

5  
26y



# UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

Facultad de Estudios Superiores  
CUAUTITLAN

"DESARROLLO DE UN SISTEMA DE ADQUISICION  
DE DATOS QUE AYUDE AL DIAGNOSTICO  
DE PROBLEMAS DORSALES"

## T E S I S

QUE PARA OBTENER EL TITULO DE  
INGENIERO MECANICO ELECTRICISTA

P R E S E N T A

**SALVADOR AVILA BARRAZA**

Director de Tesis: ING. ANTONIO HERRERA MEJIA



V N A M

1988.

TESIS CON  
FALLA DE ORIGEN



Universidad Nacional  
Autónoma de México



## **UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso**

### **DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

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

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

## PROLOGO

El presente trabajo tiene como finalidad, la de mostrar a los integrantes de la carrera de Ingeniero Mécánico Electricista de la Facultad de Estudios Superiores Cuautitlan, un ejemplo del amplio campo de aplicaciones que tienen las ayudas computarizadas en áreas de aplicación que pudieran considerarse lejanas para nuestra carrera.

Esta aplicación no solamente se apoya de soluciones de programación, sino que también proporciona una solución en electrónica, enfocándose primordialmente a una solución practica y efectiva.

Como se enuncio anteriormente esta aplicación consiste de un sistema de adquisición de datos desde un transductor de fuerza hacia una microcomputadora.

En el primer capitulo, que es la Introducción; se mencionan las generalidades de los sistemas computarizados y sus ventajas con respecto a sistemas convencionales.

El segundo capitulo nos enuncia la problemática que debia ser resuelta con la ayuda de una microcomputadora. El tercer capitulo, menciona las alternativas, algunas de ellas planteadas previamente por el usuario final del sistema, como en el caso de los

transductores los cuales ya estaban definidos.

En el cuarto capítulo de Evaluación y elección, se consideran primordialmente los aspectos de precio y funcionalidad de los equipos encontrados para satisfacer la necesidad planteada plenamente.

El quinto capítulo que es el de Implementación del sistema, se muestra la conjunción de elementos para conseguir la ayuda deseada y finalmente el sexto y último capítulo evalúa el alcance real de este trabajo, el cual afortunadamente cubre en su totalidad las expectativas planteadas inicialmente.

## MATERIALES Y METODOS

### MATERIALES

- I) Computadora personal IBM, con 640 KBytes de RAM, adaptador de graficas y teclado.
- II) Monitor a color para IBM-PC.
- III) Impresora.
- IV) Bateria para respaldo de reloj en tiempo real.
- V) Manejador de discos flexibles.
- VI) Disco duro de 20 MBytes.
- VII) Sistema Operativo MS-DOS.
- VIII) Tarjeta de interfase AIOB de Action Instruments.
- IX) Paquete de programación y licencia de Turbo Pascal.
- X) Celda de carga de compresión.

### METODOS

- I) Interconexión del sistema completo, primero con una celda de carga de voltaje y después con una de corriente.
- II) Instalación de la tarjeta de interfase de Action Instruments en una de las ranuras libres de la microcomputadora.
- III) Elaboración de un programa de Adquisición de datos y archivo de los mismos.

## **INDICE**

**I) INTRODUCCION**

**II) PLANTEAMIENTO DEL PROBLEMA**

**III) ALTERNATIVAS**

**IV) IMPLANTACION DEL SISTEMA**

**V) PRUEBAS**

**VI) CONCLUSIONES**

**APENDICE A: PROGRAMA DE APLICACION**

**APENDICE B: DATOS TECNICOS GENERALES**

**BIBLIOGRAFIA**

## ***CAPITULO I***

## ***INTRODUCCION***

## INTRODUCCION

En la actualidad, la mayoría de los sistemas que se están diseñando, incluyen ayudas computarizadas, debido a que por medio de ellas su eficiencia se ve aumentada de una manera considerable.

Estas ayudas pueden ser de varios tipos y depende del campo de aplicación en las que se deseen implantar, pudiendo ser administrativas que incluyen el manejo de bases de datos como: nóminas, personal, etc. Productivas que están dirigidas al manejo de programación de la producción; de proceso en las que se obtienen mímicos de procesos que se actualizan dinámicamente y gráficas de tendencia del comportamiento del proceso y algunas otras de acuerdo al área de interés.

Las computadoras debido al avance en el diseño y fabricación de microcircuitos, se han convertido paulatinamente en sistemas más pequeños y de manejo más sencillo, además de que son cada vez más accesibles en precio y las diferencias entre minicomputadoras y microcomputadoras se hace cada vez menor.

Basta con observar el desarrollo de los microprocesadores que fueron desarrollados en un principio con solo 4 bits y actualmente existen de 32 bits (80386 de INTEL), los cuales se pueden emplear en microcomputadoras.

Estos microprocesadores alcanzan velocidades de hasta 20 millones de hertz y los dispositivos de memoria, de disco tanto blando como duro también avanzan en capacidad constantemente.

En este trabajo se ha considerado que lo importante del funcionamiento de una microcomputadora no está limitado solamente al almacenamiento de datos, sino que debe también aprovecharse en aplicaciones de ejecución de procesos, debido a que existen equipos que permiten que intervengan en tales procesos.

Se debe entender que aún existiendo dispositivos de control que fueron diseñados para cumplir con estas funciones, las microcomputadoras o computadoras personales pueden ya competir abiertamente contra ellos ya que por medio de tarjetas de interfase pueden ejecutar labores de control y además manejar bases de datos de manera complementaria.

Con esta diversificación en su funcionamiento el costo de una computadora personal resulta una inversión, debido a que ejecuta una doble función a través de un solo equipo.

Las microcomputadoras además se construyen ya para trabajar en ambientes industriales.

Por otro lado debe considerarse que el uso de microcomputadoras tiene la ventaja adicional de no requerir a gente demasiado especializada en el manejo del sistema, ya que los sistemas operativos de las microcomputadoras son de menor complejidad que los de las minicomputadoras.

Aunado a lo anterior debe establecerse que un sistema de cómputo grande requiere de sistemas especiales como instalación de piso falso, cuarto dedicado al equipo, temperaturas del cuarto controladas, alto costo del equipo etc.

Por todo lo anterior pienso que los Ingenieros debemos de estar preparados para nuevas tendencias en el desarrollo de sistemas, los que involucrarán sin duda ayudas computarizadas, como el caso de este trabajo.

## **CAPITULO II**

### **PLANTEAMIENTO DEL PROBLEMA**

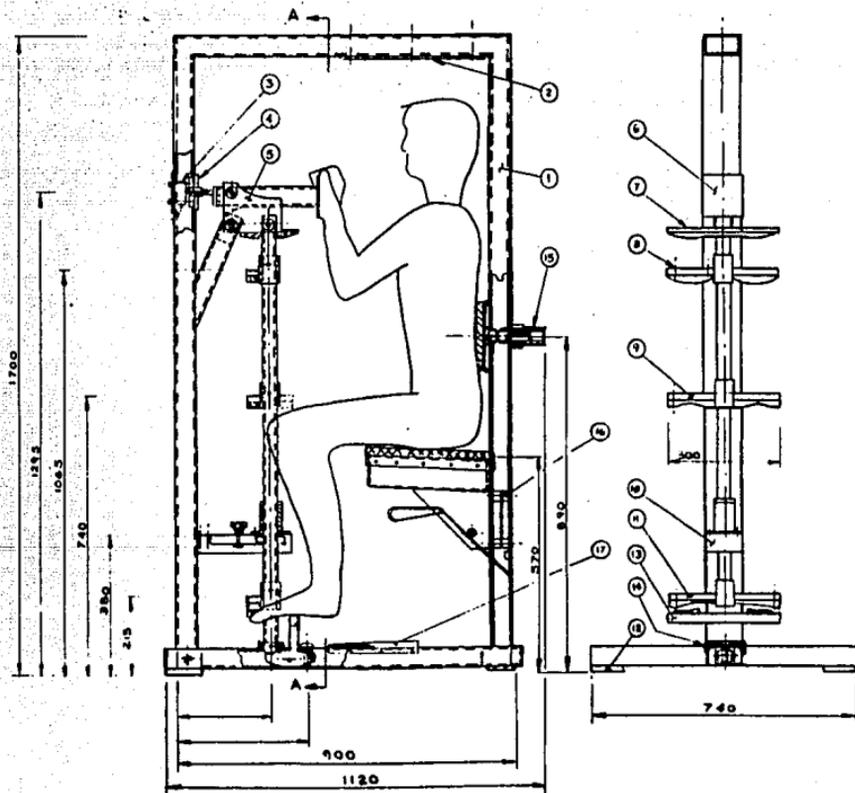
## PLANTEAMIENTO DEL PROBLEMA

### I) Antecedentes:

Se solicitó una ayuda computarizada para la adquisición de datos desde una máquina mecánica conocida como Sixalyser.

Esta máquina fue diseñada para la valoración del estado físico de personas con problemas por músculos contraídos en la parte superior del cuerpo (dorso).

La máquina en cuestión se presenta en la figura 2.1:



### LISTA DE PARTES

- 1) MARCO
- 2) COJIN DE LA CABEZA
- 3) SENSOR
- 4) TUBO PARA PRESION CON ABRASADERA ROSCADA
- 5) MECANISMO DE SUJECION DE BARRA GUIA
- 6) COJIN DE MANO PARA TRICEPS
- 7) COJIN DE BRAZO PARA DELTOIDES
- 8) BARRA MANUAL PARA BICEPS
- 9) COJIN PARA ILLIOPSOAS
- 10) GUIA
- 11) COJIN PARA PIE
- 12) COJIN PARA ESPINILLA (2 sensores adicionales)
- 13) DESCANSA PARA PIE
- 14) COJIN PARA DEDO
- 15) DESCANSA POSTERIOR
- 16) ENSAMBLE DEL ASIEN-TO
- 17) GUIA DEL PIE

**SIXALYZER STRENGTH  
TESTER**  
by LUKAS, S.A.  
Scale: none mm

Con este aparato es posible, de acuerdo a la experiencia de terapistas, evaluar los 6 músculos mas importantes en el dorso del cuerpo humano, que son:

**TRICEPS**

**DELTOIDE**

**BICEPS**

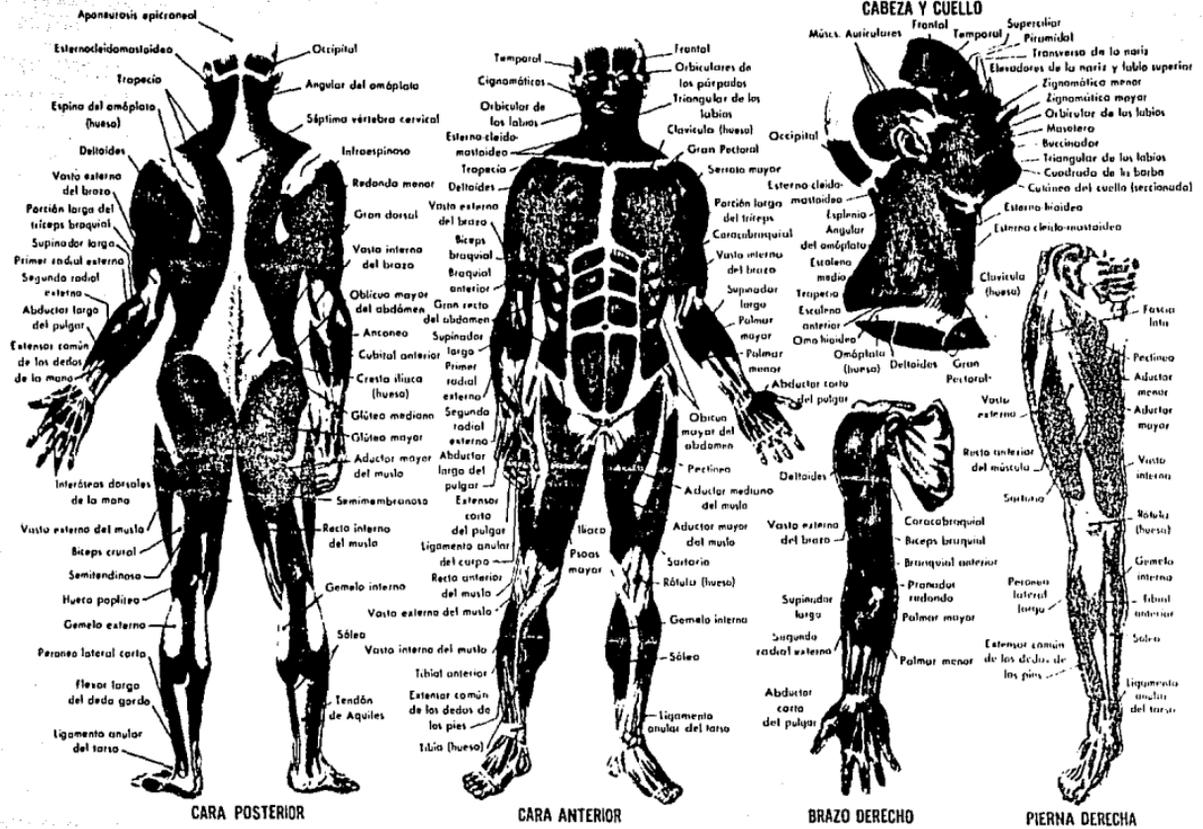
**ILLIOPSOAS**

**FLEXOR**

**EXTENSOR**

Los tres primeros músculos son irrigados por las terminaciones nerviosas provenientes de la parte alta del cuerpo (cuello), mientras que los tres últimos por los nervios de la parte media del cuerpo (cintura), en la figura 2.2. se muestra el sistema muscular y la ubicación de los músculos referidos.

# SISTEMA MUSCULAR



El aparato permite efectuar las pruebas por el método conocido como "TOMA DE FUERZAS", el cual permite el diagnóstico de problemas en las ramificaciones nerviosas y/o en los músculos directamente.

El método de la toma de fuerzas es un método comparativo entre los músculos del lado derecho y los del lado izquierdo del cuerpo humano. La teoría del método dice que "si existe un desbalance o diferencia mayor a un 15% en la fuerza que puede aplicar, el individuo al que se le está realizando la prueba tiene problemas dorsales y requiere un tratamiento".

Cuando el paciente presenta un desbalance es posible diagnosticar de manera precisa la terminación nerviosa y los músculos afectados. Antes de desarrollar el sistema de adquisición de datos por computadora existían dos versiones del sistema de medición, las cuales se describen a continuación:

En primer lugar se muestra un equipo con una báscula mecánica que se representa en la figura 2.3:

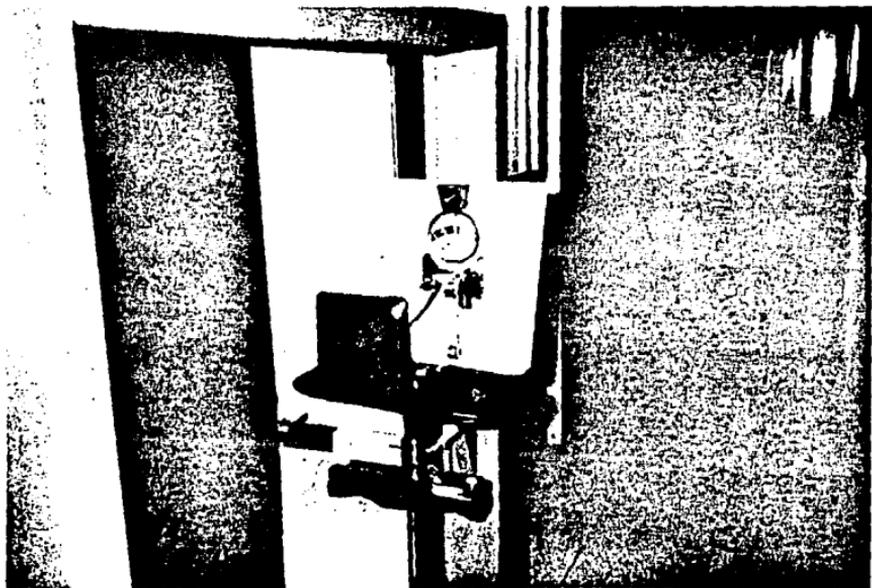


FIGURA 2.3 EQUIPO CON BASCULA  
MECANICA.

Con este sistema se tienen algunas ventajas las cuales son: un costo bajo de la báscula y retroalimentación directa al paciente, ya que éste ve directamente en la carátula de la báscula la magnitud de la fuerza aplicada. Pero también se tienen problemas como baja precisión de la báscula, ya que es muy difícil de calibrar adecuadamente, deslizamiento mecánico inercial al momento de imprimir la mayor fuerza por la construcción propia de la báscula e inestabilidad provocada por la posición vertical de la báscula.

El segundo equipo de adquisición de información consiste de un aparato registrador de fuerzas por muestreo con impresor. En la figura 2.4 se muestra este sistema.

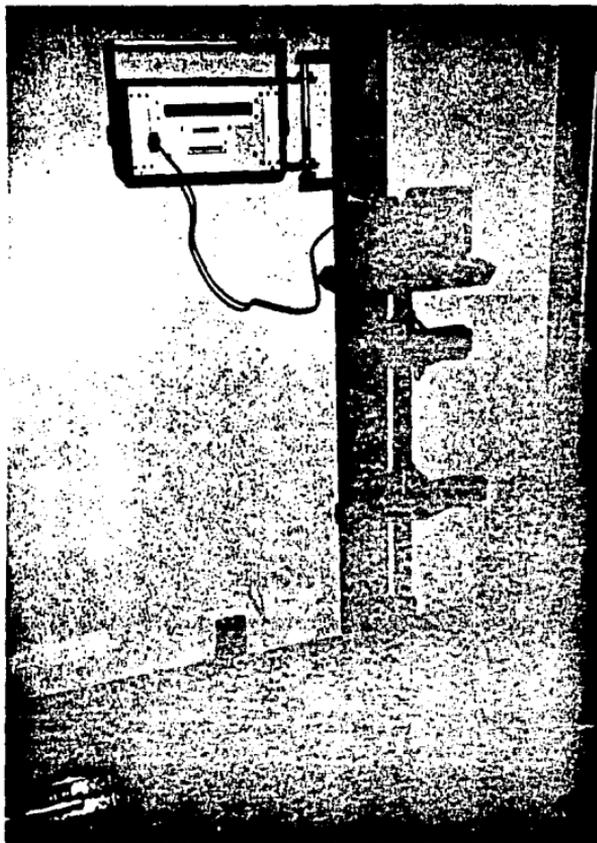


FIGURA . 2.4

Con este sistema se obtienen una serie de resultados, los cuales muestran algunos de los valores correspondientes a las fuerzas aplicadas; sin embargo, la baja velocidad de muestreo del sistema ocasiona que no se tenga la certeza de que se tiene el valor de la fuerza máxima aplicada; además es muy costoso el equipo y por si esto fuera poco, el médico tiene que identificar el cambio de miembro en la prueba basado en su experiencia para la lectura de la hoja de resultados. En la figura 2.5 se muestra una hoja de resultados típica de este aparato.

