



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

**DESARROLLO DE INTERFACES
COMPUTACIONALES Y ELECTRONICAS PARA UN
LABORATORIO DE NEUROFISIOLOGIA Y
PSICOFISICA.**

T E S I S

PARA OBTENER EL TITULO DE:

INGENIERIA EN COMPUTACION

P R E S E N T A N :

BRENDA ZORAIDA FUENTES PASTEN

YEDID ZELENE REY MENESES

DIRECTOR DE TESIS: ING. ADRIAN HERNANDEZ ALVA



CIUDAD UNIVERSITARIA

2000



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

Damos nuestro agradecimiento a:

*La Universidad Nacional Autónoma de México
Nuestra Alma Mater.*

*A la Facultad de Ingeniería y a los profesores
que contribuyeron a nuestra formación profesional
para servir a la comunidad.*

*Al Instituto de Fisiología Celular
por permitirnos formar parte de su personal.
Por la oportunidad de realizar este trabajo.*

*A todo el personal del Laboratorio del Dr. Romo,
en particular al Dr. Ranulfo Romo.*

*Al Dr. Antonio Zainos, a la Ing. Norma Ortega,
al Ing. Armando Fuentes y al Lic. Genaro Rey
por su valiosa ayuda para elaborar este trabajo
a través de sus comentarios y correcciones.*

*Y de manera muy especial
a nuestro director Ing. Adrián Hernández Alva
Por su dedicación, ayuda y paciencia en la
elaboración de este trabajo.
Por sus enseñanzas y consejos, gracias.*

El presente proyecto es el resultado del esfuerzo y la dedicación no solo de dos personas, sino de todos aquellos que nos brindaron su apoyo y cariño... en particular y de una manera muy especial quiero expresar mi agradecimiento a

Mis padres, Queta y Armando, por todo el apoyo que nos brindaron a mí y a mis hermanos, por la confianza depositada en cada uno de nosotros y principalmente por alentarnos a ser cada día mejores.

Gracias por haberme brindado su cariño y comprensión, por sus consejos, ya que sin todo esto nunca hubiese llegado al final de este camino por el cual me han guiado, nunca dejaré de agradecer a Dios por haberme dado unos padres tan maravillosos como ustedes.

A mis hermanos, Izu y Armando, por su cariño y por enseñarme tantas cosas en la convivencia cotidiana.

A Isra y Zele que forman parte importante de mi vida, por compartir los momentos difíciles y también los alegres. Por su apoyo y comprensión.

A los buenos compañeros y los amigos que hicieron de la Facultad un segundo hogar.

Brenda

Primeramente, doy gracias a Dios por poner en mi camino a tantas personas, quienes han contribuido de forma significativa a este trabajo y a mi vida.

A mis padres, Margarita y Genaro, quienes me educaron y me ayudaron a culminar mis estudios universitarios, por ser siempre alentadores y reconfortantes en los momentos difíciles y a los que es imposible pagar su cariño como se merecen.

También debo este trabajo a mis hermanos por su apoyo y cariño.

A mi familia por su amor y por estar siempre presentes.

A mis amistades por compartir alegrías conmigo.

A Todos Muchas Gracias....

Zelene



ÍNDICE

1	DEFINICIÓN DEL PROBLEMA.....	3
1.1	CONTEXTO	4
1.2	OBJETIVO	7
1.3	JUSTIFICACIÓN	8
1.4	ALCANCES	9
1.5	RESTRICCIONES	9
2	MÉTODO.....	10
3	APLICACIÓN DEL MÉTODO	13
3.1	ANÁLISIS.....	14
3.1.1	ESTUDIO PRELIMINAR	14
3.1.1.1	Sistema Actual	15
3.1.2	ANÁLISIS DETALLADO	18
3.1.2.1	Definición de estado	20
3.1.2.2	Definición de tarea	21
3.1.2.3	Definición de clase	22
3.1.2.4	Definición de ensayo.....	22
3.1.2.5	Definición de experimento.....	22
3.1.2.6	Secuencia de estados.....	22
3.1.2.7	Análisis de la construcción y ejecución de un experimento	23
3.1.2.8	Secuencia de presentación	27
3.1.2.9	Análisis del control de un experimento	28
3.1.2.10	Almacenamiento de la información	29
3.1.2.11	Graficación de datos	31
3.1.2.12	Comunicación entre computadoras.....	31
3.1.3	PLANEACIÓN	32
3.1.3.1	Propuesta.....	32
3.1.3.2	Diagrama Conceptual.....	32
3.1.4	DIAGRAMAS DE FLUJO DE INFORMACIÓN.....	33
3.1.5	EVALUACIÓN DE HARDWARE, SOFTWARE Y REQUERIMIENTOS	37
3.2	DISEÑO	39
3.2.1	DISEÑO INTERNO.....	39
3.2.1.1	Módulos principales de programación.....	39
3.2.1.2	Comunicación y control.....	40
3.2.1.2.1	Tarjeta de Comunicación de Datos GPIB	40
3.2.1.2.2	Tarjeta PC-TIO-10.....	41
3.2.1.2.3	Tarjeta LAB-PC	41
3.2.1.2.4	Configuración del Sistema.....	41
3.2.1.3	Distribución de datos	42
3.2.1.4	Control de la tarea	43
3.2.1.4.1	Secuencia de comunicación	44
3.2.1.4.2	Generación de Estímulos	44



3.2.1.5	Colección de datos	45
3.2.1.6	Graficación de datos	45
3.2.2	DISEÑO FÍSICO	46
3.2.2.1	Estándares de Diseño	46
3.2.2.2	Aplicación Constructor	48
3.2.2.3	Diseño de una tarea.....	50
3.2.2.3.1	Crear una nueva tarea	50
3.2.2.3.2	Modificar una tarea existente.....	52
3.2.2.4	Diseño del archivo de clases	53
3.2.2.4.1	Crear un nuevo archivo de clases.....	53
3.2.2.4.2	Modificar una archivo de clases existente	55
3.2.2.5	Ejecución de un experimento.....	58
3.2.3	DIAGRAMAS	61
3.2.3.1	Diagrama de transición entre estados y Esquema de eventos.....	61
3.2.3.2	Diagrama de transición de estados del Constructor.....	63
3.2.3.3	Esquema de eventos que muestra como dar inicio a un experimento desde la aplicación Constructor	64
3.2.3.4	Esquema de eventos que muestra como se controla ejecución de un experimento.....	65
3.2.3.5	Diagrama de flujo del programa de la máquina Control.....	66
3.2.3.6	Diagrama de flujo del programa de la máquina Generador	67
3.3	DESARROLLO	68
3.3.1	PROGRAMACIÓN DE MÓDULOS	68
3.3.2	PLAN DE PRUEBAS	68
3.3.3	PLAN DE INSTALACIÓN	68
3.3.4	PLAN DE LIBERACIÓN	69
3.4	MANTENIMIENTO	69
4	RESULTADOS	70
5	CONCLUSIONES	76
6	BIBLIOGRAFÍA	78



1 DEFINICIÓN DEL PROBLEMA



1.1 CONTEXTO

En este trabajo se presenta la forma en la cual se utilizan los conocimientos adquiridos al estudiar la licenciatura de Ingeniería en Computación para resolver un problema del área de Neurociencias cognitivas.

Las Neurociencias cognitivas son un conjunto de disciplinas que agrupa conocimientos y técnicas de áreas tan diversas como la Medicina, la Biología, la Psicología, y de las Ingenierías Eléctrica, Mecánica y en Computación, para tratar de resolver una de las preguntas más antiguas de la humanidad: ¿cómo el sistema nervioso es capaz de percibir el mundo externo y cómo esa información es transformada para dar lugar a procesos como el reconocimiento de patrones, la memoria, el lenguaje, el aprendizaje y generar movimientos tan complejos como el caminar?.

Las Neurociencias cognitivas utilizan diversas metodologías. Entre las más importantes se encuentra el estudio de las lesiones, que trata de asociar el deterioro de una función cognitiva con reportes clínicos de daños en un área específica del cerebro, por ejemplo, lesiones en el lóbulo frontal (área de Broca) que producen deficiencia en la articulación del lenguaje en seres humanos. La imagenología, que utiliza técnicas como los rayos X, resonancia magnética nuclear y la tomografía por emisión de positrones, que permiten mediante imágenes correlacionar la activación de ciertas áreas del cerebro con una función específica. Finalmente tenemos el registro unitario extracelular que mediante la introducción de microelectrodos (80 a 100 micras de diámetro) permite el registro de la actividad eléctrica de una neurona para posteriormente correlacionarla con alguna función cognitiva.

Las investigaciones enfocadas en esta área utilizan la Psicofísica para estudiar la relación entre las propiedades de los estímulos físicos y las reacciones psicológicas a éstas, para posteriormente utilizar técnicas de Neurofisiología que correlacionan la actividad eléctrica neuronal con esas propiedades físicas, además de las reacciones psicológicas manifestadas mediante acciones motoras. Para este fin se enseña a sujetos (primates humanos o subhumanos) a responder con ciertas acciones (movimientos motores) a la presentación de estímulos (ya sean visuales, sonoros o táctiles). A la secuencia de estímulos-respuestas se le conoce como tarea.

En el Laboratorio del Dr. Ranulfo Romo del Instituto de Fisiología Celular, UNAM, se llevan a cabo estudios, en primates subhumanos, sobre los mecanismos neurales de la percepción, para lo cual se utilizan las disciplinas experimentales antes mencionadas (la Psicofísica y la Neurofisiología).

En este Laboratorio se ha investigado la forma en que se representan algunos estímulos táctiles en ciertas áreas de la corteza cerebral de un primate subhuamno. El propósito de estos estudios ha sido descubrir los procesos neurales que permiten la representación de un estímulo táctil en el cerebro y de cómo esta representación da origen a la percepción de los mismos (Fig. 1).⁽¹⁾



También se ha tratado de establecer la forma en que las representaciones de los estímulos táctiles en la corteza cerebral son utilizadas para que un individuo sea capaz de realizar movimientos voluntarios.⁽²⁾ En la Fig. 1 se presenta el diagrama que ilustra los procesos provocados por los estímulos táctiles, desde los receptores cutáneos hasta la generación de la percepción de los mismos o de conductas mucho más complejas como lo son la memoria y la conducta motora voluntaria.⁽³⁾

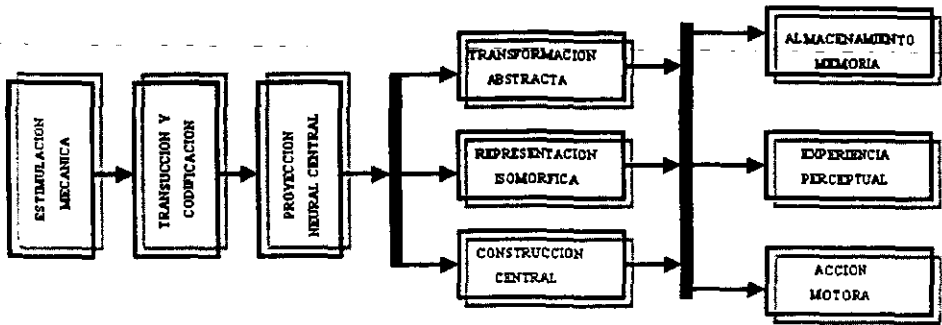


Fig. 1

Para demostrar que la velocidad y la dirección del movimiento son variables físicas que modifican la capacidad de reconocimiento de un objeto, se utilizó un estimulador táctil capaz de mover un patrón mecánico sobre las puntas de los dedos con variables cinemáticas y dinámicas controladas (velocidad, dirección, aceleración y fuerza). El propósito de este estudio fue tratar de establecer una correlación directa entre las propiedades físicas del estímulo y la actividad neuronal. Las investigaciones determinaron que la dirección del estímulo táctil es representada en la forma de un vector, generado por la actividad de una población de neuronas de la corteza somatosensorial. Además, que la magnitud del vector de dirección es modulada por la velocidad del estímulo.⁽⁴⁾

También se ha estudiado en el laboratorio la forma en que las neuronas representan al estímulo, y cómo se utiliza esta representación para agrupar y diferenciar estos estímulos, es decir, la relación entre esta representación y la decisión tomada por un sujeto para indicar su detección y categorización. Para realizar estos estudios se entrenaron monos *Macaca mulatta* en una tarea (secuencia predefinida de eventos donde se presentan estímulos y se registran las respuestas del sujeto a dichos estímulos) de detección y categorización de la velocidad de estímulos táctiles.⁽⁵⁾ La capacidad para realizar esta tarea es prácticamente idéntica a la de los humanos, lo que demuestra la utilidad de este modelo animal para investigar las bases neurales de la percepción.

(1).- Ver Bibliografía



Definición del problema: *CONTEXTO*

Para poder llevar a cabo las investigaciones el laboratorio cuenta con dispositivos electrónicos y un Sistema computacional, el cual controla a estos dispositivos y además permite la presentación de los estímulos y el almacenamiento de los datos psicofísicos y neurofisiológicos que se obtienen en una sesión experimental (Fig.2).

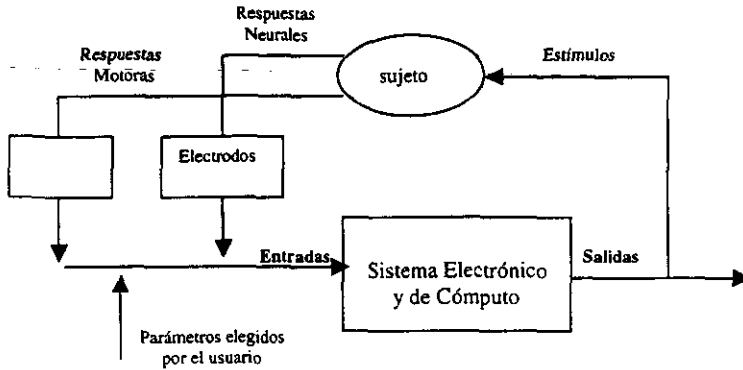


Fig. 2 Diagrama del Sistema Electrónico y de Cómputo.



1.2 OBJETIVO

Diseñar y construir un Sistema para el Laboratorio del Dr. Romo del Instituto de Fisiología Celular, UNAM, que permita realizar experimentos de neurofisiología y psicofísica. Los resultados permitirán correlacionar la actividad unitaria extracelular de la corteza cerebral, con la conducta del sujeto experimental y con los parámetros físicos de estímulos táctiles.



1.3 JUSTIFICACIÓN

Dada la complejidad y la gran cantidad de preguntas alrededor de este tipo de experimentos y considerando que para obtener resultados se requiere de largos periodos de tiempo (dos a cuatro años), el Dr. Romo solicitó la construcción de un nuevo laboratorio.

El diseño del Sistema que actualmente se utiliza ha sido de gran ayuda al Laboratorio, sin embargo, presenta una limitante muy importante, además de ser muy complejo su funcionamiento. Sólo permite realizar tres tipos de tareas: detección, categorización y discriminación de frecuencias de estímulos táctiles; y cada una estas tareas tiene un programa específico. Por lo que modificar algo en la tarea, o querer una nueva tarea implica:

- ♦ El diseño y la programación de la tarea, la modificación en el código para cada una de las aplicaciones de las máquinas que integran el Sistema, así como el acondicionamiento de los dispositivos electrónicos. Esto requeriría del trabajo de una persona con conocimiento profundo sobre la estructura del Sistema.
- ♦ Incrementar el código de programación y la complejidad del Sistema, lo cual aumenta la probabilidad de que ocurran errores, haciéndolo un Sistema poco confiable y limitando su crecimiento.
- ♦ Otro aspecto importante de considerar es el tiempo. El diseñar, programar y tener que acoplar todo el Sistema nuevamente requiere de tiempo y uso del Laboratorio, por lo que los experimentos se detienen y por consiguiente existe un atraso en la investigación.

El Sistema que actualmente se utiliza también presenta serias limitaciones con respecto a la presentación (graficación y análisis) de los datos psicofísicos dado que se dió preferencia a los datos neurofisiológicos.



1.4 ALCANCES

- ♦ Desarrollar un Sistema que permita construir y ejecutar un gran número de tareas. Así como coleccionar y graficar los datos que el investigador requiera.
- ♦ Reducir la complejidad del Sistema actual.
- ♦ El Sistema deberá permitir crear una nueva tarea de una manera sencilla y amigable.
- ♦ El Sistema permitirá la fácil modificación de una tarea. Sólo se modificarán parámetros desde una aplicación, sin necesidad de hacerlas en el código de alguno de los programas de las diferentes computadoras del Sistema.

1.5 RESTRICCIONES

El Sistema actual es capaz de ejecutar tres tipos de tareas (cada una tiene un programa diferente), controlar la realización de éstas, generar estímulos táctiles y aplicar pulsos de corriente en la corteza cerebral del sujeto; así como registrar las respuestas motoras y neuronales.

Debido a la complejidad del Sistema la construcción del nuevo Sistema se dividirá en dos proyectos. Para el desarrollo de esta tesis sólo nos correspondió implementar la parte relacionada con la ejecución de un gran número de tareas y el registro de las respuestas motoras. La aplicación de pulsos de corriente en la corteza cerebral y el registro de respuestas neuronales serán los temas de otro proyecto.



2 MÉTODO



MÉTODO

Los Sistemas de software requieren un tiempo considerable para su construcción. En este período de construcción pueden identificarse varias etapas, que juntas constituyen lo que se llama **el ciclo de construcción de los Sistemas**. Este ciclo es un enfoque por etapas de análisis y de diseño, que postula que la construcción de los Sistemas mejora cuando existe un ciclo específico de actividades del analista y de los usuarios.

Los analistas no están de acuerdo respecto al número exacto de etapas que conforman el ciclo de desarrollo de los Sistemas; sin embargo, se reconoce la importancia de su enfoque sistemático. (6)

Para construir el Sistema utilizaremos las etapas del ciclo de desarrollo de los Sistemas que en la asignatura de Ingeniería de Programación nos enseñaron, y el cual consta de las siguientes etapas:

- **Análisis**
- **Diseño**
- **Desarrollo**
- **Mantenimiento**

Análisis

En esta etapa se hace una investigación exhaustiva de los requerimientos del Sistema. Pasos que la componen:

- ◆ Estudio preliminar
- ◆ Análisis detallado
- ◆ Planeación
- ◆ Propuesta
- ◆ Diagrama de flujo de datos
- ◆ Evaluación de *hardware* y *software*

Diseño

Esta etapa de diseño se divide en:

Diseño Interno (programación): se encarga de la parte modular del Sistema de información, es aquí donde se sabe cuántos y cuáles módulos se van a desarrollar. También se crea un diagrama de modelado de datos para crear la estructura de las bases de datos del Sistema.

Diseño Físico (vista del Sistema): es la creación de una interface a usuario que incluye pantallas, menús y reportes, así como una selección de colores, modelos y distribución de funciones que haga al Sistema de información amigable y agradable, sin descuidar la funcionalidad y eficiencia del mismo.

Por lo que el diseño tiene entonces los dos siguientes pasos:

- ◆ Modelado de datos E/R
- ◆ Diagrama estructurado



Desarrollo

Esta formado por las siguientes actividades:

- ◆ Programación de módulos.- se codifican los programas.
- ◆ Plan de pruebas.- se utiliza para evaluar la seguridad y eficacia de la entrada y salida de datos en un Sistema de información.
- ◆ Plan de instalación.- se realizan todas las tareas necesarias para la implantación y proporciona un periodo inicial de soporte al Sistema. La introducción de un Sistema implica un nuevo tipo de uso y de operación que deberá ser asimilado y aprendido por los usuarios y operadores.
- ◆ Plan de liberación.- se asegura que el Sistema funcione correctamente en la mayoría de los casos. Para esto se realizan nuevas pruebas, se reevalúan los resultados y se hacen refinamientos del Sistema, los cambios necesarios deberán ser introducidos sin afectar a los usuarios. El resultado de esta etapa proporciona un Sistema listo para su operación.
- ◆ Manual de usuario.

Mantenimiento

Esta etapa se refiere al proceso de modificar un programa cuando este se encuentra en operación, esas modificaciones pueden implicar cambios sencillos en la codificación, cambios mayores de diseño o de especificación de requisitos; o bien puede tener como objetivo corregir errores, de los cuales se derivan tres tipos de mantenimiento:

- ◆ Mantenimiento correctivo: proceso que incluye el diagnóstico y la corrección de uno o más errores.
- ◆ Mantenimiento adaptativo: proceso que modifica el software para que interaccione adecuadamente con su entorno cambiante.
- ◆ Mantenimiento perfectivo: se produce cuando un software tiene éxito; con base a las recomendaciones de los usuarios se modifica las funciones ya existentes.



