\text{Ln}[yBe^{-Akx}] = \text{Ln}[A/y]$$

$$\text{Ln}y + \text{Ln}Be^{-Akx} = \text{Ln}[A/y]$$

$$\text{Ln}Be^{-Akx} = \text{Ln}[A/y] - \text{Ln}y$$

$$\text{Ln}B + \text{Ln}e^{-Akx} = \text{Ln}[(A-y)/y]$$

$$\text{Ln}[(A-y)/y] = -Akx + \text{Ln}B$$

$$y = mx + b$$

SISTEMA DE ANALISIS

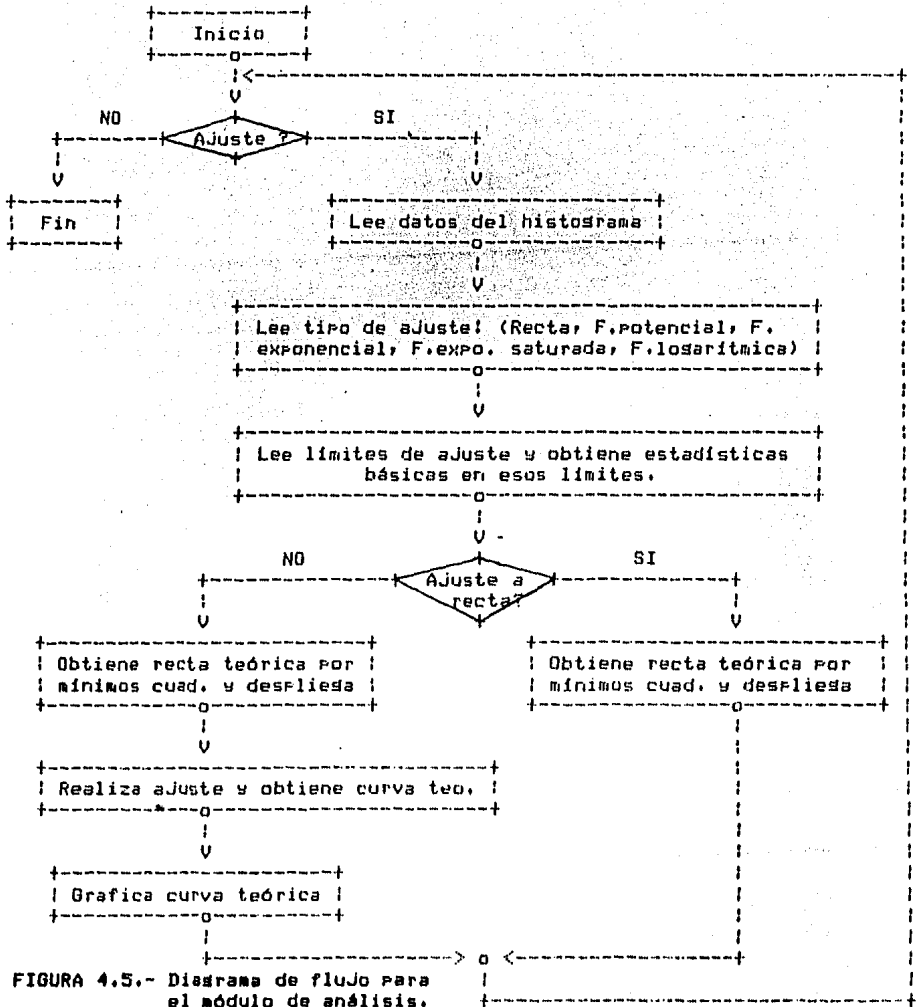


FIGURA 4.5.- Diagrama de flujo para el módulo de análisis.

SISTEMA DE ANALISIS

4.3 IMPLEMENTACION EN UNA APPLE IIE.

4.3.1 Arquitectura Del Microprocesador 6502.

El 6502 es un microprocesador de 8 bits orientado al manejo de memoria. Cuenta con 13 modos de direccionamiento, los cuales le dan herramientas que normalmente solo se encuentran en procesadores grandes. Estas capacidades de direccionamiento se ven complementadas por las altas velocidades de procesamiento de sus instrucciones. Esta rapidez de procesamiento se debe principalmente a que el 6502 se diseñó con la técnica de Pipeline, en la cual el microprocesador toma la siguiente instrucción de un programa antes de terminar de ejecutar la que tiene presente. Puede ejecutar hasta 56 diferentes tipos de instrucciones, sin embargo debido a las variadas combinaciones que sus modos de direccionamiento permiten hacer para tipos individuales de instrucciones, su conjunto de instrucciones se extiende hasta un total de 151 instrucciones ejecutables (MCS6500 Prog. manual, 1976).

El contador del programa es de 16 bits, con lo cual puede direccionar cualquier localidad en un espacio 64K-bytes. Las instrucciones se componen de uno, dos o tres bytes. El tipo específico de microprocesador 6502 que utiliza la Apple Iie como su unidad central de procesamiento (CPU) es el 6502B, que tiene una velocidad de procesamiento de 1.023 MHz para realizar hasta 500,000 operaciones de 8 bits cada segundo. Esta velocidad de procesamiento no se debe tomar como criterio de comparación con otros microprocesadores, debido a que el 6502B tiene un ciclo de instrucción más sencilla que la de muchos otros procesadores, ya que usa técnicas de pipeline para un procesamiento más veloz. La velocidad del 6502B con un reloj de 1MHz es equivalente a otros procesadores con velocidades de reloj de hasta 2.5 MHz (ver NOTA 4-6). La velocidad a la que el microprocesador procesa las instrucciones, depende del tipo de instrucción (Reference manual for Iie only, 1982). La figura 4.6 nos muestra un diagrama detallado de la arquitectura del microprocesador 6502.

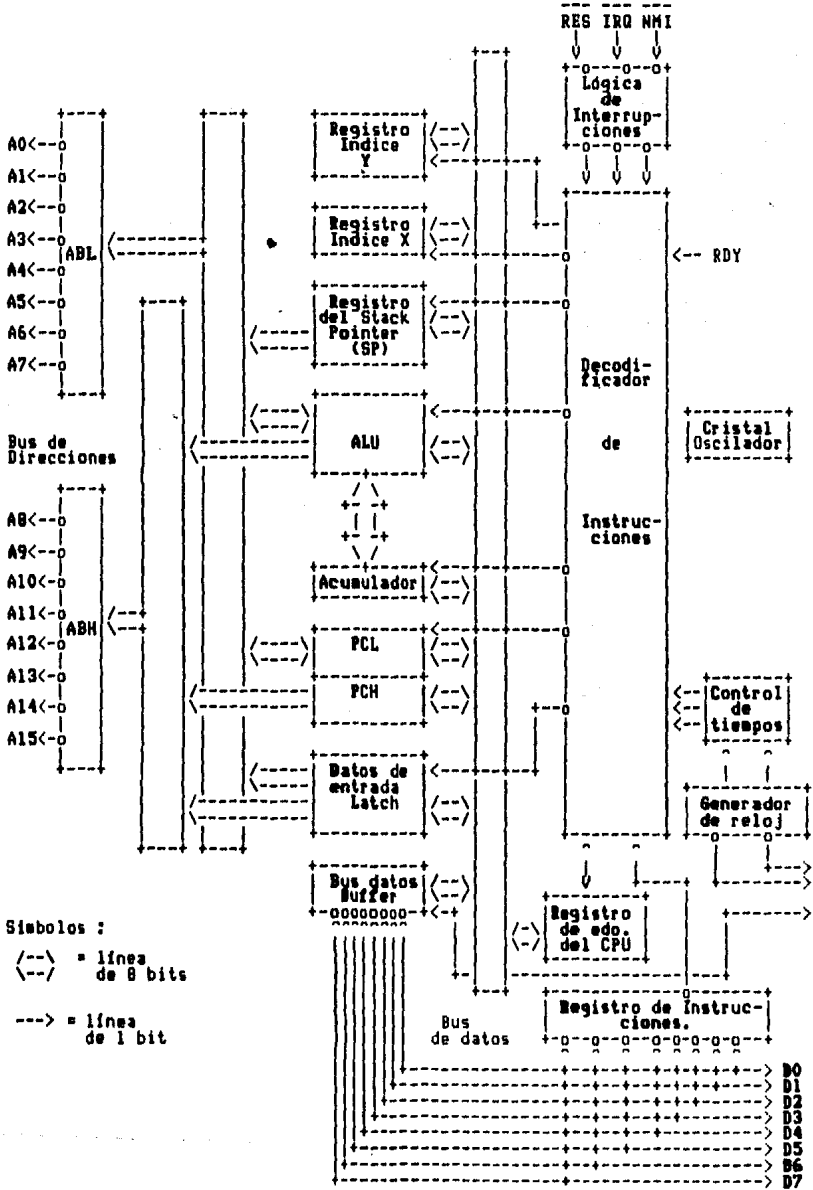
NOTA 4-6

Un punto importante para recordar es que la precisión del ciclo de tiempo del microprocesador, depende de la precisión del cristal que le está proporcionando la base de tiempo. Si el cristal es más rápido o más lento que su valor nominal, los tiempos para estos ciclos se deberán ajustar.

SISTEMA DE ANALISIS

SECCION DE REGISTROS

SECCION DE CONTROL



SISTEMA DE ANALISIS

4.3.2 Registros De Propósito General

El 6502 cuenta con tres registros de propósito general de 8 bits, que el programador puede utilizar para salvar valores temporales de datos, mantener contadores, etc. Estos registros son : acumulador, registro X y registro Y. El acumulador es el único registro en donde se pueden realizar operaciones aritméticas y lógicas, y contiene a uno de los operandos en este tipo de instrucciones. Además el 6502 cuenta con instrucciones para rotación de los contenidos del acumulador a la derecha o la izquierda, etc. Los registros X y Y son mas que nada empleados como registros indices (para acceder datos secuenciales en memoria), pero el hecho de que pueden ser incrementados o decrementados bajo el control de un programa los vuelve muy utilizados como contadores de propósito general.

4.3.3 Registro De Estado Del Procesador

Contiene siete bits utilizables, cinco de los cuales son banderas de estado, que proveen información sobre el resultado de la última instrucción ejecutada. Los otros dos bits utilizables son bits de control. La descripción de las banderas de estado y de control se da en el Apéndice B.

4.3.4 Stack

El microprocesador 6502 utiliza la página 1 de memoria para manejar su Stack, aquí se almacenan las direcciones de regreso de subrutinas, etc., en una secuencia de First-in Last-out. Los programas de usuarios también pueden hacer uso del stack para almacenamiento temporal de los registros a través de las instrucciones que para ello proporciona también este microprocesador (Leo Scanlon, 1980).

El apuntador al stack ("Stack Pointer") consta de 8 bits, por lo que el stack puede contener solo 256 bytes de información a la vez. La tabla 4.3 nos muestra un resumen de las características principales del microprocesador utilizado en la implementación actual de este sistema.

SISTEMA DE ANALISIS

Tipo	6502E
Registros utilizados	Acumulador (A) Registros índices (X,Y) Stack Pointer (S) Estado del Procesador (P)
Tamaño de Registro	8 bits
Bus de Datos	8 bits de ancho
Bus de Direcciones	16 bits de ancho
Rango de Direcciones	65,536 (64K)
Voltaje de operación	+ 5 Volts (+/- 5%)
Potencia que disipa	500 mW. (típico)

TABLA 4.3.- Características Primordiales
del microprocesador 6502.

4.3.5 Organización De La Memoria

El 6502 puede acceder hasta 65,536 localidades de memoria (64K-bytes). Todas las localidades programables (memoria RAM), de lectura únicamente (ROM) y para dispositivos de entrada/salida están contenidas en este espacio de direcciones. La memoria de la A-1 se encuentra dividida (lógicamente) en bloques de 256 bytes llamados páginas. Por ejemplo, la página cero está compuesta por las localidades de memoria de la 0 a la 255 (\$0 a \$FF en hexadecimal) inclusive. La página 1 consiste de las localidades de 256 a la 511 (\$100 a \$1FF -nótese que el número de la página lo constituye el byte de orden alto de la dirección en hexadecimal-) y así sucesivamente. El espacio de direcciones del mapa de memoria se muestra en la figura 4.7, en la cual también se observan las diferentes áreas que lo componen.

SISTEMA DE ANALISIS

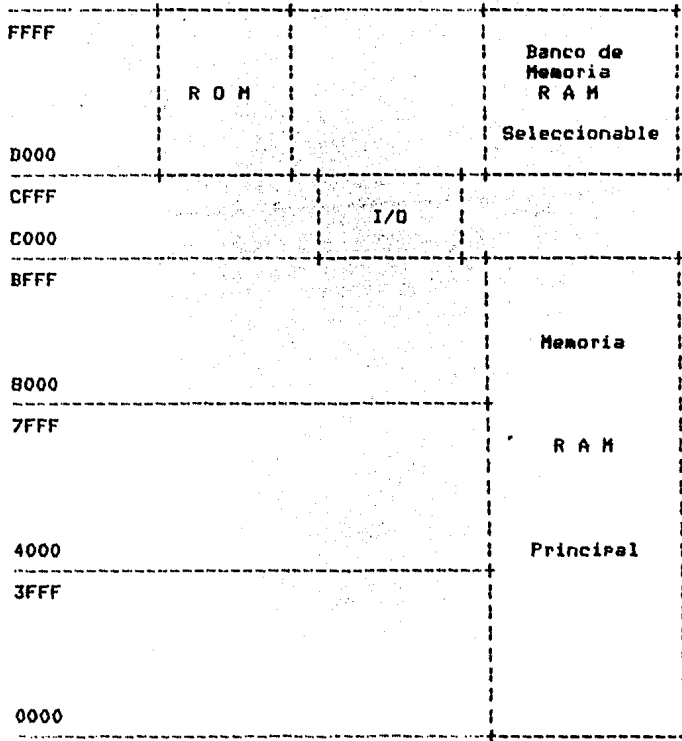


FIGURA 4.7.- Mapa de memoria del 6502.

Como se muestra en la figura 4.7 la mayor parte del espacio de memoria de la Apple IIe es para memoria de lectura y escritura. La memoria RAM principal se extiende de la localidad 0 a la 49151 (\$FFFF hexadecimal) y ocupa de la página 0 a la 191. Hay además un banco de memoria seleccionable de la dirección 53248 a la 65535 (\$D000 a \$FFFF en hexadecimal), que se utiliza cuando se cuenta con una tarjeta de 80

SISTEMA DE ANALISIS

columnas para expansión de memoria (Reference manual for IIE only, 1982).

Como se puede observar en la figura 4.8 este conjunto de direcciones de la (52K a la 64K) puede utilizarse de dos formas, para direccionar memoria ROM y la otra para direccionar memoria RAM. Los 12K-bytes de ROM direccionables por este conjunto de direcciones contienen al Monitor y al intérprete de Applesoft Basic. Alternativamente, se tienen 16K-bytes de RAM con las mismas direcciones, la cual es usada normalmente para el almacenamiento ya sea del intérprete de "Basic Inteser" o bien una parte del sistema operativo Pascal. El porqué de esta distribución de memoria es histórica, la Apple IIE es capaz de correr software de sus Predecesoras Apple II y Apple II Plus porque usa esta parte de la memoria en la misma forma que éstas lo hacen. Por esta razón le conviene tener el intérprete de Applesoft en ROM, pero también es capaz de usar este conjunto de direcciones para otras cosas cuando Applesoft no está siendo utilizado. El mapeo de 12K a 16K se hace a través de espejos de memoria.

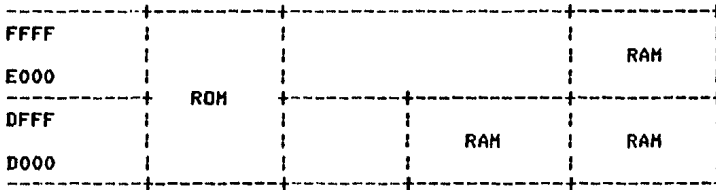


FIGURA 4.8.- Banco de memoria RAM seleccionable

4.3.5.1 Espacio De Memoria Para Periféricos --

Debido a que el microprocesador 6502 realiza toda su entrada/salida a través de localidades de memoria, parte de la memoria de la Apple IIE ha sido destinada para el uso exclusivo de tarjetas de expansión que se colocan en diferentes ranuras ("slots"). Son siete las ranuras de expansión con que cuenta la Apple IIE para la conexión de tarjetas que contengan el "Hardware" y/o "Firmware" necesario para realizar la interfaz entre la Apple IIE y dispositivos periféricos. Estas ranuras no son solamente simples puertos de I/O; las tarjetas de periféricos pueden acceder datos, direcciones y líneas de control a través de estas ranuras. Cada ranura de expansión tiene el uso exclusivo de 16 localidades de memoria para datos de entrada y de salida en el espacio de direcciones que empieza en la localidad 1C090.

SISTEMA DE ANALISIS

El mapa de memoria de I/O se muestra en la figura 4.7 y en él se pueden observar algunos espacios de memoria ROM. Estos espacios de memoria son utilizados para contener pequeños programas tales como los manejadores de entrada/salida (manejadores de I/O). Las tarjetas de periféricos que contienen sus propias rutinas de manejo en "Firmware" como éstas suelen llamarse también periféricos inteligentes. Cada tarjeta de periféricos tiene destinada una página de memoria, esto es, 256 bytes, la cual normalmente es utilizada para memoria ROM en la tarjeta con programas manejadores que controlan la operación del dispositivo periférico conectado a dicha tarjeta. La página de memoria que se destina a cada ranura de expansión comienza en las localidades \$Cn00 (donde n va de 0 a F).

SISTEMA DE ANALISIS

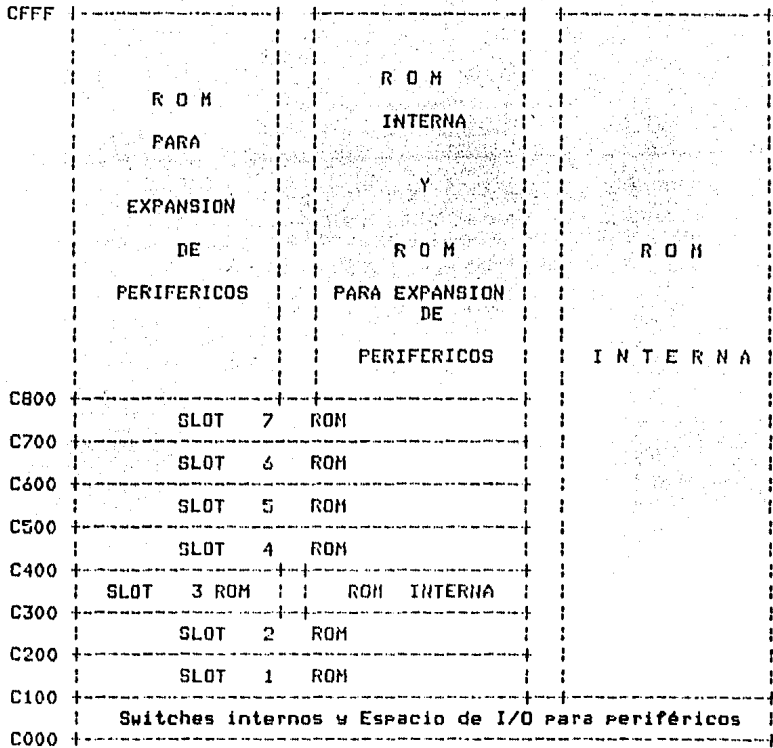


FIGURA 4.9 - Mapa de memoria de I/O

SISTEMA DE ANALISIS

4.4 INTERFAZ ANALOGICA/DIGITAL

Se utilizó un conversor Analógico Digital y Digital Analógico (A/D + D/A Mountain 11-00230-03) diseñado para operar en microcomputadoras Apple II, Apple II Plus y Apple IIe. Para su uso no se requiere de software extra al que ya contiene la Apple IIe, pero si se necesita por lo menos un disk drive y 48K de memoria.

4.4.1 Hardware

La tarjeta de A/D y D/A utiliza un conversor digital analógico de 8 bits (DAC) para las salidas analógicas y un registro de aproximaciones sucesivas de 8 bits (SAR) para la conversión analógica digital. La programación de los canales del DAC, el comienzo de las conversiones A/D, y la lectura de los canales del ADC van acompañados por la lectura o la escritura a alguna de las 16 direcciones dependientes de los slots. La tabla 4.4 nos muestra las características más importantes del A/D + D/A.

SISTEMA DE ANALISIS

Convertor	Características	Valor
D A C	Voltajes de salida analógicos	-5 a +5 V.
	Rango de dígitos de conversión	0 a 255
	Presición Absoluta	+/- 3% FSR
	Presición Relativa	+/- 1 LSB
	'Slew rate' a la salida	10 V/ms.
	Tiempo de conversión por canal	16 microses.
	Tipo de salidas	Monotónicas
	# de canales de entrada	16
A D C	# de canales de salida	16
	Impedancia de salida dinámica	10 ohms
	Rango de voltajes de entrada	-5 a +5 V.
	Resolución	8 bits
	Metodo de conversión	APROX. Suc.
	Tiempo de conversión por canal	9 microses.
	Presición Absoluta	+/- 3% FSR
	Presición Relativa	+/- 1 LSB
# de canales de entrada	16	
# de canales de salida	16	
Impedancia de entrada Para AC	1K ohm	
Impedancia de entrada Para DC	1Mes.ohm	

TABLA 4.4.- Características esenciales del convertor A/D + D/A.

4.4.2 Lectura Y Escritura A Un Canal Del A/D + D/A

Se toma o envía información a la tableta del convertor a través de comandos de software. La tableta puede ser controlada desde cualesquiera de los lenguajes que utiliza la microcomputadora Apple IIe. Para la lectura y la escritura de cada canal, ésta se realiza haciendo referencia a las direcciones específicas de cada canal, las cuales dependen (como ya se mencionó anteriormente) de la colocación de la tableta en una ranura específica y del canal a referenciar.

Al realizar la toma de datos de un canal, se debe leer dos veces el mismo canal para obtener una lectura confiable de esa dirección. Esto se debe a que durante la primer lectura se obtiene el valor de la conversión anterior y empieza el ciclo de conversión de la lectura deseada, por lo cual el valor deseado se obtiene hasta la segunda lectura (A/D + D/A Mountain Deer, Manual, 1982).

SISTEMA DE ANALISIS

Para determinar la dirección de cada canal del conversor se aplica la siguiente fórmula :

$$\text{Direc} = 49280 + (\# \text{ranura} * 16) + \text{canal}$$

en donde :

- 1.- 49280 (\$C080 en hexadecimal) es la dirección inicial de todas las localidades dependientes de las ranuras.
- 2.- El # de ranura debe ser un entero entre 1 y 7.
- 3.- El # de canal debe ser un entero entre 0 y 15.

Debido a la naturaleza del hardware con que cuenta el DAC, si se escribe un valor digital al mismo canal mas de una vez en 16 microsegundos, la salida de alguno de los otros canales estará sujeta a un posible traslape ("droop") de la información. Este efecto de "traspase" no ocurre si se está haciendo uso de pocos canales. Sin embargo, si están utilizándose la mayoría de ellos, el efecto se puede eliminar evitando el leer alguno de los canales mas rápido que 16 microsegundos. En el apéndice B se proporcionan algunas tablas útiles para la comprensión del funcionamiento de este A/D + D/A. La interfaz entre la tableta del conversor, la microcomputadora Apple IIe y la aplicación se realiza a través de un par de cables, cada uno de los cuales se conecta a la tableta del conversor. En la parte superior de la tableta hay dos conectores de 25 pins, llamados "J1" y "J2", es aquí donde se conectan ambos cables. Las puntas terminales de los cables que quedan libres tienen conectores DB-25 (de propósito general) con contactos macho y hembra. Son estas puntas las que se utilizan para realizar el acople necesario a la aplicación. En el Apéndice B se da una relación de los pins en la tableta del conversor, así como la de los cables y la señal en cada uno de ellos.