0.000
0.000
1.234
4.764
7.980
9.876
8.998
6.765
3.751
0.000
0.000
2.340
6.998
8.879
11.345
8.976
6.812
3.213
0.000
0.000

FIG. 2.5

En este sistema, para efectuar la adquisición de datos, es necesario el uso de dispositivos sensores de fuerza, que sean capaces de transmitir señales eléctricas al equipo de registro de fuerzas, se usarón en este caso; Celdas de Carga por compresión, mismas que se utilizaron en el sistema computarizado.

En cuanto al sistema a desarrollar este debe cumplir con una serie de requerimientos específicos que se enlistan a continuación:

1) Elaboración de un programa de computadora que incluya:

- Manejo fácil y rápido.
- Almacenamiento de datos personales de los pacientes, como: Nombre, edad, peso, domicilio, número del IMSS, etc.
- Elaboración de Hojas de Resultados.
- Adquisición de datos de Celdas de Carga.
- Graficación de Resultados.
- Respaldo de Información.
- Sin requerimientos de programación para su empleo.
- Búsqueda alfanumérica

b) Capaz de almacenar información anterior (consultas anteriores).

c) Impresión de resultados en papel.

d) El sistema debe tener la capacidad de poder expandirse a un máximo de tres sensores.

e) Posibilidad de manejo de información independiente (otros programas).

### **CAPITULO III**

### **ALTERNATIVAS**

### III) ALTERNATIVAS

Con las características planteadas por el usuario se inició la búsqueda de sistemas que cumplieran con los requisitos planteados.

#### CELDA DE CARGA DISPONIBLES.

Para el desarrollo del proyecto solamente se contaba con dos tipos de Celdas de Carga de compresión, una de entrada de voltaje y la otra de entrada de corriente.

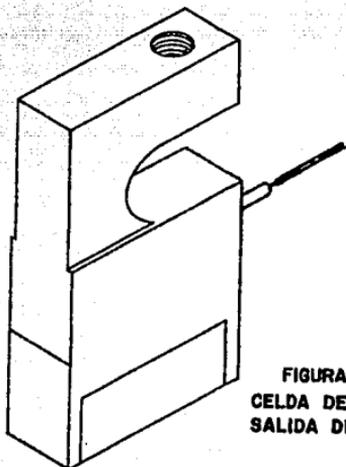
a) Celda de carga de voltaje marca INTERFACE. Las características de esta celda de carga para hacer la toma de fuerzas son:

- Diseño para mediciones de fuerza electrónica de precisión.
- Linealidad alta.
- Alta repetibilidad.
- Compensada térmicamente.
- Bajo costo.
- Fácil instalación.
- Capacidad desde 10 hasta 1000 libras (454 Kg).
- Lecturas analógicas.

**Características Electricas:**

- Voltaje de excitación recomendado +/- 10V.
- Salida nominal de voltaje= 3mV/V.
- Resistencia de entrada= 350 +/- 3.5 ohms.
- Resistencia de Salida= 350 +/- 3.5 ohms.

La figura 3.1 nos muestra esta celda de carga.



**FIGURA 3.1**  
**CELDA DE CARGA PARA**  
**SALIDA DE VOLTAJE**

Los cables de conexión de la celda de carga son:

**ROJO = +EXCITACION**

**BLANCO = -SALIDA**

**NEGRO = -EXCITACION**

**VERDE = +SALIDA**

b) Celda de carga con manejo de lazo de corriente. Esta es una Celda fabricada en Alemania. Sus características de ella son casi las mismas en cuanto a su funcionamiento que la descrita anteriormente y las diferencias principales entre ambas radican en la tensión de alimentación y en el manejo de la señal de salida, ya que con esta Celda se maneja un lazo de corriente de 4-20 mA.

En la figura 3.2 se muestra esta celda de carga.



**FIGURA 3.2**  
**CELDA DE CARGA PARA**  
**SALIDA DE CORRIENTE**

*Sus características principales de funcionamiento son:*

- Mediciones de gran precisión.*
- Linealidad alta.*
- Bajo costo.*
- Fácil instalación.*
- Voltaje de alimentación de 24 Volts.*

*Los cables de alimentación de la celda de carga son:*

*Rojo: entrada de voltaje de alimentación.*

*Negro: manejo de lazo de corriente de 4-20 mA.*

*Para poder conectar el equipo de cómputo con la celda de carga requerimos de un convertidor analógico/digital que debe ser compatible con ambos dispositivos.*

#### **TARJETAS DE INTERFASE PARA CONVERSION ANALOGICO/DIGITAL:**

*La tarjeta de interface es muy importante en nuestro sistema, ya que con ella los datos de la celda de carga se hacen comprensibles a la computadora.*

*De esta tarjeta los datos deben ser transmitidos de manera digital hacia la computadora.*

También es importante indicar que de la correcta elección de la tarjeta de interface depende en gran medida la rentabilidad del sistema. Debe observarse que la capacidad y funcionalidad de la tarjeta elegida, además de su precio y disponibilidad son los factores de mayor relevancia en la toma de decisión.

Solamente se encontraron en el mercado tarjetas compatibles con la IBM-PC. Cada una de estas tarjetas tiene características que nos permiten efectuar una elección con mayores fundamentos.

A continuación se enlistan las tarjetas que son utilizables en el desarrollo del proyecto:

- \* ADC-100
- \* ADC-200
- \* ADC-300
- \* AICP-A108
- \* AICP-A1016
- \* AICP-SG04

Las tres primeras son de la firma japonesa CONTEC y las tres últimas de la firma americana ACTION INSTRUMENTS.

Hagamos un análisis de las especificaciones, características y descripción funcional de cada tarjeta.

## ADC-100, 200 y 300

### Características:

- 16 Canales de entrada analógica sencillos u 8 diferenciales.
- 12 Bits de resolución.
- 2 Canales analógicos de salida.
- 50,000 Conversiones máximas usando comandos de acceso directo a memoria
- 24 Canales digitales programables de I/O.  
Manejo de Interrupciones.  
Rango de actualización programable.  
Operación de Voltaje Unipolar y Bipolar.  
Soporte de Programación de fácil empleo.  
Reloj calendario en tiempo real con respaldo de batería.

### Descripción Funcional.

Las tarjetas ADC-100, ADC-200 y ADC-300 son de alta velocidad, de multifunciones de entrada/salida, que puede emplearse en computadoras IBM-PC y compatibles.

Estas tarjetas son ideales en aplicaciones para adquisición de datos en laboratorio y aplicaciones de control industrial que necesitan una mezcla de entradas y salidas analógicas y digitales.

Las tres hacen las mismas funciones y manejan los mismos canales de E/S; la diferencia que tienen es que la ADC-200 maneja un lazo de corriente de 4-20 miliamperes y la ADC -300 maneja selección de ganancia de voltaje.

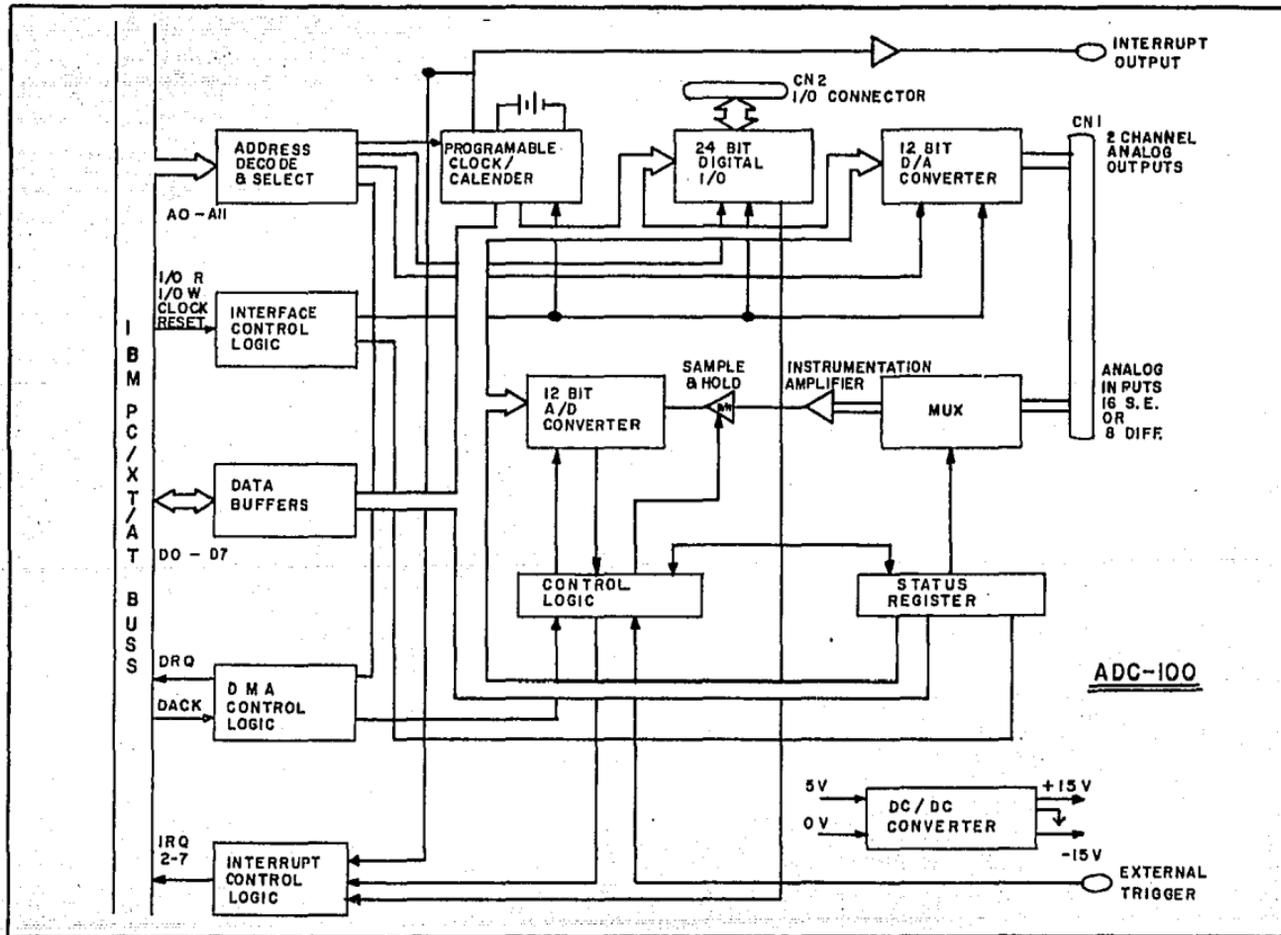
Las características adicionales de la ADC-200 y ADC-300 las hacen también de un precio mas alto.

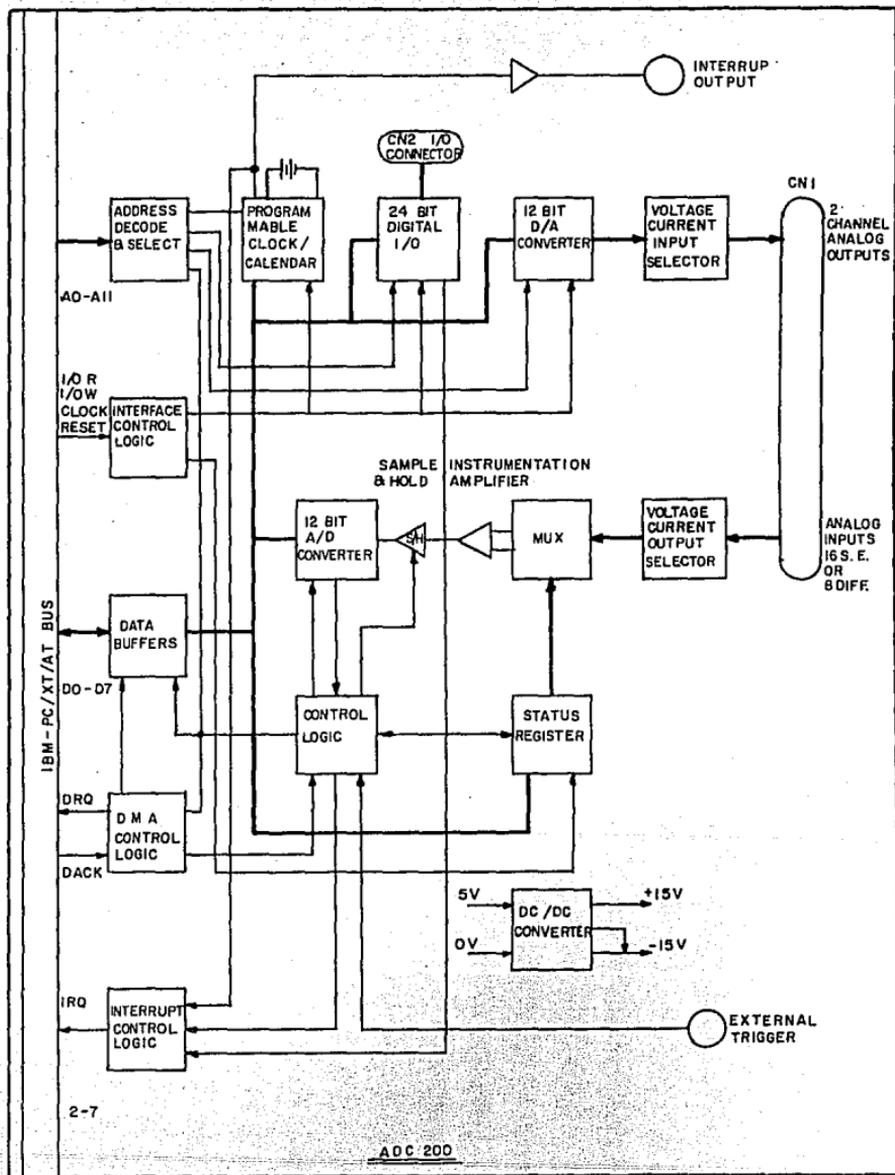
Los tres modelos manejan 16 canales de entrada analógicos (single-ended) o bien 8 canales de entrada diferenciales, 2 canales de salida analógicos, 24 canales de E/S digitales programables por el usuario y un reloj calendario respaldado con batería para aplicaciones de control en tiempo real. La transferencia de datos de alta velocidad se consigue con el acceso directo a memoria (DMA).

La conversión de 12 Bits Analógico/Digital puede ser iniciada por programa, por temporizador interno, por disparo externo o por una combinación del disparo y el temporizador. El dato convertido debe ser colectado con un comando de programación, una rutina de interrupción de servicio o por el uso de uno de los dos canales de DMA.

El software tambien proporciona comandos convenientes de alto nivel para la conversión Digital/Analógica (D/A), E/S digital y funciones de tiempo que requieran precisión en su ejecución.

A continuación se presenta un diagrama de bloques de los modelos ADC-100 y 200 de CONTEC en la figura 3.3.





## ESPECIFICACIONES DE LAS TARJETAS ADC-100, 200 Y 300

### Entradas Analógicas

16 sencillas/8 diferenciales (seleccionable)

#### Rango de Escala completa:

Bipolar: +/- 10 Volts o +/- 5V.

Unipolar: 0 a 10 Volts

Corriente: 4-20 mA (\*)

Ganancia de entrada: x1, x10, x100, x200 (\*\*)

#### Resolución:

12 Bits.

#### Presición de Conversión de entrada:

+/- 0.04 % FSR a 25 grados centigrados.

#### Velocidad de Conversión:

50,000 muestras por segundo (modo DMA).

### Salidas Analógicas:

2

#### Rango de Salida:

Unipolar : 0-10 Volts o 0-5 volts

bipolar : +/- 10 V, +/- 5V, +/- 2.5 V

Corriente: 4-20 mA (\*)

**Presición de conversión de Salida:**

**+/- 0.5 % FSR**

**Transferencia:**

**30 KHz/canal**

**Entradas/Salidas digitales:**

**24 canales de nivel TTL**

**Reloj en tiempo real**

**Rango de Direccionamiento:**

**Cualquier limite dentro de 64 KB.**

**Selección de Puertos de E/S:**

**Por DIP switches.**

**Ambiente de Operación:**

**Rango de temperatura de 0-50 Grados Centígrados.**

**Rango de Humedad 0/90 % no condensada.**

**\* Disponible únicamente para ADC-200**

**\*\* Disponible únicamente para ADC-300**

## AICF-A108

### Características:

- 8 Entradas analógicas sencillas
- 12 Bits de resolución
- 30,000 Conversiones máximas/segundo
- 3 Entradas Digitales
- 4 Salidas Digitales
- Soporte de Programación de fácil empleo
- Entrada de interrupción externa
- Contador de eventos
- Tarjeta terminal auxiliar.

### DESCRIPCION FUNCIONAL

La tarjeta A108 contiene 8 canales de entrada analógicos de voltaje y 7 canales de E/S digitales; La tarjeta se inserta en cualquiera de las ranuras de expansión de una computadora IBM-PC compatible, y es ideal para entradas comunes múltiples en adquisición de datos.

Las 8 entradas analógicas tienen un rango de +/- 5 Volts de escala completa, las señales de entrada son digitalizadas con una resolución de 12 bits (2.44 mV/cuenta) y enviadas al bus de datos de la computadora.

Si se usa BASIC compilado como lenguaje de programación de las aplicaciones, el flujo de datos es típicamente de 250 microsegundos, con lo que se obtiene un máximo de 4000 muestras/segundo. Sin embargo, el tiempo de transmisión por hardware llega a ser de 35 microsegundos, con lo cual el número de muestras/segundo se incrementa a cerca de 30,000.

Con el empleo de esta tarjeta se tiene la gran ventaja de poder soportar sobrevoltajes continuos de hasta  $\pm 30$  V en cada entrada analógica, considerando esta condición para entradas activas, las inactivas se consideran abiertas y en el caso de sobrevoltaje no son afectadas.

Se tiene además 7 canales de E/S digitales con lógica TTL. Estos se distribuyen en un puerto con 4 canales de salida y 3 canales de entrada.

Como característica adicional, se dispone de una tensión de salida de +5, +12 y -12 para aplicaciones que requieren de alguna referencia o para alimentar circuitos externos de bajo consumo de potencia.