3 APLICACIÓN DEL MÉTODO



3.1 ANÁLISIS

3.1.1 ESTUDIO PRELIMINAR

El objetivo de un proyecto en el área de Neurofisiología es poder correlacionar la actividad eléctrica de una neurona o de una población de ellas con algún aspecto de la conducta o bien con los parámetros de los estímulos que se le presentan al sujeto. Aún cuando el interés está centrado en conocer el funcionamiento del cerebro humano, la tecnología actual no permite el registro de la actividad en el hombre ya que la introducción de un electrodo en la corteza cerebral implica un daño irreparable, por lo cual se trabaja con modelos animales. Los investigadores de esta área normalmente condicionan la conducta de un primate subhumano (monos rhesus (*Macaca mulatta*), monos aulladores (del género *Alouatta*), monos macaco come cangrejos (*Macaca fascicularis*), entre otros) para ejecutar una serie de acciones como respuesta a la presentación de un grupo de estímulos. A esa secuencia de estímulos y respuestas se le conoce como una *tarea*.

Los investigadores en el área de las Neurociencias cognitivas han diseñado un gran número de tareas, pero en general se pueden clasificar en tres grandes grupos:

Detección: el objetivo de esta tarea es determinar la capacidad del sujeto (humano o mono) para percibir la presencia de un estímulo. Por lo tanto la tarea consiste en presentarle al sujeto un estímulo (puede ser táctil, visual, auditivo o de otra modalidad sensorial) de cierta magnitud y éste debe reportar si fue capaz de sentir ese estímulo; se procede a realizar en varias ocasiones o *ensayos* la misma tarea pero variando en forma controlada la magnitud del estímulo, finalmente se realiza un análisis estadístico del experimento con el fin de determinar el "umbral de detección de la variable en cuestión". Con experimentos de este tipo se ha determinado por ejemplo que el hombre tiene la capacidad de distinguir amplitudes de estímulos mecánicos aplicados sobre las yemas de los dedos del orden de dos a tres micras. (7)

Categorización: su objetivo es determinar la capacidad de un sujeto para clasificar una serie de estímulos, normalmente en dos categorías, por ejemplo, la velocidad de un estímulo táctil (velocidades altas o bajas), o bien clasificar la dirección de un grupo de puntos (estímulo visual, en una pantalla de televisión) hacia la izquierda o hacia la derecha. La tarea consiste en presentar al sujeto un estímulo para que posteriormente manifieste la categoría a la cual supuestamente pertenece, obviamente primero se le muestra como son los estímulos que pertenecen a una u otra clase. Durante las diferentes repeticiones o *ensayos* de la tarea se van presentando diferentes magnitudes del estímulo, de tal forma que existen valores de fácil clasificación, con los que el sujeto no se equivoca, sin embargo existen otros valores que es difícil distinguir si pertenecen a una u otra categoría por lo cual el sujeto se equivoca al emitir sus juicios. Al analizar estadísticamente esos resultados se logra determinar el "umbral de categorización para la variable bajo estudio". Con esta metodología se ha logrado determinar por ejemplo que los humanos poseemos un umbral de categorización de velocidades del orden de 6 mm/s.



Discriminación: en esta tarea se presentan en cada ensayo dos estímulos (que pueden ser táctiles, visuales o de otra modalidad sensorial) al sujeto y él debe compararlos para indicar cual de los dos es de mayor magnitud, o en su caso si esos estímulos son iguales o diferentes. La principal ventaja de esta tarea es que permite estudiar funciones de alto orden como son la memoria y la comparación.

(8)

La actividad de las neuronas en diversas regiones de la corteza cerebral del sujeto, que se encuentra resolviendo cierta tarea, es registrada con microelectrodos. Tanto los parámetros del estímulo como las respuestas motoras y las descargas neuronales son colectadas en tiempo real.

Es importante señalar que si bien los estudios neurofisiológicos se realizan en primates, los investigadores primero realizan exhaustivos estudios de psicofísica en humanos a fin de determinar las mejores condiciones para realizar los experimentos de tal forma que se garantice que los monos podrán resolver la tarea y por lo tanto que el experimento tendrá éxito.

3.1.1.1 Sistema Actual

El Sistema actual está constituido por seis computadoras y seis dispositivos electrónicos.

Computadoras:

- La máquina de interface de usuario (IU)
Interactúa con el usuario. Acepta los parámetros requeridos para el experimento, presenta y almacena los datos electrofisiológicos.
- La máquina control (Controlador)
Se encarga de la secuencia de la tarea activando los dispositivos que generan los estímulos (mecánicos y visuales), además de evaluar las respuestas motoras del sujeto.
- La máquina generadora de estímulos (Generador)
Produce las funciones rampa y, a través del Sintetizador, la vibración, los cuales son enviados al motor lineal.
- La máquina de microestimulación (Microestimulador)
Activa el microestimulador, el cual aplica trenes de pulsos de corriente en la corteza cerebral del sujeto, a través de los microelectrodos.
- La máquina de control de microelectrodos
Controla los dispositivos electrónicos y mecánicos que permiten introducir los microelectrodos a la corteza cerebral y por lo tanto colectar la actividad eléctrica neural.
- La máquina colector de datos (Colector de Datos)
Registra los pulsos digitales correspondientes a las respuestas motoras y neuronales, previamente acondicionadas por un dispositivo electrónico.



Dispositivos:

- **Matriz**
Es un sistema electromecánico compuesto por una unidad de potencia que mueve a un motor de pasos, éste último es el responsable de crear el desplazamiento controlado de los micro electrodos con una resolución de una micra, el sistema mecánico es un conjunto de placas, engranes, frenos y embragues que permite el control de siete micro electrodos en forma independiente.
- **Micro estimulador**
Es un dispositivo electrónico, eléctricamente aislado del resto del sistema y que es usado para inyectar pulsos de corriente a la corteza cerebral a través de los micro electrodos.
- **Sintetizador**
Es un generador de señales con la capacidad de presentar formas de onda senoidal, cuadrada y triangular; con las siguientes especificaciones:
Amplitud: rango 0 a ± 20 V, resolución 1 mV.
Frecuencia: rango 0 a 600 kHz, resolución 0.1 Hz
Fase: rango de 0 a 359.9 grados, resolución de 0.1 grado.

Este dispositivo tiene integrado una interfaz para ser programado usando el protocolo GPIB, y permite almacenar hasta dieciséis conjuntos de parámetros (forma, amplitud, frecuencia y fase) los cuales pueden ser seleccionados mediante un puerto digital (dieciséis combinaciones) con un tiempo de respuesta de ocho micro segundos. Finalmente permite, mediante una terminal especial, el establecer la fase de la onda al valor predeterminado (*phase reset*) con un tiempo de reacción de dos micro segundos.
- **Motor Lineal**
Este dispositivo es usado para aplicar las vibraciones mecánicas sobre la yema de alguno de los dedos del sujeto experimental, su funcionamiento se basa en el principio de la fuerza de Lorentz, y se compone de una bobina móvil que se encuentra dentro de un cilindro con un campo magnético permanente. Al hacer circular una corriente a través de la bobina se produce el desplazamiento de la misma y el de la flecha a la cual se encuentra unida, la punta (punta de prueba) de esa flecha es de dos milímetros de diámetro, es de cerámica y es redondeada a fin de no provocar daño sobre la piel.
- **Control del Motor Lineal**
Este dispositivo electrónico controla el desplazamiento del motor lineal mediante una relación simple entrada/salida la cual especifica que, por cada 5 mV que se apliquen a la entrada, la punta del motor lineal se desplazará una micra.
- **Registro de respuestas motoras**
Este dispositivo electrónico permite identificar las respuestas motoras del sujeto que pueden ser retirar o colocar su mano sobre una palanca inmóvil o



bien presionar uno de los cuatro botones usados para manifestar su decisión, además de generar una solicitud de interrupción de hardware para que se registre el instante en el cual esos eventos suceden.

En la Fig. 3 se ilustra el Sistema actual.

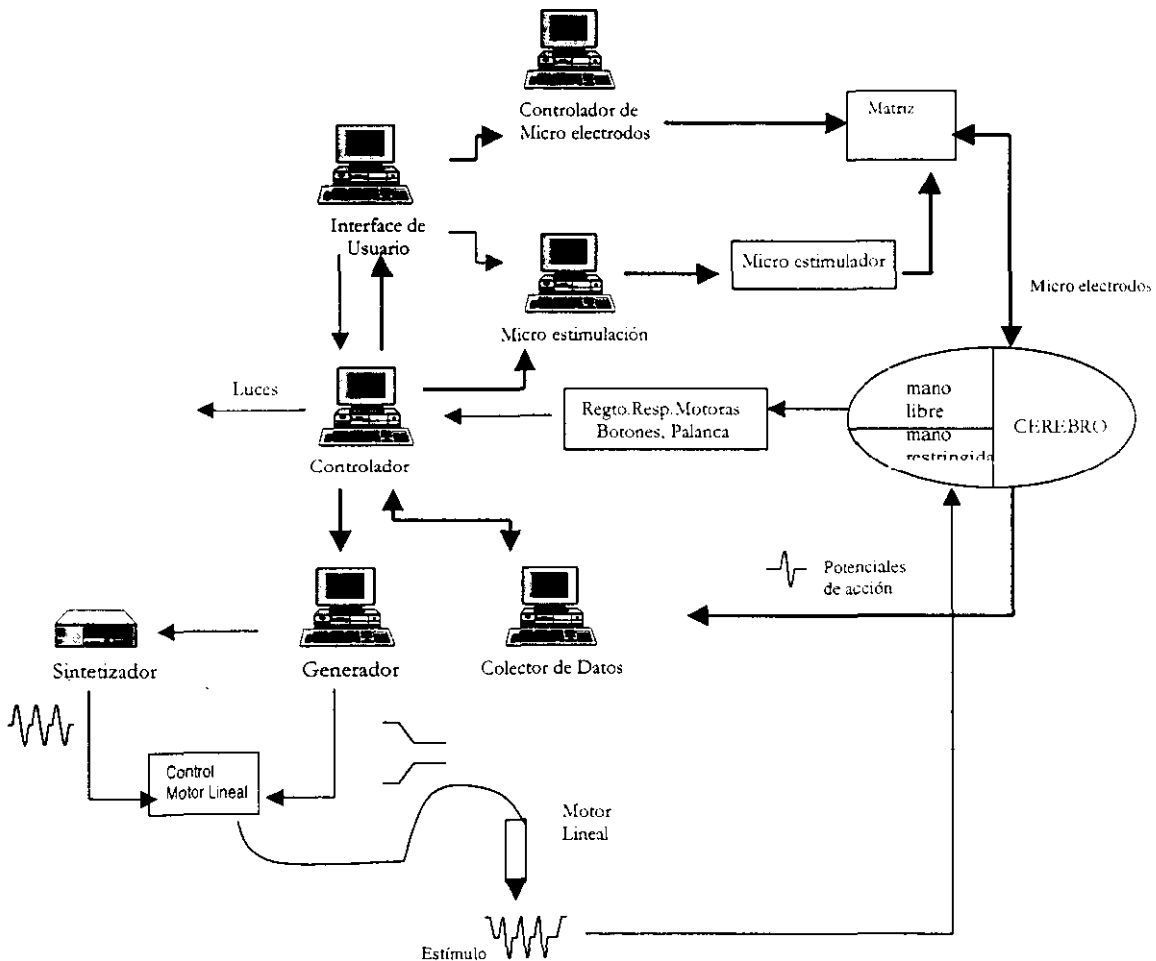


Fig. 3 Diagrama del Sistema

Para dar inicio a un experimento se debe cargar en el Controlador el programa que lleva la secuencia del experimento, y en la IU los archivos que contienen los parámetros que requiere el experimento. La ejecución de un experimento se



puede dividir en dos etapas; en la primera la IU verifica que sus archivos sean los correctos para el experimento que le fue cargado al Controlador. Si así fue la IU envía información al Micro estimulador y al Controlador. A su vez, el Controlador manda los datos correspondientes al Generador y al Colector de Datos; y del Generador al Sintetizador. En la segunda etapa una vez que la información ha sido distribuida a cada una de las máquinas, el Controlador lleva la secuencia de la tarea, dependiendo de las respuestas del sujeto, indicando en que momento se debe generar un estímulo, una micro estimulación, o coleccionar datos. Cuando finaliza la tarea el Controlador se lo indica a la IU quien pide sus datos al Control de Micro electrodos y al Colector de datos para finalmente presentar la gráfica correspondiente, y almacenar esa información en un archivo.

3.1.2 ANÁLISIS DETALLADO

La investigación en Neurociencias es demandante, dada la complejidad de los procesos cognitivos. Una vez que se ha determinado la metodología a seguir durante el experimento, la actividad neuronal puede revelar procesos que en principio no se tomaron en consideración y que exigen la modificación del experimento (tarea) a fin de obtener la mayor cantidad posible de información del fenómeno. Por esto se busca que el Sistema sea muy flexible y permita construir un gran número de tareas.

Se decidió tratar de identificar aquellos elementos que fuesen comunes a todas las tareas a fin de generar una estructura basada en bloques funcionales, es decir elementos básicos con características bien definidas a partir de las cuales se pudiesen construir estructuras de mayor complejidad. Para ello analizamos la tarea de detección y discriminación.

Tarea de Detección

En esta tarea al sujeto se le presenta un estímulo vibro-táctil en una mano, a la cual se le ha restringido su movimiento, para indicar que ha detectado el estímulo, el sujeto realiza un movimiento, que consiste en colocar su otra mano libre sobre una palanca.

La secuencia de presentación es la siguiente:

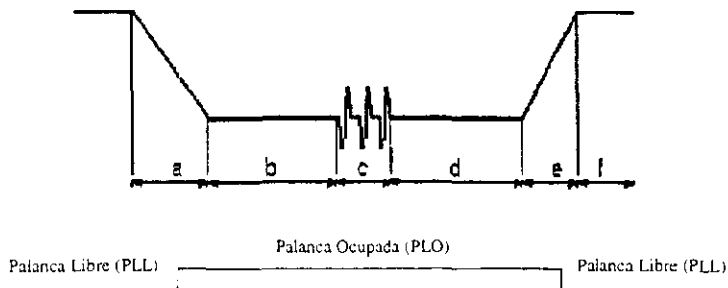


Fig. 4



Se provoca que la punta del motor lineal haga contacto con la yema de uno de los dedos del sujeto (aplicando una rampa negativa en función del tiempo al control del motor, Fig. 4.a). El sujeto debe indicar la detección de la punta de prueba colocando su mano libre sobre la palanca (PLO). Después de un intervalo de tiempo variable (Fig. 4.b) se presenta el estímulo (Fig. 4.c) el cual hace que la punta de prueba se mueva siguiendo un patrón determinado. Después de la generación del estímulo táctil se produce un intervalo de tiempo variable (Fig. 4.d), seguido de una rampa ascendente (Fig. 4.e) que hace que la punta de prueba se levante. En este momento el sujeto debe de indicar la detección del estímulo soltando su mano de la palanca (PLL) y después de esto se produce un último intervalo de tiempo en el cual se evalúa si el sujeto detectó a tiempo el estímulo. Si es así el sujeto recibe como recompensa unas gotas de agua.

Tarea de Discriminación

En esta tarea al sujeto le son presentados dos estímulos vibro-táctiles que pueden ser diferentes o iguales con respecto a cierto parámetro, de interés para el investigador, que constituye al estímulo como la frecuencia o la amplitud. Los estímulos son presentados en la mano inmovilizada y con la otra mano libre indica cual de los dos estímulos tiene la frecuencia más alta, oprimiendo uno de los botones.

Para llevar a cabo esta tarea el investigador diseña la secuencia de presentación de la siguiente forma:

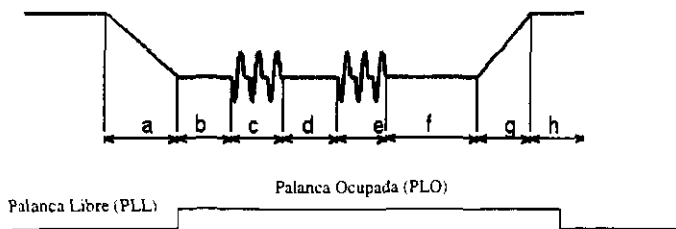


Fig. 5

La tarea comienza cuando se produce una rampa descendente (Fig. 5.a), la cual bajara la punta de prueba del motor lineal que hace contacto sobre la superficie de la piel de uno de los dedos, de la mano inmóvil (PLO). El sujeto debe indicar la detección de la punta de prueba colocando su otra mano libre sobre la palanca (PLO). Después de un intervalo de tiempo variable (Fig. 5.b), se presenta el primer estímulo (Fig. 5.c) el cual hace que la punta de prueba se mueva siguiendo cierto patrón determinado. Seguido del primer estímulo se genera otro intervalo de tiempo variable (Fig. 5.d), el cual indica la separación de primer estímulo y el segundo. El siguiente estímulo (Fig. 5.e) es generado haciendo que la punta de prueba nuevamente se mueva. Después de la generación de estímulos táctiles se produce un intervalo de tiempo variable (Fig. 5.f), seguido de una rampa ascendente (Fig. 5.g) que hace que la punta de prueba se levante. En este momento el sujeto debe de indicar el resultado de la comparación de los



estímulos soltando su mano de la palanca y proyectándola hacia uno de los botones (BTx). Después de esto se produce un último intervalo de tiempo en el cual se evalúa si la respuesta fue correcta; si es así el sujeto recibe como recompensa unas gotas de agua.

Con base en lo anterior identificamos los siguientes elementos comunes, mostrados en la Fig. 6:

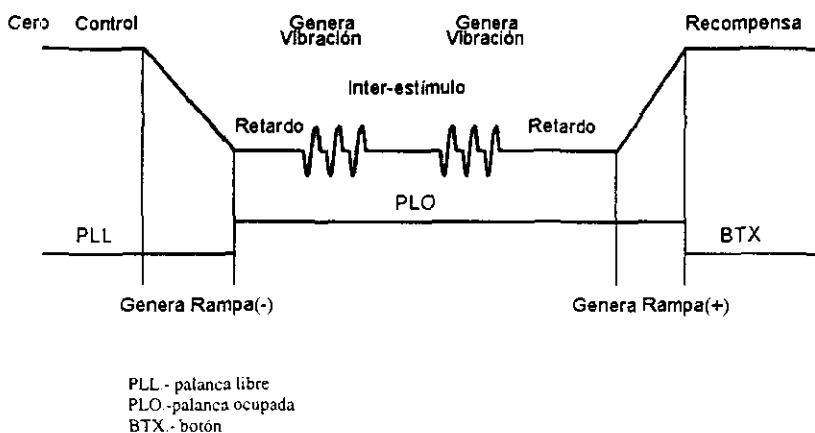


Fig. 6

3.1.2.1 Definición de estado

Encontramos que es posible dividir una tarea, ya sea de detección, discriminación o categorización, en periodos con características bien definidas ya que se puede identificar qué eventos se presentan y qué respuestas se esperan dentro de ellos. A cada uno de estos periodos le llamamos estado. Entonces podemos definir al estado como una condición estable del Sistema, así existen estados como: estimulando, esperando evento, etc.

Características de los estados:

En la tarea de Discriminación, por ejemplo, se pueden identificar los siguientes estados:

Estado Control.- En este estado aún no ha bajado la punta. Sirve para coleccionar la actitud de las neuronas, fuera del contexto de la tarea.



Estado Retardo.- Espera que aparezca el estímulo rampa. Es necesario para que las neuronas se estabilicen, y así poder evitar la interferencia con la reacción que tienen cuando la punta baja.

Estado Rampa.- Estado en donde se genera el estímulo rampa: es decir la punta se desplaza hasta hacer contacto con la piel o se desplaza dejando de hacer contacto con ésta.

Estado Vibración.- Estado en donde se generan los estímulos táctiles.

Estado Inter-estímulo.- Período para poder separar e identificar adecuadamente los eventos generados por los dos estímulos táctiles.

Estado Recompensa.- Dependiendo de la tarea se espera que la palanca se desocupe, o que se oprima alguno de los cuatro botones (Discriminación); si el sujeto da la entrada que se espera el Sistema lo recompensará con una gota de agua.

Hay un estado especial y necesario al que llamamos Estado Cero; este estado verifica que la palanca esté libre antes de iniciar la secuencia.

Estado Inter-ensayo.- Es el tiempo necesario entre el fin de una secuencia y el inicio de otra. El Sistema espera cierto tiempo antes del siguiente estado Cero.