SISTEMA DE ANALISIS

4.5 DISCUSION.

En el largo camino que ha tenido la investigación médica desde sus inicios hasta la actualidad se han logrado muchos avances, algunos de los cuales deben su desarrollo gracias al apoyo que la Ingeniería (electrónica, computación, etc.) ha dado en esta área, algunas veces en forma de equipo para la investigación, y otras automatizando procesos.

Un ejemplo de ello se tomó en este trabajo al automatizar los procesos de análisis de actividad eléctrica unitaria extracelular. Para realizar este análisis, antiguamente el investigador o el médico realizaba el registro extracelular almacenando éste en algún medio magnético, para después pasar a papel todo el registro (o algunas partes), por medio de polígrafos (instrumentos aportados por la Ingeniería electrónica). Una vez que el registro se encontraba en papel, se hacía un recuento manual de los eventos que se consideraban espigas para generar también, en forma manual, histogramas de frecuencia. La generación de histogramas de intervalos era aún más complicada debido a que se tenían que medir de alguna forma (con regla generalmente o a ojo) la amplitud de los mismos, hecho que como se ve, ocasionaba un gran error en la información, así obtenida. Una vez generado alguno de estos histogramas o los dos, por simple inspección el investigador decidía que tipo de comportamiento podía estarse presentando en la región registrada (ya que el análisis tanto determinístico como estadístico llevaba mucho tiempo).

Actualmente gracias a la Ingeniería en computación, se cuenta con la automatización de estos procedimientos, esto es, después de que el investigador ha realizado el registro extracelular de la actividad unitaria, la generación de los histogramas tanto de frecuencia como de intervalos, así como el análisis determinístico se realizan en computadora. Con esto se logran varias cosas, entre ellas podemos mencionar: un gran ahorro en cuanto al tiempo invertido en el análisis de datos. Con esto se tiene además, un aumento considerable en la confiabilidad de los mismos debido a la gran disminución en número de errores durante los procesos de captura y de análisis (por ejemplo, los errores de paralaje desaparecen por completo). Ahora bien, la capacidad del sistema de poder procesar información en línea, le permite al usuario tener un incremento (y por ende mayor productividad) en el número de experimentos confiables realizados, lo que acarrea consigo una mayor cantidad de información de calidad como apoyo en sus deducciones y conclusiones. En cuanto a carturas que serán grabadas para posteriormente ser analizadas también presenta ventajas, ya que el usuario puede seleccionar los datos a capturar pues cuenta con un sistema de monitoreo, lo que además optimiza el uso

SISTEMA DE ANALISIS

de los medios de almacenamiento analógicos utilizados.

Resumiendo, la automatización del proceso de captura y análisis de estas señales permite la obtención de resultados más confiables y más actualizados, punto que es de gran importancia en las áreas de investigación científica.

SISTEMA DE ANALISIS

BIBLIOGRAFIA

- 1.- A/D + D/A Operating Manual
Mountain Computer Incorporated 1982.
- 2.- Cramer, H.
Mathematical Methods of Statistics
Princeton University, (1946).
- 3.- Davenport, R. W.
An Introduction to the theory of Random Signals and Noise
McGraw-Hill, New York. (1973).
- 4.- Glaser, Edmund M.
Principles of Neurobiological Signal Analysis.
Academic Press Inc., New York (1976).
- 5.- Leo J. Scanlon
6502 Software Design
Howard W. Sams & Co., Inc., (1980).
- 6.- MCS6500 Microcomputer family
Programming Manual
MOS Technology, Inc., (1976).
- 7.- Reference Manual for IIE Only
Technical Manual
Apple Products, Pub. Dep., 1982

CAPITULO 5

RESULTADOS

Este capítulo presenta en su primera parte la guía de usuario del sistema y en la segunda parte una serie de resultados probados sobre distintas neuronas del Núcleo del Tracto Solitario en el Bulbo Olfatorio, sus histogramas de Frecuencia e Intervalos, ajustes en ciertas zonas de los mismos, ampliaciones de los histogramas, etc.

5.1 GUIA DE USUARIO

Convenciones : Las líneas o los caracteres remarcados representan respuestas que deberá proporcionar el usuario y las no remarcadas las de la microcomputadora. La palabra RETURN equivale a oprimir la tecla llamada RETURN en la microcomputadora.

Importante: Cada vez que el usuario dé una respuesta, ésta deberá ir acompañada de un RETURN.

5.1.1 Módulo De Captura

Para realizar una captura, primero se deben hacer las conexiones necesarias entre los dispositivos externos que se utilizan y la microcomputadora. Un diagrama de conexiones se muestra en la figura 4.2 (Capítulo 4). Posteriormente se ejecuta el programa de captura el cual deberá estar almacenado en disco flexible (en cualquiera de los dos diferentes manejadores de disco), a través de la instrucción Ejecuta (X) :

RESULTADOS

Command: E(edit), R(run), F(file), C(comp), L(link), X(ecute), A(assem), D(debug), TC1.11
X
Execute what file? CAPTURA ó #SICAPTURA

Hecho lo anterior el programa presenta su menú de entrada :

```
+-----+  
|          |  
|  MODULO I  |  
|  CAPTURA DE ESPIGAS  |  
|          |  
+-----+
```

MENU PRINCIPAL

(C) aptura
(T) erminar

OPCION :

a lo que el usuario deberá responder : C . Ahora el programa pide los datos del número de canales de entrada y la duración del registro. Si en respuesta a la duración del registro se proporciona un número menor a 1 o mayor a 400 (segundos) el programa responderá :

>>>> No se permiten 1 > muestras > 400 segundos <<<<

y regresará a pedir los datos. Habiendo proporcionado estos datos correctamente, el Programa comienza la captura hasta que el usuario así lo indica al responder S y RETURN a la pregunta que hace la microcomputadora de :

DESEA COMENZAR EL REGISTRO ... (S/N)

Inmediatamente que el usuario decide iniciar, el Programa espera el pulso de sincronía y empieza a tomar datos, mostrándolos a su vez en el osciloscopio (monitoreo). Cuando la captura concluye el usuario tiene la opción de almacenar ésta en disco o desecharla (en caso de registro no deseado) después de lo cual se presenta nuevamente el menú para continuar capturando (C) o terminar (T). Los resultados de la captura almacenada en disco constituirán la entrada al siguiente módulo.

RESULTADOS

5.1.2 Módulo De Análisis

Este módulo está constituido por dos programas que permiten :
1.- Generación de Histogramas de Frecuencia e Intervalos y
2.- Ajuste a funciones en alguna zona del Histograma.

5.1.2.1 Generación De Histogramas

Para realizarla se corre el programa GENERA de la misma forma como se corrió el de captura :

```
Command: E(edit, R(run, F(file, C(cmp, L(link, X(ecute, A(sshem, D(ebug, ?(1,1)
X
Execute what file? GENERA ó #5:GENERA
```

Después de lo cual el programa presenta su menú de entrada :

```
+-----+
|          MODULO II          |
|    HISTOGRAMAS DE ESPIGAS    |
+-----+
```

MENU PRINCIPAL

```
(F) frecuencia Histograma
(I) ntervalos Histograma
(T) erminar
```

OPCION:

a lo que el usuario puede responder con F o I dependiendo del tipo de histograma que desea obtener. De acuerdo con el tipo de respuesta el programa presentará :

```
.... Tamano del Bin (msed) ..... 500
```

ó

```
.... Tamano del Intervalo (msed) ..... 100
```

y el nombre del archivo de datos conteniendo los datos de tiempos de aparición de las espigas (captura). Es muy importante que el disco

RESULTADOS

que contenga el archivo de datos se coloque en el manejador 2, debido a que este programa tratará de leer la información de este dispositivo ;

NOMBRE DEL ARCHIVO DE DATOS CAPTURAI

En este punto el programa comienza a generar el histograma seleccionado de acuerdo con los parámetros de entrada para lo cual muestra en pantalla, una pequeña tabla de información concerniente al archivo de datos que está leyendo (como medio de control para el usuario). Esta tabla presenta la siguiente información :

Fecha de Registro	06-SEP-85
# de canales	1
Tiempo de duracion	200 seg.

Una vez que el programa termina de generar el histograma, muestra en la pantalla el menú para la presentación de el histograma :

HISTOGRAMA DE FRECUENCIA

- (B) barra simple
- (A) shurado
- (S) alir

OPCION:

(B) si se desea ver el Histograma con barras simples, (A) barras rellenas o (S) no se desea ver en pantalla. Si se contesta alguna de las dos opciones correctas para mostrar el histograma en pantalla el programa presentará su siguiente menú para ajuste del mismo en escala vertical :

- (M) axima escala
- (D) iferente escala
- (S) alir

OPCION:

RESULTADOS

Si se escoge la opción D, el programa preguntará por una nueva escala vertical :

A que Escala ...? 50

de lo contrario, con la opción M, se presenta el histograma en pantalla a su máxima escala. Cada figura generada por la microcomputadora presenta hasta 80 barras del histograma, por esta razón, si el histograma correspondiente al tiempo de captura total, no cupiera en una sola pantalla, dado el tamaño del bin, el programa lo presentará en forma seccionada mostrando 80 barras en cada despliegue. Para poder observar la siguiente sección del histograma simplemente debe oprimirse la tecla de RETURN. Antes de presentar la siguiente sección, el programa presenta otras opciones, esto es, presenta si se desea cambiar el tamaño del bin (volver a generar el histograma con nuevo valor de bin), o si se desea ampliar alguna zona del histograma (ampliación en eje horizontal) :

.... Desea cambiar tamaño del bin (S/N)N
.... Desea ampliar alguna zona del histograma (S/N) ...S

**** LIMITES DE LA AMPLIACION ****

Inicio : 0
Fin : 1000

con lo cual se amplificaría (para el ejemplo) la zona del histograma comprendida entre 0 y 1000 milisegundos. Si ya no existen mas secciones del histograma por desplegar el usuario deberá oprimir la tecla S para terminar el despliegue, con lo cual el programa preguntará si se desea guardar el histograma en disco o no (para su posterior ajuste a función) :

.... Desea guardar el histograma (S/N) S
.... NOMBRE PARA EL HISTOGRAMA histol

*** Fecha de Grabacion ***
(dd-mm-aa)
23-SEP-85

el histograma es almacenado registrando los parámetros con que cuenta en ese momento mas la fecha en que se generó. En el ejemplo el histograma se guardaría ampliado. Finalmente el programa regresa a mostrar su menú principal para seguir generando histogramas de Frecuencia (F), histogramas de Intervalos (I) o bien terminar (T).

RESULTADOS

5.1.2.1.1 Impresión De Histogramas En Papel. -

El despliegue del histograma generado por el Programa GENERA, tanto a papel como en pantalla se logra a través del uso de este programa llamado DESPLIE que proporciona la facilidad de pasar el histograma de pantalla a papel debido a que el programa anterior (GENERA) no realiza esta función (por razones de falta de memoria para su implementación). Para correrlo se siguen los mismos pasos que para el de GENERA y el de CAPTURA. Y lo primero que responderá el programa será su menú principal :

```
+-----+
|           |
|   MODULO II.1.1   |
| DESPLIEGUE DE HISTOGRAMAS |
|           |
+-----+
```

MENU PRINCIPAL

```
-----
(D) espliegue
(T) erminar
```

OPCION :D

El programa pregunta en que archivo (en disco) se encuentra almacenado el histograma para desplegarlo en pantalla y después a papel si se desea. De ahí en adelante el programa funciona de la misma manera que el de GENERA.

5.1.2.2 Ajuste A Funciones -

Con este programa se pueden realizar ajustes a curvas teóricas, para ver si la respuesta neuronal representada a través de un histograma sigue algún patrón de respuesta en alguna zona del mismo y poder proporcionar así una herramienta para la formulación de un modelo de comportamiento en particular. Para utilizar el programa se debe correr el programa AJUSTES en la misma forma como se hizo con los anteriores (CAPTURA, GENERA y DESPLIE).

```
Command: E(edit, R(run, F(file, C(omp, L(link, X(ecute, A(ssemb, D(ebug, ?(I.1)
X
Execute what file? AJUSTES ó #5:AJUSTES
```

RESULTADOS

Después de lo cual el programa presentará su menú principal :

```
+-----+
|           |
|  MODULO II.2  |
|  AJUSTE A FUNCIONES  |
|           |
+-----+
```

MENU PRINCIPAL

- (A) Juste a Funciones
- (T) erminar

OPCION:

Al seleccionar A para realizar los ajustes el programa presentará si se desea analizar una suma de trenes de espigas o un solo tren (una sola captura) :

```
... Desea analizar mas de un TREN ... (S/N) ...N
... NOMBRE DEL TREN # 1 .....NTS1A
```

Si se contesta que no, el programa pedirá el nombre del único tren a analizar presentará en pantalla la fecha de generación del histograma que contiene al mismo y seguirá al siguiente menú. Por otra parte, si se pidió analizar mas de un tren el programa pedirá el nombre de cada uno de ellos en orden secuencial para sumarlos y hacer el ajuste sobre la suma de estos. Durante la lectura de cada tren, el programa presenta información adicional sobre el archivo de datos que está leyendo con el fin de que el usuario pueda tener cierto control (chequeo) sobre sus datos :

```
+-----+
| HISTOGRAMA DE FRECUENCIA DEL >> 24-A00-85 |
+-----+
```

RESULTADOS

Inmediatamente después de la lectura, se presentará al usuario el menú de ajuste a funciones :

MENU #2

- [1] Línea Recta
- [2] Potencial
- [3] Exponencial o Malthus
- [4] Berthalanffy o Exponencial Saturada
- [5] Logarítmica o Logística
- [6] Salir del menú

OPCION : 3

El usuario escoge el tipo de ajuste que considere el más conveniente, en la zona del histograma que analizará. Una vez escogida la función, el programa preguntará la zona del histograma a analizar y presenta una tabla de estadísticas básicas sobre esta zona (estos datos se deberán proporcionar en unidades de milisegundos):

LIMITES PARA EL AJUSTE

Inicio : 0
Fin : 30000

TABLA DE ESTADISTICAS BASICAS en los límites 00 a 30000 ms.	
Media Aritmetica	8.40000
Desviacion Standard	3.61107
Moda	15
Coef. de Variacion	4.29892E-1

-- PARA CONTINUAR OPRIMA (RETURN) <--

A continuación el programa presenta el ajuste de la recta teórica (en la zona de análisis) por el método de mínimos cuadrados, remarcando con un círculo el punto de la ordenada al origen. Y finalmente muestra la función ajustada con todos los parámetros obtenidos a partir del método de m.c. Si en este punto el usuario oprime la tecla de RETURN, el programa regresará a presentar su menú 2 para seguir ajustando o salir al nivel más alto de menús (menú

RESULTADOS

principal) y terminar (T). Si en lugar de ello el usuario oprime la tecla S para salir (esto sucede tanto en la presentación de la recta como la de la función ajustada), el programa preguntará si se desea pasar las graficas a papel o no :

... Desea pasar la grafica a papel ... (S/N) ...N

Para después regresar al mismo punto que en el caso de oprimir la tecla de RETURN, esto es, a los menús.

3.2 EJEMPLOS

Los ejemplos que a continuación se presentan fueron tomados de una serie de cuatro neuronas distintas localizadas en la zona del Núcleo del Tracto Solitario. El registro de las neuronas se hizo, antes de estimulación, con estimulación y después de ella. Las estimulaciones fueron de tipo gástrico como ya se explicó ampliamente en el Capítulo 2.

Para cada ejemplo se presenta su histograma de frecuencias, y una amplificación de una zona del histograma en donde se hará el ajuste mas adecuado. En algunos casos, donde fué mas representativo, se presenta también el histograma de intervalos y el mejor ajuste sobre el mismo. Los ejemplos se presentan intentando cubrir todas las opciones que el sistema presenta (amplificaciones, histogramas de barras simples o ahuradas, etc).

RESULTADOS

5.2.1 Ejemplo #1

Como primer ejemplo se eligió el análisis de la respuesta presentada por la neurona 41 del registro obtenido del NTS. La longitud del tren es de 200 segundos; se aplicó una estimulación después de 25 segundos de comenzado el registro, y en la figura 5.1a se muestra el histograma de frecuencias con un tamaño de bin de 2.5 segundos. En él se observa que el pico máximo (17 espigas) se dio a los 65 segundos. Este histograma no parece presentar cambios muy significativos en ningún momento; sin embargo antes de aplicar la estimulación (en los primeros 25 segundos) parece haber una tendencia al incremento en la frecuencia. El histograma se presenta con la opción de barras simples y a escala máxima.

En la figura 5.1b se puede observar una amplificación y escalamiento a 17 barras para el histograma anterior en los primeros 25 segundos (este histograma se presenta con la opción de barras ahuradas); mientras que la recta teórica para ajuste a función exponencial saturada o Berthalanffy se muestra en la figura 5.1c. Esta recta tiene una pendiente de 7 grados y ordenada al origen de 1.2. Para esta zona se ajustaron curvas logarítmica, potencial y exponencial saturada; siendo la exponencial saturada, la que más se asemeja al comportamiento real (ver figura 5.1d).

La figura 5.1e muestra el histograma de intervalos para la misma neurona (opción de barra simple). Aquí se presentan solo los primeros 20 segundos; debido a que ya no apareció respuesta alguna en lo que restaba del tren (de 200 segundos). La estimulación también se aplicó a los 25 segundos de comenzada la captura sin observar cambio alguno en la amplitud de los intervalos posteriores. El tamaño del intervalo para este histograma fué de 250 milisegundos, y el tamaño de intervalo que más frecuentemente se presentó durante el tren fué el de 250 milisegundos (282 intervalos).

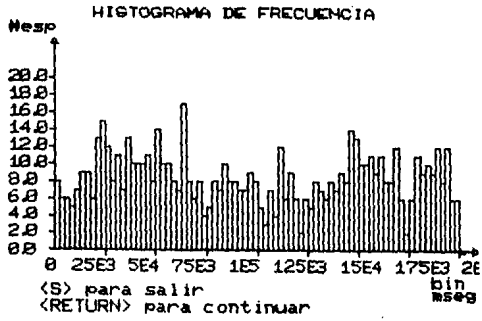
La figura 5.1f presenta otras de las opciones del sistema; esto es, la amplificación en los primeros 2 segundos del histograma anterior y a un escalamiento vertical de 101. La recta teórica para ajuste a una función exponencial sobre el primer segundo de la figura 5.1a, es mostrada en la figura 5.1g. Dicha recta tiene una pendiente de 130 grados y una ordenada al origen de 6. Finalmente, la figura 5.1h presenta el ajuste a curva exponencial del primer segundo del histograma de intervalos para esta neurona. Como se puede observar, nuevamente se tiene el mismo tipo de respuesta exponencial decreciente, sin observarse mayores cambios a partir de la estimulación. Este hecho no permite extraer mucha información de este

RESULTADOS

tipo de histograma para los casos muy particulares que aquí se manejan; lo que significa que en este experimento específico la estimulación no altera la información contenida en los intervalos interespidas. Debe notarse que en otro tipo de experimentos, bajo otro conjunto de condiciones paramétricas, Klemm y Sherry (1962) mostraron que en las neuronas del hipocampo, en ratos, existe una significancia estadística de ordenamiento serial de ciertos patrones de intervalos relativos durante estados específicos del comportamiento ('quiet wakefulness' y sueño REM), mientras que dicha significancia no se presenta en otras conductas (sueño de onda lenta). En su laboratorio la actividad espontánea de neuronas cerebelares en ratas demostró la existencia de grupos de hasta 6 intervalos adyacentes cuya incidencia estadísticamente diversifica de la independencia. Esto significa que una neurona utiliza un patrón de intervalos solo bajo ciertas circunstancias debido a que los requerimientos de acarreo de información varían con el estado funcional de la preparación biológica; esto último también implica que algunas neuronas puedan operar en códigos de patrones de espigas, mientras que otras no.

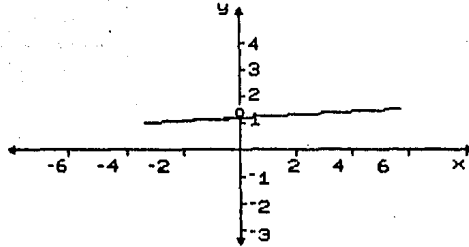
RESULTADOS

5.1.-



RESULTADOS

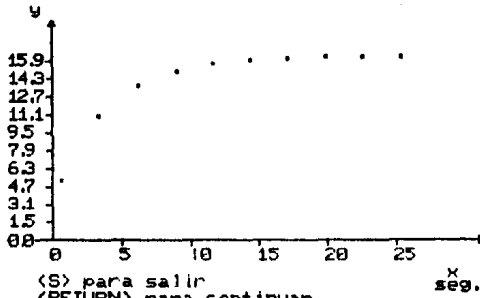
AJUSTE POR MINIMOS CUADRADOS



(c)

<S> para salir
<RETURN> para continuar

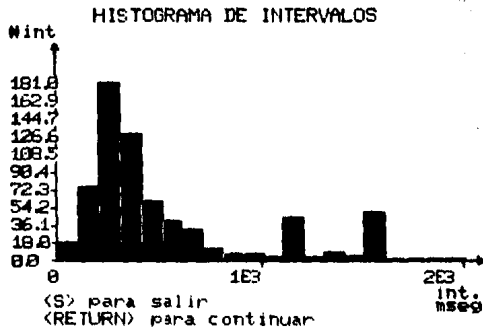
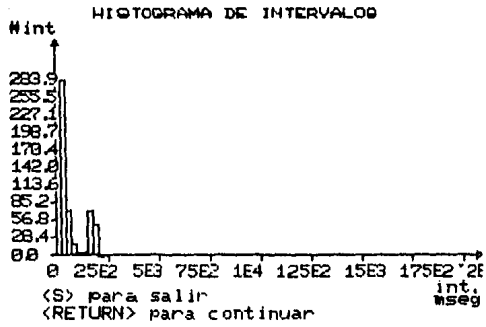
AJUSTE A CURVA BERTHALANFFV



(d)

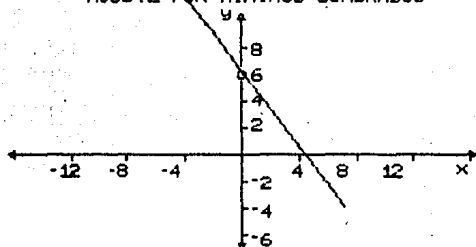
<S> para salir
<RETURN> para continuar

RESULTADOS



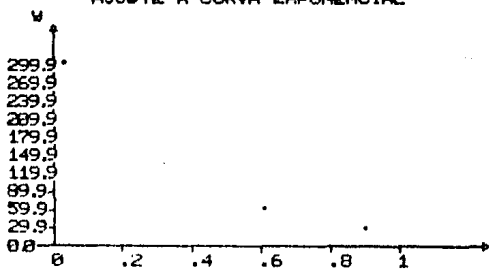
RESULTADOS

AJUSTE POR MINIMOS CUADRADOS



<S> para salir
<RETURN> para continuar

AJUSTE A CURVA EXPONENCIAL



<S> para salir
<RETURN> para continuar

RESULTADOS

5.2.2 Ejemplo #2

Aquí se presentan los resultados de el análisis del comportamiento fisiológico de una segunda neurona en la zona del NTS (ver figura 5.2a,b y c). La longitud del tren utilizado es también de 200 segundos, se aplicó una estimulación después de 25 segundos de comenzado el registro, y el bin para este histograma fué de 2.5 segundos. En la figura 5.2a se observa que el pico máximo en este histograma (34 espigas) se dió a los 60 segundos. Nuevamente se observa que después de aplicada la estimulación hay un incremento rápido en la frecuencia de aparición de las espigas, el cual dura hasta los 70 segundos de captura, para después comenzar a decaer y estabilizarse. De este histograma se observa claramente que existe una respuesta de la neurona debida a la estimulación, hecho que en el histograma anterior no había sido observado claramente. Esto permite llegar a la conclusión de que aún cuando exista un modo de operación de patrones de intervalos, o de frecuencias, su identificación puede no ser siempre evidente por sí misma.