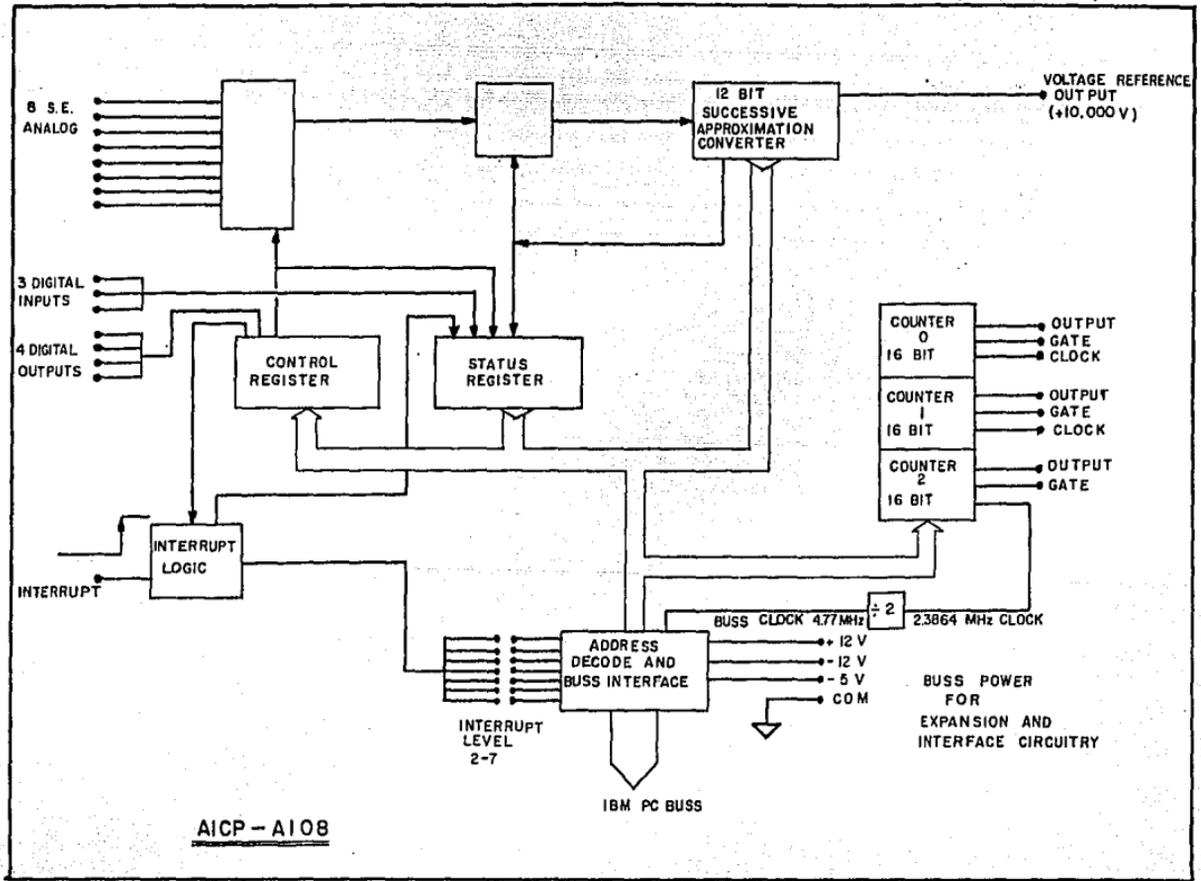
Un contador/temporizador programable proporciona interrupciones periódicas para el convertidor A/D y adicionalmente puede contar eventos, pulsos y generar formas de onda, frecuencia, periodo y medición de ancho de pulso.

Se cuenta con una entrada de interrupción externa para cualquier nivel de interrupción de la computadora (2-7), esta entrada habilita rutinas de interrupción para proporcionar datos de referencia recolectados o control de manejo de interrupción.

Una tarjeta auxiliar para conexiones con los dispositivos de E/S se incluye para facilitar al usuario las interconexiones y así evitar la manipulación de la computadora. La tarjeta principal y la tarjeta auxiliar incluyen conectores tipo D macho de 37 pines.

Existe un Disco Flexible (floppy disk) con ayudas de programación (software), que contiene entre otras una rutina de E/S la cual puede accesarse por medio de una instrucción de llamada de BASIC. Con esta llamada se pueden utilizar todas las E/S de la tarjeta tanto analógicas como digitales.

A continuación se muestra el diagrama de bloques de esta tarjeta de interfase en la figura 3.4.



AICP - A108

IBM PC BUSS

BUSS POWER FOR EXPANSION AND INTERFACE CIRCUITRY

## ESPECIFICACIONES DE LA TARJETA A108

**Número de entradas analógicas:**

**8 sencillas**

**Rango de escala completa de voltaje:**

**+/- 5 VDC**

**Resolución:**

**12 Bits (4096 cuentas, 2.44 mV/cuenta)**

**Corriente de entrada parcial:**

**100 nanoamperes máximo**

**Rechazo a modo común:**

**75 Db máximo**

**Sobrevoltaje de entrada:**

**+/- 30 Volts DC o AC de pico.**

**Estabilidad:**

**Cero: +/- 10 microvolts máxima**

**Escala completa: +/- 25 microvolts máxima.**

**Tiempo de transferencia:**

**35 microsegundos.**

**Entradas Digitales:**

0 lógico: 1 a 0.4 VDC.

1 lógico: 2.7 a 5 VDC.

**Rango de temperatura de operación:**

0 a 50 grados centígrados.

**Rango de temperatura de almacenaje:**

-20 a 70 grados centígrados.

**Humedad:**

0 a 90 % sin condensación.

**Requisitos de energía:**

+5 VDC, 180 mA

+12 VDC, 10 mA

-12 VDC, 16 mA

**Rango de direccionamiento:**

Cualquier límite de 8 bytes entre H200 y H3FF.

## TARJETA DE MULTIFUNCIONES DE E/S AICP-A1016

### Características:

- 16 Entradas analógicas sencilla u 8 diferenciales.
- 12 Bits de resolución
- 4 Entradas digitales
- 4 Salidas digitales
- 2 Salidas Analógicas
- 40,000 Conversiones/seg. máximo.
- Soporte de programación externo.
- Tarjeta terminal auxiliar.

### DESCRIPCION FUNCIONAL

Esta tarjeta puede acoplar 16 entradas de voltaje analógico sencillas ( 8 diferenciales), 2 salidas analógicas, 4 entradas y 4 salidas digitales a una computadora IBP-PC o compatible; y ocupa una ranura de expansión completa.

Puede emplearse en aplicaciones que requieran una mezcla de entradas/salidas digitales y analógicas.

Debido a la capacidad de manejo de canales sencillos y diferenciales de esta tarjeta, es posible alimentar las entradas de manera unipolar desde 0-1 volt hasta 0-10 volts y de forma bipolar desde  $\pm 1/2$  volt hasta  $\pm 10$  V, digitalizando los rangos de entrada a una resolución de 12 bits. Los rangos seleccionados son comunes para todos los canales de entrada y se eligen con un interruptor de ganancia controlando al amplificador de instrumentación.

Las ganancias que maneja el amplificador pueden aumentar en número y valores, instalando una ganancia resistiva en la tarjeta.

Las entradas analógicas pueden resistir sobrevoltajes continuos de un rango de  $\pm 30$  VDC continuo.

También cuenta con 2 canales de salida analógicos de 12 bits de resolución.

Las conversiones digitales/análogicas que generan estas salidas deben operarse con una referencia fija que esta disponible en la A/D16. Esto produce un rango de Voltaje de 0-5 volts, o bien pueden operarse con una referencia externa de AC o DC con el objeto de dar diferentes rangos de salida o señales atenuadas de AC.

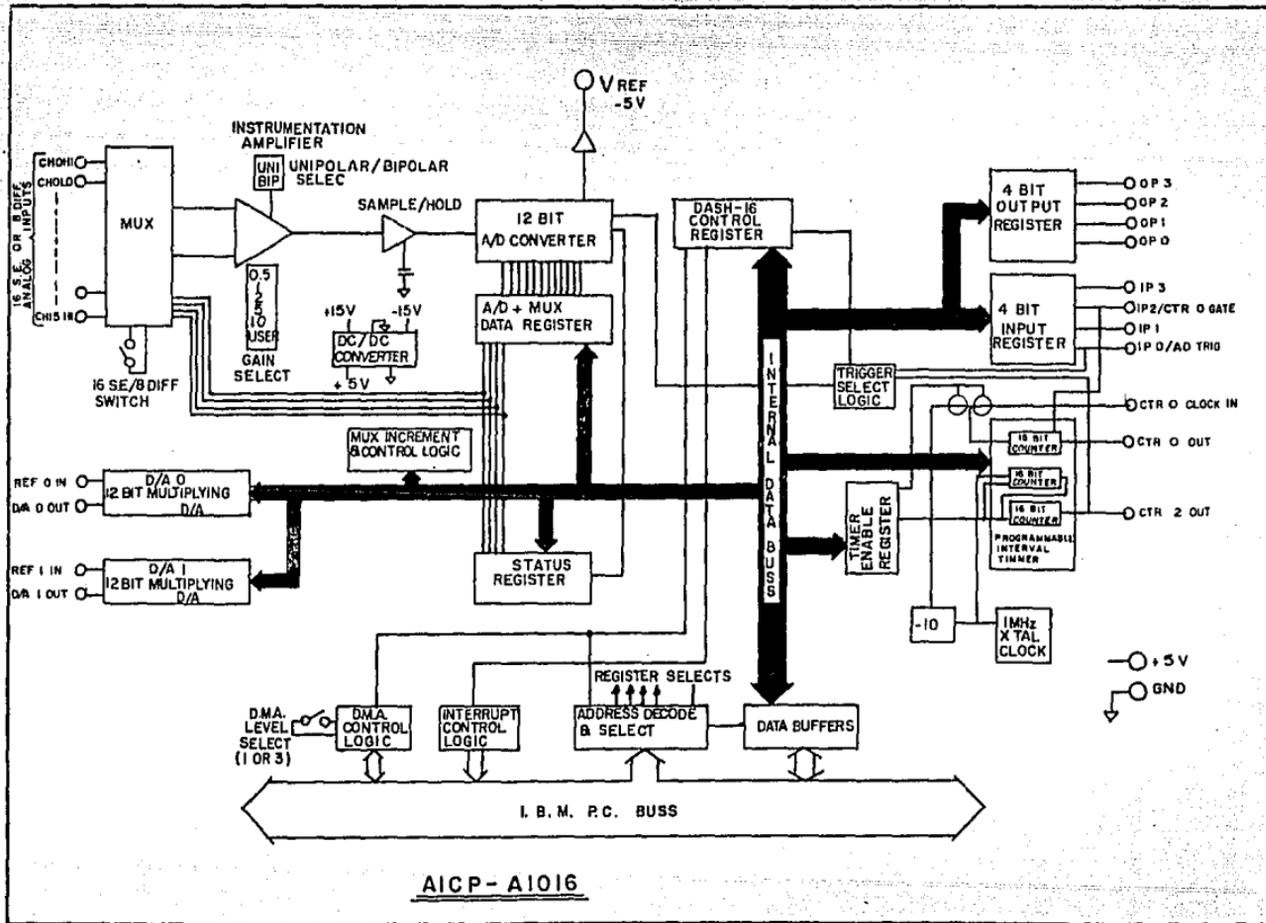
Las E/S digitales emplean niveles lógicos compatibles con TTL/DTL.

Para la conexión con los dispositivos de instrumentación se cuenta con una tarjeta auxiliar que se monta remotamente en la PC.

Tanto la A1016 como la tarjeta auxiliar tienen conectores D macho de 37 pines.

También se tiene un disco con programas de soporte, en los cuales se incluye una rutina de manejo de E/S de fácil manejo, la cual se ejecuta con una llamada desde BASIC.

En la figura 3.5 se muestra un diagrama de bloques de la tarjeta.



AICP-A1016

## ESPECIFICACIONES DE LA TARJETA A1016

### Entradas Analógicas:

16 sencillas / 8 diferenciales.

Ajuste de Ganancia (seleccionable por el usuario: 0.5,1,2,5,10).

### Rango de escala completo:

+/- 10 Volts

### Resolución:

12 Bits

### Sobrevoltaje de entrada:

+/- 30 VDC

### Rechazo a modo común:

90 dB típico

### Tiempo de transferencia (hardware):

25 microsegundos, típico

### Presición:

+/- 1 bit

### Sálidas analógicas:

2

**Rango de Salida analógica:**

5 Volts de CD con la referencia proporcionada en la tarjeta.

**Manejo de salida analógica:**

5 mA

**Entradas digitales:**

0 lógico: -0.5 a 0.8 VDC

1 lógico: 2 a 5 VDC

**Salidas Digitales:**

0 lógico: 0 a 0.4 VDC

1 lógico: 2 a 5 VDC

**Rango de temperatura de operación:**

0 a 50 grados centígrados

**Rango de temperatura de almacenamiento:**

-20 a 70 grados centígrados.

## TARJETA DE ENTRADAS DE TENSION

AICP-S604

### Características:

- 4 Entradas analógicas diferenciales
- Rango de ganancia infinito
- +/- 12 bits de resolución
- 2 Entradas de alarma por canal
- 8 Salidas digitales
- Soporte de programación
- Tarjeta de terminación auxiliar.

### DESCRIPCION FUNCIONAL

Esta fué la última de las tarjetas que se encontro para ayudar a resolver el problema, es de propósito general con entradas analógicas de bajo nivel con voltaje de excitación integral capaz de interfazar con: celdas de carga, instrumentos de medición de esfuerzos y con transductores de presión a través de una Computadora Personal compatible.

Sus atributos son especialmente útiles en aplicaciones de procesos por lotes.

Acepta un máximo de 4 entradas completamente diferenciales, con un rango desde +/- 10 milivolts hasta +/- 10 volts y digitaliza las señales de entrada a la computadora con una resolución de 12 bits (más uno de signo).

Cada uno de los canales de entrada de la S604 tiene la gran ventaja de ajustarse a cero y hasta el valor máximo de tensión de entrada individualmente y tener con ello una ganancia seleccionable.

Cuenta con la ventaja de inmunidad al ruido externo en sus canales diferenciales de entrada, debido a que posee dos filtros constantes en tiempo que se seleccionan por medio de un puente.

Tiene además dos alarmas de límite por canal y dos salidas digitales asociadas con las alarmas, lo que aumenta la capacidad de la tarjeta.

La tensión de excitación hacia la tarjeta es seleccionable por el usuario, contando con rangos de 5, 10 o 15 volts. Estas tensiones están aisladas de corto circuitos y son capaces de manejar corrientes del orden de hasta 200 miliamperes.

Cada canal de entrada diferencial proporciona dos setpoints de alarma independientes y bandas muertas ajustables. La polaridad de la banda muerta determina si la alarma opera como alarma alta (banda muerta negativa) o como una alarma baja (banda muerta positiva). Esto permite tres posibles configuraciones de alarma por entrada que son: alto-alto, alto-bajo y bajo-bajo. Cada setpoint tiene una salida digital que indica el estado de la alarma.

Las salidas digitales tienen la ventaja de operar de manera independiente desde la computadora y pueden conectarse directamente a relevadores de estado solido.

Otra posibilidad de las alarmas de límite en la tarjeta, es la habilidad de una alarma para generar una interrupción para que así las entradas diferenciales no tengan que actualizarse continuamente.

Así, en el caso de una aplicación por lotes, una vía de alarma puede usarse para interrumpir operaciones una vez que el nivel deseado (lleno o vacío) sea alcanzado.

Esta tarjeta tiene dos modos de operación seleccionables por el usuario, estos modos determinan la manera en que se harán las conversiones en las entradas.

Los modos referidos son el de conversión en demanda y el modo de actualización continua.

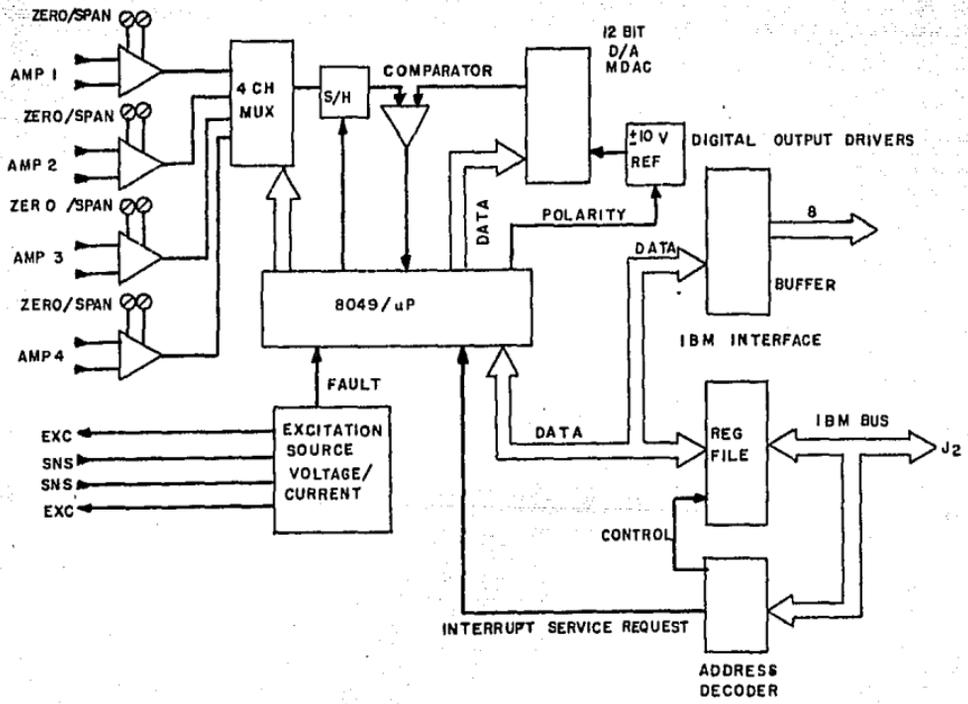
En el modo de conversión en demanda, la tarjeta se conecta en estado de espera hasta que la computadora inicia una conversión. La tarjeta entonces ejecuta la conversión para el canal indicado y envía la respuesta por el bus de datos en un tiempo total de transferencia de 2.3 milisegundos.

En el modo de actualización continua, todos los canales de entrada son monitoreados continuamente y los datos son recuperados del bus de datos, por cada canal. El dato que se envía al bus representa el promedio de los datos recuperados desde la actualización anterior y leídos previamente; con esta técnica se consigue superior rechazo al ruido.

En el modo de actualización continua, cada canal se monitorea una vez cada 9.2 milisegundos, lo que representa 108.7 actualizaciones por segundo.

Con esta tarjeta también se tiene una adicional para conexiones externas y con un programa de soporte que contiene subrutinas de manejo de las Entradas/Salidas, que puede ser accesado con una llamada BASIC.