Clasificación de los estados:

Los estados anteriores se clasificaron en cuatro tipos:

Estado Espera.- Su principal característica es que el Sistema permanece en "reposo" durante un determinado período. Este estado engloba al Estado Control, Estado Retardo, Estado Inter estímulo, y el Estado Inter ensayo.

Estado Rampa.- Es el estado en el que se genera el estímulo rampa (que baja o sube la punta), para producir o evitar el contacto con la piel del dedo.

Estado Vibración.- En este estado se generan los estímulos vibro-táctiles

Estado Recompensa.- En este estado se evalúa la decisión del sujeto y en su caso se le da recompensa.

3.1.2.2 Definición de tarea

Una tarea se define como una secuencia ordenada de estados, que permiten la presentación de los estímulos táctiles al sujeto y se registran sus respuestas neuronales y motoras.



3.1.2.3 Definición de clase

El investigador puede estar interesado en estudiar varios procesos neurales o conductuales al mismo tiempo o puede requerir estudiar detalladamente uno en particular, por lo que debe contar con la flexibilidad suficiente para modificar la mayor cantidad de características posibles durante un experimento. Por ejemplo, si el investigador quisiera estudiar las respuestas al variar dentro de un intervalo de valores cierto parámetro que define al estímulo, como la amplitud, cada uno de los diferentes valores se debe establecer en lo que llamamos clase. Los valores de la clase sirven para presentar una vez la secuencia completa de estados. El conjunto de datos que se pueden variar durante una experimento está contenido en la clase, la información del conjunto de clases lo guardaremos en un "Archivo de clases".

3.1.2.4 Definición de ensayo

Es cada presentación completa de la secuencia de estados de una tarea con los valores de determinada clase.

3.1.2.5 Definición de experimento

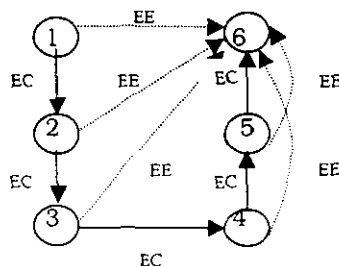
Un experimento corresponde a la presentación de m ensayos de cada una de las n clases en un tiempo determinado. Con lo cual se puede realizar un análisis estadístico de los datos, que se obtuvieron en el experimento. Generalmente se repite 10 veces cada clase.

3.1.2.6 Secuencia de estados

Como ya se ha mencionado una tarea se forma por una secuencia de estados, esta secuencia depende de las respuestas del sujeto (eventos). Por lo que cada estado requiere saber que evento es el que necesita para pasar al siguiente estado. Si se da el evento esperado el siguiente estado será el Estado Correcto, si no se da éste, el siguiente estado es el Estado Erróneo.

3.1.2.6.1.1 Diagrama de Estados

A continuación pondremos como ejemplo la tarea X, la cual inicia en el estado número 1 y termina en el número 6. Se recorren todos los estados en forma consecutiva siempre que se den los eventos correctos; al no darse estos el siguiente será el número 6, como se muestra en la Fig. 7:



Donde:
 EC.- Evento Correcto
 EE.- Evento Erróneo
 Eventos: PLL, PLO, BOTÓN, DRE

Fig. 7



3.1.2.7 Análisis de la construcción y ejecución de un experimento

Como ya vimos una tarea se constituye por una secuencia de estados; para definir ésta es necesario saber a qué estado debe pasar, y que entrada esperar.

Para construir una tarea se debe definir:

- El número de estados que conforman la tarea.

Y para cada estado:

- El evento, la respuesta que da el sujeto y que el Sistema espera para poder pasar al estado siguiente. Los eventos son:
 - Que el sujeto coloque su mano libre en la palanca, PLO.
 - Que el sujeto retire su mano libre de la palanca, PLL.
 - Que el sujeto oprima con su mano libre un botón, BTN. Se tiene hasta cuatro botones disponibles.
 - Que se cumpla la duración del estado, DRE. Cada estado tiene una duración; éste es el único evento que no lo proporciona el sujeto, sino el Sistema.
- Estado Correcto, es el estado al que se pasa siempre que se dé el evento correcto.
- Estado Erróneo, es el estado al que se pasa si no se da el evento esperado.
- Tipo de Estado, se debe establecer que tipo de estado es: una rampa, espera, vibración o recompensa. Cada estado realiza determinada función dependiendo del tipo de estado. Por ejemplo, el estado rampa genera el estímulo rampa; el estado vibración, la vibración; y el estado recompensa evalúa la respuesta, si ésta fue correcta proporciona el premio al sujeto.

En la Tabla 1 se ha representado en las columnas cada estado, en los renglones las características que definen a cada estado y la transición del estado se genera al presentarse el evento esperado.

Tabla 1

Estado 1	Estado 2	...	Estado n
Evento	Evento	Evento	Evento
Edo. Correcto	Edo. Correcto	Edo. Correcto	Edo. Correcto
Edo. Erróneo	Edo. Erróneo	Edo. Erróneo	Edo. Erróneo
Tipo de Edo.	Tipo de Edo.	Tipo de Edo.	Tipo de Edo.



Ahora bien, como ya se mencionó el investigador requiere contar con la flexibilidad suficiente para modificar la mayor cantidad de características posibles durante un experimento. Estas características se definen en las clases. Por lo tanto, para cada tarea debemos definir sus clases.

Clases

Las características que definen a un estado, y lo hacen diferente de ensayo a ensayo se establecen en las *clases*. Una tarea puede tener n clases; permitiendo así la variedad de ensayos en un experimento.

Estas características son:

- *La duración del estado*

Cada estado tiene su propia duración, y no tiene que ser igual a la de los demás estados. Además, esta duración es diferente para el mismo estado de ensayo a ensayo; pero su valor siempre estará dentro de un intervalo que especifique el investigador.

Por lo tanto cada estado requiere de un valor mínimo, un máximo y un valor que permita cuantizar este intervalo (quantum), de tal forma que: al iniciar un ensayo, se dividirá el intervalo comprendido entre los valores máximo y mínimo en segmentos de acuerdo al quantum, y se elegirá al azar uno de esos segmentos; finalmente, la duración del estado será la suma del valor mínimo y el valor seleccionado al azar. Todos expresados en milisegundos.

- *Límite inferior de respuesta*

Este valor evita respuestas anticipadas del sujeto, por lo tanto, si el evento se presenta en un tiempo menor a este valor, el Sistema considera dicha respuesta como temprana (EARLY). Existe de igual forma un límite superior de respuesta, pero ese valor corresponde a la duración del estado, si la respuesta del sujeto se manifiesta en un instante posterior a ese valor, entonces se considera como respuesta tardía (LATE).

- *Características de los estímulos mecánicos*

Estas características son necesarias siempre que se trate de un estado Rampa o Vibración, que son los dos estímulos mecánicos que se tienen.

- *Patrón de luces*

Se refiere a los instantes de encendido y apagado de uno o algunos de las cuatro luces que se encuentran dentro de los botones.

- *Evento recompensa*

Si el estado es se debe establecer qué evento se espera que el sujeto proporcione para que éste sea premiado.

Estímulos mecánicos:

♦ **Rampa** (Fig. 8)

- Pendiente de la rampa.- Puede ser positiva (ascendente) o negativa (descendente).
- Duración de la Rampa.- Expresada en milisegundos (50 – 500 ms).
- Amplitud de la rampa.- Expresada en micras (100 – 1800 micras).

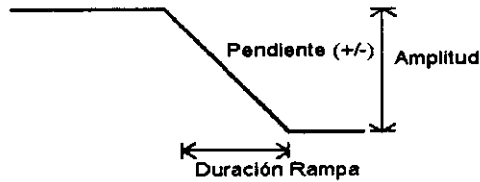


Fig. 8

♦ **Vibración**

Una vibración puede ser periódica o aperiódica. Esta clasificación indica si los intervalos entre los diferentes pulsos mecánicos que componen al estímulo son regulares o no.

Periódica (Fig. 9)

- Amplitud del tono.- Expresado en micras (0 – 200 micras).
- Frecuencia del tono.- Expresado en Hz (40 – 50 Hz).
- Frecuencia media.- Expresado en Hz (6 – 46 Hz).
- Duración del estímulo.- Expresado en milisegundos.

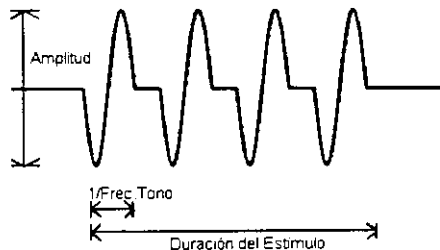


Fig. 9

Aperiódica (Fig. 10)

Este estímulo el sujeto lo percibe como pulsos distribuidos en forma aleatoria, sin embargo el investigador tiene control absoluto sobre la posición de cada uno de los pulsos.



- Amplitud del tono.- Expresado en micras (0 – 200 micras).
- Frecuencia del Tono.- Expresado en Hz (40 – 50 Hz).
- Núm. de Casillas.- Representa el número de tonos máximo que pueden haber durante la duración del estado.

Con la Frecuencia de Tono y el Núm. de Casillas se obtiene la Duración del Estímulo.

$$\text{Duración} = \text{Núm. Casillas} * \text{Frec. Tono}$$

- Distribución Temporal.- Es una cadena de 1 y 0, donde el 1 representa la presencia de un tono y un 0 un tiempo muerto.

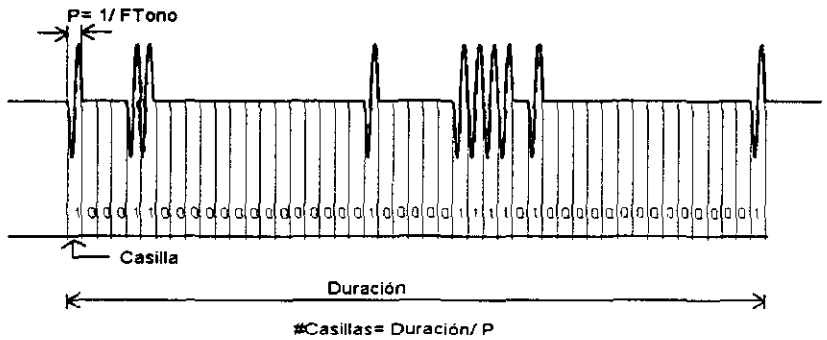


Fig. 10

Azarosa (Fig. 11)

Esta vibración es muy semejante a la Aperiódica; mientras que en la Aperiódica la presencia o ausencia de tonos en determinado momento es establecida por el investigador, en la Azarosa la posición del tono y el tiempo muerto se establecen al azar mediante un algoritmo de números pseudoaleatorios.

- Amplitud del tono.- Expresado en micras (0 – 200 micras).
- Frecuencia del tono.- Expresado en Hz (40 – 50 Hz).
- Frecuencia media.- Expresado en Hz (6 – 46 Hz).
- Duración del estímulo.- Expresado en milisegundos.

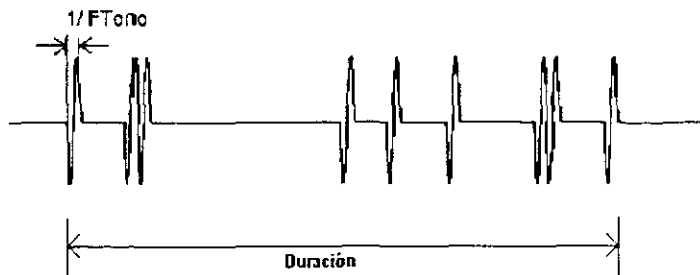


Fig. 11



Bipulso (Fig. 12)

Este tipo de vibración solo presenta dos pulsos uno al inicio y otro al final del estado. Fue necesario crear este otro estímulo debido a la tarea que requería la investigación. En los otros tipos de vibraciones se presenta un tono completo, es decir, su parte positiva y su parte negativa, en la vibración bipulso solo se presenta la parte negativa.

- Amplitud del tono.- Expresado en micras (0 – 200 micras).
- Frecuencia del tono.- Expresado en Hz (40 – 50 Hz).
- Duración del estímulo.- Expresado en milisegundos.

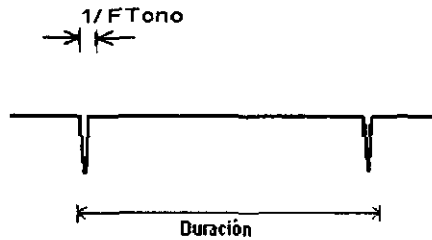


Fig. 12

Ejecución de experimentos

Para realizar un experimento se deberá establecer la tarea y las clases, el número de ensayos por clase y la secuencia de presentación.

Número de ensayos

Como ya se mencionó un experimento se forma de muchas corridas, cada corrida por n clase y *cada clase se repite m veces* de tal forma que cada corrida tiene $n*m$ ensayos. El investigador establecerá el número de ensayos al definir el número de repeticiones – m – para cada clase.

3.1.2.8 Secuencia de presentación

Al inicio del experimento (una vez que se han dado de alta la tarea y los parámetros de las clases), se procederá a crear una lista que representa la forma en la que las diferentes clases serán presentadas. Se debe contar con dos opciones:

Aleatoria.- En este modo, se utiliza un generador de números aleatorios para determinar el orden de presentación. Se debe tomar en cuenta que todas las clases se tienen que presentar antes de volver a repetir una de ellas.



Secuencial.- Aquí, el orden de presentación es secuencial, presentando primero la clase 1, luego la 2, y así sucesivamente.

Una vez que se haya dado inicio al experimento se procederá a distribuir la información al resto del Sistema; se enviará la bandera de inicio de ensayo y se espera la del fin del mismo. Para cada ensayo se colectará el resultado y ciertos parámetros de éste.

3.1.2.9 Análisis del control de un experimento

Como ya sabemos un experimento se forma con $n*m$ ensayos; donde n es el número de clases y m las repeticiones de éstas.

La aplicación que se va a encargar de controlar el experimento se encontrará en la Interface de Usuario (Constructor); establecerá cual es la clase que se debe ejecutar y la enviara al Controlador, además permitirá que el experimento pueda ser pausado o detenido. Pero quien lleva el control de cada uno de los ensayos es el Controlador.

Controlar el ensayo significa dirigir la secuencia de eventos durante la ejecución de éste. El Controlador ejecutará ensayo tras ensayo y para cada uno de ellos la Interface de Usuario le mandará los parámetros de la clase correspondiente.

Para el Controlador es importante conocer que estado es cada uno, así como cual entrada (PLL, PLO, BOTÓN ,DRE) debe esperar en estos para decidir a que estado pasar. si la entrada que se presentó es la que esperaba pasará al estado correcto, si no al estado erróneo; las entradas son interrupciones de *hardware* provocadas por la electrónica asociada a la palanca, a los botones y al temporizador. Además requiere del tiempo de duración del estado y el menor tiempo que debe esperar por una respuesta para poder decidir si la respuesta fue: en tiempo, temprana (EARLY) o tardía (LATE). Si la respuesta se da dentro del periodo adecuado, se debe evaluar si se presentó la entrada que permite la continuidad del ensayo, de lo contrario se aborta el ensayo dando lugar a un estado final adecuado. Requiere saber que tipo de estado es, ya que dependiendo de éste realizará determinadas acciones como por ejemplo, si es un estado Vibración o Rampa indicarle al Generador de Estímulos que los genere, o si es un estado de Recompensa evaluar si la respuesta es correcta (HIT) y activar la recompensa o errónea (WRONG) y no dar recompensa.



3.1.2.10 Almacenamiento de la información

Como podemos darnos cuenta son varios los parámetros que se necesitan para realizar un sólo ensayo. Para llevar a cabo un experimento completo necesitaremos de una gran cantidad de estos. Es necesario almacenar toda la información que tiene que ver con una tarea, y poderla recuperar cada vez que se requiera. Además es deseable que presente una estructura ordenada y coherente, por esto, todos los parámetros deben ser almacenados en una Base de Datos.

Por lo anterior se requieren dos tipos de tablas, la tabla tarea y la tabla clase. Los campos que cada una de estas tablas necesitan son los siguientes:

Numero	Nombre	Evento	EdoCorrecto	EdoErróneo	TipoEdo

Tabla tarea

Nombre	ValMin	ValMáx	Quantum	Evento	LimIniResp	Patrón	Pendiente	DurRampa	Amplitud	FrecTono	FrecMedia	DistTemp

Tabla clase

Los registros de la tabla tarea representan los estados que la conforman; los registros de la tabla clase, cada una de las clases que conforman el archivo de clases.

Al iniciar un experimento se abrirá la tabla que contiene la secuencia de la tarea que se quiere realizar, así como la tabla de clases. De estas tablas se tomarán los valores que se enviarán tanto a la máquina de Control como a la Generadora de estímulos.

Se requiere de un *software* que permita manipular bases de datos. Visual C++ proporciona varias formas de trabajo con bases de datos. Utilizando la biblioteca de clases MFC (*Microsoft Foundation Class*), se puede recurrir a las clases DAO (*Data Access Objects* -objetos de acceso a datos) o a las clases ODBC (*Open DataBase Connectivity*- conectividad abierta de bases de datos).



ODBC (Conectividad Abierta de Bases de Datos)

Es un estándar internacional para manipular datos relacionales que utilizan la sintaxis de consultas SQL en orígenes de datos dispares, ODBC tiene la ventaja de ser un estándar internacional que permite manipular un amplio número de orígenes de datos relacionales mediante diversos controladores ODBC de Microsoft y de otros fabricantes. El principal inconveniente de ODBC es que limita a datos relacionales basados en la sintaxis de SQL.

La interoperabilidad que proporciona permite a los desarrolladores construir y distribuir aplicaciones cliente/servidor no ligadas específicamente a una base de datos determinada.

ODBC es útil cuando se desea crear aplicaciones en las cuales se utilizarán bases de datos independientes basada en archivos para el desarrollo y que posteriormente se unirán a un modelo cliente/servidor, evitando hacer enormes cambios a nivel de código.

DAO (Objetos de Acceso a Datos)

El DAO es un modelo de acceso a datos el cual admite dos entornos diferentes de bases de datos o "espacios de trabajo."

Los espacios de trabajo Microsoft Jet permiten tener acceso a datos en bases de datos Microsoft Jet, bases de datos Microsoft conectadas a ODBC y orígenes de datos ISAM instalable en otros formatos, como Paradox o Lotus 1-2-3.

Los espacios de trabajo ODBCdirect permiten tener acceso servidores de bases de datos mediante ODBC, sin cargar el motor de base de datos Microsoft Jet. El espacio de trabajo ODBCdirect proporciona una alternativa cuando sólo necesita ejecutar consultas o procedimientos almacenados en un servidor, como Microsoft SQL Server o cuando la aplicación cliente necesita las capacidades específicas de ODBC, como las actualizaciones por lotes o la ejecución de una consulta asíncrona.

¿Cuáles son las ventajas de usar una u otra herramienta? Básicamente las ventajas dependen del tipo de usuario para el que se realice la aplicación. El motor Jet funciona bien en monousuario y en redes pequeñas; tiene sus problemas de bloqueo cuando varios usuarios acceden a una misma información o incluso informaciones colindantes. Con bases de datos grandes es aconsejable trabajar con ODBC. La ventaja de trabajar con el motor Jet es que no se necesita depender de ningún *driver* ODBC externo, ya que las aplicaciones son más sencillas y dan menos problemas de configuración.

En nuestro caso utilizamos el método DAO, ya que sólo requerimos de un manejo sencillo de la base de datos; sólo necesitamos almacenar una colección de datos clasificados y estructurados.



3.1.2.11 Graficación de datos

Los datos, como antes se mencionó, serán presentados al investigador y almacenados; sin embargo, la graficación de estos datos le permiten de manera rápida y sencilla comprender lo que está sucediendo en la realización del experimento. Por esto, además de presentar en pantalla el resultado de cada ensayo, se presentará en ésta, de manera simultánea, una grafica con estos datos; la cual irá tomando los valores del archivo que se va creando durante el experimento. Se graficará esta información cada vez que el investigador lo solicite.

La graficación de los datos será en MatLab; por lo que es necesario que nuestro Sistema se comunique con esta aplicación y le indique que debe graficar los datos que está obteniendo. Se utilizará MatLab porque en el Laboratorio se trabaja con este software para el análisis de datos, y existen diversos programas que cada investigador a creado para sus propias necesidades. Estos programas podrían de alguna manera integrarse al Sistema, y los datos obtenidos por el Sistema ayudar a los investigadores sin necesidad de modificar considerablemente sus programas.