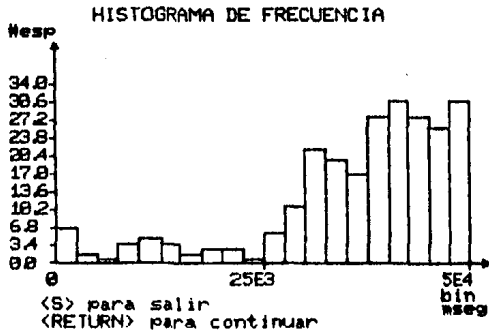
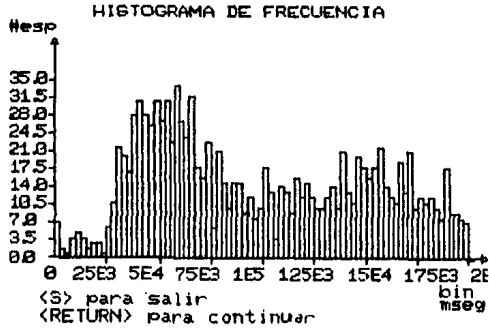
La figura 5.2b muestra una amplificación del histograma de la figura 5.2a, en los primeros 50 segundos, que fueron los del ascenso rápido, para proceder a realizar un ajuste a una curva logarítmica. La figura 5.2c presenta la recta teórica para ajuste a función logarítmica. Esta recta tiene una pendiente de 7 grados, aproximadamente, y una ordenada al origen en 1.4. La función logarítmica ajustada en la zona de 0 a 50 segundos para el histograma de la figura 5.2a se muestra en la figura 5.2d. Como se observa el comportamiento obtenido es muy similar al real (ver figura 5.2a).

La figura 5.2e presenta el histograma de intervalos para la misma neurona #2, y con los mismos parámetros del histograma de frecuencia, para un intervalo de 100 milisegundos. Aquí se puede observar que el rango de valores para los intervalos obtenidos cae entre 0 y 1 segundo, siendo los intervalos de 200 milisegundos los más frecuentes. En este histograma, solo se presenta una amplificación del primer segundo del tren, debido a que sólo en este intervalo se observó respuesta.

La figura 5.2f muestra la recta teórica para un ajuste a función exponencial de la zona presentada en la figura 5.2a. La recta tiene una pendiente de 95 grados y ordenada al origen de 6. Por último, la figura 5.2g presenta la función exponencial decreciente que sigue el comportamiento del histograma de intervalos generado para esta neurona.

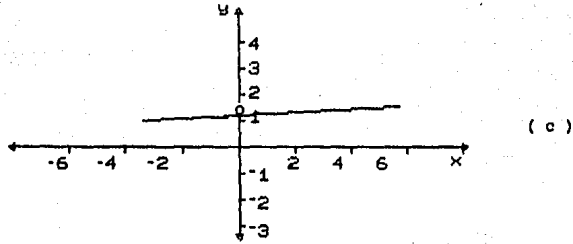
RESULTADOS

5.2.-



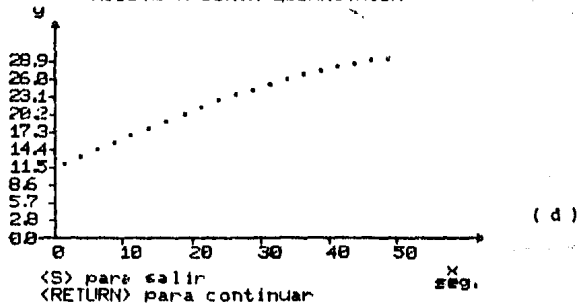
RESULTADOS

AJUSTE POR MINIMOS CUADRADOS



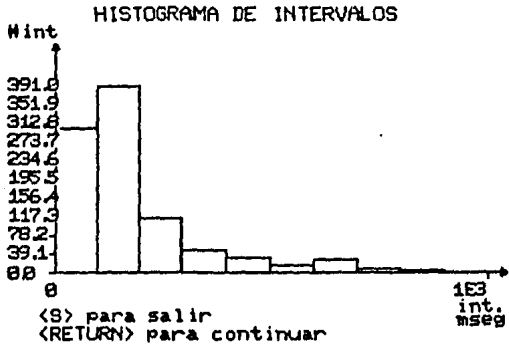
<S> para salir
<RETURN> para continuar

AJUSTE A CURVA LOGARITMICA

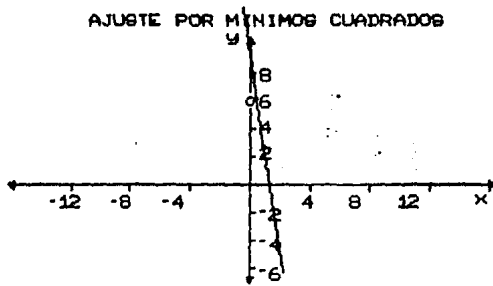


<S> para salir
<RETURN> para continuar

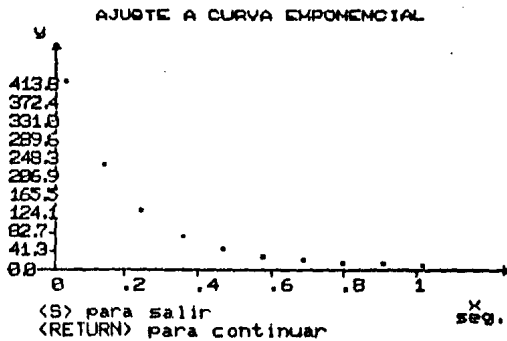
RESULTADOS



(e)



(f)



(g)

RESULTADOS

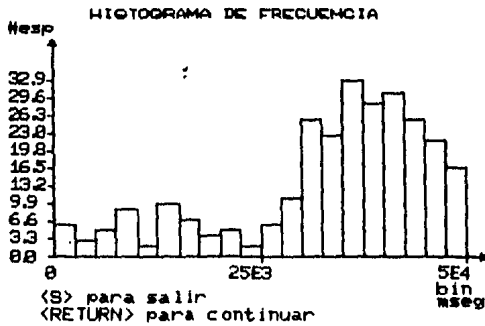
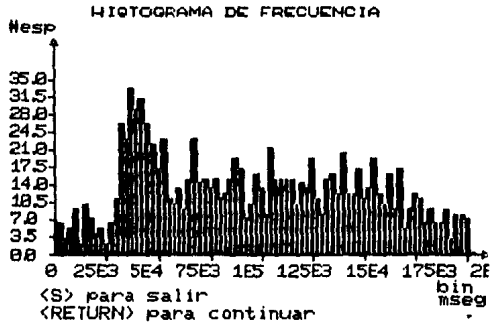
5.2.3 Ejemplo #3

Las figuras 5.3a,b,c y d muestran los resultados del análisis de la respuesta fisiológica de la neurona #3 del NTS. La duración del tren fué de 200 segundos, con un artefacto de estimulación aplicado a los 25 segundos de iniciado el registro y un tamaño de bin de 2.5 segundos. En la figura 5.3a se muestra el histograma de frecuencias donde se observa un incremento rápido de la frecuencia a partir de la aplicación del estímulo y durante los 15 segundos posteriores, para después comenzar a decaer también rápidamente durante 15 segundos hasta estabilizarse (a partir de los 50 segundos aproximadamente). El pico máximo aquí se presenta a los 32.5 segundos de iniciado el registro y fué de 33 espigas.

Una amplificación del histograma anterior en los primeros 50 segundos, puede observarse en la figura 5.3b) mientras que en la figura 5.3c se muestra la recta teórica para un ajuste a función exponencial. Dicha recta presenta una pendiente de 5 grados y una ordenada al origen de 1. Por último, en la figura 5.3d se observa la función exponencial ajustada en los primeros 50 segundos del histograma de la figura 5.3a. Se observa que esta curva se asemeja grandemente al comportamiento real que sigue la neurona en este período de tiempo.

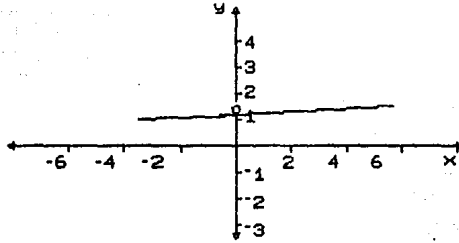
RESULTADOS

5.3.-



RESULTADOS

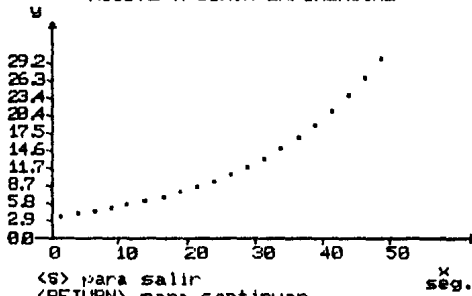
AJUSTE POR MINIMOS CUADRADOS



(c)

<S> para salir
<RETURN> para continuar

AJUSTE A CURVA EXPONENCIAL



(d)

<S> para salir
<RETURN> para continuar

RESULTADOS

5.2.4 Ejemplo #4

Como último ejemplo, presentamos el análisis de datos obtenidos en la neurona #4, registrada en la zona del NTS, con un tren de 200 segundos (200,000 milisegundos) de registro y un tamaño de bin de 2.5 segundos. En la figura 5.4a se presenta el histograma de frecuencias, donde se observa que el número máximo de espigas (32) se da en el bin de 45 segundos. La estimulación en este experimento fué proporcionada a los 25 segundos del registro, y se puede observar que a partir de ese momento existe un incremento rápido en la frecuencia de aparición de las espigas (barras más altas), la que continúa hasta los 50 segundos, donde empieza a decaer lentamente hasta estabilizarse durante los siguientes 50 segundos.

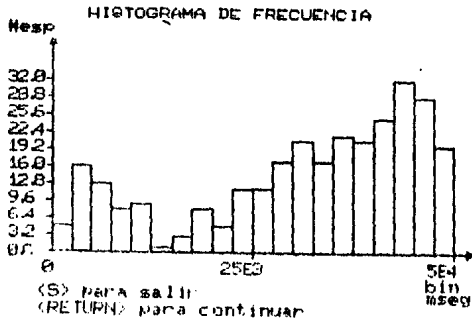
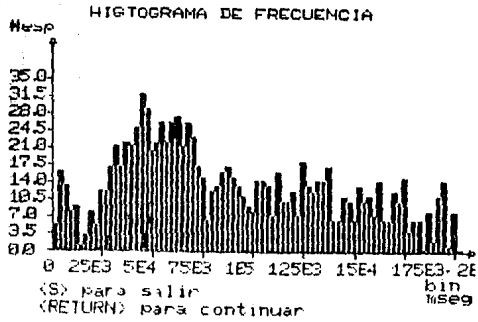
La figura 5.4b presenta una amplificación del histograma de frecuencias de la figura 5.4a en los primeros 50 segundos, que fué donde se dió el ascenso rápido. Sobre esta zona se realizará un ajuste a una curva potencial para observar la similitud entre el comportamiento real y el teórico.

En la figura 5.4c se presenta la recta teórica que resulta del ajuste a curva potencial para la zona que se presenta en la figura 5.4b. La recta tiene una pendiente de 30 grados y la ordenada al origen es igual a uno. Finalmente, en la figura 5.4d se puede observar el ajuste a curva potencial para el histograma de la figura 5.4b. Como se puede observar, el comportamiento que sigue esta función ajustada se asemeja bastante bien al comportamiento real de la neurona (figura 5.4b), lo que permite decir que en esa zona la neurona siguió un comportamiento de tipo función potencial.

De los ejemplos presentados en este trabajo se puede concluir que al menos una zona del núcleo del tracto solitario está relacionada con los procesos de ingestión de alimentos, ya que se observaron cambios significativos en la frecuencia de la respuesta neuronal a partir de estimulaciones gástricas en la zona correspondiente.

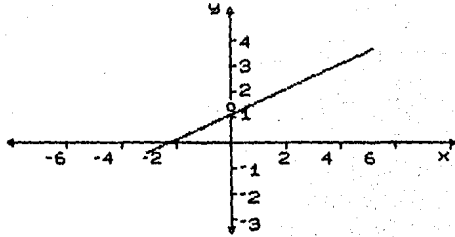
RESULTADOS

5.4.-



RESULTADOS

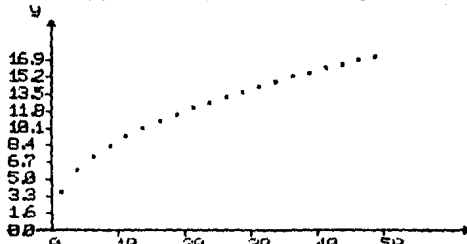
AJUSTE POR MINIMOS CUADRADOS



(c)

<S> para salir
<RETURN> para continuar

AJUSTE A CURVA POTENCIAL



(d)

<S> para salir
<RETURN> para continuar

CAPITULO 6

CONCLUSIONES

6.1 EVALUACION DEL SISTEMA

En el campo de las neurociencias (estudio del sistema nervioso), cada herramienta de análisis que permita la extracción de información significativa sobre el comportamiento neuronal, representa una fuente de datos que servirán para apoyar teorías ya existentes o bien para dar pauta a nuevas deducciones y nuevas teorías. Tanto los histogramas de frecuencia como los de intervalos proporcionan información útil sobre el comportamiento de una vía neuronal específica. Esta información es complementaria debido a que en algunas ocasiones los histogramas de frecuencia revelan cambios que en los histogramas de intervalos son impercibibles y viceversa dependiendo de las condiciones particulares del experimento del área del sistema nervioso bajo estudio.

Durante los experimentos analizados con el sistema desarrollado en este trabajo se observaron cambios significativos al aplicar un estímulo sobre una serie de siete neuronas diferentes en la zona del núcleo del tracto solitario (de las cuales se presentan cuatro en el capítulo anterior), en cuanto a la frecuencia de aparición de eventos (histograma de frecuencia), sin embargo la relación de intervalos intereventos (histograma de intervalos) para todos los casos se mantuvo más o menos constante no revelando cambios de gran importancia (ver ejemplos Capítulo 5).

El sistema de análisis no sólo proporciona, a través de los dos tipos de histogramas, los cambios de frecuencia o de intervalos intereventos existentes, sino que también realiza la medición de las tasas de cambio de estas variaciones, con lo que se obtiene información cuantificable sobre lo que sucede en el fenómeno, permitiendo proponer funciones "tipo" que semejen el comportamiento de una neurona o de un conjunto de neuronas en particular. Esto se lleva a cabo ajustando aquella función que semeje mejor el comportamiento del tren neuronal o de una porción del mismo, con lo cual se obtiene nueva información útil para el planteamiento de algún modelo teórico.

CONCLUSIONES

de comportamiento. Las dos herramientas utilizadas en este trabajo, los histogramas de frecuencia y de intervalos, sirven muy bien en algunas condiciones pero en otras no (McGregor & Lewis, 1977); esto sugiere la no individualidad de estas pruebas. El uso de cualquiera de estos histogramas, o de cualquier herramienta en general, por sí sola para la deducción de resultados resulta prácticamente inválida debido a que con ella se llegaría a conclusiones mas bien empíricas y no con una base científica. Su gran valía va acompañada indudablemente por el uso de herramientas extras (que mientras mas relacionadas estén entre sí darán resultados mas concretos) de análisis como lo constituyen por ejemplo, el análisis estadístico, un análisis Probabilístico o bien como se propone en este trabajo un estudio analítico, herramientas que si se pudieran conjuntar de la mejor manera resultarían en un arma de análisis mas completa.

De los resultados presentados en este trabajo se pudieron obtener las siguientes conclusiones :

6.1.1 Papel De La Ingeniería En Estudios Fisiológicos.

La Ingeniería a través de la instrumentación electrónica, así como del desarrollo de sistemas (equipo electrónico y Software) tiene una gran influencia en la investigación científica, en particular en el campo de las neurociencias debido principalmente a la capacidad de procesamiento de señales bioeléctricas en línea, esto es, en tiempo real. Sin embargo, hasta el momento se había descuidado un tanto, el apoyo matemático para el análisis de estas señales debido principalmente a la complejidad que este tipo de análisis representaba. El sistema en este trabajo desarrollado pretende, además de proporcionar la capacidad de procesamiento de información en tiempo real o pregrabada, el proporcionar una herramienta de análisis que hasta el momento había sido ignorada debido a las razones antes mencionadas. Además, con esto, se permite la condensación de una gran cantidad de información desplegada en forma gráfica para una mejor visualización de los fenómenos fisiológicos.

CONCLUSIONES

6.1.2 Sistema Desarrollado.

El sistema aquí desarrollado presenta las siguientes ventajas :

1. Tiene la capacidad de procesar información en línea o presabada, lo que le da una gran versatilidad.
2. Se realizó en base a módulos (siguiendo las técnicas de la programación estructurada) con lo que se obtiene mayor claridad en la programación, fácil mantenimiento y por lo tanto una mayor vida útil del sistema.
3. Filtrado de la información en todos los niveles y chequeo de errores.
4. Despliegue gráfico a todos los niveles (osciloscopios, graficadores, monitores).
5. Almacenamiento de la información para su posterior procesamiento o graficación.
6. Gran claridad de manejo del sistema, cualquier usuario puede hacer uso del sistema ya que su manejo es a base de menús.
7. Claridad en la programación, nombres de módulos y rutinas de acuerdo al papel que desarrollan.
8. Transportable, el sistema se diseñó para que su transporte a otro sistema provoque el mínimo de cambios.

CONCLUSIONES

6.1.3 Análisis Desarrollado.

El estudio realizado, como ejemplo sobre la respuesta de un grupo de neuronas en el núcleo del tracto solitario con el sistema desarrollado en este trabajo permitió llegar a las siguientes conclusiones :

1. Algunas neuronas podrían operar en códigos de patrones de espigas, mientras otras no, esto debido a que neuronas diferentes tienen demandas de acarreamiento de información diferentes sobre ellas.
2. Una neurona dada utiliza un código de patrón de intervalos solo bajo ciertas condiciones, ya que los requerimientos de acarreo de información puede ciertamente esperarse que varíen con el estado fisiológico, por ejemplo el ordenamiento serial de los intervalos puede ser evidente en estado activo de la persona, pero puede estar ausente durante el sueño (8).
3. Aún cuando exista un modo de operación de patrones de intervalos, o de frecuencias, su identificación puede no ser siempre evidente por sí misma.
4. Al menos una zona registrada dentro del área del núcleo del Tracto Solitario está relacionada con los procesos de ingestión de alimentos, ya que se observaron cambios significativos en la respuesta de las neuronas en esta zona al estimular con distensiones gástricas, durante algunos segundos.

6.2 FUTURAS IMPLEMENTACIONES DEL SISTEMA

El sistema se encuentra actualmente implementado en la Unidad de Fisiología de la Facultad de Medicina de la UNAM como base a la investigación en el área de estudio de actividad unitaria extracelular. Se tiene planeada una siguiente implementación en el Laboratorio de Bioingeniería del Departamento de Fisiología del Instituto de Fisiología Celular, debido a que este tipo de estudios

(8) Como lo demuestran Kleen y Sherry Neuroscience Marzo 1982.

CONCLUSIONES

también se llevan a cabo en este Laboratorio. Un comentario final sobre la evaluación del sistema es con respecto al módulo de captura utilizado. Sería conveniente una modificación en aquellos sistemas de cómputo que cuenten con el manejo de interrupciones para la toma de las espigas., esta modificación daría al sistema una confiabilidad más grande y una disminución en cuanto a la complejidad del módulo (programa mas corto y mas sencillo). Cabe aclarar que la única modificación que tendría que hacerse dentro del módulo que aquí se presentó sería la introducción de una subrutina dentro de la rutina principal de captura para la espera de interrupciones (lo que eliminaría algunas instrucciones con las que cuenta en este momento) pero la estructura general seguiría siendo la misma, y por ende el programa principal de captura no sufriría modificación alguna.

4.3 FUTURAS ETAPAS

Se tienen varios pasos a seguir. El primero de ellos podría ser la ampliación del sistema para analizar actividad multiunitaria también, para lo cual haría falta la utilización de mas canales de entrada al sistema provenientes de varios electrodos de registro. Otro paso sería el de conformar un tipo de análisis mas general conjuntando la naturaleza probabilística de estos procesos, para con ambos tipos de análisis (el probabilístico y el que aquí se propone, uno cualitativo y el otro cuantitativo) apoyar en forma más completa a la formulación de los correspondientes modelos teóricos generados por este tipo de análisis. También sería deseable, el registrar ahora ya no solamente en forma extracelular sino también intracelular para de esta forma estudiar conjuntamente los eventos de actividad unitaria pero desde su génesis.

```

(*****
*)
(*)      MODULO I .-  CAPTURA DE ESPIGAS      (*)
(*)      _____                          (*)
(*)
(*)      Este modulo es el primero de los tres que conforman el (*)
(*)      " Sistema de Analisis de la actividad electrica unitaria (*)
(*)      extracelular ".                          (*)
(*)      Esta compuesto a su vez de dos programas distintos, uno (*)
(*)      realizado en el lenguaje de programacion PASCAL y el otro en (*)
(*)      lenguaje Ensamblador del 6502.          (*)
(*)
(*)      Entradas : La entrada de datos se hace a traves de los 2 (*)
(*)      ----- programas. En el programa PASCAL se leen los (*)
(*)      parametros de captura que habran de pasar al (*)
(*)      programa en Ensamblador para la realizacion de la captu (*)
(*)      ra, propiamente dicha. Estos parametros son :          (*)
(*)
(*)      a) Cuantos canales del conversor A/D (*)
(*)      contendran datos a ser leidos. (*)
(*)      b) Tiempo de duracion de la captura. (*)
(*)
(*)      Salidas : Las salidas de este modulo se hacen tambien (*)
(*)      ----- a traves de los 2 programas. En el programa (*)
(*)      en ensamblador se manda una salida de los (*)
(*)      datos leidos a traves de un canal del conversor D/A, al (*)
(*)      osciloscopio para un monitoreo util al usuario; y en (*)
(*)      el programa PASCAL la salida consiste de un archivo en (*)
(*)      disco conteniendo la informacion de la captura para su (*)
(*)      analisis posterior. (*)
(*)
(*****

```

PROGRAM TOMADATOS;

```

Type
  bytes=0..255; (* Procesador de 8 bits *)
  Arreglo = Packed array[0..9000] of bytes; (* Contadores en Assmb. *)
  Grafico = Packed array[0..3000] of real; (* Tiempos de aparicion *)
(* de las espigas. *)
Var
  comando : char; (* Respuesta al menu *)

```

```

(*****
*)
(*)      Rutina para el borrado de la pantalla (*)
(*)
(*****

```

PROCEDURE BORRA(x,y:integer);

```

Var
  i : integer; (* Variable de iteracion *)
Begin
  Gotoxy(0,0);
  For i:=0 to 23 do (* 24 lineas de pantalla *)
    writeln(' ':75)
  (* endfor *)
  Gotoxy(x,y); (* Posicion del cursor *)
end
(* END-BORRA *);