**En la figura 3.6 se muestra el diagrama de bloques de esta tarjeta.**



AICP-SG04

## MICROCOMPUTADORA

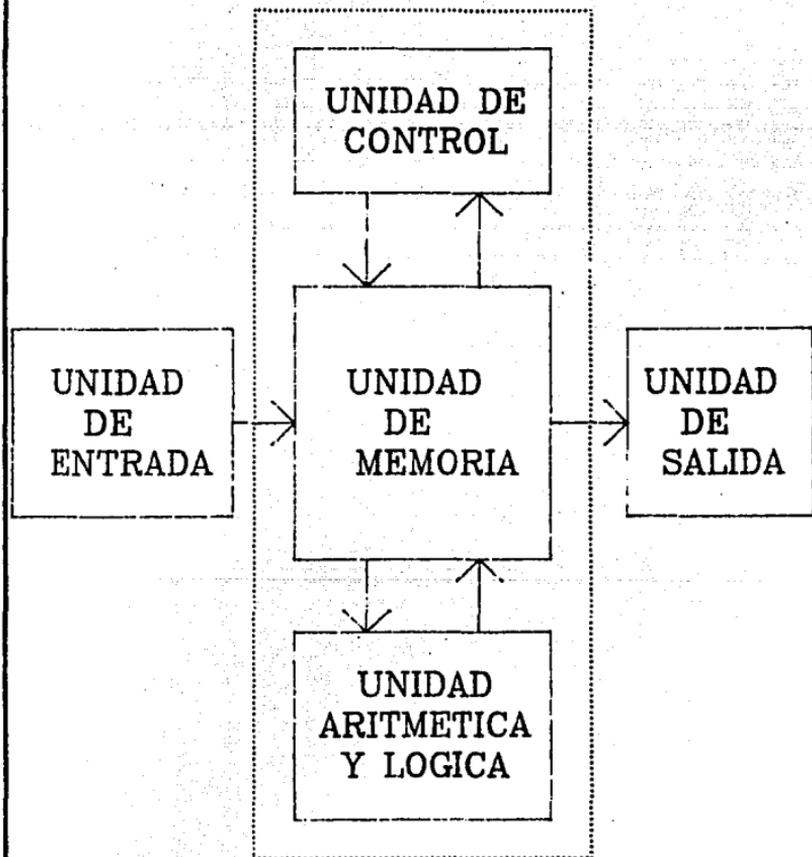
Debido a que para el desarrollo del proyecto era necesario incluir una computadora con capacidad de manejar información generada en cada consulta, se hace necesario explicar que es una computadora y como esta formada.

Una Computadora es una herramienta de proposito general que permite la obtención de ayudas mecanizadas para incrementar la productividad y hacer mas eficiente el trabajo del hombre.

A continuación se muestra un diagrama general de una Computadora en la figura 3.7:

### 37 DIAGRAMA GENERAL DE UNA COMPUTADORA

C.P.U.



Hagamos un análisis de cada una de las partes que forman a la Computadora y definamos que utilidad reportan a la función global de la misma.

Las unidades de Entrada/Salida son dispositivos que sirven para establecer la comunicación entre la Computadora y los medios externos, se les conoce comunmente como dispositivos periféricos.

Algunos de los dispositivos periféricos mas comúnmente empleados son:

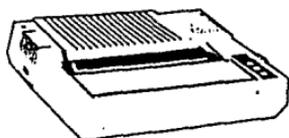
- Lectoras de tarjetas.
- Perforadoras de tarjetas.
- Terminales de video.
- Unidades de disco flexible.
- Unidades de disco rígido.
- Unidades de cinta magnética.
- Unidades de cassette.
- Graficadores.
- Impresoras.

En la figura 3.8 se muestran varios dispositivos periféricos o de Entrada/Salida.

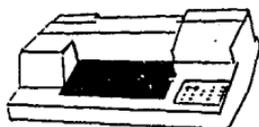
PERIFERICOS ENTRADA / SALIDA



TERMINAL  
DE  
VIDEO



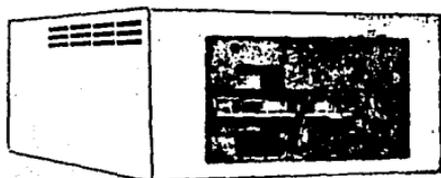
IMPRESORA



PLOTTER



UNIDAD  
DE  
DISCO



CASSETTERA

La unidad de memoria esta constituida por un gran número de localidades que pueden ser núcleos magnéticos o circuitos integrados, dependiendo del tipo de memoria que se emplee; en estas localidades es donde se almacenan los datos o las instrucciones que se generan en alguna sesión de trabajo de la computadora.

La unidad aritmética y lógica esta dividida en dos partes que son: la unidad aritmética y la unidad lógica.

La unidad aritmética se encarga de ejecutar las operaciones indicadas por la unidad de control. Las operaciones que esta unidad esta capacitada para realizar incluyen las de suma, resta, multiplicación, división, exponenciales, extracción de raíz cuadrada, etc.

La unidad lógica por su parte es la encargada de llevar a cabo operaciones del tipo Booleano, esto es operaciones con solamente dos niveles de respuesta que son ciertas o falsas.

La unidad de control por su parte se encarga del manejo y coordinación de todo el conjunto de operaciones dentro de la Computadora para conseguir un tratamiento adecuado y eficiente de la información.

De todas estas unidades se puede decir que quien forma propiamente a la Computadora es lo que se conoce como Unidad Central de Procesamiento o CPU, la cual esta formada por las unidades de memoria, aritmética y lógica y por la unidad de control.

Para que todas las unidades puedan trabajar como una Computadora se requiere de dos elementos muy importantes que son:

### Hardware y Software

El Hardware es todo lo que se refiere al aspecto físico de la Computadora como circuitos, gabinetes, teclados, dispositivos, etc.

El software por su parte se forma por un conjunto de programas realizados, que permiten a la computadora una interfase inteligente con el hombre.

Debe entenderse que un programa es un conjunto de instrucciones lógicas y estructuradas, que son escritas de una manera secuencial y que se utilizan para la solución de problemas específicos.

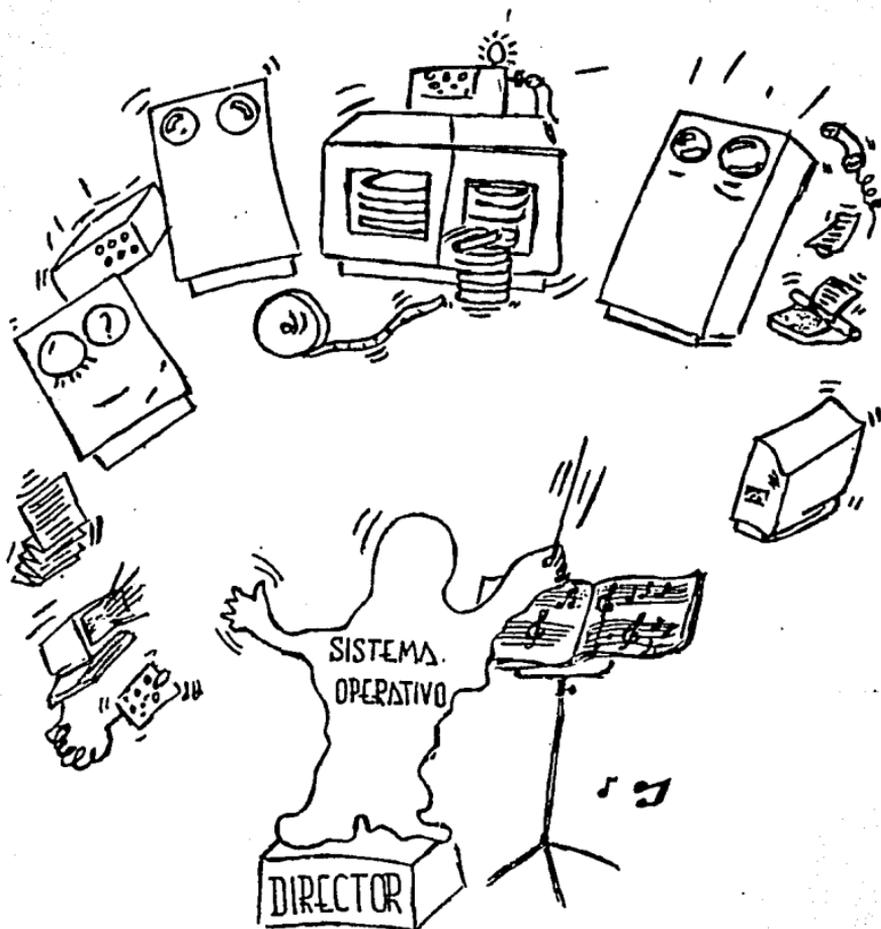
Con la creación de los lenguajes de programación se dio también origen a los Sistemas Operativos de las Computadoras.

*Un Sistema Operativo es un conjunto organizado de datos y programas, cuyo objetivo principal es la administración de todos los recursos de la Computadora, así como crear facilidades para el desarrollo de programas de computación y controlar su ejecución dentro de la Computadora.*

*Un Sistema Operativo es más eficiente si dispone de mayores facilidades para el desarrollo de programas y usos de la Computadora.*

*En la figura 3.9 se muestra un esquema representativo de la función de un Sistema Operativo.*

# DIAGRAMA REPRESENTATIVO DE LA FUNCION DE UN SISTEMA OPERATIVO



Las Microcomputadoras cuentan basicamente de los siguientes elementos:

#### La Computadora

Una o dos unidades de disco blando

Una o dos unidades de disco duro

Un teclado

Un monitor

La computadora contiene las tarjetas de memoria, el CPU, los dispositivos de manejo de Entrada/Salida, la fuente de alimentación, las unidades de disco flexible y de disco duro y tarjetas con conectores para los circuitos perifericos de Entrada/Salida como graficadores e impresoras y tarjetas de expansión.

El monitor es la pantalla del sistema en la que se visualiza toda la información procesada dentro de la computadora y puede ser monocromático o de color.

El teclado es el medio en el cual se transfieren las ordenes a la Computadora y la distribución de los caracteres es basicamente la misma que la de una máquina de escribir, aunque este teclado cuenta con una serie de teclas de funciones especiales que son empleadas generalmente para generar ayudas al usuario.

Las unidades de disco que estan contenidas en el interior de la Computadora tienen la finalidad de almacenar datos y programas de utilidad para el usuario, existiendo de varios tipos y capacidades dependiendo del tipo de Computadora que se este empleando.

Dentro de las computadoras disponibles se deben de considerar como factores importantes para su evaluación:

- Tipo de Computadora
- Sistema Operativo
- Refaccionamiento y servicio.
- Garantia.
- Precio.
- Posibilidad de Crecimiento
- Existencia de Interfaces adecuadas como convertidores A/D.

## EL LENGUAJE DE PROGRAMACION

El programa a desarrollar debe tener la capacidad de manejar varios archivos con datos independientes y además debe adquirir datos provenientes de una celda de carga en tiempo real.

Debe también incluir la posibilidad de inscribir a nuevos elementos que formen nuevos archivos, considerando la ejecución de reportes gráficos e impresos de los resultados que se obtengan en las pruebas.

Se necesita que cada archivo contenga los datos generales del paciente, siendo estos en principio: nombre, dirección, edad, peso, número de registro del Seguro Social, control de número de paciente y control de número de consulta.

Después del registro de los datos, debe ser posible realizar correcciones en cualquiera de los campos del archivo y posteriormente el programa debe conducir al usuario al módulo de pruebas de Toma de Fuerzas en los músculos. Se debe considerar que es necesario que los reportes de resultados se impriman en papel.

El resultado que se pide desplegar en el reporte es el de la mayor fuerza ejercida durante el desarrollo de las pruebas.

Se debe construir un modulo que sea capaz de traer la información de la memoria y que despliegue los resultados de consultas anteriores, ya que por medio de este modulo el médico detectara los avances que se estan obteniendo en el tratamiento de cada paciente.

Los archivos de los pacientes deben tener la capacidad de manejar consultas posteriores a la primera y guardarlas en el mismo archivo particular de cada paciente.

Este programa debe estar preparado para respaldar información en discos blandos y eliminar archivos de manera selectiva.

El último requisito en la construcción del programa consiste en la factibilidad de contar con una busqueda de pacientes por nombre.

Considerando todos los detalles que se han mencionado en las lineas anteriores, es necesario plantear las características de los lenguajes de programación para decidir cual de ellos resulta mas conveniente para efectuar el programa propuesto.

Definamos primeramente que un lenguaje de programación en computadora, traduce el código fuente del programa de aplicación en un código objeto o lenguaje de máquina que el procesador de la computadora puede entender.

Los lenguajes de programación pueden ser interpretados o compilados. Los interpretadores de programas fuerzan a las computadoras a trasladar cada instrucción como se recibe.

Los programas compilados por su parte, son procesados en su totalidad después de que todos los pasos han sido entendidos, hacen uso de toda la estructura lógica del programa. Estos programas son ejecutados de una manera muy rápida y eficiente y son mucho más convenientes que los interpretados.

Los lenguajes que se analizarán para el desarrollo del programa son: ensamblador, BASIC, FORTRAN y PASCAL.

Dentro de estos es importante observar que PASCAL por ejemplo, es un lenguaje de alta estructura y debe ser ordenado cuidadosamente y que BASIC por ejemplo, es un lenguaje no estructurado, que se considera como revuelto y en ocasiones difícil de leer.

Lenguaje ensamblador.- Es extremadamente rígido y no estructurado, está relacionado cercanamente al código de máquina, es demasiado difícil de aprender, se usa para escribir en amplio detalle y para optimizar elementos del programa.

**BASIC** (Código de instrucciones simbólico de propósito general para principiantes).- Es un lenguaje de alto nivel, no estructurado e interactivo, es popular entre no programadores debido a la similitud de sus comandos con el idioma Inglés común, pero es considerado como débil y confuso por programadores expertos. Tiene el problema de que su estructura no le permite compartir partes del programa las que deben de escribirse nuevamente cada que sea necesario.

**FORTRAN** (Trasladador de fórmula).- Estuvo entre los primeros lenguajes de alto nivel introducidos en las computadoras, no es tan eficiente como algunos de los lenguajes nuevos como FASCAL, aunque ha sido empleado comunmente por los Ingenieros y existe un gran número de programadores debido a su capacidad para ejecución de gráficas, control de procesos y otras aplicaciones comunes en Ingeniería. El uso de este lenguaje ha declinado debido a la aparición de lenguajes mas eficientes y mejor estructurados.

**PASCAL**.- Como se comento algunas lineas atras, es un lenguaje de muy alta estructura, modular y exacto, es apropiado para programas muy largos y complejos, no es interactivo y es de facil aprendizaje, esta reemplazando rapidamente al FORTRAN como elección en programas de Ingeniería. Aunque son totalmente diferentes, FASCAL ejecuta las mismas tareas de manera mas eficiente y con un programa mas fácil de entender.

## **CAPITULO IV**

### **EVALUACION Y ELECCION**

## **EVALUACION Y ELECCION**

### **TARJETA DE INTERFASE**

*Para tener un panorama general en la elección de la tarjeta de interfase, se ha construido una tabla comparativa, en la que estan resumidos los detalles más relevantes de cada una de ellas.*

*La tabla en cuestión es mostrada a continuación en la figura 4.1:*

MODELO	AICP A108	AICP A1016	SG04	ADC-100	ADC-200	ADC-300
NUMERO DE ENT. ANALOGICAS	8 SE	16SE/8 DIFF	4 DIFF	16SE/8 DIFF	16SE/8 DIFF	16SE/8 DIFF
TIPO DE ENT. ANALOGICAS	VOLTAJE DC	VOLTAJE DC	VOLTAJE DC	VOLTAJE DC	VOLTAJE DC	VOLTAJE DC
RESOLUCION	12 BITS	12 BITS	$\pm 12$ BITS Y SIGNO	12 BITS	12 BITS	12 BITS
RANGOS SELEC-CIONABLES DE VOLTAJE	$\pm 5$ VOLTS	RANGOS	INFINITO	$\pm 10V_0 \pm 5V$ .BIPOLAR 0 A 10V UNIPOLAR	$\pm 10V_0 \pm 5V$ .BIPOLAR 0 A 10V.UNIPOLAR	$\pm 10V_0 \pm 5V$ .BIPOLAR 0 A 10V.UNIPOLAR
NUMERO DE SALIDAS ANALOGICAS	—	2	—	2	2	2
NUMERO DE I/O DIGITALES	3 ENTRADAS 4 SALIDAS	4 ENTRADAS 4 SALIDAS	8 SALIDAS	24 PROGRAMABLES	24 PROGRAMABLES	24 PROGRAMABLES
PRECIO EN USA	795 USCy	1,295 USCy	1,150 USCy	1,095 USCy	1,348 USCy	1,270 USCy
TIEMPO DE ENTREGA	4 SEMANAS	6 SEMANAS	6 SEMANAS	6 SEMANAS	6 SEMANAS	6 SEMANAS
GARANTIA	1 AÑO	1 AÑO	1 AÑO	6 MESES	6 MESES	6 MESES
PERMISO DE EXPORTACION	NO REQ.	SI REQ.	SI REQ.	SI REQ.	SI REQ.	SI REQ.

Con la ayuda de esta tabla debemos replantear los requisitos de nuestro sistema, para así decidir de una manera más objetiva.

Necesitamos en principio un canal analógico de entrada que incluya la posibilidad de adquisición de datos desde una celda de carga y que pueda tener 2 sensores de carga adicionales, por lo que se requieren otros dos canales analógicos.

Por lo tanto el crecimiento mayor previsto, abarca un máximo de tres sensores analógicos que pueden ser de voltaje o de corriente indistintamente.

Considerando los datos planteados anteriormente, hagamos un análisis final de cada una de las tarjetas que fueron previamente mencionadas, pero considerando su factibilidad para esta aplicación.

#### AICP-AIOB

Con esta tarjeta se tienen 7 canales analógicos de sobra en la primera etapa del proyecto, requiere de algunas adaptaciones para manejar los transductores tanto en el caso del de tensión como el de corriente.

En el caso del sensor de tensión, el problema puede resolverse implementando un circuito de amplificación de la señal que proviene de la celda de carga, a un nivel que la computadora sea capaz de interpretar adecuadamente.

En el caso del transductor de corriente, debido a su tensión de alimentación de 24 volts, es necesario diseñar una fuente que maneje esa tensión y además debe implementarse un circuito para manejar un lazo de corriente de 4-20 mA.

En ambos casos los equipos auxiliares son de bajo costo y de fácil diseño.

Los canales digitales de la tarjeta tampoco serán usados en el desarrollo del proyecto.

Esta tarjeta es la que representa un menor costo para la realización del trabajo y se considera desde ahora como una excelente elección.

#### AICP-AI016

Con el uso de esta tarjeta, se tendrán 13 canales analógicos de entrada en exceso aún con el sistema expandido a su máxima capacidad. Las 2 salidas analógicas, así como las 4 entradas y 4 salidas digitales no serán de utilidad en esta aplicación.