3.1.2.12 Comunicación entre computadoras

En el laboratorio se utiliza como estándar para la comunicación de computadoras una interface de National Instruments, el ANSI/IEEE Standard 488.1-1987, también conocido como GPIB (General Purpose Interface Bus). Así que para el nuevo laboratorio utilizaremos el mismo estándar.

3.1.3 PLANEACIÓN

El Laboratorio del Dr. Romo requiere de un Sistema confiable y flexible que permita al investigador la libertad para generar los estímulos necesarios con los valores deseados y que registre la máxima información conductual y neuronal posible con el fin de tener una probabilidad mayor de encontrar los códigos neurales que sustentan el proceso perceptivo.

El nuevo Sistema permitirá construir y ejecutar un gran número de tareas de manera sencilla y amigable, sin tener que crear o modificar un programa. El Sistema ejecutará un solo programa capaz de controlar diferentes tareas, en lugar de ejecutar un programa para cada una.

3.1.3.1 Propuesta

La aplicación que va a interactuar con el usuario para construir, dar inicio y controlar la ejecución de una tarea correrá en una computadora, en la cual también se graficarán los resultados de los experimentos llevados a cabo.

Realizar un experimento requiere, recibir datos que especifican los pasos que constituyen una tarea determinada y recibir las respuestas que va dando el sujeto, para que de esta manera se lleve la secuencia de dicha ejecución. A la vez, debe de comunicarse con la máquina encargada de generar los estímulos y decirle cómo es el estímulo que se necesita y cuando debe generarlo.

Como la secuencia del experimento depende de lo que el sujeto hace, necesitamos conocer que respuesta está dando en el instante en que la está realizando, y requerimos que la generación de los estímulos se realice en el preciso momento que se necesita. Por ello, el programa que controla la tarea estará en una máquina que sólo se dedique a esto, y el que genera los estímulos en otra.

En la Fig.13 se muestra el diagrama del sistema que proponemos.

3.1.3.2 Diagrama Conceptual

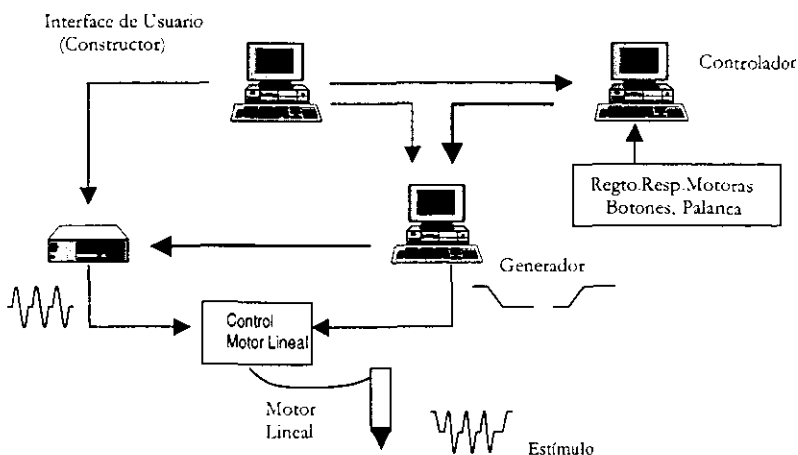


Fig.13 Diagrama del nuevo Sistema



3.1.4 DIAGRAMAS DE FLUJO DE INFORMACIÓN

En esta sección se tratará de mostrar, de manera general, como se lleva a cabo el flujo de datos durante la ejecución de un experimento.

Para que se pueda entender este proceso lo dividiremos en tres subprocesos, los cuales forman un ciclo.

Subproceso 1

Como se ha establecido, un experimento está constituido por n ensayos. El Constructor es quien lleva el control de todo el experimento, el que indica el inicio o fin de éste, el que establece la clase que se debe presentar. Sin embargo, el control del ensayo lo tiene el Controlador, quien lleva la secuencia de estados e indica al Generador el momento de generar un estímulo.

En el paso número 1 del subproceso 1, Fig. 14, el Constructor accede a la base de datos y recupera de la tabla tarea la secuencia de estados que la definen, y de la tabla clases, de los registros que corresponde a la clase que se debe presentar, los parámetros que establecerán las características particulares de cada uno de los estados del ensayo que va a iniciar.

En el paso número 2 del subproceso 1, Fig. 14, el Constructor distribuye la información de las tablas de la siguiente forma:

Constructor \Rightarrow Controlador

Número de estado	Evento esperado	Límite inferior de respuesta
Nombre del estado	Tipo de estado	Patrón de luces
Estado Correcto	Duración de estado	
Estado Erróneo	Evento recompensa	

Constructor \Rightarrow Generador

Si se trata de un estímulo Rampa le envía los siguientes parámetros:

- Amplitud
- Pendiente
- Duración de la rampa

Si el estímulo es una Vibración:

Si es Periódica:

- Frecuencia media
- Número de tonos
- Número de estímulo

Si es Aperiódica:

- Número de casillas
- Frecuencia del tono
- Número de estímulo
- Distribución Temporal



Constructor \Rightarrow **Sintetizador**

Si se trata de un estímulo Vibración:

- Frecuencia de tono
- Amplitud
- Número de tono

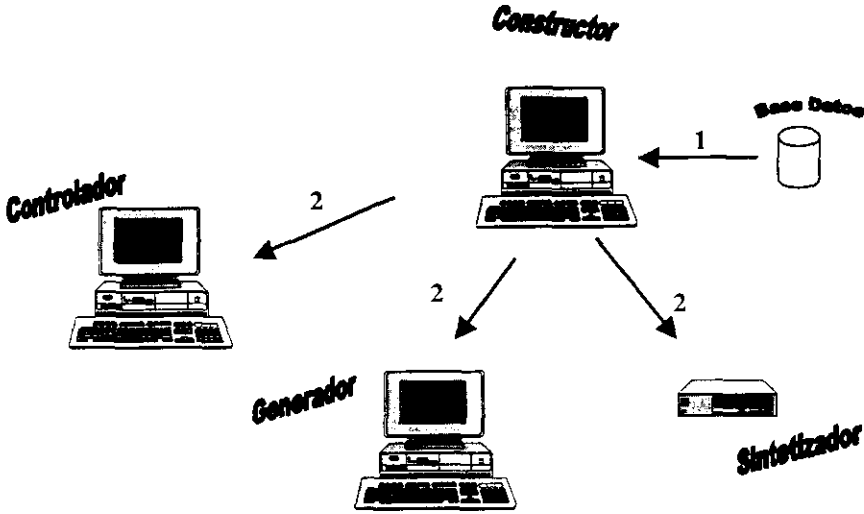


Fig.14

Subproceso 2

Una vez que el Constructor le indica al Controlador que ha terminado de enviar todos los parámetros necesarios, éste inicia el ensayo. El Controlador "ve" lo que hace el sujeto y con esta información va dirigiendo la secuencia del ensayo.

Si durante la secuencia del ensayo, es necesario generar un estímulo el Controlador se lo indica al Generador, paso número 1 del subproceso 2, Fig.15. Si el estado es una espera o recompensa el Generador no interviene y el Controlador realiza las funciones necesarias, paso 1^a del subproceso 2, Fig.15, revisa las entradas que recibe, las cuales, como ya sabemos, pueden venir del sujeto, paso 4 del subproceso 2, Fig.15, y del mismo Controlador en el caso de la duración del estado, la cual es activada por la tarjeta PC-TIO.

En el paso número 2 del subproceso 2, Fig.15, cuando se le ha indicado al Generador que genere un estímulo, éste en el caso de la Rampa interactuará



directamente con el dispositivo control del motor lineal; en caso de una Vibración, con el Sintetizador, y éste a su vez con el dispositivo de control. En el paso número 3 del subproceso 2, Fig.15, el control del motor lineal hace que la punta de prueba se mueva.

En el paso número 4, una vez que el sujeto da una respuesta está es recibida e identificada por el Controlador.

Este subproceso se realiza estado tras estado hasta que se llegue al final del ensayo, cuando se llega a éste se inicia el subproceso 3, Fig.16.

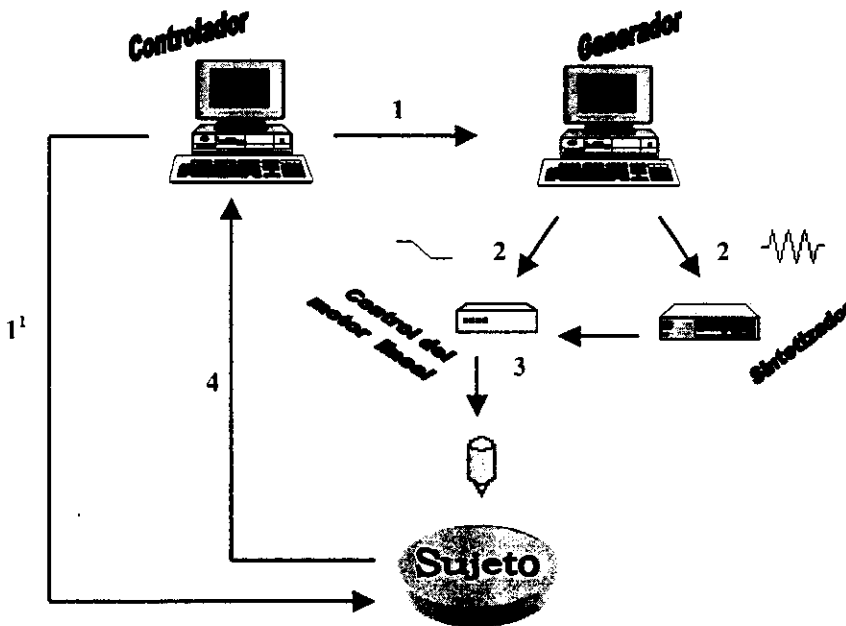


Fig.15

Subproceso 3

Una vez que se llega al final del ensayo, el Controlador se comunica con el Constructor y le envía la respuesta que se obtuvo del ensayo, que como ya se mencionó puede ser Correcta (Hit), Incorrecta (Wrong), Tardía (Late) o Temprana (Early); además le envía el tiempo, de cada uno de los estados, en el que se presentó un evento, paso 1 del subproceso 3, Fig.16.



El Constructor recibe esta información y si el investigador no le ha indicado que se detenga, ni que haga una pausa y aún quedan ensayos por hacer, decide con esta información que clase será la siguiente en presentarle al sujeto. En este momento se inicia nuevamente el subproceso 1, Fig.14.

El ciclo que se acaba de exponer se realizará ensayo tras ensayo durante todo el experimento.

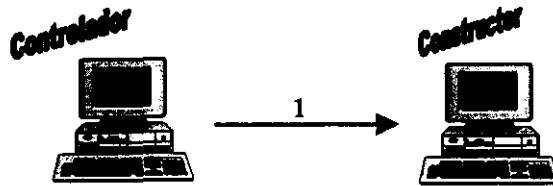


Fig.16



3.1.5 EVALUACIÓN DE HARDWARE, SOFTWARE Y REQUERIMIENTOS



Interface de Usuario (Constructor)

En esta computadora estará la aplicación con la que el investigador interactuará con el Sistema y se requiere que ésta sea lo más sencilla posible. Actualmente se cuenta con lenguajes que facilitan la creación de aplicaciones gráficas, haciendo que dichas aplicaciones hagan sencillo su manejo; es decir, que sean amigables.

Entre estos lenguajes tenemos a Visual Basic y a Visual C++, el Laboratorio del Dr. Romo cuenta con Visual C++ 5.0, por lo que con él se construirá la aplicación. Microsoft Visual C++ es un entorno de programación en el que se combinan la programación orientada a objetos (C++) y el sistema de desarrollo diseñado especialmente para crear aplicaciones gráficas para Windows.

Visual C++ 5.0

Visual C++ 5.0 presenta una tecnología de compilación avanzada para producir aplicaciones de 32 bits más rápidas y de menor tamaño. También incorpora las características y palabras claves del estándar ANSI. Proporciona accesos a bases de datos, en forma rápida e integrada y mayor soporte para el motor de datos Jet con los objetos de acceso a datos (DAO). (7)

Requerimientos:

SW:

Windows 95

Visual C++ 5.0

Access97

SW para manejar la Tarjeta de Comunicación de Datos GPIB (NI-488.2)

Los recursos de hardware se asignaron en función al equipo proporcionado por el Dr. Romo y dependiendo de los recursos que cada módulo requirió.

HW:

Una computadora Pentium a 200 MHz, 32 Mb en RAM, con un mínimo de 50 Mb de espacio libre en disco duro.

Tarjeta de Comunicación de Datos GPIB, para la transferencia de datos.

Al menos una ranura de expansión AT para la Tarjeta GPIB.



Controlador y Generador

Para controlar la ejecución del experimento se debe llevar la secuencia de la tarea, la cual depende de las entradas que da el sujeto durante la ejecución de éste. Las entradas son interrupciones de la palanca y de los botones. Como podemos darnos cuenta el tiempo en el que se dan estos es muy importante.



Por otro lado, para generar los estímulos también se requiere de una buena precisión y manejo del tiempo real.

El programa que controla el ensayo y el que genera los estímulos estarán en máquinas diferentes debido a la precisión que se requiere tanto en el control del experimento y la generación de estímulos.

Ahora bien, ya que no importa el aspecto visual de estas dos aplicaciones debido a que el investigador no interactúa directamente con estos programas, el lenguaje que utilizaremos será C.

Debido a que se manejarán interrupciones de HW y a que dicho manejo es más directo y eficiente desde DOS, ambos programas correrán en este Sistema operativo.

Lenguaje C

C es un lenguaje estructurado de propósito general conocido por bastantes programadores, ya que este lenguaje es expresivo y eficiente.

Requerimientos:

SW:

DOS

Lenguaje C

SW para manejar la Tarjeta de Comunicación de Datos GPIB (NI-488.2)

NI-DAQ (National Instruments Data Acquisition), software para interactuar con la Tarjeta LAB-PC y la Tarjeta PC-TIO-10.

HW:

Dos computadoras 486 a 33MHz, 8 Mb en RAM, con al menos 5Mb de espacio libre en disco duro.

Una de estas computadoras deberá contar con una Tarjeta PC-TIO, como temporizador, y una Tarjeta de Comunicación de Datos GPIB; la otra, además de estas dos tarjetas deberá tener una Tarjeta LAB-PC, para la generación de estímulos. Para cada una de estas tarjetas se requiere de una ranura de expansión AT.



3.2 DISEÑO

3.2.1 *DISEÑO INTERNO*

3.2.1.1 **Módulos principales de programación**

Como ya se mencionó el nuevo Sistema se conformará por tres computadoras: una tendrá la aplicación en la que se crea la tarea y sus clases y de la cual se iniciará el experimento; otra con el programa que se encargará de controlar el experimento y la tercera con el programa que genera los estímulos. Así podemos dividir la programación en tres módulos principales:

- ♦ Módulo encargado de la construcción de la tarea y sus clases, y del cual se controlará la ejecución de la corrida. (Módulo Construcción)
- ♦ Módulo encargado de controlar el experimento. (Módulo Control)
- ♦ Módulo encargado de la generación de estímulos. (Módulo Generación)

Módulo Construcción

Este módulo dividido en submódulos, deberá permitir:

- Crear una nueva tarea.
- Modificar una tarea existente.
- Crear un nuevo archivo de clases.
- Modificar el archivo de clases.
- Almacenar, recuperar y distribuir la información.
- Definir la secuencia de presentación de las clases.
- Definir el número de veces que se presentan las clases.
- Controlar la ejecución del experimento.
 - Inicio del experimento.
 - Pausa.
 - Continuar.
 - Detener.
 - Fin del experimento.
- Análisis de los resultados.- opciones que permitan visualizar de diferentes formas, los datos obtenidos desde el inicio del experimento hasta el último ensayo, a través de la:
 - Creación de un archivo con los datos obtenidos.
 - Graficación de estos datos.



Módulo Control

Consistirá en submódulos que permitan:

- Obtener la información enviada por la aplicación que da inicio a un experimento.
- Obtener las respuestas del sujeto.
- Dirigir la secuencia del experimento.
- Activar estímulos visuales.
- Activar estímulos mecánicos.
- Activar recompensa.

Módulo Generación

Tendrá los submódulos que realicen:

- La generación del estímulo Rampa.
- La generación del estímulo Vibración.

3.2.1.2 Comunicación y control

Para llevar a cabo la ejecución de experimentos en el laboratorio, se requiere que las computadoras y los dispositivos electrónicos cuenten con interfaces de comunicación adecuadas que permitan procesos de transferencia de datos, adquisición de datos y generación de señales analógicas. Utilizaremos las siguientes interfaces:

- a) Tarjeta de Comunicación de Datos GPIB.
- b) Tarjeta PC-TIO-10
- c) Tarjeta LAB-PC

3.2.1.2.1 Tarjeta de Comunicación de Datos GPIB

Los dispositivos GPIB pueden comunicarse con otros dispositivos GPIB enviando mensajes. Este tipo de dispositivos pueden ser de tipo Controller, Talker o Listener. El tipo Controller maneja el flujo de la información en la GPIB enviando comandos a todos los dispositivos. Un dispositivo tipo Talker es aquel que se encuentra en un modo de escritura hacia algún dispositivo. El dispositivo Listener se encuentra en modo de lectura, ya que solo recibe mensajes. A los dispositivos Talker y Listener también se les puede considerar como dispositivos esclavos.



3.2.1.2.2 Tarjeta PC-TIO-10

La tarjeta PC-TIO sirve para la generación de ondas y pulsos digitales, mediciones de ancho de pulsos, frecuencia y conteo de eventos. Esta tarjeta está constituida por: un temporizador, un circuito de interrupciones y dos puertos digitales, que se pueden configurar como puertos de lectura o de escritura.

3.2.1.2.3 Tarjeta LAB-PC

La LAB-PC es una tarjeta que maneja funciones analógicas y digitales, temporizadores de entrada y salida. Se utiliza para generar la rampa y como auxiliar para controlar la generación de los pulsos senoidales (estímulo vibración).

3.2.1.2.4 Distribución de Componentes del Sistema

Como se ha mencionado el Sistema está compuesto de tres computadoras y de cuatro dispositivos electrónicos: sintetizador, motor lineal, control del motor lineal y registro de respuestas motoras. Cada una de las computadoras contiene las interfaces de comunicación que se requieren para la ejecución de experimentos, de la siguiente forma:

1. Interfaz de Usuario
 - a) Interfaz de comunicación de Datos GPIB
2. Controlador
 - a) Interfaz de comunicación de Datos GPIB
 - b) Tarjeta PC-TIO-10
3. Generador
 - a) Interfaz de comunicación de Datos GPIB
 - b) Tarjeta PC-TIO-10
 - c) Tarjeta LAB-PC

La forma de comunicación entre cada una de las computadoras y el sintetizador, se puede dividir en tres etapas de comunicación:

- 1) Distribución de Datos
- 2) Control de la Tarea
- 3) Generación de Estímulos



3.2.1.3 Distribución de datos

La Fig. 17 muestra como los dispositivos se encuentran conectados basándose en una configuración estrella, por medio de la tarjeta GPIB. Se clasifica a cada dispositivo como dispositivo Controlador o como dispositivo Esclavo y se le asigna una dirección única que los identifica.

La secuencia de distribución de los datos que se representa en el esquema se hace de la siguiente forma:

- La distribución de datos da inicio cuando en la Interfaz de Usuario se da la instrucción de envío de datos, los cuales son obtenidos de la base de datos.
- La Interfaz de Usuario envía datos a la máquina Controlador, después a la máquina Generador y por último al Sintetizador, por medio de los buses GPIB que tienen conectados cada terminal.
- Una vez que cada una de las terminales y el dispositivo electrónico han recibido los datos de la Interfaz de Usuario, la máquina Controlador toma el control de la secuencia de la tarea.

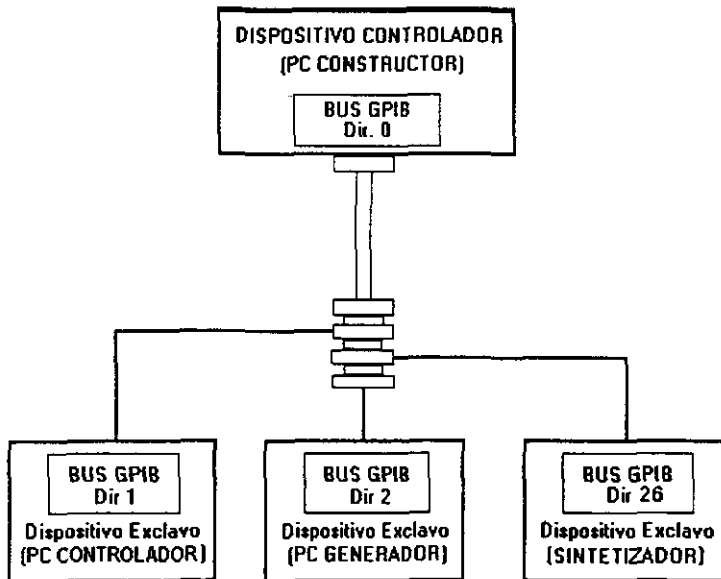


Fig. 17



3.2.1.4 Control de la tarea

El inicio del control de la tarea y la secuencia de la misma depende de las respuestas que emite el sujeto, por lo que la forma para que la máquina Controlador detecte cada respuesta del sujeto se presenta en la Fig. 18.