```

```

(*****): *****
(x
(* Rutina en Ensamblador del 6502 para la captura de las *)
(* Espigas. *)
(* *)
(*****))

PROCEDURE CAPTURA(nchan:integer;Var his:arreglo);
  external
(* END CAPTURA *);

(*****
(* *)
(* Rutina que lee los parametros de entrada para la captura : *)
(* *)
(* 1.- Numero de canales del conversor A/D que permitiran *)
(* la entrada de las espigas (en el caso de registros *)
(* con mas de un electrodo). *)
(* *)
(* 2.- Tiempo de duracion para la captura ( se restringe *)
(* a valores mayores o iguales a uno y menores o *)
(* iguales a 400 segundos ). *)
(* *)
(*****))

PROCEDURE DATOS_ENT(Var nc,d:integer;Var bien:boolean);
Var
  correc : char; (* Para datos correctos *)
Begin
  Gotoxy(5,8);
  writeln;
  write(' # de canales a muestrear.....');
  readln(nc);
  write(' Duracion del registro (seg)....');
  readln(d);
  If (d<1) or (d>400) then
    Begin
      writeln('>>>> No se permiten l > muestras > 400 segundos <<<<');
      bien:=false;
    end
  (* noelse *)
  (* endif *)
  writeln;
  write(' _____ ');
  writeln(' # DE CANALES ',nc:8);
  write(' DURACION DEL EXP.',d:5);
  writeln(' _____ ');
  writeln;writeln;
  write(' ESTAN CORRECTOS LOS DATOS...(S/N)');
  readln(correc);
  If (correc <> 'S') and (correc <> 's') then
    If (correc <> 'N') and (correc <> 'n') then
      Begin
        writeln;
        writeln(' >>>>>> P.F. solo respuesta SI o NO');
        bien:=false;
      end
    else
      bien:=false
    (* endif *)
  else
    bien:=true
  (* endif *)
end
(* END DATOS_ENT *);

```

```

(*****)
(*)
(*) Rutina que ajusta los parametros de entrada para poder ser (*)
(*) almacenados en las localidades de memoria destinadas para (*)
(*) los contadores del tiempo de duracion de la captura, en la (*)
(*) rutina de CAPTURA. (*)
(*)
(*****)

PROCEDURE ENTRADA(Var nch,dur:integer;Var his:arreglo;Var histo:grafico);
Const
  t1 = 2; (* Carga de byte alto para cuenta de min.*)
  tm = 0.041; (* Velocidad de muestreo en ensamblador *)
  t2 = 3; (* Carga de byte bajo para cuenta de seg.*)
  c2 = 1; (* # de segundos donde empieza muestreo *)
Var
  inh : integer; (* Indicador de carga para cuenta de min.*)
  valor,
  residuo,
  ci, (* Aux.en el calculo de la carga de cont.*)
  (* Da exactitud en la carga de contadores*)
  (* A partir de este valor se incrementa *)
  (* la cuenta en minutos. *)
  inci, (* Incremento de tiempo/muestra para min.*)
  inc2 : real; (* Incremento de tiempo/muestra para seg.*)
  bien : boolean; (* Bandera indicadora de datos validos *)
Begin
  ci := (65535.0*tm)/1000.0;
  inci := ci;
  inc2 := (255.0*tm)/1000.0;
  inh:=0;
  borra(8,3);
  Datos_ent(nch,dur,bien);
  While not(bien) do
    Datos_ent(nch,dur,bien)
  (* endwhile *);

  (* Entre 1 y 3 segundos no hay necesidad de *)
  (* carga de un ciclo en contador de minutos *)

  If (dur>1) and (dur<3) then
  Begin
    Valor:=(dur-c2)/inc2;
    If (valor)=255.0) then
      Begin
        inh:=1;
        his[0]:=round(valor)+t2-255;
      end
    else
      his[0]:=round(valor)+t2
    (* endif *);
    his[1]:=2+inh;
  end
else

```

```

(* Todos los demas necesitan carga para min. *)

If (dur>=3) and (dur<400) then
  Begin
    Valor:=(dur-c1)/inci;
    residuo:=trunc(valor);
    residuo:=valor-residuo;
    If residuo<=0.5 then
      his[0]:=round(residuo*100.0)
    else
      his[0]:=0
    (* endif *)
    his[1]:=round(valor)+t1;
  end
else
  If (dur=1) then
    Begin
      his[0]:=1;
      his[1]:=1;
    end
  (* noelse *)
  (* endif *)
  (* endif *)
end
(* END ENTRADA *);

```

```

(*****)
(*
(* Rutina de espera para Inicio de captura o Repeticion *)
(*
(*****)

```

```

PROCEDURE ESPERA(Var ini:boolean;orden:string);
Var
  r : char; (* Respuesta, s = inicio *)
Begin
  Gotoxy(5,22);
  write(' DESEAS ',orden,' REGISTRO...[S/N]');
  readln(r);
  If ((r = 'S') or (r = 's')) then
    ini:=true
  else
    ini:=false
  (* endif *)
end
* END ESPERA *);

```



```

(*****
(*)
(*) Rutina que inicializa los vectores de tiempos de aparicion (*)
(*) de espigas y de contadores de tiempos de aparicion (*)
(*)
(*****

```

```

PROCEDURE INICIA(n:integer;Var vector:arreglo;Var vec1:grafico);
Var
  i,                                     (* Variable de iteracion *)
  m : integer;                           (* Tamano del vector de *)
                                           (* tiempos de aparicion. *)
Begin
  borra(12,12);
  writeln('          *** INICIALIZANDO ***');
  For i:=0 to n do
    vector[i]:=0
  (* endfor *);

  (* Su tamano es 1/3 de la del vector de espigas *)

  m:=round(n/3);
  For i:=0 to m do
    vec1[i]:=0.0
  (* endfor *);
end
(* END INICIA *);

```

```

(*****X: *****
(*)
(*) Rutina que presenta el Menu Principal del Modulo (*)
(*)
(*****X: *****

```

```

PROCEDURE MENU(Var comando:char);
Begin
  Gotoxy(21,10);
  writeln('          MENU PRINCIPAL');
  Gotoxy(21,11);
  writeln('          _____');
  Gotoxy(15,14);
  writeln('          (C) aptura en linea.....');
  Gotoxy(15,15);
  writeln('          (T) erminar.....');
  Gotoxy(26,18);
  write(' OPCION : ');
  readln(comando);
  writeln;
end
(* END MENU *);

```

```

(*****): *****
(*)
(*) Rutina que obtiene la informacion de los contadores y la (*)
(*) convierte a valores reales de tiempo para ser almacenados (*)
(*) ( en caso deseado ) en un archivo en disco (Drive 2), con (*)
(*) el nombre de ---) #5:nomarch.dat (*)
(*) Esta informacion puede ser graficada posteriormente por (*)
(*) el MODULO II de este sistema : "Histogramas de Espigas". (*)
(*)
(*****)

```

```

PROCEDURE SALIDA(his:arreglo;Var histo:grafico;nch,dur:integer);
Const
  tm = 0.041; (* Velocidad de muestreo mseg.*)
  Uv = -1.0; (* Ultimo valor del archivo *)
  Hv = 9.99999E5; (* Valor alto de comparacion *)
  (* para capturas. *)
Var
  Lb, (* Cuenta baja del tiempo *)
  i,j (* Variables indices. *)
  fecha : string[9]; (* Fecha del registro *)
  resp : char; (* Bandera para grabar info. *)
  doble : boolean; (* Aux. para valores > 32767 *)
  dureal, (* Auxiliar de la duracion *)
  Hb, (* Cuenta alta del tiempo *)
  eb, (* Cuenta los minutos *)
  ultiem : real; (* Ultimo valor calculado *)
  nomarch : string[20]; (* Nombre para archivo de *)
  (* datos (Nombre.dat) *)
  nuevo : text; (* Archivo de datos (salida) *)
Begin
  borra(15,6);
  write(' DESEAS SALVAR EN DISCO LA CAPTURA...(S/N)');
  readln(resp);
  If (resp<>'n') and (resp<>'N') then
    Begin
      doble:=false;
      i:=0; j:=0;
      histo[0]:=0.0;
      dureal:=dur*1000.0;
      Hb:=0;
      ultiem:=0.0;
      While (dureal>ultiem) do
        Begin
          Lb:=his[i];
          If (his[i+1]>0) then
            Hb:=his[i+1]*256.0
          else
            Hb:=0
          (* endif *);
          histo[j]:=(hb+lb)*tm;
          If (his[i+2]>0) then
            Begin
              eb:=(his[i+2]*65535.0);
              histo[j]:=(hb+lb+eb)*tm;
            end
          (* noelse *)
          (* endif *);
          i:=i+3;
          If ((histo[j]-ultiem) > 3*tm) then
            ultiem:=histo[j]
          else

```

```

Begin
    ultiem:=histo[j];          (* evita espigas eco *)
    j:=j-1;
end
(* endif *)
If (ultiem=0.0) and (j>2) then
Begin
    ultiem:=Hu;
    j:=j-1;
end
(* noelse *)
(* endif *);
j:=j+1;
end
(* enddo *);
writeln;writeln;
write('          Nombre del archivo a grabar...');
readln(nomarch);
writeln;writeln;
write('          *** FECHA DEL REGISTRO ***');
writeln;
write('          (dd-mmm-aa)...');
readln(fecha);
writeln;
write('          *** GRABANDO LA INFORMACION AL DISCO ***');
nomarch:=concat('#5:',nomarch,'.dat');
rewrite(nuevo,nomarch);
writeln(nuevo,fecha);
writeln(nuevo,nch);
writeln(nuevo,dur);
For i:=0 to j-1 do
    writeln(nuevo,histo[i])
(* endfor *);
writeln(nuevo,Uu);
close(nuevo,lock);
end
else
Begin
    writeln;writeln;writeln;
    write('          *** INFORMACION DESECHADA ***');
end
(* endif *);
end
(* END-SALIDA *);

```

```

(*****
*)
*) Rutina que despliega en pantalla el Titulo del Modulo *)
*)
*****

```

PROCEDURE TITULO;

```

Begin
    borra(14,3);
    writeln('          +-----+');
    Gotoxy(14,4);
    writeln('          |      MODULO      |');
    Gotoxy(14,5);
    writeln('          |  CAPTURAS DE ESPIGAS  |');
    Gotoxy(14,6);
    writeln('          +-----+');
end

```

```

(* END TITULO *);

```

```

(*****
(*)
(*) Rutina que hace las llamadas en el orden necesario a (*)
(*) a las demas rutinas del modulo. (*)
(*)
(*****

```

```

PROCEDURE TOMA_ESP(Var comando:char);

```

```

Var

```

```

    Nchan,                (* # de canales a capturar      *)
    dur                   (* Duracion del registro      *)
    his                   (* Contadores para los tiempos de *)
                           (* aparicion de espigas ;      *)
                           (* his[0] = segundos de captura *)
                           (* his[1] = minutos de captura  *)
    histo                 (* Tiempos de aparicion de espigas*)
    inicio,              (* Inicio de la captura        *)
    fin                   (* Fin de la captura           *)

```

```

Begin

```

```

    inicia(9000,his,histo);
    Entrada(nchan,dur,his,histo);
    Espera(inicio,'COMENZAR EL');
    While not(inicio) do
        Espera(inicio,'COMENZAR EL')
    (* enddo *)
    Captura(nchan,his);
    Salida(his,histo,nchan,dur);
    Espera(inicio,'REALIZAR UN NUEVO');
    If not(inicio) then
        comando:='t'
    (* noelse *)
    (* endif *)

```

```

end

```

```

(* END-TOMA_ESP *)

```

```

(*****XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
(*)
(*) Programa Principal del MODULO I (*)
(*)
(*****

```

```

Begin

```

```

    comando:=' ';
    While ((comando <> 'T') and (comando <> 't')) do
        Begin
            Titulo;
            Menu(comando);
            Case comando of
                'C','c' : toma_esp(comando);
                'T','t' : writeln('119','*** FIN DE LA CAPTURA ***');
            end
        end
    (* endcase *)

```

```

    end
    (* enddo *)
end.

```

RUTINA DE CAPTURA DE ESPIGAS

Esta rutina realizada en Lenguaje ensamblador del microprocesador 6502, forma parte del Modulo 1 (Modulo de Captura), como una rutina External que se encarga de obtener las espigas directamente de un conversor Analogico/Digital y Digital/Analogico y guardar el tiempo de aparicion de las mismas.

Las espigas son almacenadas en la pagina cero de la memoria de una Microcomputadora (en este caso la Apple IIe), de donde seran tomadas por el programa en Pascal por medio de un parametro de entrada/salida. Este parametro lo constituye un vector de 9000 elementos, lo cual hace un total de 9Kbytes de almacenamiento para espigas.

Para la captura se manejan contadores de milisegundos, segundos y minutos, dos de los cuales se cargan al inicio de la rutina (segundos y minutos) atraves de dos parametros de entrada a la rutina, que son los dos primeros elementos del vector endonde se almacenaran las espigas (his[0] e his[1]). El contador de milisegundos realiza una captura minima (en caso de que los parametros de entrada sean nulos) de 26 mseg.

La entrada de datos se sincroniza con un pulso que manda el usuario del medio que el elija, con una amplitud de pulso ≥ 1.25 V.

La entrada actual de datos es por el canal 1 del conversor Analogico/Digital, y el pulso de sincronia se lee por el canal 0 de el A/D. Solo se capturan datos positivos (las espigas son siempre positivas.

Entradas : Tiempo de duracion de la captura y el # de canales de toma de datos.

Salidas : Tiempos de aparicion de las espigas y una señal de monitoreo (salida analogica que se presenta en un osciloscopio y que representa cada uno de los datos (espigas) leidos.

Al final de la captura se graba un pulso de sincronia por el canal 2 del D/A al medio de reproduccion de datos de entrada utilizado (esto es opcional).

```
.MACRO POP ; Definicion de un Macro para ;
pla ;
sta %l ; la lectura y escritura del ;
pla ;
sta %l+1 ; stack de usuario. ;
.ENDM ;

.MACRO PUSH ; Se leen byte alto y bajo ;
lda %l+1 ;
pha ;
lda %l ;
pha ;
.ENDM
```

```

-----
;
;
;                               Programa Principal
;
;
-----

return      .PROC      CAPTUR,2 ; Dos parametros de entrada ;
dir         .equ      0      ; Direcc. de regreso a Pascal ;
           .equ      2      ; Direcc. del vector de espigas (his);
param1     .equ      4      ; his[0] <---- byte bajo cuenta seg. ;
ultimac    .equ      6      ; his[1] <---- byte alto cuenta min. ;
veces      .equ      8      ; # de canales de entrada de datos ;
segundo    .equ      0a     ; Ultima espiga presentada ;
sal        .equ      0c     ; Auxiliar para cuenta del tiempo ;
           ; Cuenta de segundos <---- his[0] ;
           ; Aux. para grabacion de pulso de ;
           ; sincronia de salida ;
           ; Obten direccion de regreso a Pascal ;
pop        return     ; Obten direccion del vector de datos;
pop        dir        ; Obten # de canales de entrada ;
pop        param1    ; Limpia localidad de ultima espiga ;
lda        #000      ; presentada, byte bajo y ;
sta        ultimac   ; byte alto. ;
sta        ultimac+1; Limpia aux. de contador de tiempo ;
sta        veces     ; Inicializa indice del vector de ;
ldy        #000      ; espigas. ;
           ; Carga cuenta de milisegundos ;
lda        #0ff      ; Cuenta minima de 26 milisegundos ;
sta        cuenta    ; Carga contador de segundos ;
lda        (dir),y   ; y lo almacena. ;
sta        segundo   ;
iny        ;
lda        (dir),y   ; Carga el contador de minutos ;
sta        minuto    ; y lo almacena. ;
ldy        #000      ; Inicializa indice de vector de ;
           ; espigas. ;
espera     lda        0c080+48. ; Espera pulso de sincronia por el ;
           ; canal 0 del conversor A/D ;
           ; Voltaje del pulso de Sincronia debe ;
           ; se mayor o igual a 1.25 Volts. ;
           ; Si es menor salta a esperarlo ;
loop       bmi        espera ;
           ; Arranca canal 1 de entrada ;
           ;
           ; Espera que termine la conversion ;
           ;
           ;
           ; Lee dato del canal 1 ;
           ; Espera que estabilize el dato ;
           ;
           ;

```

```

sta      0c080+49. ; Salida a monitoreo (osciloscopio) ;
cmp      #080      ; Ve si hubo dato (compara con 0 U.) ;
beq      continua ; Si no hubo dato, elimina lo leído ;
bmi      continua ; Si dato negativo, lo elimina ;
lda      ultimac   ; Guarda tiempo en mseg. ;
sta      (dir),y
iny
bne      novet     ; Recorre hasta 255 localidades ;
inc      dir+1     ; Incrementa apuntador de localidades ;
novet    lda      ultimac+1 ; Guarda tiempo en segundos ;
sta      (dir),y
iny
bne      tseg
inc      dir+1
tseg     lda      veces   ; Guarda tiempo en minutos ;
sta      (dir),y
iny
bne      continua
inc      dir+1
continua inc      ultimac   ; Incrementa contador de mseg. ;
bne      sigue     ; Salta si no overflow ;
inc      ultimac+1 ; Incrementa contador de segundos ;
bne      sigue     ; Salta si no overflow ;
inc      veces     ; Incrementa contador de minutos ;
sigue    dec      cuenta
bne      loop
dec      segundo   ; Salta a seguir tomando datos ;
bne      loop      ; mientras tiempo de captura > 0 ;
dec      minuto
bne      loop      ; (1us)*(123 ciclos) = 0.123 mseg. ;
; Con Bit Slice --> 0.123/3 = 41 us. ;
; Carga cuenta para grabar sincronia ;
graba    lda      #00f
sta      sal
lda      #0d0      ; Pulso de sinc. de 3.12 Volts ;
sta      0c080+50. ; Salida por canal 2 del D/A ;
nop
nop      ; Espera a que haga la conversion ;
nop
nop
nop
nop
sta      0c080+50. ; Escribe el pulso de sincronia ;
dec      sal
bne      graba
fin      push     param1 ; Regresa al stack valor de param. 1 ;
push     dir        ; Coloca en stack direcc. de vector ;
; de tiempos de aparicion de espigas. ;
push     return     ; Coloca en stack direcc. de regreso ;
; a Pascal. ;
; Regresa a Pascal. ;
cuenta  .word     0    ; Localidad del contador de mseg. ;
minuto  .word     0    ; Localidad del contador de minutos ;
.end

```

(+S+)

```
(*****  
(*  
(*          MODULO II.1 .- HISTOGRAMAS DE ESPIGAS          *)  
(*  
(*  
(*  
(*      Este es el ultimo modulo de los dos que conforman el *)  
(* Sistema de Analisis de la actividad electrica unitaria extra- *)  
(* celular. A partir de la captura hecha en el Modulo I de este *)  
(* Sistema, se presentan los resultados de dicho modulo a manera *)  
(* de Histogramas de Frecuencia o de Intervalos. *)  
(* *)  
(*      Entradas : Un archivo en disco conteniendo los tiempos *)  
(* ----- de aparicion de las espigas, fecha en que *)  
(*                se efectuo el registro y el tiempo de dura- *)  
(* cion del mismo ( estos 2 ultimos solo como datos para *)  
(* el usuario ). *)  
(* *)  
(*      Salidas : Son Histogramas de Frecuencia o de Interval- *)  
(* ----- los, en pantalla ( opcionalmente ) y un *)  
(*                archivo en disco conteniendo el Histograma *)  
(* generado. *)  
(* *)  
(*****
```

PROGRAM HISTOGRAMAS;

```
USES TURTLEGRAPHICS, (* Lib. que permite graficacion *)  
    BASES; (* Libreria con rutinas comunes. *)  
  
Const  
    tam_vec = 1700; (* Tamano del vector de tiempos *)  
  
Type  
    arreglo = packed array[0..tam_vec] of integer;  
    calculos = packed array[0..100,0..1] of integer;  
  
Var  
    opcion : char; (* Opcion deseada del Menu P. *)
```



```

(*****)
(*)
(*) Rutina que amplia zonas del Histograma en el eje X (*)
(*)
(*****)
PROCEDURE AMPLIACION(numbin:integer;Var inc,i,num_his,nh,ndiv,rango:integer);
Var
  ini, (* Inicio de la ampliacion *)
  fin : integer; (* Fin de la ampliacion *)
  auxini, (* Variables auxiliares pa-*)
  auxfin, (* ra el manejo de valores *)
  auxrango : real; (* mayores a 32767 *)
Begin
  writeln;
  writeln(' **** LIMITES DE LA AMPLIACION ****');
  writeln;
  write(' Inicio :');
  readln(auxini);
  writeln;
  write(' Fin :');
  readln(auxfin);
  ini:=trunc(auxini/10);
  fin:=trunc(auxfin/10);
  inc:=trunc((24*numbin)/(fin-ini));
  i:=trunc((ini/numbin)*10);
  num_his:=trunc((fin/numbin)*10)-1;
  If (i)0) then
    nh:=num_his-i
  else
    nh:=num_his
  (* endif *)
  ndiv:=trunc((fin-ini)/numbin);
  auxrango:=ini;
  rango:=trunc(auxrango);
end
(* END-AMPLIACION *);

```

```

(*****)
(*)
(*) Rutina que limpia los vectores a usar (*)
(*)
(*****)
PROCEDURE CEROS(Var vector:arreglo);
Var
  i : integer; (* Variable de iteracion *)
Begin
  For i:=0 to tam_vec do
    vector[i]:=0
  (* endfor *)
end
(* END-CEROS *);

```

```

(*****)
(*)
(* Rutina que presenta al usuario los datos del registro *)
(*)
(*****)

```

```
PROCEDURE DATOS( fecha:string; numcan,durac:integer);
```

```

Begin
  writeln('_____');
  writeln;
  writeln('          Fecha de Registro      ',fecha);
  writeln('          # de canales            ',numcan:2);
  writeln('          Tiempo de duracion      ',durac:4,' seg. ');
  writeln('_____');
end
(* END-DATOS *);

```

```

(*****)
(*)
(* Rutina intermedia para calcular el Histograma de Intervalos *)
(* Ordena los intervalos de acuerdo a su tamaño *)
(*)
(*****)

```

```
PROCEDURE EN_ORDEN(Var vecint:calculos; fin:integer);
```

```

Var
  aux1,          (* Variables auxiliares para el ordenamiento *)
  aux2,          (* por el metodo de la burbuja. *)
  i,             (* Variable para iteracion *)
  con : integer; (* Contador de cambios en el ordenamiento *)
Begin
  con:=1;
  While (con <> 0) do
    Begin
      con:=0;
      For i:=0 to fin-1 do
        If (vecint[i,1] > vecint[i+1,1]) then
          Begin
            con:=con+1;
            aux1:=vecint[i,0];
            aux2:=vecint[i,1];
            vecint[i,0]:=vecint[i+1,0];
            vecint[i,1]:=vecint[i+1,1];
            vecint[i+1,0]:=aux1;
            vecint[i+1,1]:=aux2;
          end
          (* noelse *)
          (* endif *)
        (* endfor *);
      end
    (* enddo *);
  end
(* END-EN_ORDEN *);

```

```

(*****
(*)
(*) Rutina que Escala a un valor real el eje Y (*)
(*)
(*****)

PROCEDURE ESCA(Var vec,vecmod:arreglo;Var escala:real;
               Var fin_dat,factor_esc:integer);

Var
  i : Integer; (* Variable de iteracion *)
Begin
  ceros(vecmod);
  For i:=0 to fin_dat do
    Begin
      vecmod[i]:=trunc((vec[i]/factor_esc)*100);
    end
  (* endfor *);
  escala:=factor_esc/10.0;
end
(* END-ESCA *);

```

```

(*****)
(*      Rutina que manda graficar la division numerica del eje X      *)
(*      utilizando notacion de Ingenieria                             *)
(*      *)
(*****)

```

```

PROCEDURE ESCALAH(Var ndiv, incre, rango: integer; binw: real);