Para el uso de los sensores con ella, deberán hacerse las mismas implementaciones que en el caso anterior y el precio de la tarjeta es 500 dolares mayor que el de la tarjeta AIOB, en aplicaciones que requieran de algún tipo de control esta tarjeta puede ser ideal.

#### AICPG-8604

Para la utilización de esta tarjeta se tiene 1 canal de sobra considerando la máxima expansión, pero para este caso no se requieren canales diferenciales ya que no se trata de una aplicación de control. Las mayores ventajas con el empleo de esta tarjeta se presentan cuando se usa la celda de carga de voltaje, ya que con su empleo no es necesario amplificar los rangos de entrada desde 10 milivolts; sin embargo, para el caso de la celda de corriente se requiere también de la fuente externa y del circuito de manejo del lazo de corriente.

Su precio es 355 dolares mayor al de la AIOB y no se considera como una buena elección.

ADC-100, 200 y 300

Como se vio inicialmente al hacer la presentación de cada una de estas tarjetas, practicamente las 3 realizan las mismas funciones solo que el modelo 200 maneja directamente un lazo de corriente y el modelo 300 maneja una ganancia programada de voltaje.

De hecho estas tarjetas tienen características muy parecidas a la A1016, por lo que resultan muy excedidas para nuestro objetivo.

Por lo tanto se ha decidido realizar el proyecto con la tarjeta AICP- A108 que es la de menor precio y la que posee características más ajustadas para el proyecto.

### MICROCOMPUTADORA

La decisión del tipo de computadora para este proyecto se basó como se había anticipado previamente en la disponibilidad de interfases, en el uso de ellas dentro del mercado y de la garantía y servicio que ellas representan, precisamente por estas razones se optó por las compatibles con la IBM-PC ya que además de tener un rango de selección bastante amplio debido al gran número de fabricantes, los niveles de precios de estas computadoras son muy variados y depende de las necesidades del usuario el elegir la más adecuada.

Además como se explicó anteriormente solamente existen tarjetas compatibles con las computadoras IBM - PC capaces de adquirir datos de equipos de instrumentación y por ello la decisión de realizar el proyecto con este tipo de computadoras fue automática.

Con este tipo de máquinas se puede trabajar con monitores monocromáticos o en color, considerando que la adquisición de datos necesita del desarrollo de gráficas animadas en tiempo real.

## ELECCION DEL LENGUAJE

Tomando en cuenta que la tarjeta cuenta con un manejador para las Entradas/Salidas desarrollado para el lenguaje BASIC que ya ha sido previamente evaluado, encontramos que en el mencionado manejador se notan las siguientes limitantes:

- Adquisición de datos pobre con velocidad de muestreo sumamente baja.
- Elaboración de gráficas de muy baja calidad.
- Manejo de las rutinas por medio del interpretador de BASIC lo que hace el procedimiento muy lento.
- Necesidad de empleo de un lenguaje ensamblador para una mejor adquisición de datos, ya que con el manejador en BASIC no es posible trabajar en tiempo real.

Por todas estas limitantes se concluye que el lenguaje BASIC no es conveniente para el desarrollo de este programa.

El lenguaje ensamblador resulta de uso muy tedioso y poco práctico para un programa tan largo como el que se pretende hacer; el tiempo de la elaboración del programa resulta un problema adicional. Por lo tanto este lenguaje no representa una solución efectiva.

El lenguaje FORTRAN se considera como una buena opción, pero el disponer de un lenguaje mejor estructurado y que puede hacer el mismo tipo de programas de una manera mas efectiva nos llevan a deshechar su posible empleo.

La desición por el uso de PASCAL se debe al poderio que tiene el lenguaje en el que se incluyen las posibilidades de elaboración de gráficas, empleo de ventanas, manejo de editor, manejo de compilador, despliegue de mensajes de error durante la elaboración del programa, posibilidad de adquisición de datos en tiempo real debido a un número de conversiones alto etc.

La tarjeta de interfase esta preparada ademas para recibir comandos de Entrada/Salida desde PASCAL.

Se utilizo el programa TURBOPASCAL, el que permite el uso de todos los comandos de PASCAL y cuenta con algunas ventajas. Si se desea obtener mayor información de estos lenguajes puede consultarse al manual de usuario respectivo.

## ***CAPITULO V***

### ***IMPLEMENTACION DEL SISTEMA***

## IMPLEMENTACION DEL SISTEMA

Una vez decididos todos los elementos del sistema, pasamos a su implementación particular.

a) Las celdas de carga.

- Celda de carga de voltaje

Como se ha mencionado previamente, es necesario amplificar la señal que se recibe de la celda de carga de voltaje. Esta señal es originalmente de 20 milivolts/volt y considerando que se tiene como máximo una tensión de 5 volts disponible del canal de entrada, el diseño original puede conseguir una lectura máxima de 100 milivolts y con ella no es posible mostrar gráficamente el resultado deseado.

El circuito que se hace esta basado en un amplificador operacional con una ganancia positiva. A continuación se explica el funcionamiento de un amplificador operacional.

Su símbolo electrónico es el siguiente:

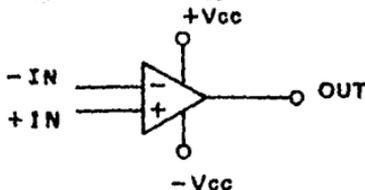


FIGURA 5.1 SIMBOLO ELECTRONICO DE UN AMPLIFICADOR OPERACIONAL

En donde:

+ Vcc = Voltaje suministrado en la entrada positiva.

- Vcc = Voltaje suministrado en la entrada negativa.

+ IN = Entrada no inversora

- IN = Entrada inversora

OUT = Salida

Sus características funcionales son:

1) Alta impedancia en la entrada

2) Baja impedancia de salida

3) Ganancia de voltaje variable

La ganancia de un Amplificador Operacional en voltaje (AV), puede fijarse por medio de un valor de resistencias externas; en la siguiente figura, se muestra un arreglo de amplificación, el que determina su ganancia de acuerdo a los valores de las resistencias externas.

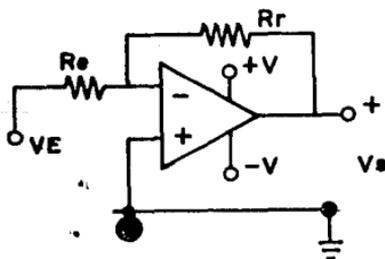


FIGURA 5.2 CIRCUITO DE AMPLIFICACION CON GANANCIA RESISTIVA

Con este circuito básico de amplificación, se hace el diseño del amplificador que se va a emplear en el proyecto para el caso de la celda de carga de voltaje.

### Celda de carga de corriente

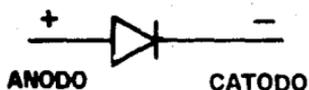
Con la celda de carga de corriente se tienen que resolver dos problemas que son:

- El suministro de una tensión de 24 volts para alimentar al sensor
- El manejo de un lazo de corriente (4 - 20 mA).

Para el manejo de la tensión de 24 volts se diseñó una fuente de energía con la ayuda de un transformador reductor, un rectificador de onda completa diseñado con diodos, una resistencia de circuito, un capacitor como filtro y un regulador de voltaje.

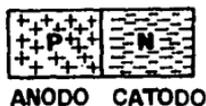
Veamos primeramente que es y como funciona un diodo para poder de esta manera entender cual es la función de un rectificador de onda completa.

El simbolo electrónico del diodo es el siguiente:



**FIGURA 5.3 SIMBOLO ELECTRONICO DEL DIODO**

*El ánodo está polarizado positivamente y el cátodo negativamente. Físicamente, el diodo está constituido por dos capas o junturas como se muestra a continuación:*



**FIGURA 5.4 JUNTURAS DE UN DIODO**

*Los diodos son semiconductores generalmente contruidos con silicio o con germanio y contienen 4 electrónes en su orbita exterior y pueden o no conducir de acuerdo a la manera en la cual son conectados (polarizados).*

Consideremos un diodo conectado a una fuente externa de corriente directa polarizado positivamente (conduciendo), como se muestra a continuación.

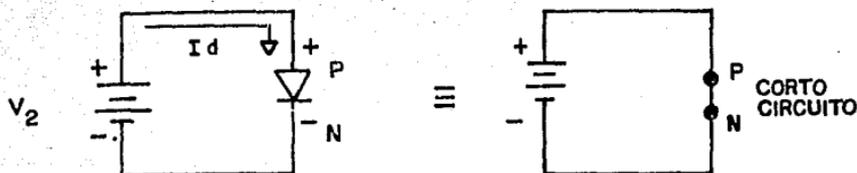


FIGURA 5.5 CIRCUITO CON UN DIODO POLARIZADO POSITIVAMENTE (CONDUciendo).

El voltaje suministrado ( $V_g$ ) al diodo, es aplicado en la juntura P y va hacia la juntura N, lo que provoca un circuito cerrado y en consecuencia la conducción de corriente por el diodo.

Considerando ahora el mismo diodo conectado de manera invertida a la fuente externa. Tenemos el siguiente circuito:

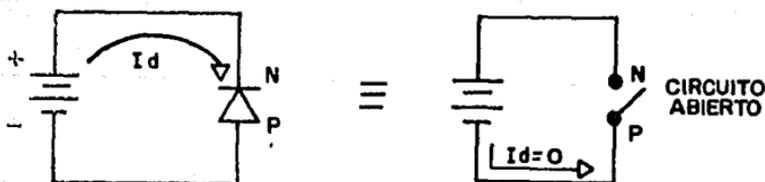


FIGURA 5.6 CIRCUITO CON UN DIODO POLARIZADO NEGATIVAMENTE (ABIERTO).

Para un voltaje negativo ( $-V_g$ ) aplicado de N hacia P (polarización negativa), no existe generación de corriente negativa  $I_d$  que circule de N hacia P, independientemente de la magnitud de tensión aplicada y por lo tanto el diodo se comporta como un circuito abierto.

Un diodo puede considerarse por lo tanto como un interruptor controlado por la polaridad de la tensión aplicada. Siendo un interruptor cerrado para el caso de polarización directa o un interruptor abierto para el caso de polarización inversa.

Una vez hecho el análisis del funcionamiento elemental del diodo pasemos al diseño del rectificador de onda completa tipo puente, que nos ayudara para alimentar con 24 volts de CD. al sensor, este circuito se muestra a continuación.

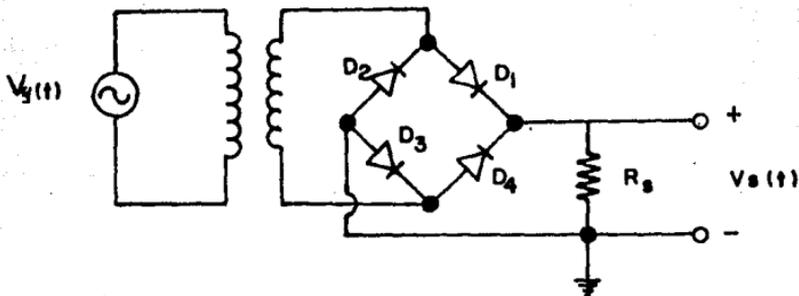


FIGURA 6.7 CIRCUITO RECTIFICADOR DE ONDA COMPLETA

Hacemos el análisis del circuito para al semiciclo positivo y lo mostramos en el circuito que se presenta a continuación, considerando también la forma de onda resultante.

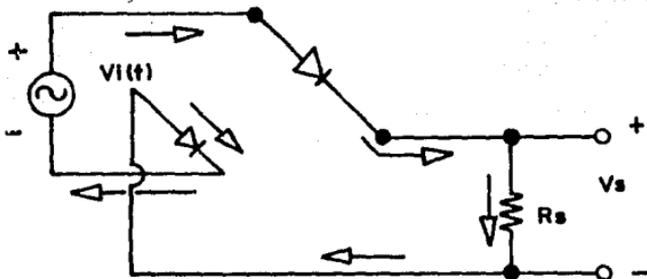


FIGURA 5.8 ANALISIS DEL SEMICICLO POSITIVO DE UN PUEBTE RECTIFICADOR DE ONDA COMPLETA.

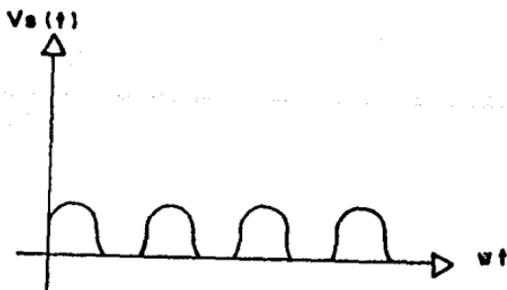


FIGURA 5.9 FORMA DE ONDA RESULTANTE DEL SEMICICLO POSITIVO DEL RECTIFICADOR DE ONDA COMPLETA.

Hacemos ahora el análisis para el semiciclo negativo de la misma forma que en el caso anterior.

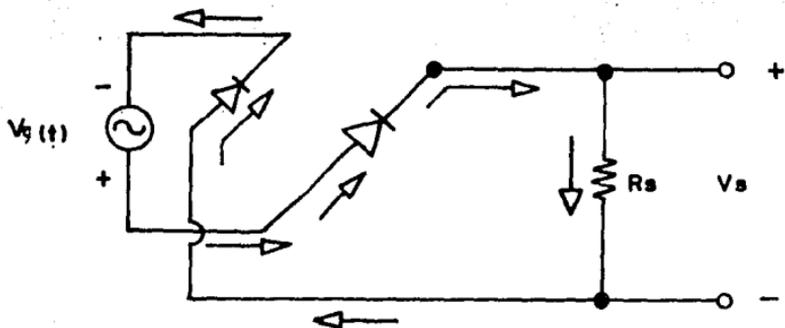


FIGURA 5.10 ANALISIS DEL SEMICICLO NEGATIVO DE UN RECTIFICADOR DE ONDA COMPLETA.

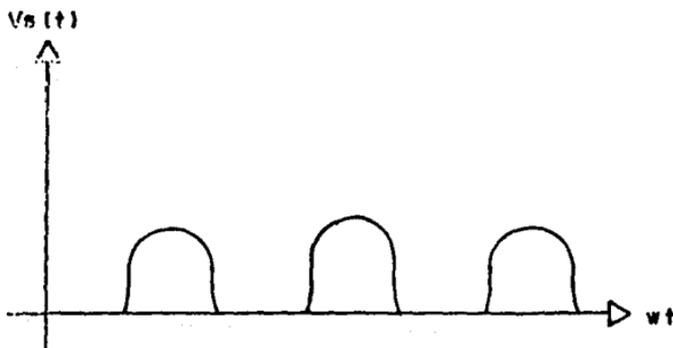


FIGURA 5.11 ANALISIS DE LA FORMA DE ONDA RESULTANTE DEL SEMICICLO NEGATIVO DEL RECTIFICADOR DE ONDA COMPLETA.

ESTA TESIS NO DEBE  
SALIR DE LA BIBLIOTECA

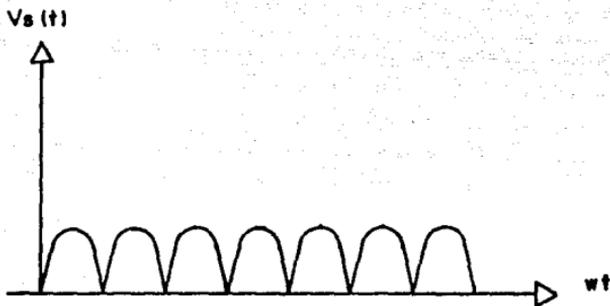


FIGURA 5.12 . FORMA DE ONDA RESULTANTE DE  
LOS DOS CIRCUITOS DE RECTIFICACION  
CONJUNTADOS.

Tenemos en este momento una onda rectificada de tipo positiva y continua, para conseguir una onda continua pura de CD, emplearemos un circuito que se conoce como rectificador de pico; este circuito se ilustra a continuación:

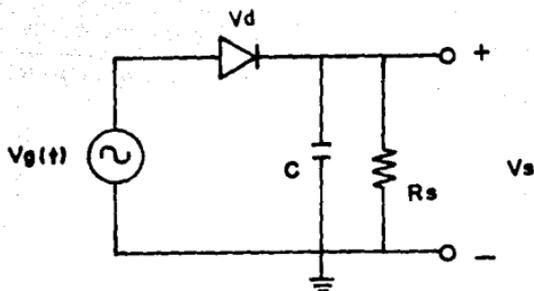
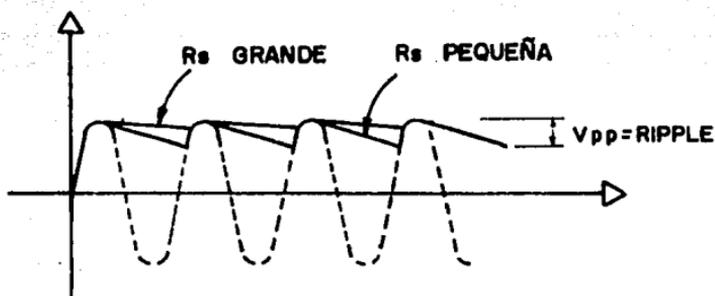


FIGURA 5.13 CIRCUITO RECTIFICADOR DE PICO

Durante el primer cuarto de ciclo,  $V_i(t)$  aumenta hasta su valor máximo positivo, polarizando el diodo directamente, circulando  $I_d$  y almacenando carga en el capacitor  $C$ , hasta que  $V_c(t)$  llega a su valor máximo cuando  $V_i(t)$  es máximo, la forma de onda resultante se muestra a continuación.



**FIGURA 5.14** FORMA DE ONDA DE SALIDA DE UN RECTIFICADOR DE PICO.

Después del primer cuarto de ciclo, la carga  $R_s$  se utiliza para descargar al capacitor. Esto es porque  $R$  y  $C$  están en paralelo y por teoría de circuitos:

$$V_s(t) = V_C E^{-t/R_s C} = V_{im} E^{-t/\tau}$$

donde  $\tau = R_s C$  es la constante de tiempo del circuito.

Si la resistencia  $R_s$  es grande, entonces  $C$  se descarrga lentamente y si  $R_s$  es pequeña la descarga es más rápida. En ambos casos, durante un breve intervalo cerca del pico positivo de  $V_g(t)$ , un pulso de corriente pasa a través del diodo, lo que le devuelve al capacitor la carga perdida en  $R_s$ .

Durante el tiempo que  $D$  no conduce o  $C$  se descarga, se produce a la salida una tensión de corriente continua pulsante conocida como ripple, rizo o ruido. Este ripple, no es senoidal y su valor se mide de pico a pico  $V_{pp}$  (ripple).