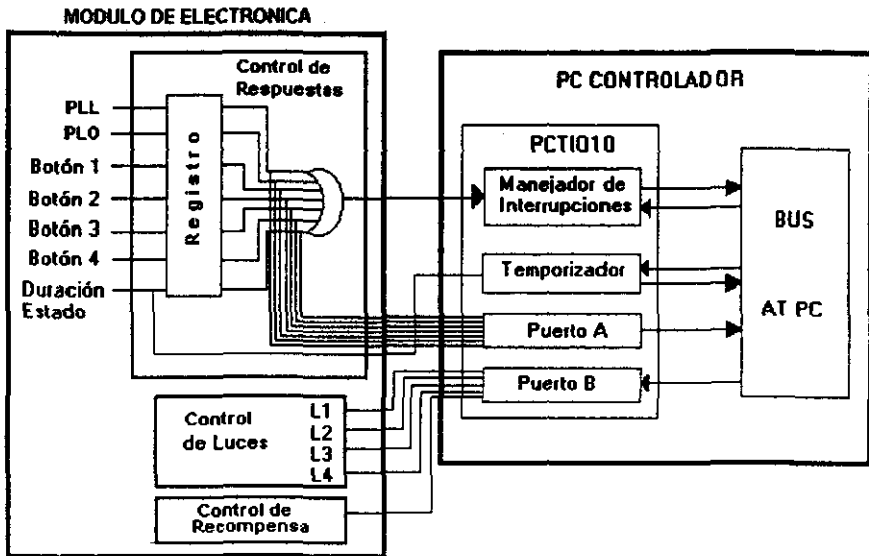


Fig. 18

Primeramente se tiene a la máquina control, donde su bus AT tiene conectado la Tarjeta PC-TIO-10. Esta tarjeta tiene como elementos que la constituyen: Un temporizador, que tiene como función llevar la cuenta del tiempo, de la duración de cada estado que compone a la tarea, es así que por medio de este circuito se puede obtener el tiempo en que se presenta cierta respuesta, o bien determinar el instante en que finaliza la duración de un estado. La PC-TIO-10 también cuenta con dos puertos digitales: el puerto A que se configuró de lectura y el puerto B de escritura.

En la Fig.18 también se presentan tres submódulos dentro del módulo de Electrónica. El submódulo de Control de Respuestas tiene como entradas a un registro las respuestas que el sujeto puede generar, y conectado a las salidas del registro se encuentran las entradas del puerto A de la PC-TIO-10 y también a una OR cuya salida activa al controlador de interrupciones de la PC-TIO-10. Por otro lado se tiene el submódulo de Control de Luces y Control de Recompensa los cuales son controlados por medio del puerto B de la PC-TIO-10.



3.2.1.4.1 Secuencia de comunicación

El proceso de Control de la tarea da inicio una vez que se haya generado una respuesta (PLO, PLL, BOTON1, BOTON2, BOTON3, BOTON4, DRE). Cuando se da una respuesta, ésta es detectada por el Módulo de Control de Respuestas, el cual activa al Manejador de Interrupciones, para que en dicho momento se interrumpan los procesos que está ejecutando la máquina Controlador y de esta forma se pueda obtener del temporizador el tiempo en que se generó la respuesta y leer del puerto A, que tipo de respuesta es la que entró. Una vez que la máquina Controlador ha obtenido los datos por medio de la Tarjeta PC-TIO-10, evalúa la respuesta obtenida para continuar con la secuencia de la tarea.

La máquina Controlador escribirá al puerto B en los siguientes casos:

- a) Cuando un ensayo haya terminado y su respuesta final sea la correcta, es entonces cuando se activa la recompensa, por lo que es necesario escribir al puerto B para que el submódulo de Control de Recompensa la genere.
- b) Cuando en la secuencia de estados de un ensayo se requiera que algunos estados estén acompañados con indicaciones visuales, por medio del puerto B se le indica al submódulo de Control de Luces.

3.2.1.4.2 Generación de Estímulos

La última forma de comunicación empleada es para permitir la Generación de estímulos, la cual permite que se creen el estímulo rampa y los estímulos Vibración. Como se muestra en la Fig. 19.

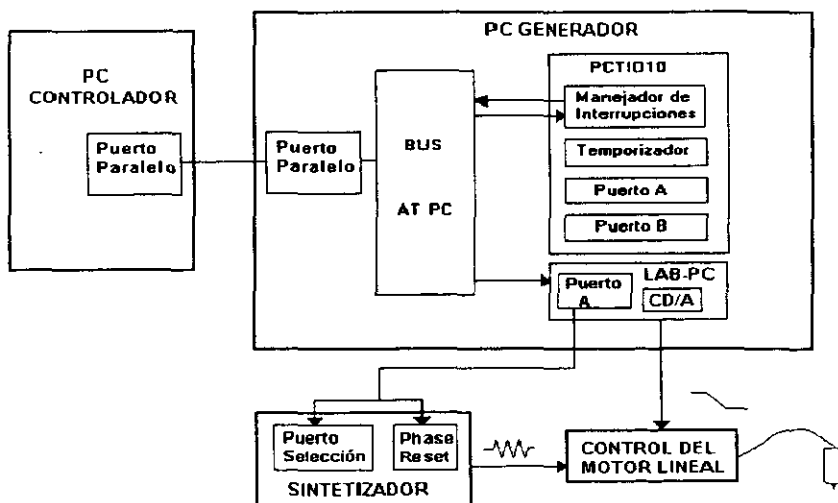


Fig. 19

La comunicación inicia cuando la máquina Controlador indica por medio de su puerto paralelo a la máquina Generador que se ejecute un estímulo, al recibir la indicación el Generador envía las instrucciones necesarias a la Tarjeta LAB-PC. Si



se trata de un estímulo Rampa la Tarjeta LAB-PC la genera, si se requiere un estímulo vibración la Tarjeta LAB-PC envía los comandos necesarios por medio del puerto A, al Sintetizador para que éste active su *phase reset*, y entonces pueda generar el estímulo; el estímulo se establece a través del puerto de selección del Sintetizador.

3.2.1.5 Colección de datos

Para el investigador es importante conocer ciertos parámetros de cada uno de los ensayos de un experimento, por lo que es necesario registrar las respuestas conductuales del sujeto durante el experimento, tales respuestas son registradas por el Controlador y enviadas por éste al Constructor quien finalmente las almacenará en un archivo. Además, en este archivo, se guardarán también ciertos parámetros de los estímulos presentados en el ensayo. Esta información permitirá al investigador correlacionar la actividad eléctrica de la corteza cerebral con la conducta del sujeto experimental y con los parámetros de los estímulos.

Durante la ejecución del experimento es necesario para el investigador saber cómo se está llevando a cabo éste y como está respondiendo el sujeto, por lo que además de almacenar la información en el archivo, ésta será presentada al investigador en la pantalla por medio del "Historial".

3.2.1.6 Graficación de datos

MATLAB tiene funciones que le permiten acceder a otras aplicaciones de Windows y a otras aplicaciones de Windows acceder a MATLAB. Estas funciones utilizan el intercambio dinámico de datos (dynamic data exchange, DDE), tecnología que permite a aplicaciones de Microsoft Windows comunicarse una con otra por medio del intercambio de datos.

La comunicación entre aplicaciones se realiza a través del establecimiento de una *conversación* DDE. La aplicación que inicia la conversación es llamada *cliente* y la aplicación que responde a la aplicación cliente es llamada *servidor*.

Una aplicación cliente puede acceder a MATLAB como un servidor DDE de las siguientes maneras, dependiendo de la aplicación cliente:

- Si se utiliza una aplicación que provee de funciones o macros para dirigir conversaciones DDE, se pueden usar éstas.
- Si se crea una aplicación propia, se puede hacer uso de la Librería del Motor de MATLAB (MATLAB Engine Library)

El motor de MATLAB soporta tres operaciones que pueden ser utilizadas por las aplicaciones. Estas operaciones incluyen enviar comandos a MATLAB para evaluar, solicitar y enviar datos a MATLAB.

Para graficar los datos obtenidos durante el experimento usaremos MATLAB, como ya se había establecido, y para la comunicación con esta aplicación utilizaremos la herramienta de la Librería del Motor de MATLAB.



3.2.2 DISEÑO FÍSICO

3.2.2.1 Estándares de Diseño

Los estándares que se plantean a continuación son elementos muy particulares de diseño gráfico, los cuales de no seguirse, repercutirían en la uniformidad de apariencia que el usuario final estaría viendo en la aplicación.

I. Formas

El título de cada pantalla que derive de la pantalla principal, se escribirá en mayúsculas. La letra tendrá como fuente el default (MS Sans Serif) y tamaño 8.

Todas las pantallas tendrán el color gris claro.

La pantalla principal tendrá un tamaño establecido, de tal forma que ocupe únicamente la mitad de la pantalla, esto con el fin de que en la otra mitad la ocupe MATLAB.

A las pantallas que no deben cambiar de tamaño se les debe quitar la opción de maximizarse y minimizarse.

II. Fuentes

En el tipo de letra de cajas de diálogos (tales como etiquetas, cajas de texto, grid's, etc.) usaremos la fuente por default, así como el tamaño; el color de las letras será negro.

En el caso de etiquetas, en el texto la letra inicial deberá escribirse con mayúscula. Los formas que tengan funciones iguales (como el caso de nuevo y de modificación) deberán tener las etiquetas en el mismo orden.

III. Botones

Utilizaremos botones que consten de texto, de un bitmap, o ambos. En general se utilizarán botones que utilicen texto y bitmap, esto con el fin de hacer un sistema fácil de entender, sobre todo en el caso de aquellas pantallas en donde se requerirán de varios botones para diversas funciones.

Los botones de Aceptar y Cancelar no llevarán bitmap alguno.

Los botones deben seguir los siguientes lineamientos:

- Las formas que tengan funciones iguales (como el caso de nuevo y de modificación) deberán llevar los mismos botones.
- Estos deberán pintarse en el mismo orden y ubicarlos en el mismo lugar, para evitar que el usuario los esté buscando.
- El color del fondo del botón será gris claro, excepto los cuatro botones que representan el tipo de estado, estos tendrán el fondo azul y la imagen en negro.

IV. Barra de título

La pantalla principal tendrá como título el nombre de la aplicación. Al abrirse una tarea se cambiará por el nombre de la tarea. Fig.20.

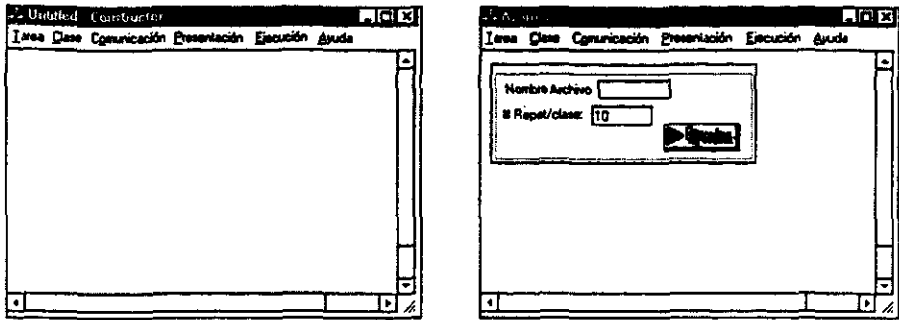


Fig.20

V. Barra de menús

Visualiza el conjunto de los seis menús disponibles para la aplicación Constructor. Cada uno de los menús tendrá su conjunto de órdenes que lo constituyen.

A cada uno de los seis menús se le asignará una letra que permita el acceso rápido a su respectivo menú. El investigador podrá seleccionar un menú pulsando la tecla *Alt* más la tecla que da acceso al menú, la cual aparecerá subrayada. Los elementos que componen cada uno de los menús no contarán con esta opción.

La propiedad *Prompt* de las ordenes de los menús contendrá el texto que aparece en la barra de estado cuando el elemento es seleccionado. Dicho texto dará una breve explicación de la función de esa orden.

VI. Barra de estado

Su objetivo es indicar la función que realiza el control en el cual esté posicionado el puntero del ratón y comentarios oportunos. Fig.21.

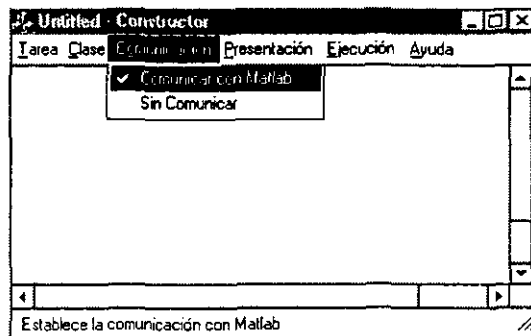


Fig.21



3.2.2.2 Aplicación Constructor

La aplicación Constructor es donde se creará la tarea y sus clases y donde se iniciará a un experimento.

El diseño de esta aplicación es como se muestra en la Fig.22.

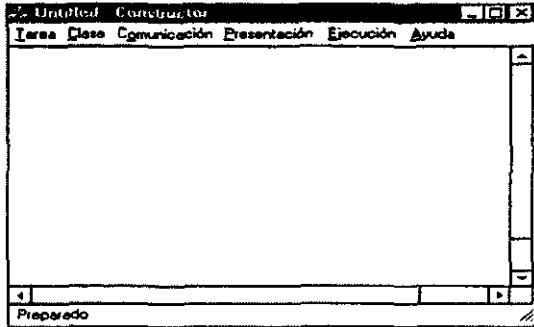


Fig.22

En la opción **Tarea** se encuentran los comandos que permitirán crear, abrir y modificar una tarea. Y la opción para salir del Sistema. Fig.23.

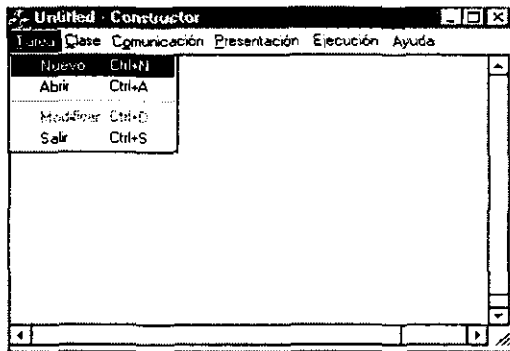


Fig.23

En la opción **Clase**, están los comandos que permitirán crear, abrir, modificar un archivo de clases y ver el nombre del archivo de clases abierto. Fig.24.

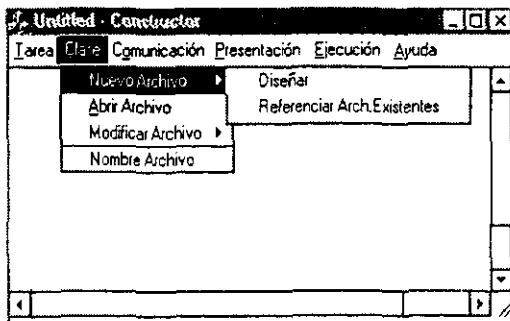


Fig.24



La opción **Comunicación** permitirá o no la interacción con MatLab, para que éste grafique las respuestas que se vayan obteniendo durante el experimento. Fig.25.

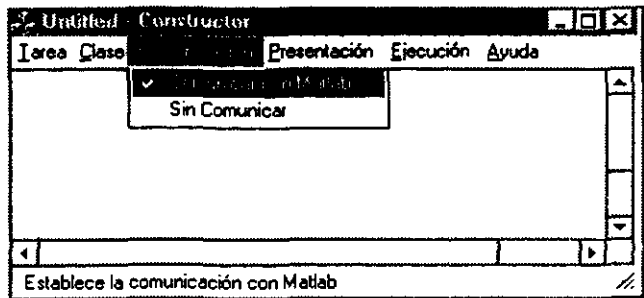


Fig.25

La opción **Presentación**, establecerá el orden en que se presentarán las clases de un archivo de clases, se tienen dos opciones: aleatoria y secuencial. Fig.26.

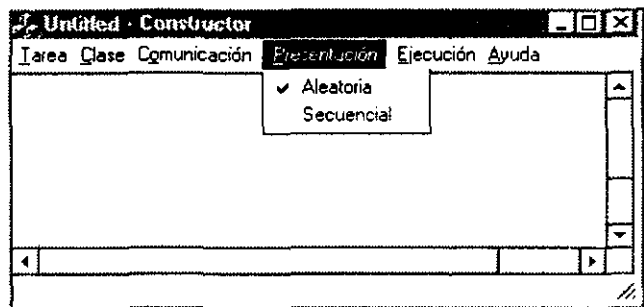


Fig.26

La opción **Ejecución**, tiene las opciones que controlarán el experimento (corrida): inicio de experimento, pausa, continuar, fin del experimento. Fig.27.

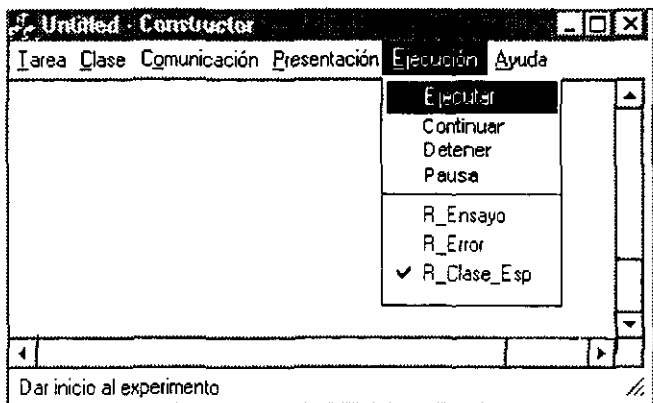


Fig.27



3.2.2.3 Diseño de una tarea

Debe consistir en un conjunto de funciones que permitan definir:

- ♦ Nombre de la tarea
- ♦ Número de estados (el mayor número de estados es quince).
- ♦ Características de cada estado
 - Nombre
 - Evento correcto
 - Estado correcto
 - Estado erróneo
 - Tipo de estado
- ♦ Poder recuperar y modificar esta información

3.2.2.3.1 Crear una nueva tarea

Para crear una tarea se seleccionará la opción Tarea del menú de la aplicación y se elegirá la opción Nueva. Aparecerá una ventana que pide el nombre de la tarea y el número de estados. Fig.28.

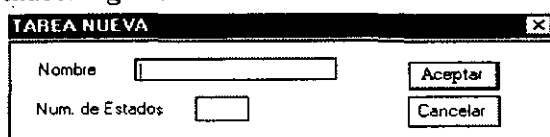


Fig.28

Una vez que se ha definido el nombre de la tarea y el número de estados, y después de oprimir el botón de Aceptar se presentará otra ventana en la que se deberá introducir las características de cada estado. Fig.29.

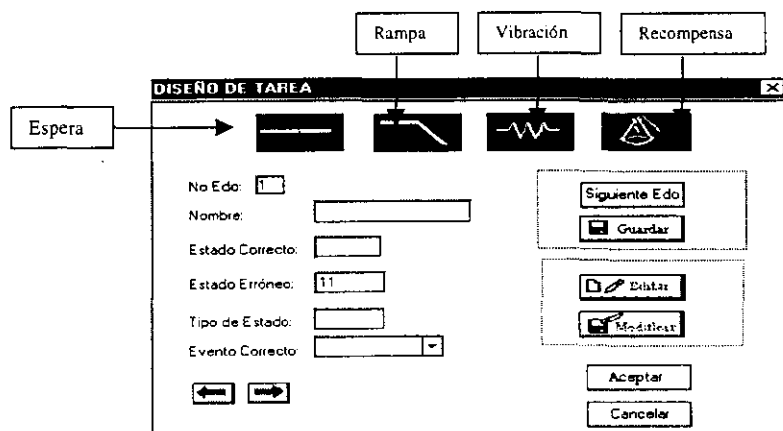


Fig.29

Los cuatro botones que aparecen de color oscuro en el tope de la ventana representan a los cuatro tipos de estímulos que se tiene: espera, rampa, vibración y recompensa. Al pulsar cualquiera de estos botones se presentará otra ventana llamada *Valores por omisión*, en la que se introducirán los parámetros correspondientes a dicho tipo de estímulo y los demás parámetros que se requieren en las clases. Fig.30, 31, 32 y 33. Estos valores servirán para crear fácilmente un archivo de clases.