```

```

Var
  i,                (* Variable para iteracion *)
  j,                (* Auxiliar del # de caracteres a graficar *)
  con,              (* Numero de caracteres a graficar *)
  cadabin,         (* Tamaño del bin o del intervalo *)
  expo,           (* Auxiliar del exponente del # a graficar *)
  inc,            (* Espacio entre marcas del eje X *)
  x : integer;   (* Espacio entre caracteres graficados *)
  car : charac;  (* Vector de caracteres a graficar *)
  exp : char;    (* Exponente del numero a graficar *)

```

```

Begin
  pencolor(none);
  moveto(0,0);
  x:=22;
  cadabin:=trunc(binw);
  inc:=cadabin;
  For i:=1 to ndiv+1 do
    Begin
      convierte(rango,con,car);
      j:=con;
      While (car[j]<>'0') do
        Begin
          moveto(x,32);
          wchar(car[j]);
          x:=x+7;
          j:=j-1;
        end
      (* enddo *)
      If (con=0) then
        Begin
          moveto(x,32);
          wchar('0');
          x:=x+2;
        end
      else
        Begin
          expo:=j+2;
          moveto(x,32);
          wchar('E');
          x:=x+5;
          moveto(x,32);
          exp:=chr(expo+ord('0'));
          wchar(exp);
        end
      (* endif *)
      x:=x+incr*10-16;
      rango:=rango+inc;
    end
  (* endfor *)
  rango:=rango-inc;
end
(* END-ESCALAH *)

```

```

(*****)
(*)
(*) Rutina que calcula los valores para graficar el Histograma (*)
(*) de Frecuencias a partir del bin deseado (*)
(*)
(*****)

```

```

PROCEDURE FRECUENCIA(Var binw:real;Var grafica:arreglo;Var fin:integer);
Const

```

```

  Uv = -1.0; (* Ultimo valor del archivo de entrada *)
Var
  bin, (* Tamaño del bin en milisegundos *)
  durtotal, (* Duracion del registro en miliseg. *)
  tiempo : real; (* Tiempo de aparicion de espigas mseg. *)
  i, (* Contador del # de espigas/bin *)
  numcan, (* Numero de canales de entrada *)
  durac, (* Duracion del registro en segundos *)
  J : integer; (* Indice del vector a graficar *)
  arch : text; (* Archivo de datos de entrada *)
  nomarch, (* Nombre del archivo de entrada *)
  fecha : string; (* Fecha en que efectuo el registro *)

```

```

Begin
  writeln;
  write(' ..... Tamano del bin (mseg) ..... ');
  readln(binw);
  zeros(grafica);
  fin:=0;
  j:=0;
  i:=0;
  bin:=binw;
  writeln;
  write(' ..... NOMBRE DEL ARCHIVO DE DATOS..... ');
  readln(nomarch);
  nomarch:=concat('#5:',nomarch,'.dat');
  reset(arch,nomarch);
  readln(arch,fecha);
  readln(arch,numcan);
  readln(arch,durac);
  datos(fecha,numcan,durac);
  tiempo:=0.0;
  durtotal:=durac*1000.0;
  While (tiempo >= 0) and (tiempo <= durtotal) do
    Begin
      While (tiempo <= bin) and (tiempo <> Uv) do
        Begin
          readln(arch,tiempo);
          i:=i+1;
        end
        (* enddo *);
        if (i=0) then
          grafica[j]:=i
        (* noelse *)
        (* endif *);
          grafica[j]:=i-1;
          i:=1;
          j:=j+1;
          bin:=bin+binw;
        end
        (* enddo *);
        fin:=j-1;
      end
    end
  (* END-FRECUENCIA *);

```

```

(*****
(*                                                                    *)
(*      Rutina que grafica en pantalla el Histograma elegido        *)
(*      en forma de barra simple o ashurado                         *)
(*                                                                    *)
(*****

PROCEDURE GENERA(Var a:arreglo;Var i, incre, num_his:integer;Var ashu:boolean);
Var
  x,                                     (* Coordenadas " x " y " y " de *)
  y : integer;                            (* la pantalla de graficacion *)
Begin
  If (ashu) then
    Begin
      y:=28;
      x:=y+incre;
    end
  else
    y:=incre
  (* endif *);
  moveto(27,45);
  While (i <= num_his) do
    Begin
      If (a[i] > 100) then      (* para valores escalados *)
        a[i]:=100
      (* noelse *);
      (* endif *);
      If (ashu) then
        Begin
          viewport(y,x,46,a[i]+46);
          fillscreen(reverse);
          y:=x;
          x:=x+incre;
        end
      else
        Begin
          move(a[i]); turn(-90);
          move(y);   turn(-90);
          move(a[i]); turn(180);
        end
      (* endif *);
      i:=i+1;
    end
  (* enddo *);
end
(* END-GENERA *);

```

```

(*****
(*)
(*) Rutina que permite la opcion de almacenar en disco el
(*) Histograma para analizarlo posteriormente mediante el
(*) Modulo III de este Sistema.
(*)
(*)
(*****

```

```

PROCEDURE GUARDA_HIST(fin_dat,numbin,inc,nh,ndiv:integer;escala:real;
que_hist:boolean);

```

```

Const
  Uv = -1; (* Ultimo valor que contendra el archivo *)
Var
  nomarch, (* Nombre del archivo de datos de salida *)
  fecha : string; (* Fecha de generacion del Histograma *)
  arch : text; (* Archivo de datos de salida --> Histog. *)
  k : integer; (* Variable de iteracion *)
  guarda : char; (* Bandera para guardar el Histograma *)
Begin
  writeln;
  write('..... Desea guardar el Histograma (S/N) ',
'.....');
  readln(guarda);
  If (guarda = 'S') or (guarda = 's') then
    Begin
      writeln;
      write('..... NOMBRE PARA EL HISTOGRAMA .....');
      readln(nomarch);
      nomarch:=concat('#5:',nomarch,'.his');
      rewrite(arch,nomarch);
      writeln;
      write(' *** Fecha de Grabacion ***');
      writeln(' (dd-mmm-aa)');
      write(' ':23);
      readln(fecha);
      writeln(arch,fecha);
      If (que_hist) then
        writeln(arch,'FRECUENCIA')
      else
        writeln(arch,'INTERVALOS')
      (* endif *)
      writeln(arch,numbin);
      writeln(arch,inc);
      writeln(arch,nh);
      writeln(arch,ndiv);
      writeln(arch,escala);
      For k:=0 to fin_dat do
        writeln(arch,vecmod[k])
      (* endfor *)
      writeln(arch,u);
      close(arch,lock);
    end
    (* noelse *)
  (* endif *)
end
(* END-GUARDA_HIST *)

```

```

(*****
(*)
(*) Rutina que calcula los valores para graficar el Histograma (*)
(*) de Intervalos a partir del intervalo dado (*)
(*)
(*****X)

PROCEDURE INTERVALO(Var tam_int:real;Var grafico:arreglo;Var fin:integer);
Const
  Uv = -1.0; (* Ultimo valor del archivo de entrada *)
Var
  durtotal, (* Duracion del registro en milisegundos *)
  tiempo, (* Tiempo de aparicion de espigas en mseg*)
  tiempoant: real; (* Tiempo de aparicion de la anterior *)
  ya : boolean; (* espiga en milisegundos. *)
  dif, (* Bandera de intervalo ya existente *)
  intervalo, (* Numero de intervalos diferentes *)
  inter, (* Tamano del intervalo calculado *)
  auxiti, (* Auxiliar para tamano de intervalo calc*)
  durac, (* Auxiliar para tamano de intervalo base*)
  numcan, (* Duracion del registro en segundos *)
  j,i : integer; (* Numero de canales de entrada *)
  vecint : calculos; (* Variables de iteracion *)
  arch : text; (* Vector aux. para calculo de intervalos*)
  nomarch, (* Archivo de datos de entrada *)
  fecha : string; (* Nombre del archivo de datos de entrada*)
  (* Fecha en que se realizo el registro *)
Begin
  writeln;
  write(' ..... Que amplitud de intervalo (ms) .....');
  readln(tam_int);
  auxiti:=trunc(tam_int);
  tiempo:=0.0;
  dif:=0;
  tiempoant:=0.0;
  ceros(grafico);
  For i:=0 to 100 do
    For j:=0 to 1 do
      vecint[i,j]:=0
    (* endfor *)
  (* endfor *)
  writeln;
  write(' ..... NOMBRE DEL ARCHIVO DE DATOS.....');
  readln(nomarch);
  nomarch:=concat('#5',nomarch,'.dat');
  reset(arch,nomarch);
  readln(arch,fecha);
  readln(arch,numcan);
  readln(arch,durac);
  datos(fecha,numcan,durac);
  durtotal:=durac*1000.0;

```



```

While (tiempo >= 0) and (tiempo<=durtotal) do
  Begin
    i:=0;
    ya:=false;
    readln(arch,tiempo);
    inter:=round(tiempo-tiempoant);
    intervalo:=trunc(inter/auxti);
    If ((inter mod auxti) <> 0) then
      intervalo:=intervalo+1
    (* noelse *)
    (* endif *);
    While (i<=dif) do
      If (vecint[i,1]=intervalo) then
        Begin
          vecint[i,0]:=vecint[i,0]+1;
          ya:=true;
          i:=dif+1;
        end
      else
        i:=i+1
      (* endif *)
    (* enddo *)
    If not(ya) then
      Begin
        vecint[dif,1]:=intervalo;
        vecint[dif,0]:=vecint[dif,0]+1;
        dif:=dif+1;
      end
    (* noelse *)
    (* endif *);
    tiempoant:=tiempo;
  end
  (* enddo *);
  fin:=dif-1;
  en_orden(vecint,fin);
  For i:=0 to fin do
    grafico[i]:=vecint[i,0]
  (* endfor *);
end
(* END-INTERVALO *);

```

```

(*****#)
(*                                     *)
(* Rutina que manda a la pantalla de graficacion todos los *)
(* letreros ( caracteres de graficacion ) de identificacion *)
(* del Histograma tales como : Tipo de Histograma, escala *)
(* horizontal y vertical, y divisiones numericas de los ejes *)
(*                                     *)
(*****#)

```

```

PROCEDURE LLAMA(Var rango,inc,ndiv:integer;binw,escala:real;
                tit,unidad:string);

```

```

Var
    leth2,                                     (* Segundo letrero horizontal *)
    letv : string;                             (* Letrero para el eje Y *)
Begin
    escalah(ndiv,inc,rango,binw);
    escalav(escala);
    leth2:='mseg';
    If (unidad='bin ') then
        letv:='Wesp'
    else
        letv:='#int'
    (* endif *);
    letreros(tit,unidad,leth2,letv);
end
(* END-LLAMA *);

```

```

(*****#)
(*                                     *)
(* Rutina que calcula la ocurrencia mas grande de espigas para *)
(* escalar el eje Y a dicho valor *)
(*                                     *)
(*****#)

```

```

PROCEDURE MAXIMO(Var vec:arreglo;Var fin_dat,num_max:integer);

```

```

Var
    i : integer;                               (* Variable de iteracion *)
Begin
    num_max:=vec[0];
    For i:=0 to fin_dat do
        Begin
            If (vec[i] > num_max) then
                num_max:=vec[i]
            (* noelse *)
            (* endif *);
        end
    (* endfor *);
end
(* END-MAXIMO *);

```

```

(*****
(*
(*          Rutina que presenta el Menu Principal del Modulo
(*
(*
(*****

```

```

PROCEDURE MENU_PRINC(Var opcion:char);
Begin
  gotoxy(0,0);espacio(23);
  gotoxy(0,0);espacio(2);
  writeln(' :24,'+-----+');
  writeln(' :24,'|          MODULO II          |');
  writeln(' :24,'| HISTOGRAMAS DE ESPIGAS I |');
  writeln(' :24,'+-----+');
  espacio(2);
  writeln(' :28,'MENU PRINCIPAL');
  writeln(' :28,'_____');
  espacio(2);
  writeln(' :23,'(F) recuencia Histograma');
  writeln;
  writeln(' :23,'(I) ntervalos Histograma');
  writeln;
  writeln(' :23,'(T) erminar');
  espacio(2);
  write(' :28,'OPCION: ');
  readln(opcion);
end
(* END-MENU_PRINC *);

```

```

(*****
(*
(*          Rutina para realizar modificaciones sobre un mismo Histograma
(*          o bien continuar a los siguientes Histogramas
(*
(*
(*****

```

```

PROCEDURE MISMO_HIST(numbin:integer;Var inc,i,num_his,nh,ndiv,rango:integer);
Var
  amplia : char;          (* Bandera de ampliacion de Histograma *)
Begin
  write('..... Desea ampliar alguna zona del Histograma ',
        '(S/N) ..... ');
  readln(amplia);
  If ((amplia='S') or (amplia='s')) then
    ampliacion(numbin,inc,i,num_his,nh,ndiv,rango)
  else
    Begin
      i:=num_his+1;
      num_his:=num_his+nh;
    end
  (* endif *);
end
(* END-MISMO_HIST *);

```

```

(*****
(*)
(*) Rutina que selecciona la escala a la que quedara el eje Y (*)
(*)
(*****)

PROCEDURE MOD_ESCALA(Var vec,vecmod:arreglo;Var fin_dat:integer;
                    Var escala:real);
Var
  op2      : char;          (* Bandera de escala maxima o dif. *)
  nummax,  : integer;      (* Base para la escala maxima *)
  escala_ent : integer;    (* Base para diferente escala *)
Begin
  menu_op2(op2);
  Case op2 of
    'm','M': Begin
      maximo(vec,fin_dat,nummax);
      esca(vec,vecmod,escala,fin_dat,nummax);
    end;
    'd','D': Begin
      writeln;
      write('A que escala...? ');
      readln(escala);
      escala_ent:=round(escala);
      esca(vec,vecmod,escala,fin_dat,escala_ent);
    end;
  end
  (* endcase *);
end
(* END-MOD_ESCALA *);

```

```

(*****)
(* *)
(* Rutina que manda todos los procesos de graficacion a *)
(* la pantalla de graficacion *)
(* *)
(*****)

PROCEDURE PANTALLA(opl:char;Var inc,ndiv,rango,i,num_his:integer;binw:real;
Var vecmod:arreglo;Var escala:real;tit,unidad:string);
Var
  cual : boolean; (*'Bandera del tipo de Histograma a *)
(* graficar, barra simple o asurado *)
Begin
  cual:=false;
  Case opl of
    'b','B': cual:=false;
    'a','A': cual:=true;
  end
  (* endcase *);
  initturtle;
  eje;
  lineas(inc,ndiv);
  If (cual) then
    Begin
      llama(rango,inc,ndiv,binw,escala,tit,unidad);
      genera(vecmod,i,inc,num_his,cual);
    end
  else
    Begin
      genera(vecmod,i,inc,num_his,cual);
      llama(rango,inc,ndiv,binw,escala,tit,unidad);
    end
  (* endif *);
end
(* END-PANTALLA *);

```

```

(*****)
(* *)
(* Rutina que manda a llamar a todas las demas rutinas en el *)
(* orden adecuado para su proceso *)
(* *)
(*****)

```

```

PROCEDURE PROCESO(Var binw:real;Var vec:arreglo;Var fin_dat:integer;
Var que_hist:boolean;Var tit,unidad:string);

```

```

Var
  ndiv, (* Numero de divisiones del eje X *)
  i, (* Indice del valor a partir del cual *)
(* se mandan a graficar las barras. *)
  inc, (* Incremento de puntos en eje X *)
  nh, (* # total de barras del Histograma *)
  num_his, (* # total de barras para una pantalla *)
  numbin, (* Tamano del bin o del intervalo *)
  rango : integer; (* Inicio de la grafica en eje X *)
  escala : real; (* Escala real del eje Y *)
  haya : boolean; (* Bandera de fin de la corrida *)
  opl, (* Que tipo de Histograma se obtendra *)
  respu : char; (* Bandera de cambio de bin o intervalo *)
  vecmod : arreglo; (* Vector a graficar *)

```

```

Begin
  numbin:=trunc(binw);
  menu_opl(opl,tit);
  while (opl <> 'S') and (opl <> 's') do
    Begin
      mod_escal(a(vec,vecmod,fin_dat,escala);
      haya:=true;
      i:=0;
      rango:=0;
      num_his:=80;
      ndiv:=8;
      inc:=3;
      nh:=80;
      While (haya) do
        Begin
          pantalla(opl,inc,ndiv,rango,i,num_his,binw,vecmod,escala,
            tit,unidad);
          If (resp) then
            Begin
              Textmode;
              haya:=false;
              guarda_hist(fin_dat,numbin,inc,nh,ndiv,escala,que_hist);
            end
          else
            Begin
              textmode;
              writeln;
              write('... Desea cambiar tamaño del ',unidad,' (S/N) ',
                '.....');
              readln(respu);
              If ((respu='S') or (respu='s')) then
                Begin
                  If (que_hist) then
                    frecuencia(binw,vec,fin_dat)
                  else
                    Intervalo(binw,vec,fin_dat)
                  (* endif *);
                  numbin:=trunc(binw);
                  haya:=false;
                end
              else
                mismo_hist(numbin,inc,i,num_his,nh,ndiv,rango)
                (* endif *);
              end
            (* endif *);
          end
        (* enddo *);
        textmode;
        menu_opl(opl,tit);
      end;
    (* enddo *);
  end
  (* END-PROCESO *);

```

```

(*****
*)
*) Rutina que prepara los parametros para el Histograma de
*) de Frecuencias
*)
*)
(*****

```

```

PROCEDURE PROC_FREQ;

```

```

Var
  binw      : real;          (* Tamano del bin *)
  fin_dat   : integer;      (* Total de datos leidos *)
  que_hist  : boolean;      (* Tipo de Histograma *)
  tit,      (* Letrero para el Histograma *)
  unidad    : string;       (* Unidad a manejar = bin *)
  vec       : arreglo;      (* Vector de datos de entrada *)

```

```

Begin
  frecuencia(binw,vec,fin_dat);
  que_hist:=true;          (* True para Frecuencias *)
  tit:= 'HISTOGRAMA DE FRECUENCIA';
  unidad:= 'bin';
  proceso(binw,vec,fin_dat,que_hist,tit,unidad);
end

```

```

(* END-PROC_FREQ *)

```

```

(*****
*)
*) Rutina que prepara los parametros para el Histograma de
*) Intervalos
*)
*)
(*****

```

```

PROCEDURE PROC_INT;

```

```

Var
  tam_int   : real;         (* Tamano del Intervalo base *)
  fin_dat   : integer;      (* Total de datos leidos *)
  que_hist  : boolean;      (* Tipo de Histograma *)
  tit,      (* Letrero para el Histograma *)
  unidad    : string;       (* Unidad a manejar = Intervalo *)
  vec       : arreglo;      (* Vector de datos de entrada *)

```

```

Begin
  intervalos(tam_int,vec,fin_dat);
  que_hist:=false;        (* False para Intervalos *)
  tit:= 'HISTOGRAMA DE INTERVALO';
  unidad:= 'int.';
  proceso(tam_int,vec,fin_dat,que_hist,tit,unidad);
end

```

```

* END-PROC_INT *

```

```

(*****)
(*                                           *)
(*           Programa Principal para el MODULO II           *)
(*                                           *)
(*****)

Begin
  menu_princ(opcion);
  While (opcion<>'T') and (opcion<>'t') DO
    Begin
      Case opcion of
        'f','F': proc_frec;
        'I','i': proc_int;
      end
      (* endcase *);
      menu_princ(opcion);
    end
    (* enddo *);
end.

```



```
(*****)  
(*  
(*          Programa Principal para el MODULO 11          *)  
(*  
(*****)
```

```
Begin  
  menu_princ(opcion);  
  While (opcion<>'T') and (opcion<>'t') DO  
    Begin  
      Case opcion of  
        'f','F': proc_freq;  
        'I','i': proc_int;  
      end  
      (* endcase *);  
      menu_princ(opcion);  
    end  
  end  
  (* enddo *);  
end.
```

(**S**)

```
(*****  
(*  
(*          MODULO II.1.1.- DESPLIEGUE DE HISTOGRAMAS          *)  
(*  
(*          _____          *)  
(*          Este es el complemento del modulo II.1 ( modulo de *)  
(*          de histogramas ) encargado de sacar los resultados de este *)  
(*          a papel, a partir de un archivo en disco.          *)  
(*          *)  
(*          Entradas : Un archivo en disco conteniendo el numero *)  
(*          ----- de espigas que aparecio en cada bin (his *)  
(*          grama de frecuencias) o bien el numero de *)  
(*          intervalos que aparecio en cada bin.          *)  
(*          *)  
(*          Salidas : Histogramas de Frecuencia o de Intervalos *)  
(*          ----- en pantalla y/o en papel.          *)  
(*          *)  
(***** : *****
```

PROGRAM RECONSTRUYE;

```
USES UCALIW,          (* Libreria para el paso de pantalla a *)  
                    (* a papel. *)  
    TURTLEGRAPHICS,  (* Libreria que permite la graficacion *)  
    BASES;           (* Libreria con rutinas para graficar *)
```

```
Const  
    tam_vec = 500;    (* Tamano del vector de # de espigas o *)  
                    (* intervalos. *)
```

```
Type  
    grafico = packed array[0..tam_vec] of integer;
```

```
Var  
    opcion : char;    (* Opcion deseada del Menu Principal *)  
    histog : grafico;
```

```

(*****)
(*)
(*) Rutina que manda graficar la division numerica del eje X (*)
(*) utilizando notacion de Ingenieria. (*)
(*)
(*****)

```

```

PROCEDURE ESCALAH(Var ndiv, incre, rango: integer; numbin: integer);

```

```

Var
  i, (* Variable para iteración *)
  j, (* Auxiliar del # de caracteres a graficar *)
  con, (* Numero de caracteres a graficar *)
  expo, (* Auxiliar del exponente del # a graficar *)
  inc, (* Espacio entre marcas del eje X *)
  x : integer; (* Espacio entre caracteres graficados *)
  car : char; (* Vector de caracteres a graficar *)
  exp : char; (* Exponente del numero a graficar *)
Begin
  pencolor(none);
  moveto(0,0);
  x:=22;
  inc:=numbin;
  For i:=1 to ndiv+1 do
    Begin
      convierte(rango,con,car);
      j:=con;
      While (car[j]<>'0') do
        Begin
          moveto(x,32);
          wchar(car[j]);
          x:=x+7;
          j:=j-1;
        end
      (* enddo *);
      If (con=0) then
        Begin
          moveto(x,32);
          wchar('0');
          x:=x+2;
        end
      else
        Begin
          expo:=j+2;
          moveto(x,32);
          wchar('E');
          x:=x+5;
          moveto(x,32);
          exp:=chr(expo+ord('0'));
          wchar(exp);
        end
      (* endif *);
      x:=x+incre*10-16;
      rango:=rango+inc;
    end
  (* endfor *);
  rango:=rango-inc;
end
(* END-ESCALAH *);

```

```

(*****)
(*
(* Rutina que grafica en pantalla el Histograma elegido *)
(* en forma de barra simple o ashurado. *)
(*
(*
(*****)

```

```

PROCEDURE GENH(Var aigrafico;Var i, incre, num_his:integer;Var ashur:boolean);
Var
  x, (* Coordenadas " x " y " y " de *)
  y : integer; (* la pantalla de graficacion *)
Begin
  If (ashur) then
    Begin
      y:=28;
      xi:=y+incre;
    end
  else
    y:=incre
  (* endif *);
  moveto(27,45);
  While (i <= num_his) do
    Begin
      If (a[i] > 100) then (* para valores escalados *)
        a[i]:=100
      (* noelse *)
      (* endif *);
      If (ashur) then
        Begin
          viewport(y,x,46,a[i]+46);
          fillscreen(reverse);
          y:=x;
          xi:=x+incre;
        end
      else
        Begin
          move(a[i]); turn(-90);
          move(y); turn(-90);
          move(a[i]); turn(180);
        end
      (* endif *);
      i:=i+1;
    end
  (* enddo *);
end
(* END-GENH *);