Ahora conjuntamos los circuitos de rectificación de onda completa y el de rectificación de pico en la parte resistiva y capacitiva en uno solo y tendremos como resultante el circuito que nos proporciona la forma de onda mostrada adicionalmente, en las siguientes figuras:

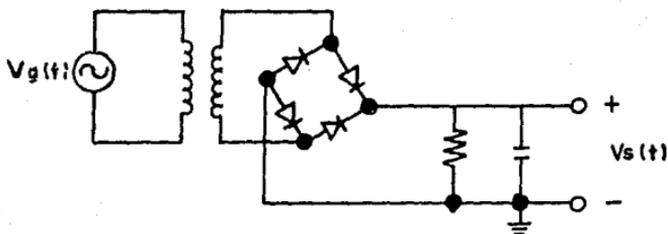


FIGURA 5.15 RECTIFICADOR DE ONDA COMPLETA CON CAPACITOR EN LA SALIDA

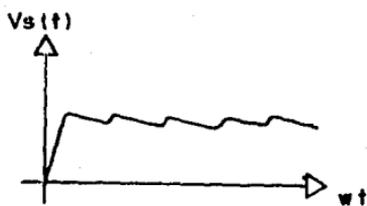


FIGURA 5.16 FORMA DE ONDA RESULTANTE DE UN RECTIFICADOR DE ONDA COMPLETA CON CAPACITOR EN LA SALIDA

Como último paso, acoplamos en nuestra salida un regulador de tensión que nos proporciona 24 volts regulados en su salida.

La forma de onda a la salida del regulador es la siguiente:

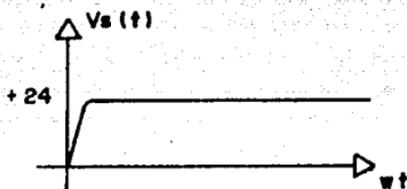


FIGURA 5.17 FORMA DE ONDA DE UN RECTIFICADOR DE +24 VOLTS CON SALIDA REGULADA

### Manejo del lazo de corriente de 4-20 mA

Una señal análogica continua dentro del rango de amplitud representando una condición del "mundo real" que tiene un rango desde 4 hasta 20 miliamperes, posee la característica de inmunidad de la corriente a la inductancia de cables adyacentes en el caso alto y de un cero "vivo" en el caso bajo.

El control de transductores de proceso por lazo de corriente, es fácilmente interfazado por la AIQB, agregando una resistencia shunt a través de la entrada. Debido a que la máxima corriente es de 20 mA y el rango máximo de tensión que maneja el bus de la computadora es de 5 volts, tenemos por ley de ohm:

$$R = 5V/20mA = 250 \text{ ohms}$$

y también por ley de ohm para el caso de la corriente mínima tenemos:

$$V_1 = RI = (250 \text{ ohms})(4mA) = 1 \text{ Volt}$$

Con esto el rango de voltaje que se maneja en la entrada de la tarjeta es desde 1 hasta 5 volts.

Usando la interfase de 4-20 mA, el rango de trabajo del lazo de corriente corresponde a 1638 bits de entrada, con una resolución de aproximadamente 0.06 %.

Tenemos ahora resuelto el problema para la parte del manejo de señales de las dos interfases de carga para la computadora.

En el siguiente esquema se muestra la conexión de la celda de carga con la tarjeta de interfase:

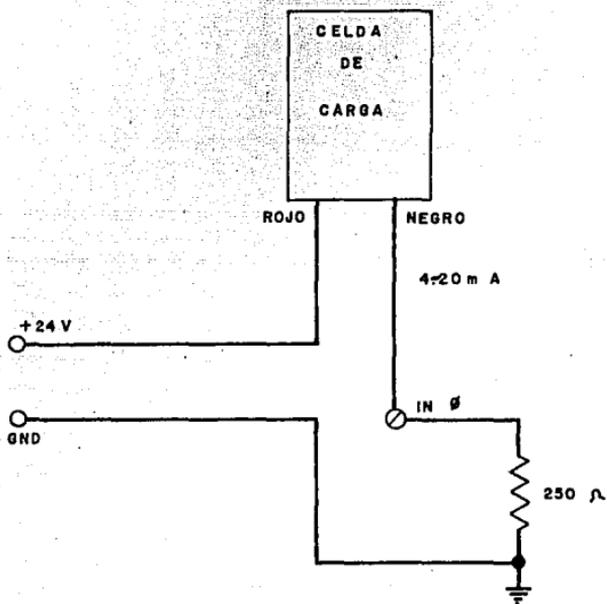


FIGURA No. 5.18

DIAGRAMA DE CONEXION PARA UNA CELDA DE CARGA

### Conversión Analógica/Digital

Como se ha mencionado previamente, el transductor de carga emplea señales analógicas, que deben ser acopladas a un sistema digital.

Una señal digital, es aquella que tiene solamente dos valores discretos de voltaje que son uno o cero.

Por su parte una señal analógica, varía en forma continua de un mínimo a un máximo en corriente o tensión.

Como se desea acoplar las señales analógicas con las digitales es necesario entender como se realiza este tipo de conversión.

Un convertidor analógico/digital proporciona un voltaje analógico desconocido al convertidor y este en su salida nos muestra la salida binaria correspondiente. La salida binaria será proporcional a la entrada analógica, en la siguiente figura se muestra un diagrama de bloques para un convertidor A/D.

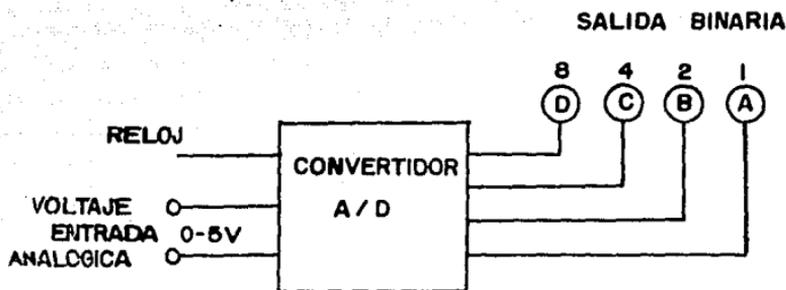


FIGURA 5.19 DIAGRAMA DE BLOQUES DE UN CONVERTIDOR ANALOGICO / DIGITAL.

Este convertidor tiene un voltaje analógico de entrada de 0 a 5 volts. La salida binaria se leera en un rango de 0000 a 1111. El convertidor A/D tiene además una entrada de reloj.

En la siguiente tabla de verdad se detalla la operación del convertidor A/D. El lado de las entradas en la tabla muestra los voltajes analógicos de entrada, mientras que el lado de las salidas muestra las lecturas binarias que corresponden.

ENTRADA ANALOGICA VIN	SALIDA BINARIA			
	8 D	4 C	2 B	1 A
0	0	0	0	0
0 . 333	0	0	0	1
0 . 666	0	0	1	0
1 . 000	0	0	1	1
1 . 333	0	1	0	0
1 . 666	0	1	0	1
2 . 000	0	1	1	0
2 . 333	0	1	1	1
2 . 666	1	0	0	0
3 . 000	1	0	0	1
3 . 333	1	0	1	0
3 . 666	1	0	1	1
4 . 000	1	1	0	0
4 . 333	1	1	0	1
4 . 666	1	1	1	0
5 . 000	1	1	1	1

FIGURA 5.20 TABLA DE VERDAD PARA UN CONVERTIDOR A/D DE 4 BITS.

Una vez explicado el funcionamiento de un convertidor analógico digital, pasamos a una explicación más detallada de la tarjeta de interfase AIO8.

Esta tarjeta mide 5 pulgadas de longitud (12.7 cms.) y cabe en una ranura de longitud media. Todas sus conexiones se hacen a través de un conector normal de 37 pines tipo macho que se proyecta por la parte trasera de la computadora. Esta tarjeta es capaz de realizar las siguientes funciones.

**Conversión Analógica/Digital.**— Hay un convertidor Analógico/Digital de 8 canales, con 12 bits de aproximación sucesiva. La escala de entrada completa de cada canal es de  $\pm 5$  Volts, con una resolución de 2.44 milivolts. Cada entrada es sencilla con una tierra común y puede soportar una sobrecarga continua de  $\pm 30$  volts y breves transitorios de varios cientos de volts. Todas las entradas están abiertas cuando la energía de la computadora no está siendo suministrada. El tiempo típico de conversión A/D es 25 microsegundos, llegando a un máximo de 35 microsegundos por lo que transferencias de hasta 30,000 canales/segundo pueden obtenerse dependiendo de la velocidad del manejador del programa.

**Contador/Temporizador.**- Un Contador/Temporizador programable INTEL 8254 proporciona interrupciones periódicas para el convertidor Analógico/Digital. Hay aparte 3 contadores de bajada de 16 bits en el 8254. Uno de ellos se conecta a un submúltiplo del reloj del sistema y todas las funciones de Entrada/Salida de las dos restantes son de acceso externo. El 8254 maneja frecuencias de hasta 2 MHz.

**Voltaje de referencia.**- Un voltaje de referencia de salida de precisión de  $\pm 10$  volts ( $\pm 0.1$  volt) se deriva desde el convertidor. Esta salida puede aceptar o entregar 2 miliamperes.

**Interrupción.**- Se dispone de una entrada externa de interrupción, la cual puede seleccionar cualquiera de los niveles de interrupción (2-7) y permite programar rutinas de interrupción para adquisición de datos o manejo de control de interrupción. Se incluyen registros de control y estados, con lo cual se simplifica el manejo de las interrupciones; la entrada de interrupción puede conectarse externamente al Contador/Temporizador o a cualquier otra fuente de disparo.

**Energía.**- El bus de energía de la computadora suministra rangos de energía de +5, +12, y -12 volts, que son accesibles para adicionar interfases diseñadas, circuitos de acondicionamiento de señal de entrada (como Amplificadores Operacionales en el caso de este proyecto), etc.

Se cuenta además con un disco que contiene una serie de programas de utilería para facilitar el manejo de la tarjeta. Estos programas se enlistan a continuación, aunque en nuestro caso no serán empleados.

*AIOBDVR.BIN.* - Es un manejador de Entradas/Salidas escrito en lenguaje de máquina para controlar las funciones A/D de tiempo y las de E/S digitales. Se accesa con la instrucción CALL desde BASIC.

El manejador de E/S puede seleccionar canales de multiplexor, poner límites de actualización, ejecutar software de comandos de conversión A/D, manejo de interrupción de conversiones y actualización, puesta y lectura del Temporizador/Contador y mediciones de frecuencia y periodo.

*INSTALL.BAS.* - Este programa ayuda a poner el interruptor de dirección de base, verifica la dirección contra una tabla de direcciones reservadas y previene contra posibles conflictos. El programa despliega una gráfica de la colocación de los interruptores del DIP switch de acuerdo a valores de dirección definidos en el programa.

*POLYNOM.BAS.* - Esta es una rutina de linealización de datos de acuerdo al algoritmo de un polinomio definido en el programa.

*LINPLT.BAS.* - Es un programa de graficación de datos lineales. Las gráficas pueden verse en la pantalla. Las gráficas pueden salvarse en disco o imprimirse oprimiendo SHIFT FRT SCR en la computadora.

*LUADSCRN.BAS.* - Este programa despliega las gráficas salvadas previamente desde *LINPLT.BAS.*

*MAKEDATA.BAS.* - Esta rutina permite construir manualmente un archivo de datos que puede graficarse con *LINPLT.BAS.*

*MAKEGRAF.BAS.* - Esta rutina permite crear una gráfica lineal X, Y en la pantalla con los ejes X e Y marcados.

*MAKPLTCF.BAS.* - Permite crear o leer un archivo de gráfica de control y ejecutarla.

*MEASFREQ.BAS.* - Este programa muestra como operar la tarjeta como un contador de frecuencia.

*MODE18.BAS.* - Este programa muestra la operación de la AIOB en modo 18, el cual corresponde a un arreglo de conversión.

*MODE5.BAS.* - Muestra la operación en modo 5, arreglo de conversión.

*PLOT18.BAS.* - Es un programa de demostración de gráficas para el modo 18.

*DISKLOG8.BAS.* - Es un ejemplo de programación de carga de datos. Los datos se almacenan en disco en intervalos de tiempo seleccionados y pueden imprimirse o verse en pantalla.

*EXINT.BAS.* - Demostración del modo 8 en interrupción.

*XFER.BAS.* - Es una utilería para transferencia de datos. Mueve datos canal por canal desde archivos *DISKLOG.BAS* a paquetes de gráficas comunes en formato de archivo de datos.

El sistema de adquisición de datos se compone por los siguientes elementos:

*Computadora personal compatible con IBM.*

*Teclado.*

*Monitor.*

*Impresora.*

*Tarjeta de interfase AIOB.*

*Software TURBOPASCAL.*

*Celda de carga (corriente y voltaje).*

Primeramente hacemos las conexiones de los equipos de Entrada/Salida hacia la computadora y por otro lado conectamos la celda de carga que corresponda a la entrada analógica deseada ayudandonos con la tarjeta de terminación con que cuenta la tarjeta de Interfase, como se muestra en la figura 5.21.

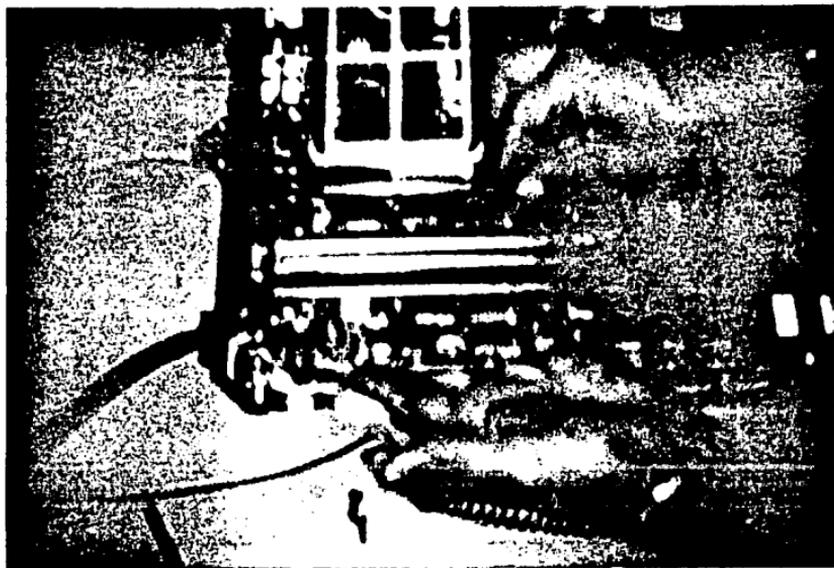


FIGURA . 5.21

*Insertamos en algun conector de expansión de la computadora que este disponible la tarjeta de Interfase (preferentemente el más alejado de la fuente de energía), como se muestra en la siguiente figura.*



FIGURA. 5.22

Despues hacemos la declaración de las direcciones que van a ser utilizadas por la computadora por medio del programa INSTALL.BAS que viene incluido en el manejador de la tarjeta, con la dirección H300 (decimal 768) de fábrica.

La instalación del hardware de la tarjeta requiere de 8 localidades de dirección consecutivas en el espacio de dirección de E/S. La dirección base de la AIOB esta restringida por las siguientes condiciones.

1) La dirección base debe estar dentro del rango de 200 a 3FF en hexadecimal.

2) La dirección base (y las siguientes 7 direcciones consecutivas) no deben entrar en conflicto con cualquier dirección reservada de E/S por IBM.

3) La dirección base (y las 7 siguientes) no deben entrar en conflicto con cualquier dirección asignada a otro dispositivo activo.

4) La dirección base caera siempre automáticamente en un limite de 8 bytes. Esto significa que la dirección base terminara en un 0 o en un 8 hexadecimal.

La rutina de BASIC llamada INSTALL.BAS proporciona un despliegue visual de la correcta colocación para el DIP-switch de la dirección base.

En este programa se incluye una ayuda por pasos.

Al hacer la declaración de dirección, tenemos preparado el sistema para la adquisición de datos con el manejador del sistema, pero como se ha explicado anteriormente el programa se va a elaborar en PASCAL.

Se elabora el programa y se pasa a la parte de pruebas que constituye el siguiente capítulo.

El programa fuente se muestra en el apéndice A de este trabajo.

## **CAPITULO VI**

### **PRUEBAS**

## PRUEBAS

En este capítulo se mostraran los resultados que fuerón obtenidos despues de elaborar el programa de desarrollo de menus y adquisición de datos.

Para hacer el programa de fácil acceso se le asignó una ejecución directa como comando desde el sistema operativo, esto significa que desde la entrada del sistema se puede teclear directamente su nombre y la corrida del programa es automática.

Se le dio el nombre de SIXS.BAT, por lo que con solo anotar desde el teclado el nombre SIXS el programa iniciara su proceso.

Despues de anotar la instrucción anterior, se nos mostrara la siguiente pantalla:

**SISTEMA DE DIAGNOSTICO**

**OPRIMA CUALQUIER TECLA PARA CONTINUAR**

**ALL RIGHTS RESERVED**

Como vemos, en esta pantalla solo se nos muestra el objetivo del programa y se nos indica que oprimamos una tecla para continuar con su ejecución.

El siguiente menú que se despliega despues de oprimir una tecla, nos despliega la siguiente información:

ESCOGE UNA OPCION: < >

- 1) ALTA DE UN NUEVO PACIENTE
- 2) VERIFICACION DE INFORMACION
- 3) CONSULTAS POSTERIORES A LA PRIMERA
- 4) RESPALDO DE INFORMACION A DISKETTE
- 5) ELIMINACION SELECTIVA DE INFORMACION
- 6) EDICION Y ACTUALIZACION DE INFORMACION
- 7) BUSQUEDA ALFABETICA DE UN PACIENTE
- 8) IMPRESION DE MANUAL DE USUARIO
- 9) SALIDA DEL SISTEMA

Este menú sera designado como el menú principal en lo sucesivo porque en el se generan todas las opciones del sistema de diagnostico.

Veamos que se puede hacer con cada una de las opciones mostradas por el sistema iniciando con la primera, al anotar el número 1 en la opción que corresponde al alta de pacientes nuevos.

En este modulo se nos muestra el siguiente formato, el cual debe irse llenando de acuerdo a los datos de los pacientes.

#### SISTEMA DE DIAGNOSTICO

##### MODULO DE ALTAS

NUM. DE CONTROL DEL PACIENTE: < 1 >

NOMBRE DEL PACIENTE: < >

NUMERO DEL IMSS: < > EDAD: < >

PESO: < > CONSULTA NUMERO: < >

DOMICILIO: < >

ESTAN CORRECTOS LOS DATOS? : < > [S/N]

En los espacios disponibles se anota la información particular del paciente en cuestión y por último se ratifica o se rectifica si la información anotada es correcta.