Espera

VALORES POR OMISION

ESPERA

Valor Mín.:

Valor Máx.:

Quantum:

Lim. Inf. Resp.:

Patron:

Luces:

Luz1 Luz2 Luz3 Luz4

Aceptar Cancelar

Fig.30

Rampa

VALORES POR OMISION

RAMPA

Valor Mín.:

Valor Máx.:

Quantum:

Lim. Inf. Resp.:

Patron:

Amplitud:

Dur. Rampa:

Pendiente:

Luces:

Luz1 Luz2 Luz3 Luz4

Aceptar Cancelar

Fig.31

Recompensa

VALORES POR OMISION

RECOMPENSA

Valor Mín.:

Valor Máx.:

Quantum:

Lim. Inf. Resp.:

Patron:

Evento:

BT1 BT2 BT3

Luces:

Luz1 Luz2 Luz3 Luz4

Aceptar Cancelar

Fig.32



Vibración

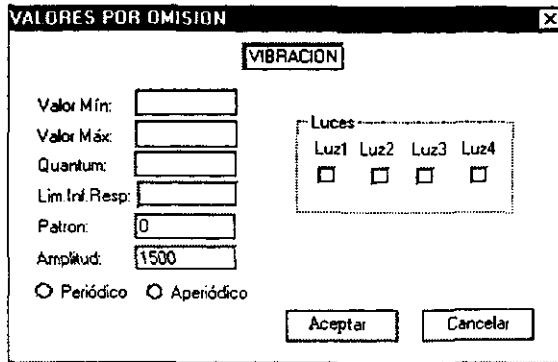


Fig.33

3.2.2.3.2 Modificar una tarea existente

La modificación de una tarea no requerirá de corregir el código. La aplicación permitirá hacerlo de forma sencilla. Una vez abierta la tarea se puede modificar. La opción **Modificar** presenta la pantalla mostrada en la Fig.34, en donde las flechas permiten moverse al estado que se desea modificar.

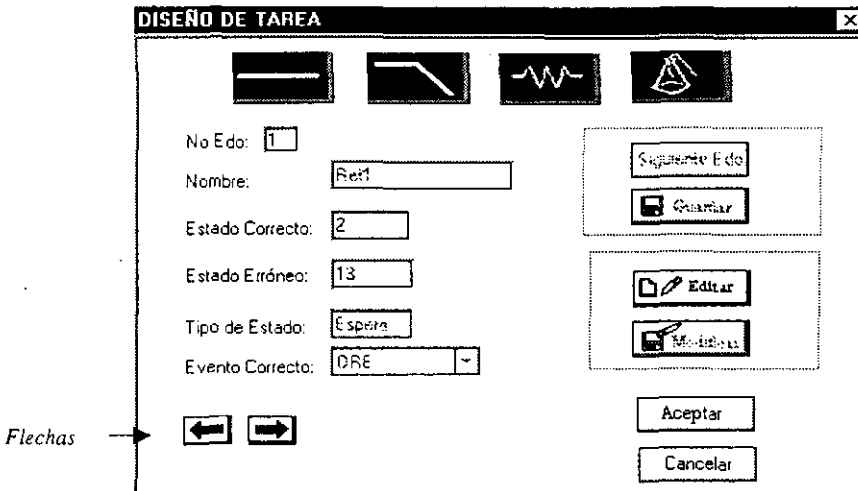


Fig.34

El botón Editar permitirá escribir y hacer las correcciones en el estado; el botón Modificar guardará las correcciones hechas.



3.2.2.4 Diseño del archivo de clases

Se debe definir:

- ◆ Número de clases (el máximo número de clases es cuarenta).
- ◆ Para cada estado de las diferentes clases especificar:
 - Duración del estado (mínima, máxima, quantum)
 - Características de los estímulos mecánicos
 - Cuáles luces se van a encender, si se requiere del uso de éstas
 - Si es un estado recompensa, establecer que entrada es la que se evaluará como correcta (PLL, DRE, BOTON).
- ◆ Poder recuperar y modificar esta información

3.2.2.4.1 Crear un nuevo archivo de clases

Para crear un nuevo archivo de clases se deberá seleccionar la opción **Clase** del menú de la aplicación y elegir la opción **Nuevo**; habrá dos opciones: por diseño y haciendo referencia a otro archivo de clases o a los valores por omisión. Al seleccionar cualquiera de estas opciones, aparecerá una ventana que pide el nombre del archivo de clases y el número de clases que tendrá éste, Fig.35.

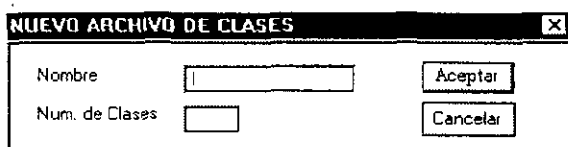


Fig.35

La opción **Diseñar**, permitirá crear el archivo de clases estado por estado y clase por clase. Fig.36.

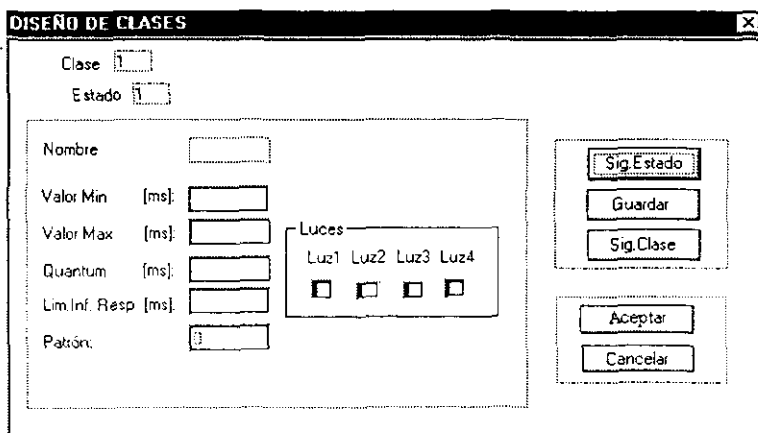


Fig.36

Los parámetros que se piden dependen del tipo de estado, de la misma forma que en la parte de Valores por omisión.



La opción **Referenciar Archivos Existentes** presentará dos maneras de crear el archivo, utilizando los valores por omisión o un archivo de clases ya existente. Fig.37.

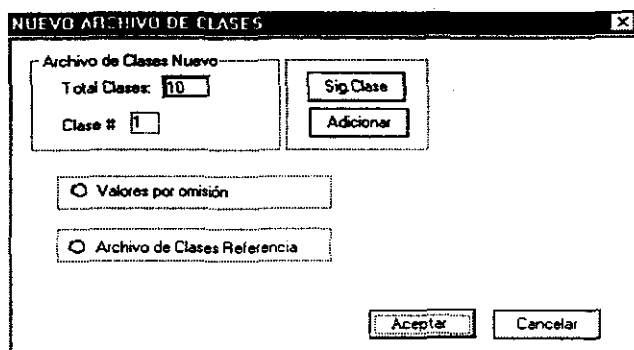


Fig.37

La opción **Valores por omisión** copiará los valores que fueron establecidos para cada estado, en la ventana de Valores por omisión, en cada una de las nueva clase que forman el Archivo de clases que está creando.

La opción **Archivo de Clases Referencia**, permitirá copiar determinada clase de cualquiera de los archivos de clases que pertenecen a esa tarea y a cada una de las clases nuevas que conforma el nuevo archivo de clases. Se abre un archivo de clases que ya existe, el que el investigador quiera; se cargan todas las clases que este archivo tiene y se elige la que se va a copiar en la clase actual del nuevo archivo de clases. Fig.38.

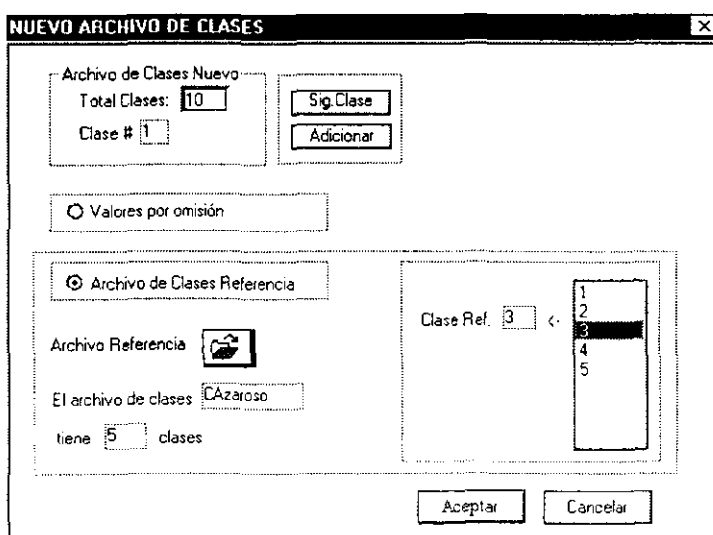


Fig.38



3.2.2.4.2 Modificar una archivo de clases existente

Una vez abierto el archivo de clases se podrá modificar. La opción **Modificar** presentará las opciones de: adicionar clases, eliminar clases y modificar datos. Fig.39.

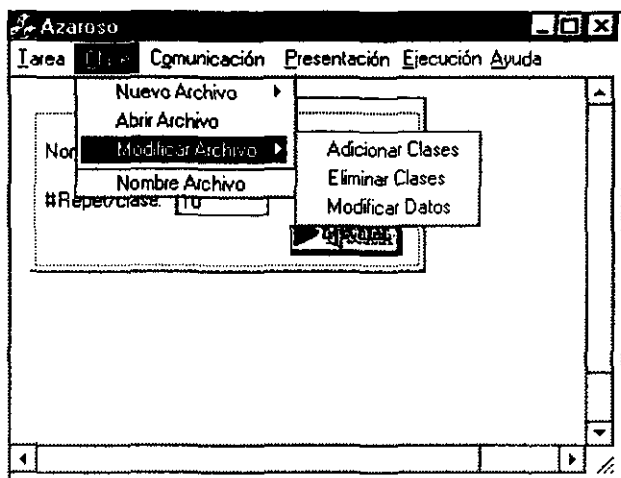


Fig.39

La opción **Adicionar** permitirá aumentar n clases en el archivo de clases; se tendrán dos opciones: Llenar con parámetros predefinidos y Escoger valores de clases existentes. Fig.40.

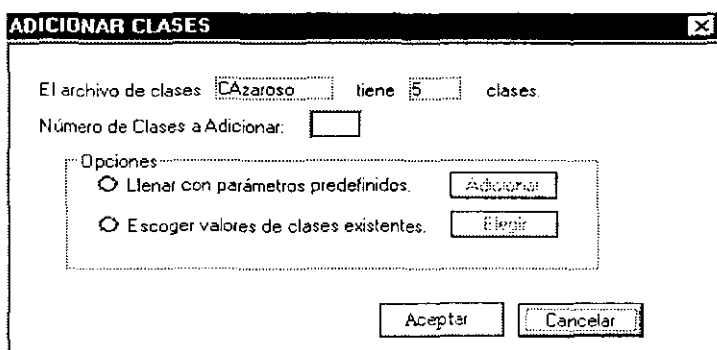


Fig.40

Llenar con parámetros predefinidos.- las n clases nuevas tienen los valores predefinidos.

Escoger valores de clases existentes.- permite copiar una clase ya existente del mismo archivo de clases, a la nueva clase que se está creando. Fig.41.

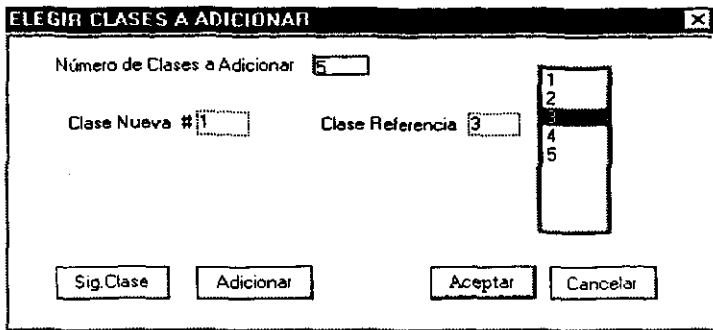


Fig. 41

La opción **Eliminar** permitirá borrar una clase en particular o un rango de clases del archivo de clases abierto. Fig. 42.

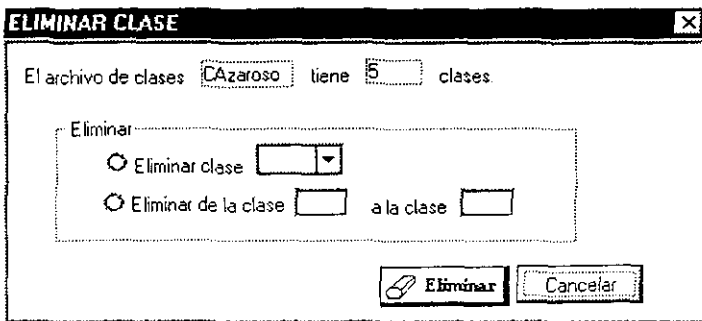


Fig. 42

La opción **Modificar datos** permitirá modificar un parámetro en particular de determinado estado de cierta clase, o de todas las clases. Fig. 43.

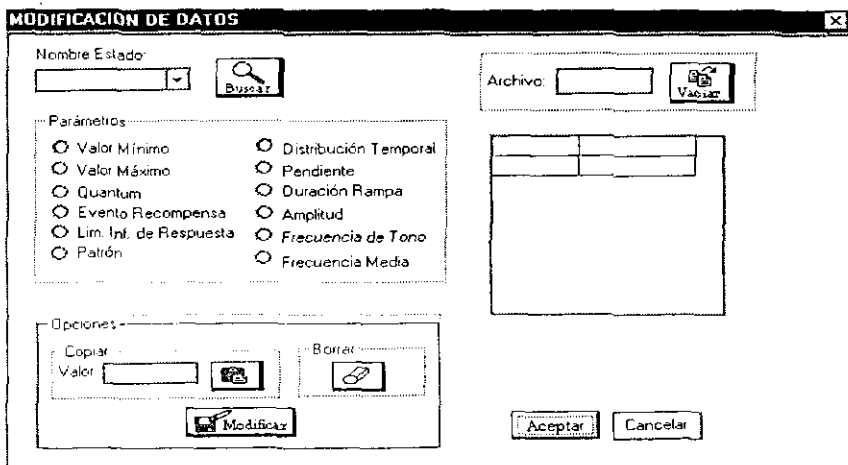


Fig. 43



Para modificar los parámetros de una o varias clases se deberá primero abrir el Archivo de Clases, elegir el estado al cual pertenece el parámetro que se requiere modificar, y seleccionar el parámetro de interés; una vez hecho esto se deberá oprimir el botón Buscar. Este botón desplegará en una rejilla (grid) el valor que tiene ese parámetro, en el estado especificado, para cada una de las clases que conforman ese Archivo. Fig.44.

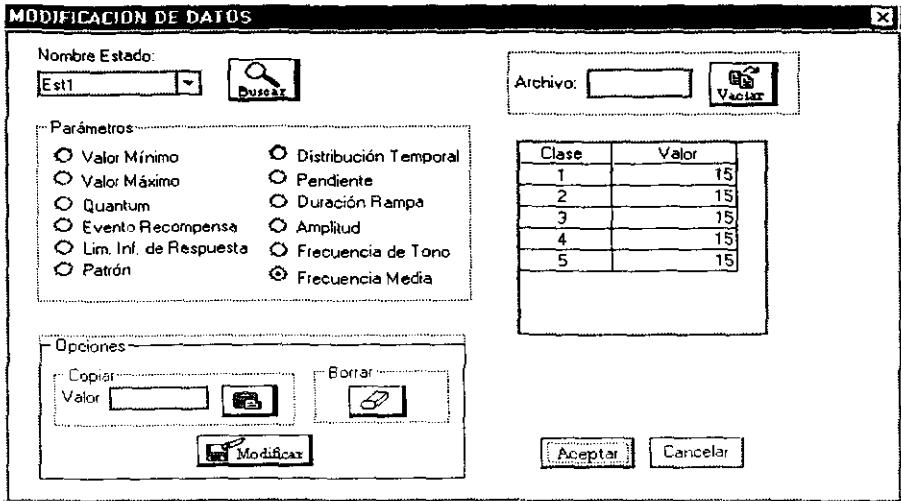


Fig.44

Una vez que se tiene en la rejilla los parámetros que se requieren modificar, se puede corregir directamente en ésta la clase deseada al colocarse en la celda correspondiente. O bien pueden utilizarse las opciones *Borrar* o *Copiar*, las cuales permiten respectivamente, borrar una celda o un rango de éstas; y copiar un valor en una o varias celdas.



Además se cuenta con el botón, *Vaciar*, el cual permite importar un conjunto de parámetros de un archivo txt, creado por el investigador y como su nombre lo indica, vaciar cada uno de estos parámetros en su correspondiente clase.



El botón *Modificar* guarda en la tabla del archivo de clases los cambios efectuados.



3.2.2.5 Ejecución de un experimento

Una vez abierta la tarea y el archivo de clases se podrá iniciar el experimento. La presentación de las clases por omisión será Aleatoria; sin embargo, antes de iniciar el experimento podrá cambiarse a Secuencial.

Para abrir el archivo de clases se deberá abrir primero la tarea; una vez abierto el archivo de clases aparecerá un recuadro en donde se introducirá el número de veces que se repetirá cada clase del archivo; aparecerá el nombre del archivo de clases abierto y el botón que da inicio al experimento (Ejecutar). Fig.45.

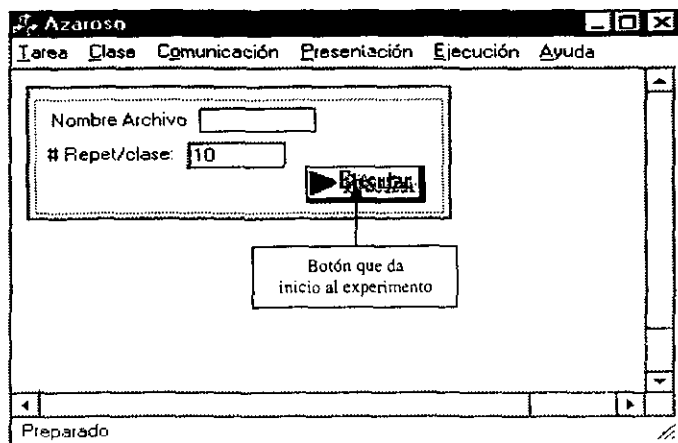
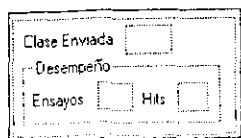


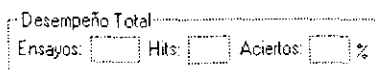
Fig.45

Una vez que se oprima el botón de Ejecutar se comenzará a mandar los datos al Controlador, Generador y Sintetizador.

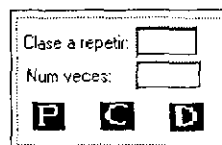
Aparecerán otros tres recuadros, uno en donde se muestre la clase que se ha enviado y el desempeño del sujeto, indicando el número de ensayos y el número de aciertos que haya realizado durante el experimento que se está efectuando (Recuadro 1). Otro recuadro mostrará el número total de ensayos, el total de hits y el porcentaje de aciertos que lleva durante los experimentos realizados hasta ese momento, sin importar con que tarea y archivo de clases se han realizado estos (Recuadro 2); el último permitirá que el usuario indique al Sistema que durante el experimento se desea que cierta clase en especial se repita n veces o se le indique cuando se requiera que haga una pausa, que continúe o detenga el experimento (Recuadro 3).



Recuadro 1



Recuadro 2



Recuadro3

- P** Pausa
- C** Continuar
- D** Detener



Aparecerá además en la aplicación una rejilla (Historial) que indique para cada ensayo la clase que se envió, la respuesta que dió el sujeto, y ciertos parámetros de los estímulos presentados. Fig. 46.