```

```

(*****
(*
(*          Rutina que imprime en papel el Histograma desplegado          *)
(*
(*
(*****

PROCEDURE IMPRIME;
Var
  pasa : char;
Begin
  Textmode;
  writeln;writeln;
  write(' ..... Desea pasar el Histograma a papel (S/N) .....');
  readln(pasa);
  If (pasa = 'S') or (pasa = 's') then
    Begin
      Calca(18);
      rewrite(gra,'printer');
      write(gra,chr(27),chr(99));
      close(gra);
    end
  (* noelse *)
  (* endif *)
  .Initturtle;
end
(* END-IMPRIME *);

```

```

(*****
(*
(*          Rutina que limpia el vector de datos          *)
(*
(*
(*****

```

```

PROCEDURE INICIA(Var histog:grafico);
Var
  j : Integer;
Begin
  For j:=0 to tam_vec do
    histog[j]:=0
  (* endfor *);
end
(* END-INICIA *);

```

```

(*****
(*
(*          Rutina de Lectura del Histograma          *)
(*
(*
(*****

PROCEDURE LEE(Var histog:grafico;Var fin_dat,numbin,inc,i,num_his,
              ndiv:integer;Var escala:real;Var tipo:string);

Const
  Uv = -1;          (* Valor de fin de archivo          *)
Var
  tit,              (* Tipo de Histograma          *)
  fecha,           (* Fecha en que se grabo el Histograma *)
  nomarch : string; (* Nombre del archivo en disco      *)
  arch : text;     (* Archivo que contiene Histograma  *)
  k,             (* Variable de iteracion          *)
  dato : integer;  (* Tamano de barra leido          *)
Begin
  writeln;writeln;
  write(' ..... QUE HISTOGRAMA A ANALIZAR .....');
  readln(nomarch);
  nomarch:=concat('#5:',nomarch,'.his');
  reset(arch,nomarch);
  k:=0;
  fin_dat:=0;
  dato:=0;
  readln(arch,fecha);
  readln(arch,tipo);
  readln(arch,numbin);
  readln(arch,inc);
  readln(arch,num_his);
  readln(arch,ndiv);
  readln(arch,escala);
  writeln(' _____');
  writeln(' | _____ |');
  writeln(' | HISTOGRAMA DE ',tipo,' DEL >> ',fecha:10,' |');
  writeln(' | _____ |');
  While (dato < Uv) do
    Begin
      readln(arch,dato);
      histog[k]:=dato;
      k:=k+1;
    end
    (* enddo *);
  fin_dat:=k-1;
end
(* END-LEE *);

```

```

(*****
(*)
(*) Rutina que manda a la pantalla de graficacion todos los (*)
(*) letreros (caracteres de graficacion) de identificacion (*)
(*) del Histograma tales como : Tipo de Histograma, escalas (*)
(*) horizontal y vertical, y divisiones numericas de los (*)
(*) ejes. (*)
(*)
(*****

PROCEDURE LLAMA(Var rango,inc,ndiv:integer;numbin:integer;escala:real;
                tipo:string);
Var
  letp,                (* Letrero principal del histograma *)
  leth,                (* Letrero para el eje X *)
  leth2,               (* Segundo letrero horizontal *)
  letv : string;      (* Letrero para el eje Y *)
Begin
  escalah(ndiv,inc,rango,numbin);
  escalav(escala);
  letp:=concat('HISTOGRAMA DE ',tipo);
  leth2:='mseg';
  If (tipo='FRECUENCIA') then
    Begin
      leth:='bin ';
      letv:='#esp';
    end
  else
    Begin
      leth:='int.';
      letv:='#int';
    end
  (* endif *);
  letreros(letp,leth,leth2,letv);
end
(* END-LLAMA *);

```

```

(*****A*****
(*)
(*) Rutina que presenta el menu principal del modulo (*)
(*)
(*****A*****

```

```

PROCEDURE MENU_PRINC(Var opcion:char);
Begin
  gotoxy(0,0);espacio(23);
  gotoxy(0,0);espacio(2);
  writeln(' :23,'+-----+');
  writeln(' :23,'| MODULO II.1.1 |');
  writeln(' :23,'| DESPLIEGUE DE HISTOGRAMAS |');
  writeln(' :23,'+-----+');
  espacio(2);
  writeln(' :29,'MENU PRINCIPAL');
  writeln(' :29,'-----');
  espacio(2);
  writeln(' :23,'(D) espiega Histograma');
  writeln;
  writeln(' :23,'(T) erminar');
  espacio(2);
  write(' :30,'OPCION: ');
  readln(opcion);
end
(* END-MENU_PRINC *);

```

```

(*****A*****
(*)
(*) Rutina que manda todos los procesos de graficacion a (*)
(*) la pantalla de graficacion (*)
(*)
(*****A*****

```

```

PROCEDURE PANTALLA(opi:char;Var inc,ndiv,rango,i,num_his,numbin:integer;
  Var histog:grafico;Var escala:real;tipo:string);
Var
  cual : boolean; (* Bandera para aschurar histograma *)
Begin
  cual:=false;
  Case opi of
    'b','B': cual:=false;
    'a','A': cual:=true;
  end
  (* endcase *);
  initturtle;
  eje;
  lineas(inc,ndiv);
  If (cual) then
    Begin
      llama(rango,inc,ndiv,numbin,escala,tipo);
      genh(histog,i,inc,num_his,cual);
    end
  else
    Begin
      genh(histog,i,inc,num_his,cual);
      llama(rango,inc,ndiv,numbin,escala,tipo);
    end
  (* endif *);
end
(* END-PANTALLA *);

```



```

(*****
(*)
(*)          Rutina principal de despliegue del histograma          (*)
(*)
(*)
(*****

PROCEDURE PRO_DESPLIEGA(Var histog:grafico);
Var
  opl,                                (* Bandera de despliegue continuo *)
  si_ver : char;                       (* Bandera para ver el histograma *)
  ndiv,                                (* Numero de divisiones del mismo *)
  i,                                    (* Indice para barras del histog. *)
  inc,                                  (* Incremento de puntos en eje X *)
  nh,                                   (* # total de barras a graficar *)
  num_his,                              (* # total de barras en una pantalla *)
  numbin,                               (* Tamano del bin o del intervalo *)
  fin_dat,                              (* Numero total de datos leidos *)
  rango : integer;                     (* Inicio de la grafica en eje X *)
  escala,                               (* Escala real del eje Y *)
  binw : real;                          (* Tamano real del bin o intervalo *)
  haya : boolean;                       (* Bandera de fin de la corrida *)
  tipo,                                 (* Tipo de histograma a desplegar *)
  tit : string;                         (* Titulo para graficar en pantalla *)
Begin
  inicia(histog);
  lee(histog,fin_dat,numbin,inc,i,num_his,ndiv,escala,tipo);
  nh:=num_his;i:=0;
  rango:=0;
  tit:=concat('HISTOGRAMA DE ',tipo);
  menu_opl(opl,tit);
  While (opl <> 'S') and (opl <> 's') do
    Begin
      haya:=true;
      While (haya) do
        Begin
          pantalla(opl,inc,ndiv,rango,i,num_his,numbin,histog,escala,
                    tipo);
          If (resp) then
            haya:=false
          else
            Begin
              imprime;
              i:=num_his+i;
              num_his:=num_his+nh;
            end
            (* endif *);
          end
          (* enddo *);
        textmode;
        menu_opl(opl,tit);
      end
    end
  (* enddo *);
end
(* END-PRO_DESPLIEGA *);

```

```

(*****)
(*)
(*)          Programa Principal          (*)
(*)
(*)
(*****)

Begin
  menu_princ(opcion);
  While (opcion<>'T') and (opcion<>'t') DO
    Begin
      Case opcion of
        'D','d': pro_despliega(histog);
      end
      (* endcase *);
      menu_princ(opcion);
    end
    (* enddo *);
  end.

```

(**S**)

```
(*****  
(*  
(*          MODULO II.2.- ANALISIS DE TRENES DE ESPIGAS          *)  
(*  
(*          _____          *)  
(*  
(*          Este es el ultimo modulo de los dos que conforman el *)  
(*          Sistema de Analisis de la actividad electrica unitaria extra *)  
(*          celular. El analisis se hace a partir de los resultados obte *)  
(*          nidos en el modulo II.1 (Histogramas de espigas). Sobre una *)  
(*          o mas zonas del Histograma se ajusta por el metodo de mini- *)  
(*          mos cuadrados a una funcion especifica, que el usuario elige *)  
(*          previamente. *)  
(*          *)  
(*          Entradas : Archivo en disco conteniendo el Histograma *)  
(*          ----- ( Frecuencias o Intervalos ).Opcionalmente *)  
(*          un numero de trenes de espigas diferente *)  
(*          de uno, para realizar el analisis sobre la suma de *)  
(*          los trenes. Por ultimo, la zona del histograma sobre *)  
(*          la que se hara el analisis. *)  
(*          *)  
(*          Salidas : Tabla de estadisticas basicas sobre una zo *)  
(*          ----- na del Histograma, conteniendo varianza, *)  
(*          desviacion standard, coeficiente de varia- *)  
(*          cion,y moda. Grafica de la recta teorica obtenida por *)  
(*          metodo de minimos cuadrados, en pantalla y (opcion- *)  
(*          mente en papel), y grafica de ajuste a curva teorica *)  
(*          en pantalla y (opcionalmente) papel. Ademas si se us- *)  
(*          cogio la opcion de analizar una suma de trenes de es- *)  
(*          pigas, se puede almacenar el resultado de la suma a *)  
(*          manera de histograma, en disco. *)  
(*          *)  
(*****
```

PROGRAM AJUSTES;

```
USES UCALIW,          (* Libreria para graficacion en papel *)  
    TURTLEGRAPHICS, (* Libreria para graficacion en pantalla *)  
    BASES,           (* Libreria extra para graficacion " *)  
    TRANSCEND;      (* Libreria con funciones trascendentales**
```

```
Const  
    tam_vec = 120;    (* Tam. del vector para barras del Histo.**)
```

```
Type  
    grafico = packed array[0..tam_vec] of real;
```

```
Var  
    histog : grafico; (* Vector conteniendo histograma *)  
    opcion : char;    (* opcion del menu principal *)
```

```

(*****)
(*                                     *)
(*      Rutina que obtiene la moda en estadísticas básicas      *)
(*                                     *)
(*****)

```

```

PROCEDURE ARREGLA(histog:grafico;Var cero,xpuntos,moda:integer);
Var
  i      : integer;      (* Índice de datos en región de ajuste *)
  max    : real;        (* # máximo de eventos = moda *)
Begin
  max:=histog[cero];
  For i:=cero to xpuntos-1 do
    If (histog[i] > max) then
      max:=histog[i]
    (* endif *)
  (* endfor *)
  moda:=round(max);
end
(* END-ARREGLA *);

```

```

(*****)
(*                                     *)
(*      Rutina que coloca las subdivisiones numéricas del eje X   *)
(*                                     *)
(*****)

```

```

PROCEDURE COLOCA(Var con,x,y:integer;Var car:carac);
Var
  i      : integer;      (* Índice del vector de caracteres a graficar *)
Begin
  i:=con;
  While (con >= 0) do
    Begin
      moveto(x,y);
      wchar(car[i]);
      x:=x+6;
      i:=i-1;
      con:=con-1;
    end
  (* enddo *)
end
(* END-COLOCA *);

```

```

(*****
(*)
(*) Rutina que presenta la tabla de estadísticas básicas del (*)
(*) ajuste (*)
(*)
(*****)
PROCEDURE CUADRO(media,desvstd,coefv:real;moda,inicio,fin:integer);
Begin
  espacio(2);
  writeln(' :14, ' _____ ');
  writeln(' :14, | _____ | ');
  writeln(' :14, | TABLA DE ESTADÍSTICAS BÁSICAS | ');
  writeln(' :14, | en los límites : | ');
  writeln(' :14, | , ' :11,Inicio:4,'0', ' :14,'a ' ,fin:4,'0', ' :13,
    ' ms. | ');
  writeln(' :14, | _____ | ');
  writeln(' :14, | _____ | ');
  writeln(' :14, | Media Aritmética ,media:12, ' | ');
  writeln(' :14, | Desviación Standard ,desvstd:12, ' | ');
  writeln(' :14, | Moda ,moda:12, ' | ');
  writeln(' :14, | Coef. de Variación ,coefv:12, ' | ');
  writeln(' :14, | _____ | ');
  writeln;
end
(* END-CUADRO *);

```

```

(*****)
(*
(* Rutina que manda graficar las divisiones numericas del eje *)
(* cartesiano para la recta teorica, en los 4 cuadrantes *)
(* *)
(*****)

```

```

PROCEDURE EJEXY(b:integer;flagx:boolean);

```

```

Var
  increm,          (* Incremento en la subdivision *)
  inc,             (* Auxiliar del incremento *)
  J,              (* Variable de iteracion *)
  x,y,           (* Coordenadas en la pantalla grafica *)
  hasta,        (* # de divisiones numericas a graficar *)
  signo,        (* Signo del numero a graficar *)
  con          : integer; (* # de caracteres a graficar *)
  car          : carac;   (* Vector de caracteres a graficar *)

```

```

Begin
  If (b=1) or (b=2) then
    b:=3
  else
    If (b=-1) then
      b:=-2
    (* endif *)
  (* endif *)
  If flagx then
    Begin
      x:=160;
      y:=82;
      hasta:=3;
      signo:=4;
    end
  else
    Begin
      x:=145;
      y:=108;
      hasta:=4;
      signo:=6;
    end
  (* endif *)
  If (b>0) then
    (* eje positivo=3 divisiones *)
    If flagx then
      increm:=round(b/3)*2
    else
      increm:=round(b/3)
    (* endif *)
  else
    (* eje negativo=2 divisiones *)
    If flagx then
      increm:=round(abs(b)/2)*2
    (* division de eje X = doble Y *)
    else
      increm:=round(abs(b)/2)
    (* endif *)
  (* endif *)

```

```

inci:=increm;
For j:=1 to hasta do
  Begin
    convierte(increm,con,car);
    coloca(con,x,y,car);
    If flagx then
      x:=x+20
    else
      Begin
        x:=145;
        y:=y+14;
      end
    (* endif *);
    increm:=increm+inci;
  end
  (* endfor *);
  If flagx then
    Begin
      increm:=increm-inc;
      x:=30;
    end
  else
    Begin
      increm:=increm-(2*inc);
      y:=43;
    end
  (* endif *);
  For j:=1 to 3 do
    Begin
      convierte(increm,con,car);
      If not(flagx) then
        x:=147
      (* endif *);
      moveto(x,y+signo);pencolor(white);
      move(2);x:=x+3;pencolor(none);
      coloca(con,x,y,car);
      If flagx then
        x:=x+20
      else
        y:=y+16
      (* endif *);
      increm:=increm-inc;
    end
  (* endfor *);
end
(* END-EJEXY *);

```

```

(*****)
(*
(* Rutina que manda graficar la division numerica del eje X
(* en segundos, para curva teorica.
(*
(*
(*****)

```

```

PROCEDURE ESCALAH(inicio,binseg:real;ceros,xpuntos:integer);

```

```

Var
  i, (* Variable de iteracion *)
  con, (* # de caracteres a graficar *)
  nceros, (* # de ceros del caracter a graficar *)
  cadabin, (* # a graficar *)
  origen, (* Origen de la grafica o inicio *)
  x : integer; (* Coordenada x de pantalla de graf. *)
  car : carac; (* Arreglo que contiene todos los *)
  abciza, (* caracteres del # a graficar *)
  inc : real; (* Valor real de c/division del eje *)
  (* Tamano del bin o del intervalo *)
Begin
  penclor(none);
  x:=26;
  moveto(x,33);
  origen:=round(cero*binseg);
  convierte(origen,con,car);
  While (con)=0 do
    Begin
      wchar(car[con]);
      con:=con-1;
    end
    (* enddo *);
    x:=66;
    moveto(x,33);
    abciza:=(binseg*((xpuntos-cero)/5))+origen; (* grafica 5 divisiones*)
    inc:=abciza-origen;
    For i:=1 to 5 do
      Begin
        nceros:=0;
        While (abciza<1.0) do (* caso de division en mseg. *)
          Begin
            abciza:=abciza*10;
            nceros:=nceros+1;
          end
          (* enddo *);
          cadabin:=trunc(abciza);
          convierte(cadabin,con,car);
          If (nceros <> 0) then
            Begin
              wchar('.');
              While (nceros>1) do
                Begin
                  wchar('0');
                  nceros:=nceros-1;
                end
                (* enddo *);
              end
            end
            (* endif *);
          While (con)=0 do
            Begin
              wchar(car[con]);
              con:=con-1;
            end
          end
          (* enddo *);

```



```

        x:=x+40;
        moveto(x,33);
        abciza:=abciza+inc;
    end
    (* endfor *);
end
(* END-ESCALAH *);

```

```

(*****
(*
(* Rutina que graba un archivo en disco conteniendo la suma
(* de varios trenes de espigas
(*
(*
(*****

```

```

PROCEDURE ESCRIBE(histog:grafico;fin_dat,numbin,inc,num_his,ndiv:integer;
    escala:real;fecha,tipo:string);

```

```

Const
    Uv = -1; (* Ultimo valor del archivo de datos *)

```

```

Var
    guarda : char; (* Bandera para guardar suma de trenes *)
    nomarch : string; (* Nombre con que quedara el archivo *)
    arch : text; (* Arch. de salida con tren de espigas *)
    j,
    dato : integer; (* Indice de los datos a escribir *)
    (* c/u de los valores del tren *)

```

```

Begin
    writeln;
    write('... Desea almacenar la suma de TRENES ...(S/N)..');
    readln(guarda);
    If (guarda='s') or (guarda='S') then
        Begin
            writeln;
            write('... NOMBRE DEL ARCHIVO DE TRENES .....');
            readln(nomarch);
            nomarch:=concat('#5:',nomarch, '.his');
            rewrite(arch,nomarch);
            writeln(arch,fecha);
            writeln(arch,tipo);
            writeln(arch,numbin);
            writeln(arch,inc);
            writeln(arch,num_his);
            writeln(arch,ndiv);
            writeln(arch,escala);
            For j:=0 to fin_dat do
                Begin
                    dato:=round(histog[j]/escala)*10;
                    writeln(arch,dato);
                end
            end
            (* endfor *);
            writeln(arch,Uv);
            close(arch,lock);
        end
    end
    (* endif *);
end
(* END-ESCRIBE *);

```

```

(*****)
(*                                           *)
(*   Rutina que calcula las estadísticas básicas en la región   *)
(*           de ajuste                                           *)
(*                                           *)
(*****)

PROCEDURE ESTADIS(histograma:grafico;Var auxini,auxfin:real;Var numbin,
                 cero,xpuntos,moda:integer);

Var
n,                               (* # de datos para el ajuste *)
l,                               (* Índice del vector de datos a ajustar *)
inicio,                          (* Inicio del ajuste *)
fin,                             (* Fin del ajuste *)
bin      : integer;             (* Tamaño del bin o intervalo *)
media,                             (* Media aritmética *)
coefv,                             (* Coeficiente de variación *)
desvstd  : real;                (* Desviación standard *)
tecla    : string;             (* Continuar el proceso *)

Begin
desvstd:=0;
media:=0;moda:=0;
Gotoxy(0,0);
espacio(23);
gotoxy(13,3);
writeln('    *** LIMITES PARA EL AJUSTE ***');
writeln;
write(' 125, 'Inicio : ');
readln(auxini);
writeln;
write(' 125, 'Fin   : ');
readln(auxfin);
n:=0;
inicio:=trunc(auxini/10);
fin:=trunc(auxfin/10);
bin:=trunc(numbin/10);
xpuntos:=trunc(fin/bin);
cero:=trunc(inicio/bin);
For i:=cero to xpuntos-1 do
  media:=histog[i]+media
(* endfor *);
n:=xpuntos-cero;
media:=media/n;
For i:=cero to xpuntos-1 do
  desvstd:=(histog[i]-media)*(histog[i]-media)+desvstd
(* endfor *);
desvstd:=sqrt(desvstd/n);
coefv:=desvstd/media;
arregla(histograma,cero,xpuntos,moda);
cuadro(media,desvstd,coefv,moda,inicio,fin);
auxini:=auxini/1000;          (* trabajar en segundos *)
auxfin:=auxfin/1000;
writeln;
write(' 110, ' --) PARA CONTINUAR OPRIMA < RETURN > (--');
readln(tecla);
end
(* END-ESTADIS *)

```

```

(*****)
(*
(*      Rutina que dibuja las flechas del eje cartesiano para      *)
(*      la recta teorica                                           *)
(*
(*
(*****)

```

```

PROCEDURE FLECHAS;
Begin
  turn(-90); move(2); turn(120);
  move(4); turn(120); move(4);
  turn(120); move(2);
end
(* END-FLECHAS *);

```

```

(*****): (*
(*
(*      Rutina que gradua los 4 cuadrantes del eje cartesiano      *)
(*
(*
(*****)

```

```

PROCEDURE GRADUA;
Var
  i:integer;          (* Variable de iteracion *)
Begin
  turn(-90);move(5);
  For i:=1 to 8 do
    Begin
      turn(-90); move(3);turn(180);
      move(3); turn(-90);move(15);
    end
    (* endfor *);
    turn(180); pencolor(none);
    move(75); turn(90);
    move(133); turn(180);
    pencolor(white);
    move(1);
    For i:=1 to 8 do
      Begin
        turn(90); move(3);
        turn(180);move(3);
        turn(90); move(33);
      end
      (* endfor *);
    end
  end
  (* END-GRADUA *);

```

```

(*****
(*)
(*)          Rutina que grafica el eje cartesiano          (*)
(*)
(*****

```

```

PROCEDURE GRA_EJEXY;

```

```

Begin
  move(132); pencolor(white); flechas;
  turn(-90); move(264);      flechas;
  turn(-90); pencolor(none); move(132);
  turn(90); move(76);       pencolor(white);
  flechas; turn(-90);      move(126);
  flechas;
end
(* END-GRA_EJEXY *);

```

```

(*****
(*)
(*)          Rutina que dibuja la pantalla para la recta teorica          (*)
(*)
(*****

```

```

PROCEDURE GRAF(b,ang:integer);

```

```

Begin
  Initturtle;
  gra_ejexy;
  gradua;
  letreros('AJUSTE POR MINIMOS CUADRADOS',' ',' ',' ' y');
  moveto(0,0); moveto(1,41);
  wstring(' ');moveto(7,41);
  wstring(' ');moveto(10,41);
  wstring(' ');moveto(257,84);
  wstring(' x');
  ejexy(b,false);
  ejexy(b,true);
  Pencolor (none);
  If (b=1) then
    moveto(136,113)
  else
    If (b=2) then
      moveto(136,127)
    else
      If (b=-1) then
        moveto(136,86)
      else
        If (b>0) then
          moveto(136,137)
        else
          moveto(136,68)
        (* endif *)
      (* endif *)
    (* endif *)
  (* endif *);
  pencolor(white); wstring('o');
  turn(ang); move(60);
  turn(180); move(150);
end
(* END-GRAF *);

```

```

(*****)
(*)
(*) Rutina que imprime en papel los resultados de (*)
(*) la recta y la curva teorica (*)
(*)
(*****)

```

```

PROCEDURE IMPRIME;
Var
  pasa : char;
Begin
  Textmode;
  writeln;writeln;
  write(' ..... Desea pasar la grafica a papel (S/N) .....');
  readln(pasa);
  If (pasa = 'S') or (pasa = 's') then
    Begin
      Calca(18);
      rewrite(gra,'printer:');
      write(gra,chr(27),chr(99));
      close(gra);
    end
  (* endif *);
end
(* END-IMPRIME *);

```

```

(*****)
(*)
(*) Rutina que inicializa a ceros el vector de datos (*)
(*)
(*****)