En caso afirmativo pasamos al siguiente modulo que es el de Toma de Fuerzas, pero en caso contrario si la respuesta es < N > que significa no, se nos despliega en la parte baja de la pantalla un modulo de edición que esta formado como se muestra enseguida.

#### SISTEMA DE DIAGNOSTICO

##### MODULO DE ALTAS

NUM. DE CONTROL DEL PACIENTE: < 1 >

NOMBRE DEL PACIENTE: < PEDRO PEREZ >

NUMERO DEL IMSS : < 84-89-0056 > EDAD: < 26 >

PESO: < 88 > CONSULTA NUMERO: < 1 >

DOMICILIO: < CALLE DE LA CUEVA # 627, SECCION EL DORADO >

CAMPO A EDITAR? : < >

- 1) NOMBRE
- 2) NUMERO DEL IMSS
- 3) EDAD
- 4) PESO
- 5) DOMICILIO
- 6) FIN DE EDICION

Después de anotar las correcciones deseadas terminamos con la edición de datos anotando el número 6 que sería el equivalente de una afirmación antes de entrar al módulo de edición, por lo que pasamos de esta manera al módulo de Toma de Fuerzas.

Es importante notar que los números de consulta y de control del paciente son generados automáticamente por el programa, con lo que se evitan confusiones y duplicidad en la formación de la base de datos.

Los campos de edad y peso solamente aceptan caracteres numéricos y todos los demás alfanuméricos.

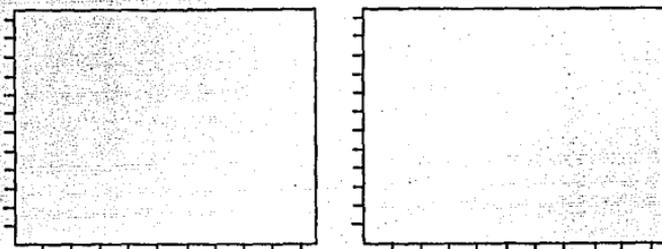
Al pasar al módulo de Toma de Fuerzas encontramos la siguiente información en la pantalla:

OPRIMA LA TECLA CORRESPONDIENTE A  
LA OPERACION QUE DESEA EFECTUAR

TECLA	MUSCULO
<F1>	TRICEPS C6-8
<F2>	DELTOIDE C5,6
<F3>	BICEPS C5,6
<F4>	ILIOPSOAS L1,3
<F5>	EXTENSOR LA,5,S1
<F6>	FLEXOR L5,S1,2

<End> FINALIZA PRUEBAS.

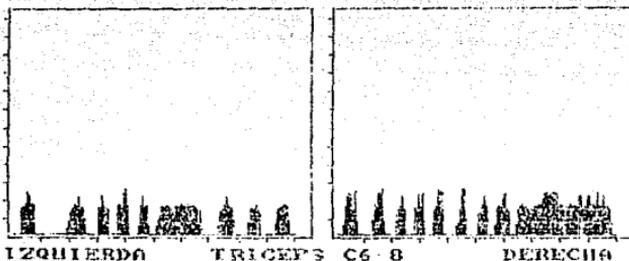
Oprimos la tecla de función de la prueba que se desee por ejemplo <F1> encontramos ahora en pantalla la siguiente figura:



**OPRIMA UNA TECLA PARA INICIAR ( IZQ. )**

En esta pantalla se tienen 2 ventanas que se encuentran vacías, al oprimir una tecla se escuchara un sonido que indica el inicio de la prueba y en ese momento el paciente deba ejercer fuerza en el dispositivo correspondiente, la fuerza ejercida hacia la celda de carga es adquirida en tiempo real por la computadora y si se desea puede ser gráfícada con la tecla <F1> una vez terminada la prueba para ambos lados del cuerpo.

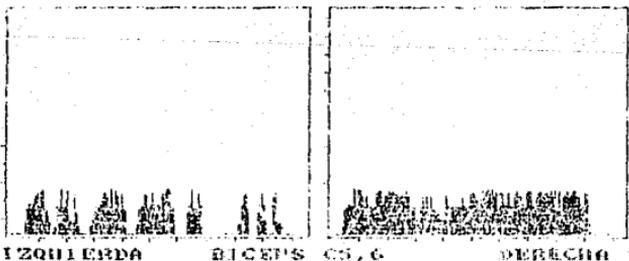
A continuación se muestra un reporte de resultados gráfícos de todas las pruebas tomadas a un paciente.



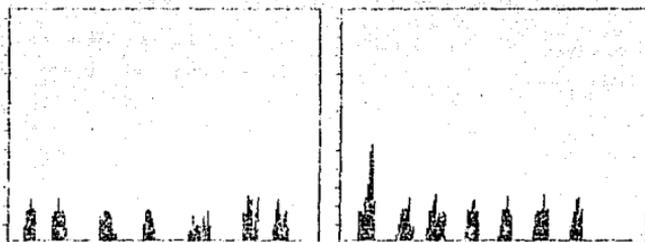
0.450 RESULTADOS DE LA PRUEBA      0.452  
 DEFICIENCIA % :      0.540



1.610 RESULTADOS DE LA PRUEBA      2.002  
 DEFICIENCIA % :      9.145



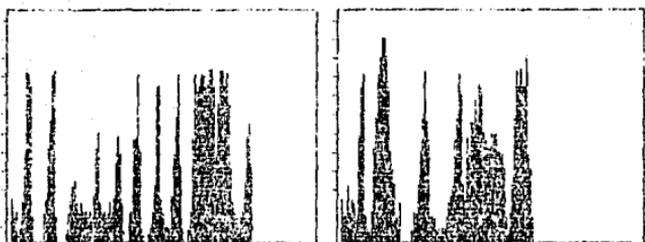
0.417 RESULTADOS DE LA PRUEBA      0.452  
 DEFICIENCIA % :      1.009



IZQUIERDA BICEPS 65,6 DERECHA

0.433 RESULTADOS DE LA PRUEBA 0.926

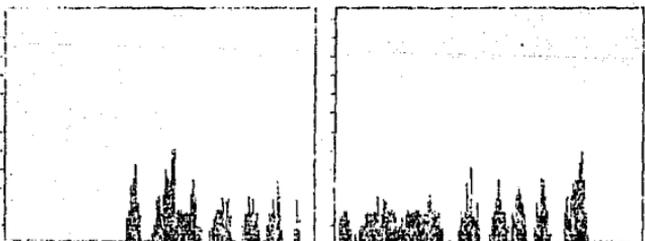
DEFICIENCIA % : 53.276



IZQUIERDA TRILOPSOAS 11,3 DERECHA

1.697 RESULTADOS DE LA PRUEBA 2.019

DEFICIENCIA % : 15.958



IZQUIERDA EXTENSOR L4,5,S1 DERECHA

0.960 RESULTADOS DE LA PRUEBA 0.882

DEFICIENCIA % : 2.167

Una vez terminadas las pruebas se oprime la tecla <End>, esto nos lleva a una pantalla que contiene los mismos datos con que el paciente fue registrado y además en la parte inferior de la pantalla un resumen de los resultados de las pruebas realizadas, se pensó en la posibilidad de repetir alguna o algunas de las pruebas a criterio del terapeuta y se incluyó como pregunta final en esta pantalla. Un ejemplo de esta pantalla se muestra a continuación.

### SISTEMA DE DIAGNOSTICO

#### MODULO DE ALTAS

NUM. DE CONTROL DEL PACIENTE: < 1 >

NOMBRE DEL PACIENTE: <PEDRO PEREZ >

NUMERO DEL IMSS : <84-89-0056 > EDAD: <26>

PESO: < 88 > CONSULTA NUMERO: < 1 >

DOMICILIO: <CALLE DE LA CUEVA # 627, SECCION EL DORADO>

PRUEBAS	TRICEPS	DELTOIDE	BICEPS	PSOAS	EXTENSOR	FLEXOR
IZQUIERDA	14.50	25.20	0.00	0.00	0.00	0.00
DERECHA	14.25	23.15	0.00	0.00	0.00	0.00
DEFICIENCIA%	1.72	8.13	0.00	0.00	0.00	0.00

DESEAS REPETIR ALGUNA(S) PRUEBAS? < > [S/N]

Si se hacen algunas pruebas más, al oprimir <End> en el teclado de la computadora, se regresara a la pantalla mostrada anteriormente y despues se entrara al modulo de comentarios en el que se dispone de 320 espacios para que el terapeuta haga las anotaciones correspondientes a cada consulta.

Después de que se han hecho los comentarios se tecléa <Enter> y la computadora nos da un reporte automático de la consulta en papel, con los datos del paciente, los resultados de las pruebas y los comentarios, después de la impresión de comentarios en papel, todos los resultados de la consulta son guardados en el disco duro en un archivo propio.

Con lo anterior se cubre la parte de altas en el programa de Toma de Fuerzas.

La segunda opción que nos da el menú principal es la de verificación de información, esto se hace tecléando el número 2 al escoger opción.

La pantalla nos muestra el siguiente formato:

### SISTEMA DE DIAGNOSTICO

#### MODULO DE VERIFICACION DE INFORMACION

INTRODUZCA EL NUM. DE CONTROL : < 1> [O=FIN]

Se anota al número de control deseado y si existe, en pantalla se nos despliega la información completa de cada una de las consultas de ese paciente como se muestra a continuación.

SISTEMA DE DIAGNOSTICO  
DESPLIEGUE DE INFORMACION.

NUM. DE CONTROL DEL PACIENTE: < 1 >

NOMBRE DEL PACIENTE: <SALVADOR AVILA BARRAZA >

NUMERO DEL IMSS: < 8984580083 > EDAD: <29>

PESO: < 85 > CONSULTA NUMERO: < 1 >

DOMICILIO: <CALLE DE LA CUEVA #627, SECCION EL DORADO>

FECHA DE LA CONSULTA: <87-12-20>

PRUEBAS	TRICEPS	DELTOIDE	BICEPS	PSOAS	EXTENSOR	FLEXOR
IZQUIERDA	14.50	25.20	0.00	0.00	0.00	0.00
DERECHA	14.25	23.15	0.00	0.00	0.00	0.00
DEFICIENCIA%	1.72	8.13	0.00	0.00	0.00	0.00

DESEAS VER LOS COMENTARIOS ? : < > S/N

Si se desean ver los comentarios, se oprime la letra S que significa si y por el contrario si el terapeuta no desea ver los comentarios anota la letra N.

Cuando los comentarios son desplegados, se emite en la parte baja de la pantalla el mensaje siguiente:

DESEAS VER LOS RESULTADOS ? : < > [S/N]

*Esto se hace con el objeto de poder pasar libremente de resultados a comentarios y de esta manera comprobar los avances del paciente y corroborar las instrucciones que se le han ido proporcionando.*

*Cuando el usuario ya no desea permanecer en la primera consulta, teclea la letra N y en la parte inferior de la pantalla encontrara la siguiente cuestión.*

**CONTINUA CON OTRO REGISTRO ? : < > [S/N]**

*Cuando contestamos afirmativamente encontraremos el despliegue de la segunda consulta y procedemos de la misma manera en la que se hizo anteriormente para emplear la información.*

*En caso de que se conteste afirmativamente y no existan ya más consultas registradas, se nos enviara el siguiente mensaje:*

**FIN DEL ARCHIVO <RETURN>**

*Regresando al menú mostrado en primera instancia en esta opción, se puede dar el caso de que se anote un número de control no registrado y si este es el caso el mensaje que nos dara la computadora sera:*

**NO EXISTE INFORMACION DISPONIBLE**

Tal como se indica en la primera pantalla al oprimir el 0 se regresará al menú principal.

Desde el menú principal y tecleando el número 3 se llega al módulo de consultas posteriores a la primera. En este módulo se registran nuevas consultas de los pacientes, ya que después de que son dados de alta su número de control y su número de consulta permiten que sean fácilmente identificados.

El primer despliegue inmediatamente después de oprimir el número 3 es el siguiente:

SISTEMA DE DIAGNOSTICO  
VISITA PERIODICA DE PACIENTES REGISTRADOS  
NUM. DE CONTROL DEL PACIENTE: < 1> [0=FIN]  
NOMBRE DEL PACIENTE: < SALVADOR AVILA BARRAZA>  
NUMERO DEL IMSS: < 8984580083> EDAD: <29>  
PESO: <85> CONSULTA NUMERO: < 3>  
DOMICILIO: <CALLE DE LA CUEVA # 627, SECCION EL DORADO>  
FECHA DE LA CONSULTA: <88-03-09>

OPRIMA CUALQUIER TECLA PARA PROSEGUIR...

Esta pantalla fue creada con el objeto de que el médico tuviera la facilidad de verificar que el paciente al cual esta consultando, corresponda con el que esta registrado en la base de datos. Puede

verse que el número de consulta es avanzado automáticamente por medio de un contador de eventos.

Después de oprimir cualquiera de las teclas de la computadora, pasamos al módulo de pruebas, el que ya fue explicado cuando se reviso la primera opción en el menú principal. Por lo tanto procedemos de la misma manera como lo hicimos en la opción mencionada para la obtención de resultados y comentarios.

Anotando el número 0 desde la primera pantalla de esta opción regresamos al menú principal, donde anotamos el número 4 para pasar a la opción de Respaldo de información a diskette.

Esta opción se incluyo, debido a que en ocasiones la información almacenada en disco duro puede pasarse también a disco flexible y con ello se tiene un respaldo confiable de información.

Cuando anotamos el número de esta opción la pantalla nos mostrara el siguiente mensaje:

#### **SISTEMA DE DIAGNOSTICO**

**MODULO DE RESPALDO DE INFORMACION.**

**A CONTINUACION PODRA SALVAR EL ARCHIVO  
DE DATOS COMPLETO A UN DISKETTE**

**COLOQUE EL DISKETTE EN LA UNIDAD DE FLOPPYS.**

**OPRIMA UNA TECLA CUANDO ESTE LISTA**

Una vez que el disco flexible se ha introducido en el manejador de discos flexibles, se oprime cualquier tecla y el inicio de la transferencia se hace automáticamente.

Cuando los registros de datos estan siendo transferidos, se nos muestra el siguiente mensaje:

### REGISTROS TRANSFERIDOS

N

En donde N muestra el número de registros transferidos al momento del despliegue.

Una vez terminada la transferencia, el programa nos regresa al menú principal.

Si se intenta iniciar la transferencia sin tener el disco blando en el manejador, el sistema operativo de la computadora nos enviara el siguiente mensaje de error:

**ERROR: Unidad no preparada, leyendo la unidad A**  
**Cancelar, Reintentar, Seguir ?**

Este error seguira apareciendo al oprimir cualquier tecla sin introducir el disco blando, si se eligen las opciones de R Reintentar o de S Seguir. En caso de anotar la opción de C

Cancelar, la computadora nos sacara del programa de diagnóstico y sera necesario repetir el procedimiento de entrada para acceder el sistema.

El siguiente modulo que analizaremos es el de Eliminación selectiva de información, en este módulo, se pueden dar de baja a los pacientes que el medico desee, ya sea por que han finalizado su tratamiento o por que ya no asisten a consulta.

Oprimiendo el número 5 desde el menú principal, obtendremos el siguiente despliegue:

#### SISTEMA DE DIAGNOSTICO

##### MODULO DE SELECCION Y BAJA DE INFORMACION.

INTRODUZCA EL NUMERO DE CONTROL : < > [O=FIN]

Como se ha mencionado previamente, solamente se podran acceder números que hayan sido registrados.

Se anota el número deseado y si existe nos aparece el siguiente mensaje:

**SISTEMA DE DIAGNOSTICO.**

**MODULO DE SELECCION Y BAJA DE INFORMACION**

**NUMERO DE CONTROL DEL PACIENTE: < 1 >**

**NOMBRE DEL PACIENTE: < SALVADOR AVILA BARRAZA >**

**NUMERO DEL IMSS: < 8984580083 > EDAD: < 29 >**

**PESO: < 85 > CONSULTA NUMERO: < 2 >**

**DOMICILIO: < CALLE DE LA CUEVA # 627, SECCION EL DORADO >**

**ESTA ES LA INFORMACION REQUERIDA, HAY 2 CONSULTAS  
CON ESTA INFORMACION, ES LA INFORMACION QUE DESEAS BORRAR?**

**[S/N]: < >**

*En caso de que se teclee N, regresaremos a la pantalla inicial de esta opción, pero si anotamos la letra S, entonces el programa nos enviara un nuevo mensaje que dice:*

**SISTEMA DE DIAGNOSTICO**

**MODULO DE RESPALDO DE INFORMACION.**

**ANTES DE PROSEGUIR INTRODUZCA LA CLAVE PARA BORRAR**

*Se ha introducido una clave (password) que solamente el médico conoce, para evitar que ninguna persona que no este autorizada elimine la información del disco duro. Al solicitar la clave, se dan un máximo de 3 intentos para que esta sea correcta, en caso de que no lo fuera, se regresa al primer despliegue dentro de esta opción.*

*Cuando hay error en la introducción de la clave, se nos despliega el siguiente mensaje:*

**CLAVE ERRONEA, LE QUEDAN 2 INTENTOS**

*Cuando la clave que se anota es correcta, entonces el despliegue es el siguiente:*

**SISTEMA DE DIAGNOSTICO**

**MODULO DE RESPALDO DE INFORMACION.**

**A CONTINUACION PODRA ELIMINAR LOS REGISTROS  
SELECCIONADOS, PARA POSTERIORES CONSULTAS.**

**COLOQUE EL DISKETTE EN LA UNIDAD DE FLOPPYS.**

**OPRIMA UNA TECLA CUANDO ESTE LISTA..**

*Cuando se inserta el disco en la unidad de control y se oprime una tecla, los registros son transferidos y la información desaparece del disco duro.*

*Una vez transferida la información, regresaremos a la primera pantalla que se mostro en esta alternativa. Para poder salir de esta opción, es necesario oprimir el número 0 y <RETURN> desde la primera pantalla.*

La opción del menú principal que esta asignada con el número 7 es la de Edición y actualización de información. Este modulo fue diseñado con el objeto de poder realizar cambios en los datos de los pacientes, ya que pueden existir cambios en su edad, domicilio y peso.

El menú que se presenta en esta opción es el mismo que se mostro en el modulo de altas para la parte de la edición en caso de error al introducir la información, los cambios se hacen de la misma forma en la que se indico en el módulo mencionado.

Al salir de la edición estaremos de regreso en el menú principal y desde ahí anotaremos la opción de Búsqueda alfabética de información tecleando el número 7.

El despliegue inicial sera:

#### SISTEMA DE DIAGNOSTICO

#### MODULO DE BUSQUEDA ALFABETICA DE INFORMACION

INTRODUZCA LA CADENA A BUSCAR [0=FIN]:

NOMBRE DEL PACIENTE: <

>

Se anota en el espacio disponible el nombre a encontrar, por ejemplo: SALVADOR AVILA BARRAZA y aparece la siguiente información.