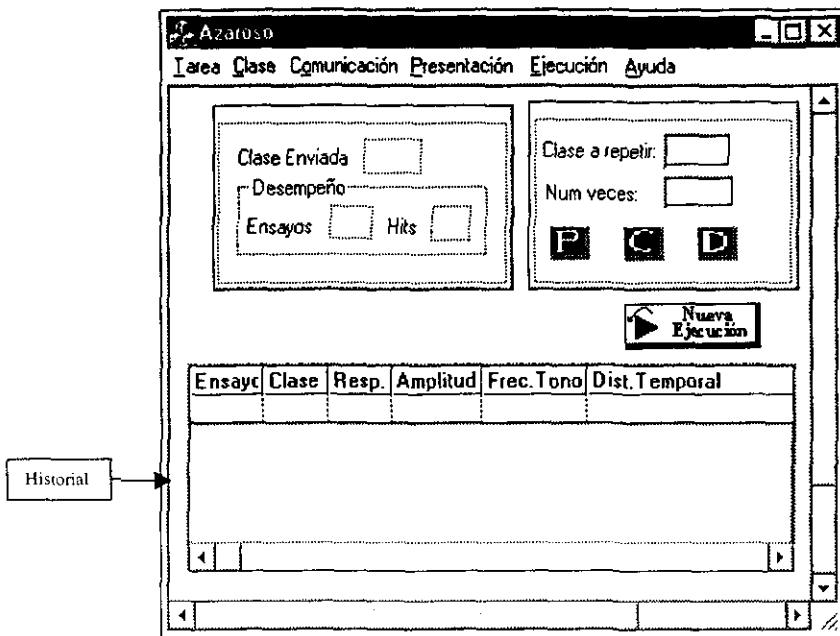


Fig. 46



Se cuenta también con el botón *MatLab*, el cual indica a esta aplicación que grafique los datos que hasta ese momento se han obtenido. Este botón estará habilitado siempre y cuando se haya activado del menú de la aplicación, la opción **Comunicar con MatLab** que se encuentra en **Comunicación**. Fig. 47

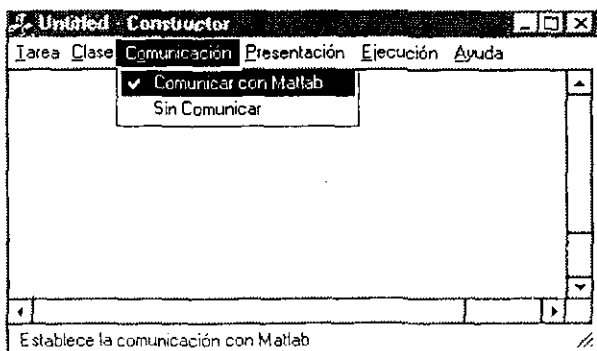
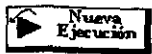


Fig. 47



El botón *Nueva Ejecución* inicia nuevamente el experimento con la misma tarea y con el mismo archivo de clases. Es útil ya que una vez que el experimento finalizó, si se deseara realizar uno más con la misma tarea y archivo de clases no se necesitará abrir cada una de éstas nuevamente.

Además en el menú de **Ejecutar** se tendrán tres opciones para modificar la manera en que se presentan las clases. Fig.48. La opción *R_Ensayo* repetirá la última clase enviada, mientras no se desactive esta opción; la opción *R_Error*, repetirá la clase que fue evaluada como Wrong; la opción *R_Clase_Esp*, habilitará los controles que permiten la repetición de una clase especial.

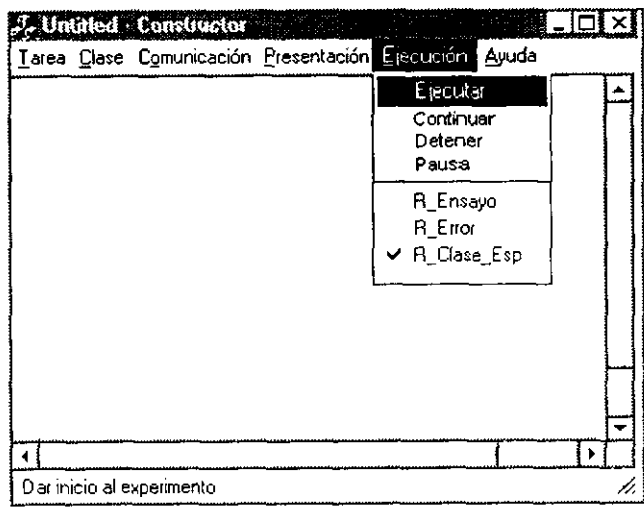


Fig.48



3.2.3 DIAGRAMAS

3.2.3.1 Diagrama de transición entre estados y Esquema de eventos

La mayoría de los objetos tiene un ciclo vital en el que una sucesión de eventos (cambio de estado de un objeto) pueden ocurrirle y cada uno de estos eventos modifica su estado (colección de los tipos de objetos que se aplican a él).

El **diagrama de transición entre los estados** (o diagrama de reja) muestra el ciclo vital de un objeto; es decir, los estados posibles de los objetos y los cambios de estados permísibles. Considerando al Constructor como un objeto, construimos su diagrama de transición para mostrar sus estados (el diagrama se muestra en la sección 3.2.3.2).

Además utilizamos los **esquemas de eventos** (sección 3.2.3.3 y 3.2.3.4), los cuales, como su nombre lo indican, muestran eventos, la secuencia en que ocurren y cómo estos cambian el estado de los objetos. En la construcción de los esquemas usaremos los siguientes términos:

Operaciones

El término *operación* se refiere a una unidad de procesamiento que puede ser solicitada. El procesamiento se implanta mediante un *método*. Éste es la especificación de cómo llevar a cabo la operación. A nivel de programa, el método es el código que implanta la operación. Una operación puede o no cambiar el estado de un objeto. Si lo cambiara, ocurriría un *evento*. Las operaciones se representan mediante cuadros con esquinas redondeadas. Los tipos de eventos se representan mediante triángulos sólidos negros conectados a la caja. Fig.49.

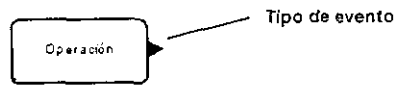


Fig.49

Los eventos son cambios de estado a los que un Sistema debe conocer y reaccionar ante ellos de algún modo. Muchas de las operaciones que producen estos eventos suelen ser *externas* al Sistema. En estos casos, el simbolo de operación se dibuja como una caja sombreada con esquinas redondeadas. Fig.50.

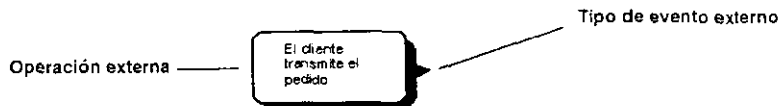


Fig.50

Reglas de activación

Las *reglas de activación* definen la relación entre la causa y el efecto, de esta manera, siempre que ocurra un evento de cierto tipo, la regla de activación invoca a una operación ya definida. Un tipo de evento puede tener varias reglas de activación, cada una de las cuales invoca a su operación en paralelo. Las



operaciones paralelas pueden producir diferentes cambios de estado en forma simultánea.

Condiciones de control

Una operación puede ser invocada por una o varias reglas de activación. Sin embargo, antes de invocar a la operación, se verifica su *condición de control*. Si los resultados de la evaluación de la condición son verdaderos, se invoca su operación. Si son falsos no.

Siempre que haya que verificar una condición de control antes de invocar a una operación, ésta se representa mediante un símbolo de forma de rombo antes de la operación, Fig.51. (9)

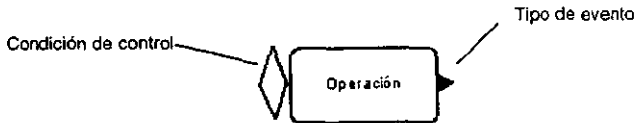
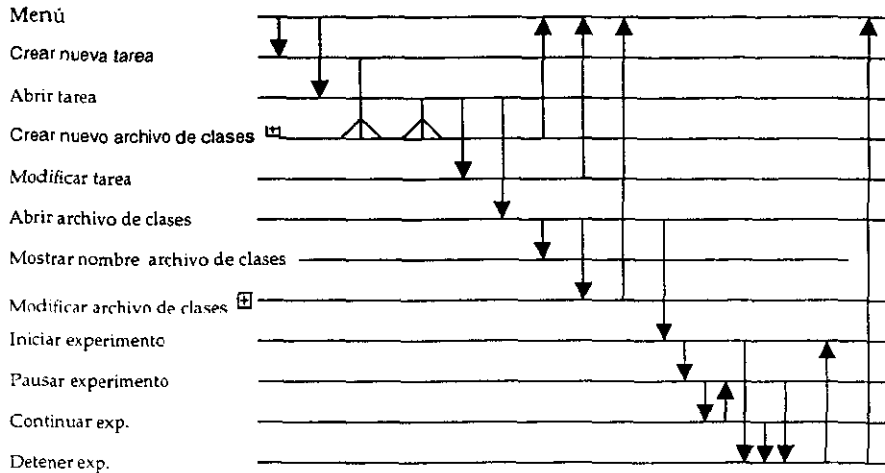


Fig.51

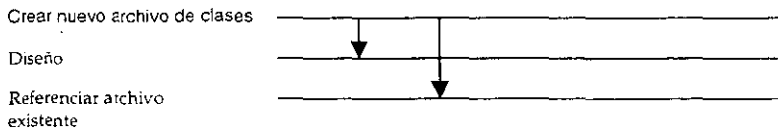


3.2.3.2 Diagrama de transición de estados del Constructor

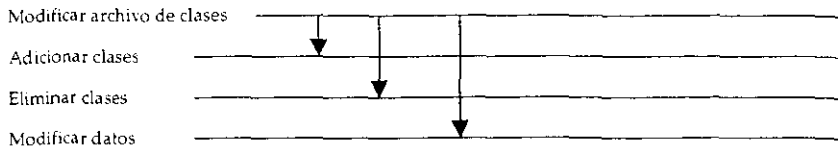


⊕ Indica que el estado se puede descomponer en subestados

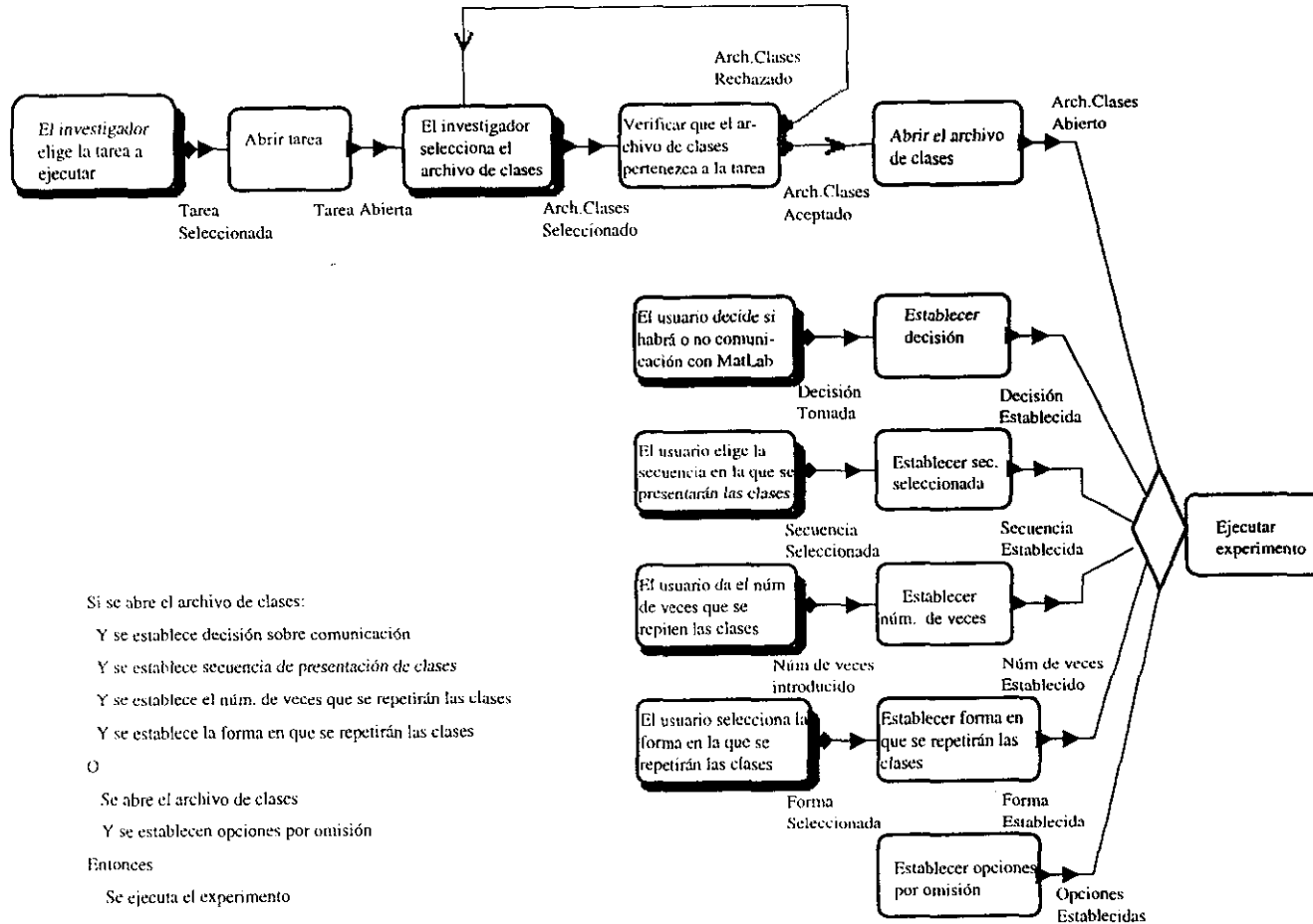
Crear nuevo archivo de clases



Modificar archivo de clases



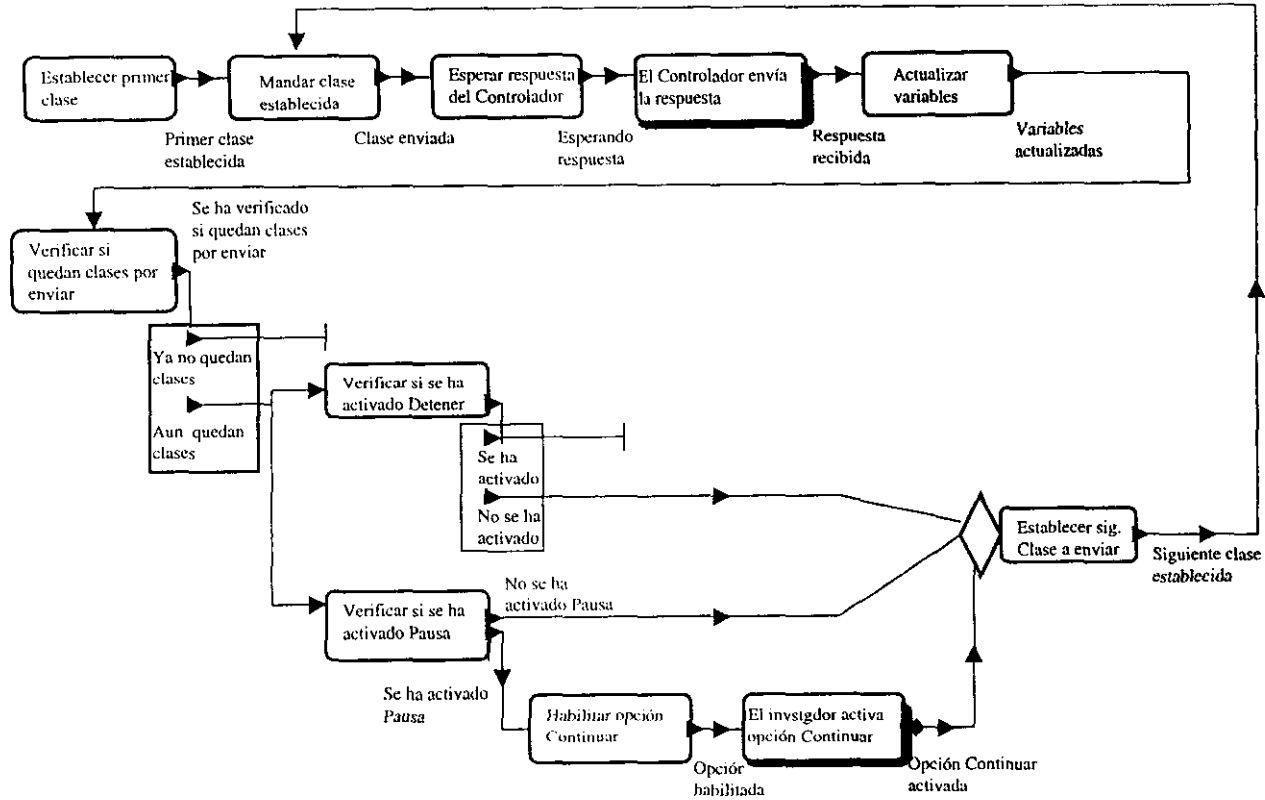
3.2.3.3 ESQUEMA DE EVENTOS QUE MUESTRA COMO DAR INICIO A UN EXPERIMENTO DESDE LA APLICACIÓN CONSTRUCTOR



Si se abre el archivo de clases:
 Y se establece decisión sobre comunicación
 Y se establece secuencia de presentación de clases
 Y se establece el núm. de veces que se repetirán las clases
 Y se establece la forma en que se repetirán las clases
 O
 Se abre el archivo de clases
 Y se establecen opciones por omisión
 Entonces
 Se ejecuta el experimento