```

```

PROCEDURE INICIA(Var histog:grafico);
Var
  j : integer;
Begin
  For j:=0 to tam_vec do
    histog[j]:=0
  (* endfor *);
end
(* END-INICIA *);

```

```

(*****
(*
(* Rutina que obtiene los datos del histograma a analizar *)
(*
(*
(*****

PROCEDURE LEE(Var histog:grafico;Var numbin:integer);
Const
  Uv = -1; (* Ultimo valor del archivo de datos *)
Var
  tit,tipo, (* Tipo de histograma (Frec. o Int.) *)
  fecha, (* Fecha en que se grabo el histograma*)
  nomarch : string; (* Nombre del archivo de datos *)
  mas : char; (* Bandera para analizar mas de 1 tren*)
  arch : text; (* Archivo con los datos de entrada *)
  k, (* Indice de vector datos de entrada *)
  fin_dat, (* # total de datos leidos *)
  varios, (* # de trenes a analizar *)
  j, (* Variable de iteracion *)
  inc, (* separacion entre divisiones *)
  num_his, (* # de barras del histograma *)
  ndiv, (* # de divisiones del histograma *)
  dato : integer; (* Cada barra del histograma leido *)
  escala, (* Valor de escala de los datos *)
  esc : real; (* Prom. de escalas en suma de trenes *)
  muchos : boolean; (* Bandera de grabar suma de trenes *)

Begin
  varios:=1;
  esc:=0;
  muchos:=false;
  writeln;write('... Desea analizar mas de un TREN ...(S/N)...');
  readln(mas);
  If (mas='s') or (mas='S') then
    Begin
      write('... # de TRENES .....');
      readln(varios);
      muchos:=true;
    end
  (* endif *);
  For i:=1 to varios do
    Begin
      writeln;
      write('... NOMBRE DEL TREN # ',i,'.....');
      readln(nomarch);
      nomarch:=concat('#5:',nomarch,'.his');
      reset(arch,nomarch);
      k:=0;dato:=0;
      readln(arch,fecha);
      readln(arch,tipo);
      readln(arch,numbin);
      readln(arch,inc);
      readln(arch,num_his);
      readln(arch,ndiv);
      readln(arch,escala);
      writeln(' :13, ');
      writeln(' :13, ');
      writeln(' :13, ' HISTOGRAMA DE ',tipo,' DEL >> ',fecha:10, ' ');
      writeln(' :13, ');
      While (dato <> Uv) do
        Begin
          readln(arch,dato); (* Se desnormalizan las barras *)
          histog[k]:=((dato/100)*(escala*10))+histog[k];
          k:=k+1;
        end
      (* enddo *);
    end
  end

```

```

        fin_dat:=k-1;
        If (esc<escala) then
            esc:=escala;
            close(arch);
        end
    (* endfor *);
    If muchos then
        escribe(histog,fin_dat,numbin,inc,num_his,ndiv,esc,fecha,tipo)
    (* endif *);
end
(* END-LEE *);

```

```

(*****)
(*
    Rutina que muestra las funciones de ajuste disponibles
*)
(*****)

```

```

PROCEDURE MENU_2(Var cual:integer);
Begin

```

```

    gotoxy(0,0);
    espacio(23);
    gotoxy(0,4);
    writeLn(' ':25,'MENU # 2');
    writeLn(' ':25,'_____');
    espacio(2);
    writeLn(' ':20,'[1] Linea Recta');
    writeLn;
    writeLn(' ':20,'[2] Potencial');
    writeLn;
    writeLn(' ':20,'[3] Exponencial o Malthus');
    writeLn;
    writeLn(' ':20,'[4] Berthalanffy o Exponencial Saturada');
    writeLn;
    writeLn(' ':20,'[5] Logaritmica o Logistica');
    writeLn;
    writeLn(' ':20,'[6] Salir del menu');
    espacio(3);
    write(' ':25,'OPCION : ');
    readLn(cual);
end

```

```

(* END-MENU_2 *);

```

```

(*****)
(*                                           *)
(*      Rutina que presenta el Menu principal del modulo      *)
(*                                           *)
(*****)

```

```

PROCEDURE MENU_PRINC(Var opcion:char);

```

```

Begin

```

```

    gotoxy(0,0);espacio(23);
    gotoxy(0,2);
    writeln('  :23,'+-----+');
    writeln('  :23,'|      MODULO II.2      |');
    writeln('  :23,'|      AJUSTE A FUNCIONES      |');
    writeln('  :23,'+-----+');
    espacio(2);
    writeln('  :29,'MENU PRINCIPAL');
    writeln('  :29,'_____');
    espacio(2);
    writeln('  :23,'(A) Juste a Funciones');
    writeln;
    writeln('  :23,'(T) erminar');
    espacio(2);
    write('  :30,'OPCION: ');
    readln(opcion);

```

```

end

```

```

(* END-MENU_PRINC *)

```



```

(*****)
(*                                     *)
(*      Rutina que grafica en pantalla la curva teorica ajustada      *)
(*                                     *)
(*****)

PROCEDURE OBTEN_CURVA(tipo:string;cer0,xpuntos:integer;
                     Var histog:grafico;inicio,binseg:real);
Var
punto,                (* Punto de la curva                *)
x,                    (* Coordenada X de la pantalla de graf. *)
j,                    (* # de subdivisiones del eje X      *)
inc,                  (* Inc. en la separacion de las subdivi- *)
                      (* siones del eje X                  *)
i      : integer;    (* Indice del vector de datos en regio de *)
                      (* ajuste.                            *)
titu   : string;     (* Tipo de curva teorica ajustada      *)
nummax : real;       (* Valor mas grande del eje Y          *)
Begin
nummax:=histog[cer0];
For i:=cer0 to xpuntos-1 do
  If (histog[i]>nummax) then
    nummax:=histog[i]
  (* endif *)
(* endfor *)
nummax:=nummax/10;      (* Para normalizar eje Y            *)
Initturtle;
eje;
i:=4;j:=5;
lineas(i,j);
titu:=concat('AJUSTE A CURVA ',tipo);
letreros(titu,' x','seg.',' y');
pencolor(white);
escalau(nummax);
escalah(inicio,binseg,cer0,xpuntos);
pencolor(none);
moveto(27,46);x:=30;
pencolor(none);
inc:=trunc(240/(xpuntos-cer0))-2;
For i:=cer0 to xpuntos-1 do
  Begin
    punto:=round((histog[i]/nummax)*10);
    moveto(x,punto+46);
    wchar('.');x:=x+inc;pencolor(none);
  end
  (* endfor *)
  If (resp) then
    imprime
  (* endif *)
end
(* END-OBTEN_CURVA *)

```

```

(*****
(*)
(*) Rutina que realiza las transformaciones necesarias de 'x'
(*) y 'y' para el proceso de ajuste a curva
(*)
(*)
(*****)

PROCEDURE PROC_AJUSTE(tipo:string;Var histog:grafico;tip:char;cero,
xpuntos,moda:integer;m,b,inicio,binseg:real);
Var
i : integer; (* Indice de vector de datos en region *)
abciza, (* de ajuste *)
inc, (* Valor de la abciza *)
a : real; (* Incremento en valor de abciza *)
(* Parte constante de la curva teorica *)
Begin
inc:=0.0;
a:=exp(b);
For i:=cero to xpuntos-1 do
Begin
abciza:=inicio+(binseg/2)+inc;
inc:=binseg+inc;
Case tip of
'E' : histog[i]:=a*exp(m*abciza);
'P' : histog[i]:=a*exp(m*ln(abciza));
'B' : histog[i]:=(moda+1)*(1-exp(a*m*abciza));
'L' : histog[i]:=(moda+1)/(1+a*exp(m*abciza));
end
(* endcase *);
end
(* endfor *);
obten_curva(tipo,cero,xpuntos,histog, inicio,binseg);
end
(* END-PROC_AJUSTE *);

```

```

(*****
(*)
(*)      Rutina que realiza el ajuste a la recta teorica      (*)
(*)
(*)
(*****):
)

PROCEDURE RECTA(Var histog;grafico;Var numbin;integer;tipos:string);
Const
  dpi = 6.283185;          (* Constante PI multiplicada por dos *)
Var
  cero,                    (* Indice para inicio de ajuste *)
  xpuntos,                 (* Indice de fin de ajuste *)
  i,                       (* Indice del vector de datos en reg.aj. *)
  n,                       (* # de datos para ajuste *)
  ang,                     (* Inclination de la pendiente de recta *)
  b,                       (* Auxiliar de la ordenada al origen *)
  moda : integer;         (* Moda estadistica *)
  maux,                    (* Pendiente real *)
  m,                      (* Auxiliar de la pendiente *)
  baux,                   (* Ordenada al origen real *)
  inicio,                 (* Inicio de ajuste en segundos *)
  inc,                    (* Incremento en abciza *)
  binseg,                 (* Bin o intervalo en segundos *)
  fin,                    (* Fin de ajuste en segundos *)
  t1,t2,                  (* Variables auxiliares para calculo de *)
  t22,t3,                 (* pendiente y ordenada al origen *)
  abciza : real;         (* Valor de las abcizas *)
  tip : char;            (* Tipo de curva a ajustar *)
Begin
  estadis(histog,inicio,fin,numbin,cero,xpuntos,moda);
  n:=(xpuntos-cero);
  tipos:=tipos[i];
  For i:=cero to xpuntos-1 do
    If histog[i]>0 then
      Case tip of
        'E','P' : histog[i]:=ln(histog[i]);
        'B' : histog[i]:=ln(((moda+1)-histog[i])/histog[i]);
        'L' : histog[i]:=ln(((moda+1)-histog[i])/(moda+1));
      end
      (* endcase *)
      (* endif *)
    (* endfor *)
    inc:=0; abciza:=0;
    t1:=0; t2:=0;
    t22:=0; t3:=0;
    binseg:=numbin/1000;
    For i:=cero to xpuntos-1 do
      Begin
        If (tip='P') then
          abciza:=ln(inicio+(binseg/2)+inc)
        else
          abciza:=inicio+(binseg/2)+inc
        (* endif *)
        inc:=binseg+inc;
        t1:=(abciza*histog[i])+t1;
        t2:=abciza+t2;
        t3:=(abciza*abciza)+t3;
        t22:=histog[i]+t22;
      end
      (* endfor *)
    maux:=(((n*t1)-(t2*t2))/((n*t3)-(t2*t2)));
    baux:=(((t22*t3-t2*t1)/(n*t3-t2*t2)));
    b:=round(baux);
  n:=numbin;

```

```

maux:=atan(maux);
angi:=round((360*maux)/dpi);          (* ajuste a grados *)
graf(b,angi);
If not(resp) then
  If (tipo='RECTA') then
    (* ajuste a recta = recta teorica *)
  else
    proc_ajuste(tipo,histog,tip,cero,xpuntos,moda,m,baux,inicio,binseg)
  (* endif *)
else
  imprime
  (* endif *);
  Textmode;
end
(* END-RECTA *);

```

```

(*****
(*)
(*) Rutina inicial del ajuste, llama a cada una de las rutinas (*)
(*) destinadas para con dicho objeto (*)
(*)
(*****

```

```

PROCEDURE SIST_ANALIS(Var histog:grafico);
Var
  cual,                (* tipo de curva a ajustar *)
  numbin : integer;    (* Tamano del bin del histograma *)
  salir : boolean;     (* Bandera de continuar ajustando *)
Begin
  salir:=false;
  inicia(histog);
  lee(histog,numbin);
  While not(salir) do
    Begin
      menu_2(cual);
      Case cual of
        1 : recta(histog,numbin,'RECTA');
        2 : recta(histog,numbin,'POTENCIAL');
        3 : recta(histog,numbin,'EXPONENCIAL');
        4 : recta(histog,numbin,'BERTHALANFFY');
        5 : recta(histog,numbin,'LOGARITMICA');
        6 : salir:=true;
      *end
      (* endcase *);
    end
  (* enddo *);
end
(* END-SIST_ANALIS *);

```

```

(*****):
(*
(*          Programa Principal          *)
(*
(*****): (****)
Begin
  menu_princ(opcion);
  While (opcion<>'T') and (opcion<>'t') DO
    Begin
      Case opcion of
        'A','a': sist_analis(histog);
      end
      (* endcase *);
      menu_princ(opcion);
    end
  end
  (* enddo *);
end.

```

```

(*$S*)
(*****
(*)
(*)          LIBRERIA BASES          (*)
(*)          _____          (*)
(*)          La siguiente serie de rutinas constituye una de las (*)
(*) librerias de las que hace uso el Sistema de Analisis de la ac- (*)
(*) tividad electrica unitaria extracelular. (*)
(*) Estas rutinas permanecen como parte de la libreria que (*)
(*) pueda manejar la microcomputadora debido a que son comunes a (*)
(*) varios programas y sirven a dos de los modulos de este sistema. (*)
(*) Con esto se ahorra el tener que incluir el codigo de la (*)
(*) rutinas en cada modulo, con el consiguiente ahorro de memoria, (*)
(*) ademas de que la libreria no necesita permanecer todo el (*)
(*) tiempo en memoria, sino solo cuando se haga una llamada a (*)
(*) alguna de las rutinas de la libreria, y lo unico que se carga (*)
(*) a memoria es la porcion de codigo que sera utilizada. (*)
(*)
(*****

```

```
UNIT BASES;
```

```
INTRINSIC CODE 25 DATA 26;          (* El unit es ligado auto- *)
                                      (* maticamente despues de *)
                                      (* la compilacion del prog. *)
```

```
INTERFACE
```

```
USES
  TURTLEGRAPHICS;                    (* Libreria de graficacion *)
```

```
TYPE
  carac = packed array [0..12] of char; (* Vector de caracteres a *)
                                           (* graficar. *)
```

```

(*****
(*)
(*)          Definicion de rutinas y sus parametros          *)
(*)
(*****

```

```

PROCEDURE CONVIERTE(Var cadabin,cont:integer;Var caricarac);
PROCEDURE EJE;
PROCEDURE ESCALAV(escala:real);
PROCEDURE ESPACIO(numlinea:integer);
PROCEDURE LETREROS(letp,leth,leth2,letv:string);
PROCEDURE LINEAS(Var inc,ndiv:integer);
FUNCTION RESP:BOOLEAN;
PROCEDURE MENU_OP1(Var op1:char;tit:string);
PROCEDURE MENU_OP2(Var op2:char);

```

IMPLEMENTATION

```
(*****
(*)
(*)      Definicion de variables locales de todas las rutinas      (*)
(*)
(*****)
```

```
VAR
i,          (* Variable de iteracion          *)
j,          (* Variable de iteracion          *)
x,          (* Incremento en eje X para graficacion *)
y,          (* Incremento en eje Y para graficacion *)
con,        (* Numero de caracteres a graficar      *)
cadabin,    (* Tamano de incremento para escala en Y *)
aux,        (* Aux. para tamano de incremento en Y *)
residuo,    (* Aux. para conversion de # a caracter *)
divi       : integer;
lee        : char;
           (* Respuesta de Salida o Continuacion en *)
           (* un Histograma.                          *)
num        : charac;
esc,       (* Vector de caracteres a graficar    *)
auxreal    : real;
           (* Escala numerica para el eje Y      *)
           (* Aux. para conversion de # a caracter *)
```

```
(*****
(*)
(*)      Rutina que convierte una cadena de numeros a caracteres   (*)
(*)      para la graficacion                                       (*)
(*)
(*****)
```

PROCEDURE CONVIERTE;

Begin

con:=0;

car[0]:='0';

aux:=cadabin;

While (aux)0) do

Begin

residuo:=aux mod 10;

auxreal:=aux/10;

aux:=trunc(auxreal);

If (auxreal < 1.0) then

Begin

aux:=trunc(auxreal*10.0);

car[con]:=chr(aux+ord('0'));

aux:=0;

con:=con+1;

end

else

car[con]:=chr(residuo+ord('0'))

(* endif *);

con:=con+1;

end

(* endwhile *);

end

(* END-CONVIERTE *);

```

(*****
(*
(*          Rutina que grafica el eje para los Histogramas          *)
(*                                                                 *)
(*****

```

```

PROCEDURE EJE;

```

```

Begin

```

```

  move(-112); turn( 270);
  move( 50 ); turn(-270);
  pencolor ( white );
  move( 248); turn( 90 );
  pencolor ( black );
  move( 120); turn( 90 );
  move( 248); turn( 90 );
  pencolor ( white );
  move( 120);
  pencolor ( none );
  turn( 90 ); move( 248);
  pencolor ( white );
  turn( -90); move ( 2 );
  turn( 120); move ( 4 );
  turn( 120); move ( 4 );
  turn( 120); move ( 2 );
  pencolor ( none );
  turn( -90); move( 248);
  turn( -90); move( 120);
  pencolor ( white );
  turn( -90); move ( 2 );
  turn( 120); move ( 4 );
  turn( 120); move ( 4 );
  turn( 120); move ( 2 );

```

```

end

```

```

(* ENDPRO-EJE *);

```



```

(*****X: *****)
(*                               *)
(* Rutina que manda graficar la division numerica del eje Y           *)
(* para los Histogramas                                               *)
(*                               *)
(*****)

```

```

PROCEDURE ESCALAV;
Begin
  pencolor(none);
  moveto(0,0);
  y:=52;
  esci:=0;
  For i:=1 to 10 do
    Begin
      xi:=1;
      con:=0;
      esc:=esc+escala;
      cadabin:=trunc(esc*10);
      convierte(cadabin,con,num);
      If (con=0) then
        Begin
          moveto(x,y);
          wchar('0');
          xi:=x+6;
        end
      (* endif *);
      For j:=con downto 0 do
        Begin
          moveto(x,y);
          If (j=0) then
            Begin
              wchar('.');
              xi:=x+4;
              moveto(x,y);
              wchar(num[j]);
            end
          else
            Begin
              wchar(num[j]);
              xi:=x+6;
            end
          (* endif *);
        end
      (* endfor *);
      y:=y+10;
    end
  (* endfor *);
end
(* END-ESCALAV *);

```

```

(*****
(*)
(*)          Rutina que limpia la pantalla          (*)
(*)
(*****

```

```

PROCEDURE ESPACIO;
Begin
  For i:=1 to numlinea do
    writeln(' ');
  (* endfor *);
end
(* END-ESPACIO *);

```

```

(*****
(*)
(*)          Rutina que coloca letreros de identificacion de la grafica
(*)          en la pantalla de graficacion          (*)
(*)
(*****

```

```

PROCEDURE LETREROS;
Begin
  pencolor(none); moveto (0,0);
  moveto (1,41); wchar ('0');
  moveto (7,41); wchar ('.');
  moveto (10,41); wchar ('0');
  moveto(50,180); wstring(lets);
  moveto (1,171); wstring(letu);
  moveto(250,23); wstring(lets);
  moveto(250,16); wstring(lets2);
  moveto (18,8); wstring('RETURN para continuar');
  moveto (18,18); wstring('S para salir');
end
(* END-LETREROS *);

```

```

(*****
(*)
(*) Rutina que coloca las marcas para las divisiones en los (*)
(*) ejes X e Y (*)
(*)
(*****)

```

```

PROCEDURE LINEAS;

```

```

Begin
  turn(-90);
  move(10);
  For i:=1 to 11 do
    Begin
      move(10);turn(-90);
      move (3);turn(180);
      move (3);turn(-90);
    end
  (* endfor *);
  turn(90);
  divi:=inc#10;
  For i:=1 to ndiv do
    Begin
      move(divi);turn(-90);
      move (3);turn(180);
      move (3);turn(-90);
    end
  (* endfor *);
  turn(180);
  move(240);
  turn(-90);
end
(* END-LINEAS *);

```

```

(*****
(*)
(*) Rutina que muestra el menu para Tipo de Histograma (*)
(*)
(*****)

```

```

PROCEDURE MENU_OP1;

```

```

Begin
  gotoxy(0,0);espacio(23);
  gotoxy(0,0);espacio(2);
  writeln(' :22,'tit);
  espacio(2);
  writeln(' :26,'(B) arra simple');
  writeln;
  writeln(' :26,'(A) schurado');
  writeln;
  writeln(' :26,'(S) alida');
  espacio(2);
  write(' :31,'OPCION: ');
  readln(op1);
end
(* END-MENU_OP1 *);

```

```

(*****
(*)
(*) Rutina que muestra el menu para tipo de Escala del eje Y
(*)
(*)
(*****)

```

```

PROCEDURE MENU_OP2;
Begin
    espacio(2);
    writeln(' :26,'(M) axima escala');
    writeln;
    writeln(' :26,'(D) iferente escala');
    espacio(2);
    write(' :31,'OPCION: ');
    readln(op2);
end
(* END-MENU_OP2 *);

```

```

(*****
(*)
(*) Funcion booleana de Salida o Continuacion, en la presentacion
(*) de Histogramas
(*)
(*)
(*****)

```

```

FUNCTION RESP;
Begin
    read(keyboard, lee);
    resp:=(lee in ['s','S']);
end
(* END-RESP *);

```

```

(*) INICIALIZACION (*)

Begin
    (* No hay nada de inicializacion *);
end.