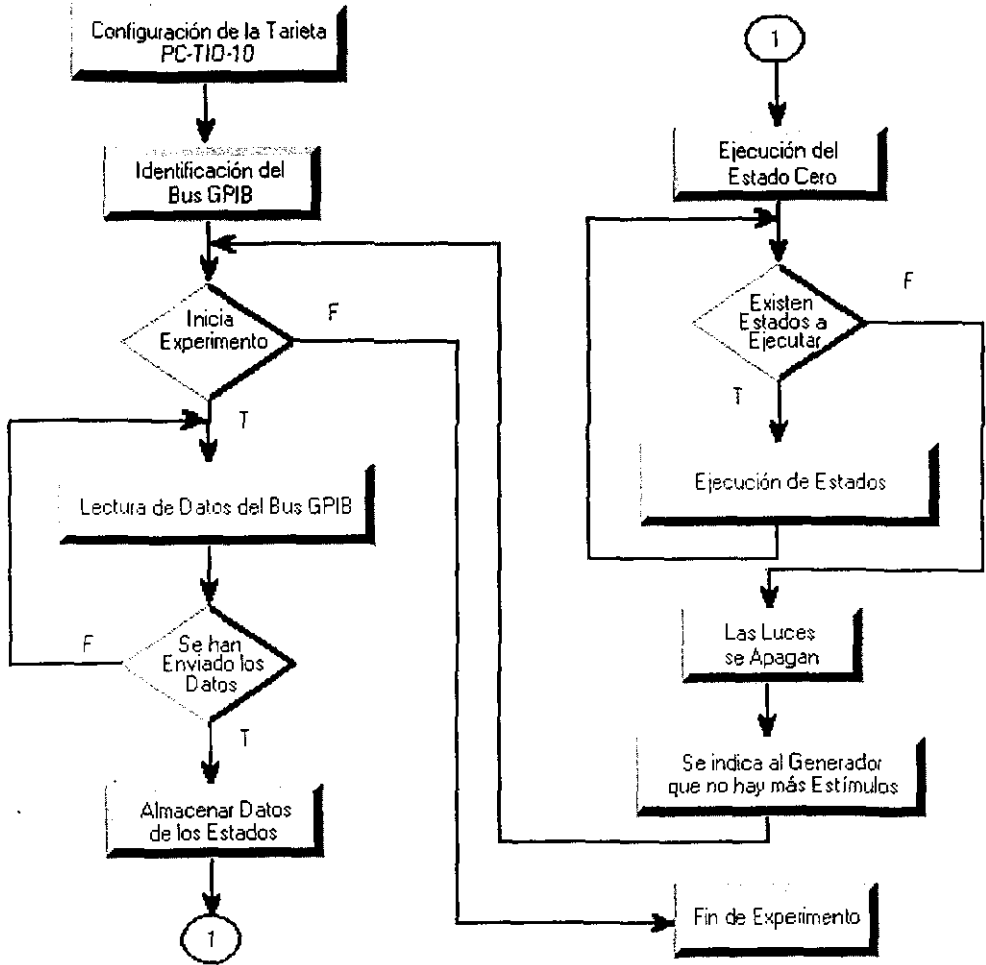


3.2.3.4 ESQUEMA DE EVENTOS QUE MUESTRA COMO SE CONTROLA LA EJECUCIÓN DEL EXPERIMENTO



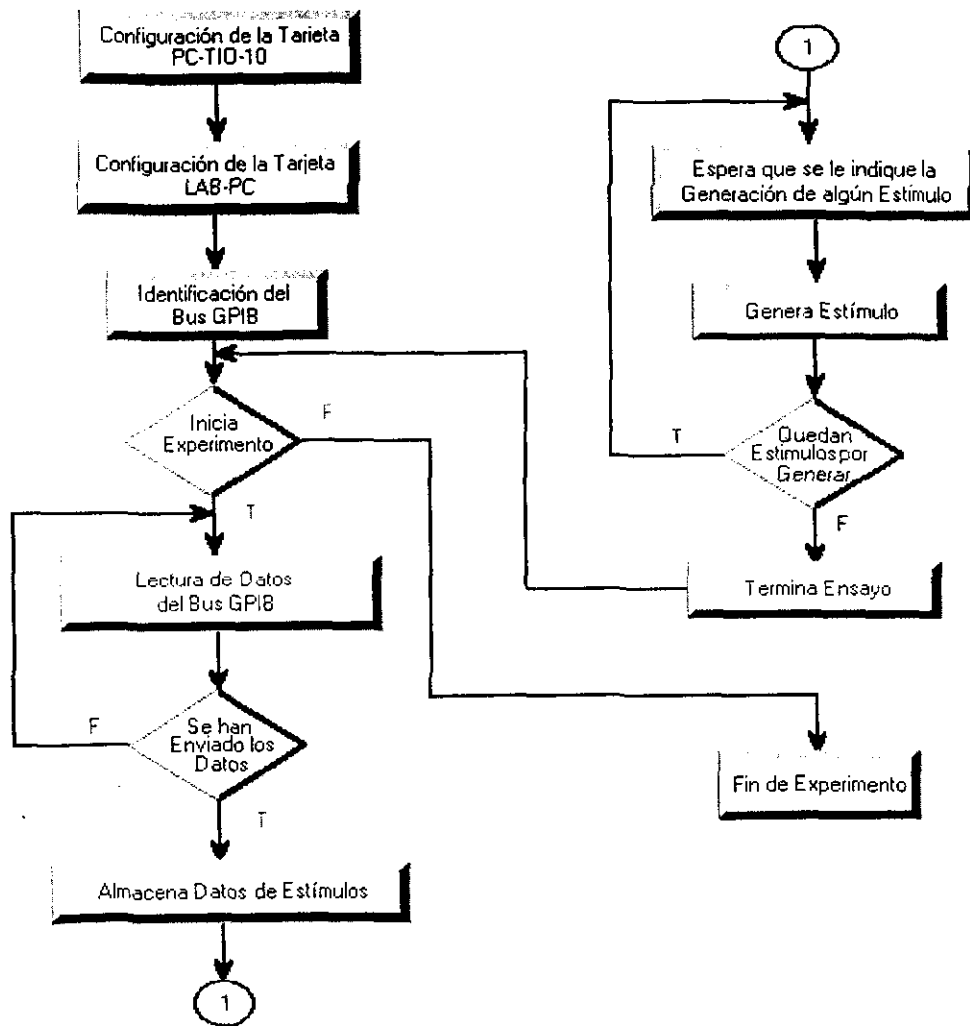


3.2.3.5 Diagrama de flujo del programa de la máquina Control





3.2.3.6 Diagrama de flujo del programa de la máquina Generador





3.3 DESARROLLO

3.3.1 PROGRAMACIÓN DE MÓDULOS

Se codificaron los programas de cada uno de los módulos principales: el de la construcción de la tarea y sus clases, desde el cual se controlará la ejecución del experimento; el módulo encargado del control del experimento y el módulo encargado de la generación de estímulos.

Una vez que se programó lo referente a la construcción de tareas y clases, se pasó a lograr la comunicación del Constructor con las demás máquinas, por medio de la tarjeta GPIB. Cuando se consiguió pasar los datos correctos al Controlador se codificó lo referente al control del ensayo, lo cual implica la comunicación y manejo de la tarjeta PC-TIO-10 así como la comunicación con el Sintetizador y con el Generador de estímulos. Se codificó entonces la generación de estímulos.

Una vez que el Sistema fue capaz de realizar un experimento completo se codificó lo referente a manipular la clase que se debe enviar según la respuesta del sujeto y la opción que haya indicado el investigador, como es el caso de repetir aquellos ensayos cuya respuesta es correcta o incorrecta, o repetir un determinado número de veces cierta clase. Finalmente se codificó la parte de colección y graficación de datos.

3.3.2 PLAN DE PRUEBAS

Como ya lo hemos señalado el Sistema es formado por tres módulos principales y cada uno de estos módulos por submódulos que a su vez están formados por rutinas. Por lo que durante la programación se realizaron pruebas en las rutinas. La prueba de los programas es la técnica de confirmación del Sistema, ésta la debemos hacer antes de entregar el Sistema al usuario. Las pruebas consisten en ejercitar el Sistema utilizando datos similares a los reales y observar los resultados e interpretar estos para detectar errores o insuficiencias en el Sistema. Es importante comprender que las pruebas nunca demuestran que un programa está correcto; es probable que existan errores aún después de la prueba más completa.

Una vez terminado el Sistema se hicieron varias pruebas para revisar que cumpliera con los requerimientos establecidos. Para ello creamos una tarea y un archivo de clases. Esto se documenta en los resultados.

3.3.3 PLAN DE INSTALACIÓN DEL SOFTWARE

La implantación del Sistema no necesariamente implica la sustitución total de los antiguos Sistemas y de sus bases de datos correspondientes. En ciertos casos, por razones económicas, los nuevos Sistemas integran algunos de los antiguos. En nuestro caso todo el desarrollo se realizó directamente sobre las máquinas que conforman el Sistema para el nuevo laboratorio por lo que no fue necesario una instalación. Sin embargo, se hicieron discos de instalación. Uno para instalar la aplicación del Controlador y otro para la aplicación del Generador. En estos discos se creó un archivo *instalar.bat* el cual crea un directorio y copia en éste el archivo ejecutable y agrega líneas en el *autoexec.bat*. Y para la aplicación del Constructor se copiaron los archivos correspondientes así como la base de datos, el usuario sólo deberá copiar estos en un directorio.



3.3.4 *PLAN DE LIBERACIÓN*

Para asegurarse de que el Sistema funcione correctamente se deben realizar nuevas pruebas para re-evaluar los resultados y hacer refinamientos del Sistema. Por esta razón, después de la etapa de instalación y de realizar varios experimentos, decidimos que era necesario que alguno de los usuarios finales probara el Sistema, así que un investigador comenzó a utilizar el Sistema. Durante esta etapa el investigador nos fue indicando ciertos detalles que modificar o agregar. Estas nuevas modificaciones no afectaron la estructura del Sistema y sirvieron para refinar su funcionamiento.

3.4 MANTENIMIENTO

Una vez que el Sistema estuvo en operación, se presentaron algunas modificaciones como fue que en el archivo de salida y en el historial se agregara un nuevo parámetro, o se quitara alguno que no era necesario. Esto en cuanto al mantenimiento adaptativo y perfectivo.



4 RESULTADOS



RESULTADOS

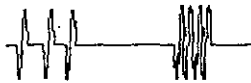
Para ejemplificar los resultados que se obtienen con el sistema presentamos una tarea diseñada por el Dr. Romo y los resultados que se obtuvieron al experimentar con varios sujetos. Tanto la creación de la tarea, la realización de los experimentos y la colección de datos obtenidos de estos, se hicieron utilizando el nuevo sistema.

Tarea "Diferenciar patrones"

El problema es el siguiente: frecuentemente nos vemos en la situación de comparar dos objetos o estímulos y debemos decidir si esos estímulos son iguales o si son diferentes. Por otro lado sabemos que el cerebro tiene una capacidad impresionante para manejar la variable tiempo, pero: ¿Cuáles son los procesos neurales que soportan ambas capacidades?. Para abordar estos problemas el Dr. Romo propuso el siguiente experimento:

A un sujeto se presentan un par de estímulos separados temporalmente, estos estímulos deben compartir la misma amplitud, frecuencia y duración; pero la posición en la que aparecen los diferentes pulsos puede o no ser la misma (con excepción del primer y el último pulso que siempre son iguales ya que indican el inicio y el fin del estímulo).

Por ejemplo:



Vibración 1

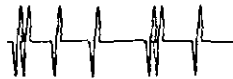


Vibración 2

Los patrones son distintos



Vibración 1



Vibración 2

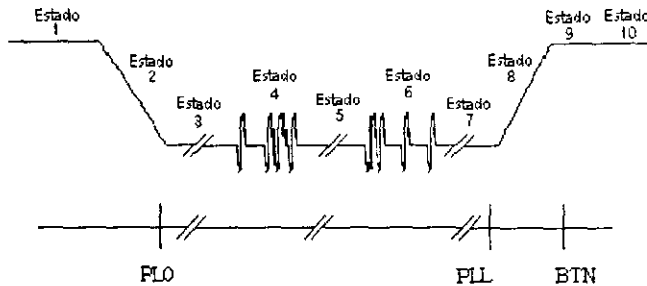
Los patrones son iguales



La tarea está formada por 10 estados, de la siguiente manera:

	Estado 1	Estado 2	Estado 3	Estado 4	Estado 5
Nombre	Espera1	Rampa1	Retardo1	Estimulo1	Inter_Estim
Evento	DRE	PLO	DRE	DRE	DRE
Edo. Correcto	2	3	4	5	6
Edo. Erróneo	11	11	11	11	11
Tipo de Edo.	Espera	Rampa	Espera	Vibración	Espera
	Estado 6	Estado 7	Estado 8	Estado 9	Estado 10
Nombre	Estimulo 2	Retardo 2	Rampa 2	Retardo 3	Recompensa
Evento	DRE	DRE	DRE	PLL	BTN
Edo. Correcto	7	8	9	10	11
Edo. Erróneo	11	11	11	11	11
Tipo de Edo.	Vibración	Espera	Rampa	Espera	Recompensa

La secuencia de presentación es la siguiente:



Se provoca que la punta del motor lineal haga contacto con la yema de uno de los dedos del sujeto (aplicando una rampa negativa en función del tiempo al control del motor), el sujeto coloca su mano sobre la palanca al detectarla. Después de un intervalo de tiempo variable se presenta la primera vibración, la cual hace que la punta de prueba se mueva siguiendo un patrón predeterminado. A continuación de esta primera vibración se genera otro intervalo de tiempo variable, el cual indica la separación entre la primera y segunda vibración. Se genera entonces la segunda vibración, se produce un nuevo intervalo de tiempo variable, se levanta la punta de prueba (rampa ascendente), en este momento el sujeto debe indicar su decisión soltando su mano de la palanca y proyectándola hacia uno de los botones. Después de esto se produce un último intervalo de tiempo en el cual se evalúa si la respuesta fue correcta, si es así el sujeto recibe unas gotas de agua como recompensa.



En esta tarea los botones tiene el siguiente significado:

Botón verde (BT1).- Los patrones fueron iguales.

Botón amarillo (BT2).- Los patrones fueron diferentes.

A fin de poder generar una curva psicométrica se procedió de la siguiente forma: Podemos asumir que, si dos patrones son iguales (coinciden todos los instantes donde aparecen los distintos pulsos en ambos estímulos) entonces la probabilidad de indicar que son diferentes es muy baja. Ahora bien, si desplazamos los pulsos en alguno de los estímulos entonces la probabilidad de indicar que son diferentes irá aumentando hasta que logremos una condición en la cual ambos estímulos sean claramente diferentes. Se propuso una medida para cuantificar la separación entre los dos estímulos mediante la siguiente expresión:

$$d = \sum_{i=2}^{n-1} P_i - S_i$$

donde:

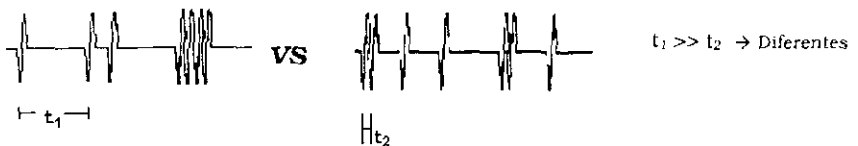
P_i es la posición del i -ésimo pulso del primer estímulo

S_i es la posición del i -ésimo pulso del segundo estímulo

Y asignando cero a la posición del primer pulso de cada estímulo.

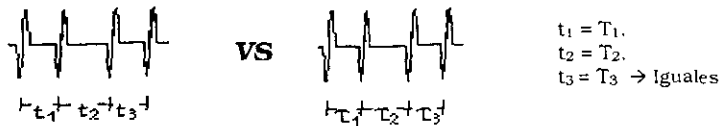
Después de experimentar con cinco sujetos se llegó a los siguientes resultados:

- Las estrategias que utiliza el sujeto para resolver la tarea pueden ser las siguientes:
 - Tratar de comparar el tiempo que cuantificó entre el primer y segundo tono de la primer vibración contra el tiempo entre el primer y segundo tono de la segunda vibración, la cual se le está presentando. Si el sujeto detecta alguna diferencia aquí, decide que las vibraciones son diferentes y no presta tanta atención al resto de los tonos de la segunda vibración. Para ejemplificar esto presentamos el siguiente par de vibraciones:



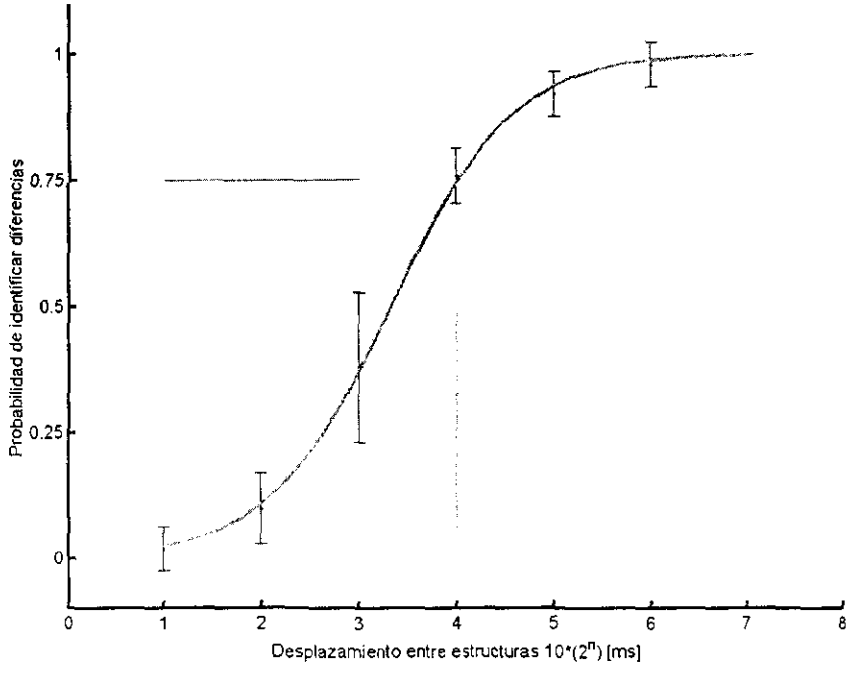


- Si no detectó diferencia en la comparación que hizo con el tiempo entre el primer y segundo tono de la primer vibración, y el tiempo entre el primer y segundo tono de la segunda vibración, compara ahora el tiempo entre el segundo y tercer tono de la primer vibración con el tiempo entre el segundo y el tercer tono de la segunda vibración. Si no son iguales ahora comparará el tiempo entre el tercer y el cuarto tono de ambos estímulos; así sucesivamente hasta detectar alguna diferencia o hasta que termine el segundo estímulo. Ejemplos:



Se puede plantear que la estrategia que usa el sujeto para resolver esta tarea es: comparar los intervalos que hay entre cada uno de los tonos de las vibraciones, es decir, trata de cuantificar el tiempo que existe entre las estructuras del patrón temporal.

- En promedio los sujetos mostraron la capacidad de encontrar diferencias entre patrones de distribución temporal cuyo desplazamiento (d) entre estructuras fue igual o mayor a 160 ms ($n=4, P=0.75$). La gráfica que a continuación se presenta es la curva psicométrica que se obtuvo con los resultados obtenidos con el promedio de los cinco sujetos, la cual muestra lo antes mencionado.





5 CONCLUSIONES



CONCLUSIONES

El sistema cumple con los requerimientos establecidos. La creación y modificación de una tarea, así como la ejecución de ésta pueden ser realizadas por cualquiera de los miembros del laboratorio sin requerir de ellos un conocimiento profundo del sistema lo cual acelerará la obtención de resultados.

Se consiguió que el sistema construya y ejecute un número mayor de tareas al identificarse aquellos elementos comunes a todas éstas; ya que se logró establecer una estructura basada en elementos básicos, bien definidos, los cuales permiten construir al elemento de mayor complejidad, la tarea.

El Sistema permite experimentar con diferentes tipos de vibraciones. Al principio se establecieron sólo dos tipos, periódicas y aperiódicas, finalmente se agregaron otras dos, azarosas y bipulsos. Al observar estas vibraciones nos dimos cuenta de que otra forma de resolver este punto hubiese sido encontrando las características comunes que definen a estos cuatro tipos de vibraciones y estableciendo un elemento general que permitiera la creación de cualquiera de estos.

Al construir este Sistema nos dimos cuenta de la importancia de conocer y seguir una metodología de desarrollo de Sistemas, ya que nos permitió establecer claramente los requerimientos y llevar un orden en cuanto al desarrollo de éste, evitando así ocupar tiempo importante en cosas no necesarias.

Conocimos instrumentación y equipos de uso común en la industria y el medio científico como la tarjeta GPIB, LAB-PC, y PC-TIO. Aprendimos a manejarlas, conocimos sus ventajas, desventajas y costos.

Además conocimos y utilizamos algunas de las herramientas que proporciona MatLab y Visual C++. Con respecto a MATLAB, este *software* tiene funciones que le permiten acceder a otras aplicaciones de Windows y a otras aplicaciones de Windows acceder a MATLAB. Estas funciones utilizan intercambio dinámico de datos (dynamic data exchange, DDE), tecnología que permite a aplicaciones de Microsoft Windows comunicarse una con otra por medio del intercambio de datos.
(10) Gracias a esta herramienta nos fue posible establecer la comunicación con MATLAB.

Hasta febrero del 2000, un año después de iniciar este proyecto, se han realizado tres estudios de psicofísica en humanos y se han entrenado a tres monos rhesus; uno de ellos en discriminación de frecuencias y dos en la tarea de diferenciar patrones, lo cual muestra que este sistema está en operación.



6 BIBLIOGRAFÍA

**BIBLIOGRAFÍA**

- (1) Romo, R., Ruiz, S., Crespo, P., Zainos, A. And Merchant, H. (1993c). *Representation of tactile signals in primate supplementary motor area*. J. Neurophysiol.
- (2) Romo, R., Ruiz, S., Crespo, P., and Hsiao, S.S. (1993b) *A tactile stimulator for studying motion processing in the somatic sensory system of primates*. J. Neuroscience. Methods 46:139-146.
- (3) Romo, R., Ruiz, S. y Crespo, P., *Cortical representation of touch*. Neuroscience: from Neural networks to artificial intelligence
- (4) Ruiz, S. y Crespo, P., Romo, R., *Representation of moving tactile stimuli in the somatic sensory cortex of awake monkeys*. J. Neurophysiol.
- (5) Romo, R., Ruiz, S. y Crespo, P., Zainos, A. And Merchant, *La actividad sensorial como reflejo de la representación de la realidad en el cerebro*.
- (6) Kendall, Kendall. *Análisis y Diseño de Sistemas* Prentice-Hall Hispanoamericana, S.A.
- (7) Maltin, Margaret W., Foley, Hugh J. *Sensación y Percepción* Prentice-Hall Hispanoamericana, S.A.
- (8) Hernández H, Salinas E, García R, Romo R, *Discrimination in the sense of flutter: new psychophysical measurements in monkeys*. J. Neurophysiol.
- (9) Ceballos Fco. Javier. *Visual C++ Aplicaciones para Win32* 1998 Alfaomega Grupo Editor, S.A. de C.V.
- (10) *PC-TIO-10 User Manual*. National Instruments Corporation. *NI-DAQ User Manual for PC Compatibles*. National Instruments.
- (11) James Martín, James J. Odell. *Análisis y Diseño Orientado a Objetos* Prentice Hall Hispanoamericana, S.A.
- (12) *MATLAB Application Program Interface Guide* The Math Works Inc.

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