```


IMPLEMENTATION

```

Var
  i,                (* Contador de margen izquierdo *)
  j,                (* Iteracion para # de renglon *)
  k,                (* Contador de bits *)
  l,                (* Indice para arreglo de bytes *)
  ij,              (* Iteracion para margen derecho*)
  iq,              (* Itera para # de alambres *)
  pri : integer;   (* Contador de bytes en pantalla*)
  estimp,          (* Byte de estado de impresora *)
  salimp : ttrio;  (* Reg. de salida de impresora *)
  q,              (* Apuntador a un byte *)
  s : tbait;      (* Rastreador de bytes *)
  pris : array[0..191] of integer; (* Arreglo con # de columnas *)
  z : array[0..nodealamb] of ttrio; (* Arreglo de # de alambres *)

```

```

(*****
(*)
(*)          Rutina encargada de pasar byte por byte toda la
(*)          pantalla de la microcomputadora a papel
(*)
(*****)

```

```

PROCEDURE LOS77;
  external
  (* END-LOS77 *);

```

```

(*****
(*)
(*)          Indica fin de Informacion
(*)
(*****)

```

```

FUNCTION LIBRE;
Begin
  si:=8;
  qi:=estimp.apb^[0];
  libre:=(ord(odd(s) and odd(q))=0) (* cierto si bit 3 es cero *)
end
(* END-LIBRE *);

```

```

(*****
(*)
(*)          Rutina que lee todo el byte para pasarlo a papel
(*)
(*****)

```

```

PROCEDURE PON;
Begin
  repeat until libre;
  salimp.apb^[0]:=g;
end
(* END-PON *);

```

```

(*****)
(*
(*          Rutina que prepara el campo para la transferencia          *)
(*
(*
(*****)

```

```

PROCEDURE PREPA;
Begin
  salimp.dir:=-16232;          (* $C088, p. 54 SSC, dir de ACIA *)
  estimp.dir:=-16231;        (* $C089, dir de byte de estado *)
end
(* END-PREPA *);

```

```

(*****)
(*
(*          Rutina encargada de todo el proceso de traspaso          *)
(*
(*
(*****)

```

```

PROCEDURE CALCA;
Begin
  prepa;
  rewrite(gra,'printer:');
  write(gra,chr(27),'n',chr(27),'T16');  (* 72 puntos pp horizontal *)
                                          (* 9 lineas pp, 72 pppv *)

  pri:=0;
  For j:=0 to 23 do
    Begin
      (* Por cada 8 lineas *)
      If marg>0 then
        For ij:=1 to marg do
          write(gra,' ')  (* 8 puntos por blanco *)
        (* endfor *)
      (* noelse *)
      (* endif *)
      write(gra,chr(27),'G0280');
      For iq:=0 to nodealamb do
        z[iq].dir:=pris[pri+iq]  (* Direcciones de 8 lineas *)
      (* endfor *)
      pri:=pri+8;
      For i:=0 to 39 do
        (* para c/byte de la banda *)
        Begin
          For iq:=0 to nodealamb do
            (* para cada pila de bytes *)
            cont[iq]:=z[iq].alb[i]  (* lee valor y pasa a cont *)
          (* endfor *)
          los77;
          For k:=0 to 6 do
            pon(casa[k])
          (* endfor *)
        end
      (* endfor *)
      writeln(gra);
    end
  (* endfor *)
  close(gra);
end
(* END-CALCA *);

```

(*

INICIALIZACION

*)

Begin

 i:=0;

 For i:=0 to 2 do

 For j:=0 to 7 do

 For k:=0 to 7 do

 Begin

 pri[s[i]]:=8192+40*i+128*j+1024*k;

 i:=i+1;

 end

 (* endfor *)

 (* endfor *)

 (* endfor *);

end.


```

-----
;
;                               Rutina auxiliar LOS77
;
; Procedimiento auxiliar en el vaciado de la pantalla grafica de
; Apple II a una impresora de matriz. Cont es un vector de 8 bytes
; que corresponde a una region de la pantalla con 8 puntos de altura.
; El ancho de dicha region es de 7 puntos en el caso de Apple,
; pues de cada byte solo se usan los 7 bits menos significativos en la
; generacion de graficas. El resultado del procedimiento son 7 bytes
; en el vector Casa que se mandan sucesivamente a la impresora, la
; cual se debe encontrar en modo grafico.
;
-----

```

```

-----
;
;                               Programa Principal
;
-----

```

```

      .PROC   LOS77
      .Public cont
      .Public casa
; Arreglo de 8 bytes, para cada alambre
; Arreglo de 7 bytes para cada columna
; del caracter grafico.
      ldy    #00
; Y, indice del caracter de salida
; o del bit del contador CONT.
cero   lda    #00
      sta    casa,y
; Limpia el caracter

      ldx    #00
; X, indice del contador CONT o del
; bit del caracter de salida.

uno    ror    cont,x
      bcs    llena
dos    inx
      cpx    #08
; Ya termino con los 8 renglones de 7
; bits ?
      bne    uno
      iny
      cpy    #07
      bne    cero
      rts
; Regresa a Pascal

llena  clc
      lda    eigen,x
      adc    casa,y
; Carga columna y
      sta    casa,y
; Esta suma nunca llena el carry
      jmp
eigen  .byte  01,02,04,08,010,020,040,080
      .end

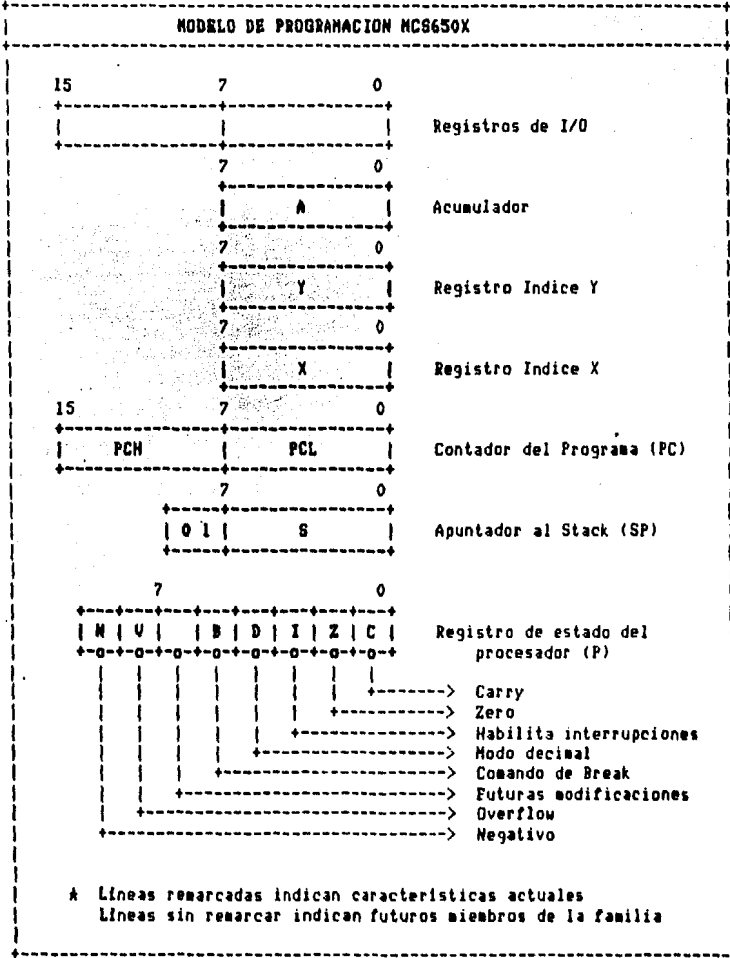
```

APENDICE B
TABLAS AUXILIARES

En este apéndice se proporcionan algunas tablas e información adicional sobre el manejo y funcionamiento del microprocesador 6502, así como del convertor analógico-digital y digital-analógico (A/D + D/A). Para mayor información sobre ambos referirse a la bibliografía que se proporciona en el capítulo 4.

TABLAS AUXILIARES

B.1 MODELO DE PROGRAMACION 6502



TABLAS AUXILIARES

B.2 INSTRUCCIONES DEL MICROPROCESADOR 6502

Mnemónico	Operación de la instrucción
ADC	Sumar el acumulador con Carry
AND	And de memoria y el acumulador
ASL	Desplaza a la izquierda al acumulador
BCC	Salto si la bandera de Carry en clear
BCS	Salto si la bandera de Carry encendida
BEQ	Salto si resultado zero
BIT	Prueba bits de memoria con el acumulador
BMI	Salta si resultado negativo
BNE	Salta si resultado diferente de zero
BPL	Salta si resultado positivo
BRK	Realiza un Break
BVC	Salta si la bandera de overflow en clear
BVS	Salta si la bandera de overflow encendida
CLC	Limpiar la bandera de Carry
CLD	Limpiar la bandera de modo decimal
CLI	Limpiar bit de deshabilitar interrupciones
CLV	Limpiar la bandera de overflow
CMP	Compara memoria con acumulador
CPX	Compara memoria con registro índice X
CPY	Compara memoria con registro índice Y
DEC	Decrementa memoria en uno
DEX	Decrementa registro índice X en uno
DEY	Decrementa registro índice Y en uno
EOR	OR exclusivo memoria contra acumulador
INC	Incrementa memoria en uno
INX	Incrementa registro índice X en uno
INY	Incrementa registro índice Y en uno
JMP	Salto incondicional
JSR	Salto a subrutina
LDA	Carga el acumulador con memoria
LDX	Carga el registro índice X con memoria
LDY	Carga el registro índice Y con memoria
LSR	Desplazamiento lógico a la derecha
NOP	No operación
ORA	OR de memoria con el acumulador
PHA	Guarda acumulador en el Stack
PHP	Guarda estado del procesador en Stack
PLA	Saca el acumulador del Stack
PLP	Saca estado del procesador del Stack
ROL	Rotación a la izquierda
ROR	Rotación a la derecha
RTI	Resceso de interrupción
RTS	Resceso de subrutina

TABLAS AUXILIARES

SBC	Resta del acumulador con carry
SEC	Enciende la bandera de Carry
SED	Enciende la bandera de modo decimal
SEI	Deshabilita interrupciones
STA	Almacena el acumulador en memoria
STX	Almacena registro índice X en memoria
STY	Almacena registro índice Y en memoria
TAX	Transfiere acumulador a registro índice X
TAY	Transfiere acumulador a registro índice Y
TSX	Transfiere stack pointer a res. índice X
TXA	Transfiere res. índice X al acumulador
TYA	Transfiere res. índice Y al acumulador
TXB	Transfiere res. índice X al stack pointer

TABLAS AUXILIARES

INSTRUCCIONES DEL PROCESADOR 6502
(Modos de direccionamiento y tiempos de ejecucion)

Inst.	Acum.	Modo de direccionamiento											
		In- med.	Pag. 0	Pag. 0,X	Pag. 0,Y	Abs. X	Abs. Y	Impl.	Rel.	(Ind.,X)	(Ind.) Y	Abs. Ind.	
ADC	.	2	3	4	.	4	4A	4A	.	.	6	5A	.
AND	.	2	3	4	.	4	4A	4A	.	.	6	5A	.
ASL	2	.	5	6	.	6	7
BCC	2AA	.	.	.
BCS	2AA	.	.	.
BEQ	2AA	.	.	.
BIF	.	.	3	.	.	4
BHI	2AA	.	.	.
BNE	2AA	.	.	.
BPL	2AA	.	.	.
BRK
BVC	2AA	.	.	.
BVS	2AA	.	.	.

TABLAS AUXILIARES

CLC	2
CLD	2
CLI	2
CLV	2
CMP	.	2	3	4	.	4	4A	4A	.	.	6	5A	.
CPX	.	2	3	.	.	4	6	5A	.
CPY	.	2	3	.	.	4	6	5A	.
DEC	.	.	5	6	.	6	7
DEX	2
DEY	2
EOR	.	2	3	4	.	4	4A	4A	.	.	6	5	.
INC	.	.	5	6	.	6	7
INX	2
INY	2
JMP	3	5
JSR	6

TABLAS AUXILIARES

LDA	.	2	3	4	.	4	4A	4A	.	.	6	5A	.
LDX	.	2	3	.	4	4	.	4A
LDY	.	2	3	4	.	4	4A
LSR	2	.	5	6	.	6	7
NOP	2
ORA	.	2	3	4	.	4	4A	4A	.	.	6	5A	.
PHA	3
PHP	3
PLA	4
PLP	4
ROL	2	.	5	6	.	6	7
ROR	2	.	5	6	.	6	7
RTI	6
RTS	6
SBC	.	2	3	4	.	4	4A	4A	.	.	6	5A	.
SEC	2

TABLAS AUXILIARES

SED	2
SEI	2
STA	.	.	3	4	.	4	5	5	.	.	6	6	.	.
STXA	.	.	3	.	4	4
STYAA	.	.	3	4	.	4
TAX	2
IAY	2
TSX	2
TXA	2
TXS	2
IYA	2
<p>A Añada un ciclo si el direccionamiento causa cruce de página A Añada un ciclo si la instrucc. es salto y uno más si cruza página</p>														

TABLAS AUXILIARES

INSTRUCCIONES DE SALTO CONDICIONAL		
Instrucción	Bandera usada	Valor en que ocurre el salto
BCC	Carry	0
BCS	Carry	1
BNE	Zero	0
BEQ	Zero	1
BPL	Nesativo	0
BMI	Nesativo	1
BVC	Overflow	0
BVS	Overflow	1

Si la bandera especificada no tiene el valor indicado, no hay ningún salto.

INSTRUCCIONES DE COMPARACION				
Condición	Nesativo	Zero	Carry	
A: X: o Y < Memoria	1*	0	0	
A: X: o Y = Memoria	0	1	1	
A: X: o Y > Memoria	0*	0	1	

* = es valida solo en complemento a dos

TABLAS AUXILIARES

B.2.1 Modos De Direccionamiento Para El 6502

El microprocesador 6502 cuenta con 13 modos de direccionamiento que se describen de manera breve a continuación :

(1) **Direccionamiento inmediato.**- Permite especificar una constante de 8 bits como el operando, el cual será identificado por el prefijo " #".

(2) **Direccionamiento Absoluto.**- Permite el direccionamiento directo de cualquiera de las 65,536 localidades de memoria en el espacio de direcciones del 6502.

(3) **Direccionamiento por página cero.**- Es una forma de direccionamiento absoluta en la cual el microprocesador accesa sólo las primeras 256 localidades en memoria. Son direcciones las hexadecimales 0000 a 00FF. Debido a que el byte de orden alto de las direcciones en página cero es siempre cero, las instrucciones que usan este tipo de direccionamiento son de sólo dos bytes. Considerando la optimización de espacio y tiempo de ejecución que se obtiene con este tipo de direccionamiento, es deseable utilizarlo lo más posible para guardar datos que son usados con mucha frecuencia.

(4) **Direccionamiento Implícito.**- Se utiliza por instrucciones que no necesitan de operando, esto es, que sólo tienen el código de operación de la instrucción.

(5) **Direccionamiento Absoluto Indirecto.**- Es utilizado sólo por la instrucción JMP. El operando de esta instrucción es la dirección de las dos primeras localidades de memoria que contienen la dirección destino de 16 bits. Este modo de direccionamiento nos permite trabajar con direcciones destino variables.

(6) y (7) **Direccionamiento Indexado Absoluto.**- La dirección efectiva del operando se calcula sumando los contenidos del registro X o Y a la dirección absoluta en la instrucción.

(8) y (9) **Direccionamiento Indexado Página cero.**- La dirección efectiva de página cero del operando se calcula sumando los contenidos de los registros X o Y a la dirección base de página cero contenida en el segundo byte de la instrucción. Un punto importante por recordar es que la dirección efectiva está restringida a la página cero.

TABLAS AUXILIARES

(10) Direccionamiento Indirecto Indexado.- Es una combinación de los modos de direccionamiento indexado y del indirecto. Un desplazamiento en el registro X es sumado al operando de la página cero en la instrucción para producir una dirección indirecta de página cero. La dirección absoluta efectiva se queda contenida en la localidad de memoria direccionada por la dirección indirecta calculada (byte bajo) y la siguiente localidad de memoria (byte alto). Lo mejor de este tipo de direccionamiento es que como la dirección efectiva es una dirección absoluta de 16 bits, el espacio completo de memoria (64K-bytes) del microprocesador 6502 puede ser accesado con una instrucción de tan solo dos bytes. Sin embargo, nada se da gratis, todas las instrucciones con direccionamiento indirecto indexado se llevan 6 ciclos para su ejecución, tres mas que los de página cero de la misma instrucción y dos mas que los de absoluta.

(11) Direccionamiento Indexado Indirecto.- Aquí, el índice es añadido a la dirección de página cero en el segundo byte de la instrucción después de que el direccionamiento indirecto se realice. Por esto, esta técnica también se conoce como "Post-indexado".

(12) Direccionamiento Relativo.- La dirección efectiva en este tipo de direccionamiento, se calcula sumando un desplazamiento positivo o negativo al valor actual del "Program Counter".

(13) Direccionamiento por Acumulador.- No es mas que un tipo de direccionamiento implicado que es único para cuatro instrucciones de desplazamiento y rotación.

TABLAS AUXILIARES

B.2.2 Formato Para Modos De Direccionaliento

Modo	Formato de operando(s)
Inmediato	#aa
Absoluto	aaaa
Basina cero	aa
Implicado	-
Absoluto Indirecto	(aaaa)
Indexado absoluto,X	aaaa,X 0 aaaaX
Indexado absoluto,Y	aaaa,X 0 aaaaY
Indexado Basina cero,X	aa,X 0 aaX
Indexado Basina cero,Y	aa,Y 0 aaY
Indirecto Indexado	(aa,X) 0 (aaX)
Indexado Indirecto	(aa),Y 0 (aa)Y
Relativo	aa 0 aaaa
Acumulador	A
a --> representa un dígito hexadecimal	

TABLAS AUXILIARES

B.3 CARTA DE CONVERSION DEL A/D + D/A

(Salida) DAC	Numero	(Entrada) ADC
-5.00 Volts	0	-5.00 Volts
.	.	.
-2.50 Volts	64	-2.50 Volts
.	.	.
0.00 Volts	128	0.00 Volts
.	.	.
+2.50 Volts	192	+2.50 Volts
.	.	.
+5.00 Volts	255	+5.00 Volts

Cuando el voltaje cambia 39 mV, el ADC produce un cambio de un valor digital. De la misma manera cuando el valor digital siendo convertido cambia en uno, el voltaje cambiará en 39 mV. a la salida.

TABLAS AUXILIARES

B.3.1 Organización De La Tableta Del A/D + D/A Y Su Interfaz DB-25

DAC-J1 pin	ADC-J2 pin	DB-25 pin	Señal
01	01	01	canal 15
02	02	02	canal 14
03	03	03	canal 13
04	04	04	canal 12
05	05	05	canal 11
06	06	06	canal 10
07	07	07	canal 9
08	08	08	canal 8
09	09	09	canal 7
10	10	10	canal 6
11	11	11	canal 5
12	12	12	canal 4
13	13	13	canal 3
14	14	no conec.	no conexión
15	15	25	canal 2
16	16	24	canal 1
17	17	23	canal 0
18	18	22	ver nota
19	19	21	tierra
20	20	20	tierra
21	21	19	tierra
22	22	18	tierra
23	23	17	tierra
24	24	16	tierra
25	25	15	tierra
26	26	14	tierra

Nota.- En el DAC el pin #18 es una señal de no conexión; en el ADC el pin #18, es una señal de -5 V. de referencia.

El Pin 10 del circuito integrado U9 en la tableta del convertor, es una señal de inicio de conversión. Esta señal está presente alrededor de 500 nanosegundos antes del comienzo de la conversión. Este sirve como una de las entradas al circuito "Sample and Hold" con que cuenta el convertor para su funcionamiento Junto con la señal de fin de conversión.

APENDICE C
GLOSARIO MEDICO

Actividad Controlada.- Un proceso espontáneo puede, cuando se le sujeta a un estímulo externo, presentar dependencia del estímulo. Si esto sucede, durante esta faceta de su comportamiento se dice que es un proceso controlado y sus propiedades pueden ser estudiadas en relación con las diversas propiedades temporales, espaciales e intensivas de los parámetros del estímulo.

Actividad Electroencefalográfica.- Actividad eléctrica encefálica de fondo, fué descrita en animales no anestesiados desde el siglo XIX.

Actividad Espontánea.- Los procesos espontáneos también son conocidos como procesos continuos o no controlados y son aquellos en los cuales los eventos observados ocurren aun en la ausencia de cualquier estímulo externo. De esta manera el proceso es una manifestación de alguna actividad sostenida en el sistema nervioso, actividad que puede ser observada esencialmente en cualquier periodo de tiempo.

Axón.- Ramificación eferente de la célula nerviosa (neurona).

Bin.- División de tiempo, sobre el cual se desea obtener la cuantificación de algún evento.

Bulbo Olfatorio.- Extremo distal abultado de ambos tractos olfatorios situados a cada lado de la cisura longitudinal bajo la cara ventral de los lóbulos anteriores del cerebro.

GLOSARIO MEDICO

Célula Mitral.- Célula principal del Bulbo Olfatorio.

Diencéfalo.- Porción del cerebro que se encuentra entre el telencéfalo y el mesencéfalo y que incluye los tálamos y la mayor parte del tercer ventrículo.

Electroencefalograma (EEG).- Registro de las variaciones de potencial recogidas del encéfalo. El EEG puede ser registrado con electrodos sobre el cuero cabelludo sin abrir el cráneo, o con electrodos sobre la superficie del encéfalo o dentro de él.

Electrofisiología.- Rama de la Fisiología basada en el estudio de actividad eléctrica.

Entrada Aferente.- Entrada dirigida hacia el centro o centripeta.

Espiga.- Véase potencial de acción neural.

Fascículo.- Haz de fibras nerviosas, musculares separadas por tejido conjuntivo. y que además parecen tener conexiones y funciones comunes.

Fisiología.- Ciencia que trata sobre las funciones de los organismos vivos o de sus partes, a diferencia de la morfología.

Función endócrina.- Función relativa a las glándulas endócrinas o a sus secreciones. Estas glándulas secretan substancias hormonales directamente dentro del torrente sanguíneo, por ejemplo la hipófisis, páncreas, tiroides, paratiroides, glándulas suprarrenales, ovarios y testículos, etc.

Glomérulo.- Masa pequeña redondeada llena de asas capilares que se proyectan dentro de un corpúsculo renal.

Hiperexcitabilidad.- Excitabilidad excesiva. Umbral bajo para la excitación.

GLOSARIO MEDICO

Hipotálamo.- Región del diencéfalo que forma el piso del tercer ventrículo y que incluye a los núcleos circundantes y asociados.

Histología.- Rama de la biología que trata sobre las estructuras de los tejidos, incluyendo el estudio de las células y de los órganos.

Intervalo interespiga.- Intervalo que existe entre la aparición de una espiga y la siguiente.

Legrado (o raspado).- Es aquella intervención quirúrgica mediante la cual se extirpan por raspado los tejidos de producción anormal; se practica en dermatología y en ginecología (raspado de la pared interna del útero para extraer los residuos de las membranas fetales (después de abortos o partos prematuros) o ciertas neoformaciones tumorales).

Membrana presináptica.- Membrana eléctricamente excitable que limita la terminal axónica adyacente a la dendrita.

Mielina.- Sustancia blanca grasosa que forma la vaina de algunos nervios. Mezcla compleja de lípidos extraída del sistema nervioso; es doblemente refráctil y contiene fosfátidos y colesterol.

Neoformación.- Es la formación de un tejido nuevo, tanto si es de carácter normal como si es patológico.

Nervio Vago o Neumogástrico.- Es el que inerva muchos músculos faríngeos y todos los extrínsecos de la laringe, y al descender por el tórax y el abdomen envía ramificaciones nerviosas al corazón, a los vasos sanguíneos, a la tráquea, a los bronquios, a los pulmones, al estómago, intestinos, etc., regulando las importantes funciones de la circulación sanguínea, la respiración y la digestión. Su nombre se debe a que 'vaga' lejos de su origen.

Neuronas.- Unidades celulares del sistema nervioso encargadas de la transmisión de la información a lo largo del mismo, especializada como conductora de impulsos.

GLOSARIO MEDICO

Neurotransmisor.- La transmisión de la información en la sinapsis se realiza por medio de una substancia química a la que se le ha denominado 'Neurotransmisor', que se libera por medios eléctricos y que propicia cambios en la permeabilidad iónica de la membrana de la neurona receptora.

Núcleo del Tracto Solitario.- Punto de unión de todas las fibras gustativas en el Sistema Nervioso.

Oradar.- Perforar sobre una zona.

Perfundir.- Introducir un líquido (inyectado) en alguna zona del sistema nervioso

Periostio.- Es aquella fina membrana de estructura fibroelástica, de aspecto blanquecino y de espesor no uniforme que rodea al hueso; asegura la sensibilidad y la nutrición del tejido óseo por medio de los nervios y vasos sanguíneos que llegan al mismo a través de esta membrana.

Potenciales de Acción.- Cambios en el potencial eléctrico de las membranas que se presentan con la depolarización y la polarización de una célula excitable.

Sinapsis.- Punto de contacto entre dos neuronas. Existen de dos tipos : a) Eléctrica y b) Electroquímica. En la primera las neuronas están conectadas entre sí, actuando como un cable. En las sinapsis químico-eléctricas existe un espacio entre las células al que se le ha llamado espacio intersináptico, que impide el paso directo del potencial eléctrico.

Sinápsis Axo-dendrítica.- A la sinapsis de un axón con una dendrita se le denomina sinápsis axo-dendrítica.

Sistema Nervioso.- Todas las estructuras nerviosas del cuerpo, incluyendo el cerebro, tallo cerebral, médula espinal, nervios craneales y periféricos, y ganglios.

GLOSARIO MEDICO

Sonda de Foley.- Sonda utilizada para introducción de líquidos en alguna zona del sistema nervioso.

Sondaje.- Es la introducción de un tubo especial de goma en un conducto natural de nuestro organismo con un fin diagnóstico o terapéutico.

Transmisión sináptica.- Transmisión de la información a través de las sinapsis entre neuronas.

Traqueostomía.- Incisión quirúrgica en la tráquea desde el exterior, o sea, por el cuello. Generalmente se incide verticalmente la parte media anterior del cuello (porción inferior) y se alcanza la pared anterior de la tráquea que también se incide; después se introduce a través de la incisión un tubo metálico corto y curvo, llamado cánula traqueal; a través de dicha cánula los pulmones pueden recibir y expeler el aire.