**SISTEMA DE DIAGNOSTICO.**

**MODULO DE BUSQUEDA ALFABETICA DE INFORMACION**

**NUM. DE CONTROL DEL PACIENTE: < 1>**

**NOMBRE DEL PACIENTE : < SALVADOR AVILA BARRAZA>**

**NUMERO DEL IMSS: < 8984580083> EDAD: <29>**

**PESO: < 85> CONSULTA NUMERO: < 2>**

**DOMICILIO: <CALLE DE LA CUEVA # 627, SECCION EL DORADO>**

*Con este despliegue el médico conoce ya el número de control del paciente y puede pasar a una consulta si así lo desea al anotar el fin de la cadena con el número 0 desde la primera pantalla de esta opción.*

*En el caso de que la información de algún paciente no sea localizada, se nos mostrara el siguiente aviso:*

**NO EXISTEN REGISTROS QUE CUMPLAN CON LA CONDICION....**

*La opción marcada con el número 8 en el menú principal corresponde a la Impresión del manual de usuario, la cual contiene una guía detallada de todos los procedimientos y facilidades con los que cuenta el sistema, ya que se supone que el personal que hará uso del sistema no posee conocimientos dentro del área de computación y es por eso que esta guía representa una gran ayuda para entender la manera de proceder en cada caso.*

*Esta guía se imprime automáticamente en una impresora y debe tenerse cuidado de que esta este conectada y puesta en línea para ejecutar la mencionada impresión.*

*La última opción que presenta el sistema de diagnóstico es la salida del sistema y se logra oprimiendo la tecla marcada con el número 9, esto se hace cuando ya no se desea seguir en el sistema de diagnóstico y por el contrario se desea emplear la computadora para otras tareas.*

*Todo lo anterior muestra la capacidad del sistema de adquisición de datos y las facilidades que se proporcionan a los posibles*

**CAPITULO VI**

**CONCLUSIONES**

## CONCLUSIONES

Los objetivos que se plantearon al iniciar este trabajo fueron plenamente alcanzados, ya que el paquete realizado cubre el total de las necesidades del usuario.

Estas soluciones conjugan la ayuda de un programa que va guiando paso a paso al médico y le permiten tomar datos del usuario, realizar pruebas, emitir comentarios, repetir pruebas, cambiar datos, respaldar información, eliminar pacientes de las bases de datos y buscar información alfabéticamente sin necesidad de saber nada acerca de computadoras y programación.

La aplicación de este tipo de ayudas puede dirigirse hacia ramas de otra índole, ya que por medio de una celda de carga se puede tomar el peso de algún cuerpo u objeto y se puede además ir verificando las variaciones que se tengan de una manera instantánea, lo cual puede ser muy útil en dosificaciones para cualquier producto con varios componentes, o bien se puede medir la fuerza aplicada en un resorte, etc.

Lo importante para cada aplicación, es la elección adecuada de interfases, ya que por medio de ellas mandamos información hacia dispositivos inteligentes que son manejados con microprocesadores.

Además en este caso, de pretender hacer un diseño de hardware para la tarjeta de adquisición de datos de la computadora hubiera llevado un tiempo largo de desarrollo, entorpeciendo de esta manera la obtención del resultado deseado. Por esta razón se busco y se consiguió una tarjeta ya probada, con un tiempo largo de diseño, con dispositivos de producción en serie y de pruebas funcionales, lo que difícilmente podría lograrse de no contar con una infraestructura adecuada.

Por lo tanto considero que este trabajo es realmente una aplicación práctica de conocimientos que fueron adquiridos en la Facultad, sin los que no hubiese sido posible obtener este resultado.

Adicionalmente a lo ya expuesto, el incursionar en áreas de trabajo como las de rehabilitación y detección de problemas de índole físico, los que no tienen una relación estrecha con los de la Ingeniería, nos permiten pensar que los campos de desarrollo para los egresados de nuestra área están limitados solamente por la capacidad de cada Ingeniero.

**APENDICE A**

**PROGRAMA DE APLICACION**

( PROGRAMA PARA LA DETECCION Y TRATAMIENTO DE LESIONES EN LA COLUMNA  
 VERTEBRAL ELABORADO PARA: LUKAS, S.A.)

```

programa trat_medico;
const
  IText61b:integer=6;
  YText61b:integer=6;
  GrafBase:integer=4800;
  Color61b:integer=15;
  rett:integer=1;
  convx:real=0.630063;
  btx:real=7.0;
  by:real=143;
  XMax61b=79;
  XScreenMax61b=639;
  YMax61b=199;
  IVStep61b=2;
  ScreenSize61b=9191;
  HardwareGrafBase=4800;
  FontLoaded:boolean=false;
  MinForeground:integer=0;
  MaxForeground:integer=15;
  MinBackground:integer=0;
  MaxBackground:integer=0;
  AspectFactor=0.44;
  SaveState61b:integer=10;
  ForegroundColor61b:integer=15;

```

```

type
  recscr = record
    posx, posy, long:integer;
    tipo :char;
  end;

  registro = record
    numcont : integer;           (NUMERO DE CONTROL)
    patient : string(40);       (NOMBRE DEL PACIENTE)
    numless : string(12);       (NUMERO DEL IMSS)
    edad : integer;            (EDAD DEL PACIENTE)
    peso : integer;            (PESO DEL PACIENTE)
    consult : integer;         (NUM. DE CONSULTA)
    domicili : string(40);     (DIRECCION)
    fecha : string(8);        (FECHA DE CONSULTA)
    elimin : char;            (ELIM? (S/N))
    result : array(1..12) of real; (RESULTADOS DEL EXAMEN)
    coment : array(1..4) of string(80) (COMENTARIOS)
  end

  (end record registro);
  archivo = file of registro;
  arr = array (1..13) of recscr;
  cad90=string(80);
  cad6 =string(6);
  cad8 =string(8);
  cad=char;

  wrkstring=cad90;

```





```

goxory(21+1)+1)+w,wherey);
write(porc:5:2)
end
(end for)
end
(end if);
if bor=1 then
while not keypressed do
(endif)
end
(end procedure ioptabla);

procedure escpanlst(regesc:registro);
var
porc:real;
i,j:integer;
begin
writeln(lst);
writeln(lst);
writeln(lst);
writeln(lst,' :29,'SISTEMA DE DIAGNOSTICO');writeln(lst);
writeln(lst,' :27,'DESPLIEGUE DE INFORMACION');writeln(lst);
with regesc do
begin
writeln(lst,' :20,'NUMERO DE CONTROL DEL PACIENTE:(' ,numcont:5,')');writeln(lst);
writeln(lst,' :9,'NOMBRE DEL PACIENTE:(' ,pacient:40,')');writeln(lst);
writeln(lst,' :9,'NUMERO DEL IMS:(' ,numass:12,') ' :24,'EDAD:(' ,edad:2,')');writeln(lst);
writeln(lst,' :9,'PESO:(' ,peso:3,') ' :33,'CONSULTA NUMERO:(' ,consult:2,')');writeln(lst);
writeln(lst,' :14,'DOMICILIO:(' ,domicil:40,')');writeln(lst);
end
(enwith);
writeln(lst,' :24,'FECHA DE LA CONSULTA:(' ,regcap.fecha:8,')');writeln(lst);
writeln(lst,' :7,'PRUEBAS: ' :7,'TRICEPS ' :8,'DELTOIDE ' :8,'BICEPS ' :8,'
PSOAS ' :8,'EXTENSOR ' :8,'FLEXOR ');writeln(lst);
write(lst,' :8,'IZQUIERDA ');
for i:=1 to 6 do
writeln(lst,regcap.result(2*(i-1)+1):5:2,' :5)
(end for);writeln(lst);
write(lst,' :8,'DERECHA ');
for i:=1 to 6 do
writeln(lst,regcap.result(2*(i-1)+2):5:2,' :5)
(end for);writeln(lst);
write(lst,' :8,'DEFICIENCIA 2 ');
with regcap do
begin
for i:=1 to 6 do
begin
if
result(2*(i-1)+1)<0) and (result(2*(i-1)+2)<0) then
begin
if result(2*(i-1)+1)/result(2*(i-1)+2) then
porc:=abs(result(2*(i-1)+1)-result(2*(i-1)+2))/result(2*(i-1)+1)
else
porc:=abs(result(2*(i-1)+1)-result(2*(i-1)+2))/result(2*(i-1)+2)
endif);
porc:=porc*100

```

```

end
else
  if (result(2*(i-1)+1)<0) or (result(2*(i-1)+2)<0) then
    porc:=100
  else
    porc:=0
  (endif)
(endif);
write(lst,porc:5i2,' ');
end
(end for)
end
(end if);writeln(lst);writeln(lst);writeln(lst)
end
end procedure escpanlst);

```

```

procedure suena(frec:integer);
begin
  sound(frec);
  delay(500);
  nosound
end
{procedure suena};

```

```

procedure adogra(musc:cad80;factor:real;lara:real;var maxval,maxval2:real);
const nummus=2223;
var
  esc,lect:byte;
  aux,xant,yant,i,x1,y1,dio:integer;
  porc,x1,x2,y1,y2:real;
  xval,yval:array [1..nummus] of real;
  lecho:char;
  inverse:boolean;
  mode:byte;

```

```

  procedure grafica(bx,by:real);
  var caract:char;
  begin
    suena(500);
    for i:=1 to nummus do
      begin
        xval[i]:=0;
        yval[i]:=0
      end
    (end for);
    xant:=round(bx);
    yant:=round(by);
    i:=1;
    while i<nummus do
      begin
        port($302):=0;
        port($301):=0;
        xl:=port($300);

```

```

xh:=port($301);
yval[i]:=((port($301)*16+port($300)/16)*10/4096)-51-lara;
xval[i]:=i;
delay(10);
if (yval[i]<convy+by) then
  draw(xant,yant,round(xval[i]&convx+bx),round(by),1)
else
  draw(xant,yant,round(xval[i]&convx+bx),round(yval[i]&convy+by),1)
endif);
xant:=round(xval[i]&convx+bx);
yant:=round(by) (round(yval[i]&convy+by));
if keypressed then
  begin
    read(kbd,carac);
    if carac=chr(27) then
      i:=nummes
    endif)
  end
endif);
i:=i+1;
end
end for);
suena(100);
end (procedure grafica);

```

```

begin
  lara:=2.50;
  graphcolormode;
  palette(3);
  draw(5,5,155,5,2); draw(165,5,315,5,2);
  draw(5,5,5,145,2); draw(165,5,165,145,2);
  draw(155,5,155,145,2); draw(315,5,315,145,2);
  draw(5,145,155,145,2); draw(165,145,315,145,2);
  for i:=1 to 10 do
    begin
      aux:=round(i*222&convx+bx);
      draw(aux,145,aux,147,2)
    end
  end);
  for i:=1 to 10 do
    begin
      aux:=round(i*222&convx+bx+169);
      draw(aux,145,aux,147,2)
    end
  end);
  gotoxy(1,20);writel' IZQUIERDA ',musc:20,' DERECHA';
  gotoxy(1,25);writel' OPRIMA UNA TECLA PARA EMPEZAR (IZD.)';
  while not keypressed do;
    read(kbd,lecho);
    gotoxy(1,25);writel' ':40);
    grafica(bx,by);
    maxvall:=0;
    for i:=1 to nummes do
      if yval[i]>maxvall then maxvall:=yval[i]

```

```

(end for);
gotoxy(1,25);write(' OPRIMA UNA TECLA PARA EMPEZAR (DER.)');
while not keypressed do;
read(kbd,lecho1);
gotoxy(1,25);write(' ':40);
grafica(bx+60,by);
maxval2:=0;
for i:=1 to nummes do
if yval(i)*maxval2 then maxval2:=yval(i)
(end for);
gotoxy(1,25);write(' ':40);
gotoxy(1,23);write(maxval1:6:3, ' RESULTADOS DE LA PRUEBA ',maxval2:6:3);
if (maxval1<0) and (maxval2<0) then
begin
if maxval1>maxval2 then
porc:=abs(maxval1-maxval2)/maxval1
else
porc:=abs(maxval1-maxval2)/maxval2
(endif);
porc:=porc*100
end
else
if (maxval1<0) or (maxval2<0) then
porc:=100
else
porc:=0
(endif)
(endif);
gotoxy(1,25);write(' DEFICIENCIA: ',porc:6:3);
inverse:=false; mode:=1;
Hardcopy(inverse,mode);
while not keypressed do;
read(ybd,lecho1);
suena(300);
textode;
textcolor(2)
end
(end procedure adogra);

```

```

procedura inicializa(var arreglo1arr);
begin
with arreglo1 do
begin
posx:=53;posy:=5;long:=5;tipos:='N'
end
(end with);
with arreglo2 do
begin
posx:=31;posy:=7;long:=40;tipos:='C'
end
(end with);
with arreglo3 do

```

```

begin
  posx:=27;posy:=9;long:=12;tipo:='C'
end
(end with);
with arreglo(4) do
  begin
    posx:=69;posy:=9;long:=2;tipo:='N'
  end
(end with);
with arreglo(5) do
  begin
    posx:=16;posy:=11;long:=3;tipo:='N'
  end
(end with);
with arreglo(6) do
  begin
    posx:=69;posy:=11;long:=2;tipo:='N'
  end
(end with);
with arreglo(7) do
  begin
    posx:=28;posy:=13;long:=40;tipo:='C'
  end
(end with);
with arreglo(8) do
  begin
    posx:=54;posy:=20;long:=1;tipo:='C'
  end
(end with);
with arreglo(9) do
  begin
    posx:=54;posy:=24;long:=1;tipo:='C'
  end
(end with);
with arreglo(10) do
  begin
    posx:=1;posy:=20;long:=80;tipo:='C'
  end
(end with);
with arreglo(11) do
  begin
    posx:=1;posy:=21;long:=80;tipo:='C'
  end
(end with);
with arreglo(12) do
  begin
    posx:=1;posy:=22;long:=80;tipo:='C'
  end
(end with);
with arreglo(13) do
  begin
    posx:=1;posy:=23;long:=80;tipo:='C'
  end
(end with);
end

```

```
(procedure inicializa);
```

```
procedure inicreg;
```

```
begin
```

```
with reqcap do
```

```
begin
```

```
numcont:=0;
```

```
pacient:=
```

```
numless:=
```

```
edad:=0;
```

```
peso:=0;
```

```
consult:=0;
```

```
domicil:=
```

```
fecha:=
```

```
end
```

```
(end with)
```

```
end
```

```
(procedure inicreg);
```

```
procedure leeaa(var cadena:cad80;var posic:integer;arreglo:array[1..maxv] of integer);
```

```
var
```

```
tipoi,aux1,aux2:cad1;
```

```
longi,diri,dirx,diry,cont:integer;
```

```
begin
```

```
cont:=1;
```

```
aux1:= '';
```

```
cadena:= '';
```

```
with arreglo[posic] do
```

```
begin
```

```
gotox(y(posx,posy);
```

```
writel('longi);
```

```
gotox(y(posx,posy);
```

```
longi:=long;
```

```
diri:=wherex;
```

```
tipoi:=tipo
```

```
end
```

```
(end with);
```

```
while (cont<=longi) and (aux1<>chr(13)) do
```

```
begin
```

```
read(kbd,aux1); aux1:=upcase(aux1);
```

```
if aux1=chr(8) then
```

```
begin
```

```
if cont<1 then
```

```
begin
```

```
cont:=cont-1;
```

```
cadena(cont):= '';
```

```
gotox(y(wherex-1,wherex); writel(' ');
```

```
gotox(y(wherex-1,wherex)
```

```
end
```

```
else
```

```
if diri<>wherex then
```

```
begin
```

```
cadena(cont):= ' ';
```



```

        aux1:=chr(13)
    end;
    B0: begin
        if (posic)=maxva then
            posic:=posic-1
        (endif);
        aux1:=chr(13)
        end;
    else ;
    end
    (end case)
    end
    (endif)
    (endif)
    (endif)
    end
    (end while);
    cadena(0):=chr(cont-1)
end
(procedure leeaa);

```

```

procedure esccon;
var
    i:integer;
begin
    gotoxy(1,16);writeln(' :00');
    gotoxy(1,17);writeln(' :00');
    gotoxy(1,18);writeln(' :00');
    gotoxy(1,19);writeln(' :00');
    gotoxy(1,20);writeln(' :00');
    gotoxy(1,21);writeln(' :00');
    gotoxy(1,22);writeln(' :00');
    gotoxy(1,23);writeln(' :00');
    gotoxy(1,24);writeln(' :00');
    gotoxy(22,19);writeln('COMENTARIOS CON RESPECTO A LA CONSULTA');
    gotoxy(1,21);
    for i:=1 to 4 do
        writeln(iregap,coment(i))
    (end for)
    end
    (procedure esccon);

```

```

procedure leecon;
var
    lee, leeant:integer;
begin
    gotoxy(1,17);writeln(' :00');
    gotoxy(1,18);writeln(' :00');
    gotoxy(1,19);writeln(' :00');
    gotoxy(1,20);writeln(' :00');
    gotoxy(1,21);writeln(' :00');
    gotoxy(1,22);writeln(' :00');
    gotoxy(1,23);writeln(' :00');

```

```

gotoxy(1,24);write(' ':80);
gotoxy(22,19);write('COMENTARIOS CON RESPECTO A LA CONSULTA');
for lee:=1 to 4 do
  regcap.coment(lee):='
lee:=10;
while (lee<>13) do
  begin
    repeat
      leecan(cadena,lee,arreglo,13);
      if cadena='' then
        lee:=14
      else
        case lee of
          10: regcap.coment(1):=cadena;
          11: regcap.coment(2):=cadena;
          12: regcap.coment(3):=cadena;
          13: regcap.coment(4):=cadena;
        end (end case)
      (end if);
      lee:=lee+1;
    until (lee>13);
  end
(end while)
end
(procedure leecan);

```

```

function entero(cadena:cad80):integer;
var
  i,j,k:integer;
  num,numaux :integer;
begin
  numaux:=0;
  for i:=1 to length(cadena) do
    begin
      num:=ord(cadena(i))-48;
      for j:=1 to length(cadena)-1 do
        num:=num*10
      (end for);
      numaux:=numaux+num
    end
  (end for);
  entero:=numaux;
end
(end function entero);

```

```

function existe(filename:cad80):boolean;
var
  fil:archivo;
begin
  assign(fil,filename);
  {$I-}
  reset(fil);

```

```

(i+);
exister=(i=resultado);
close(fil);
end
(end function existe);

procedure pruebas;
var
  car1,car2:char;
  fin,si:boolean;
  factor,tara,maxval1,maxval2:real;
  mode,i:integer;
  muscnam: array [1..6] of cad80;
  inverse:boolean;
  contpru:integer;
begin
  contpru:=0;
  muscnam[1]:=' TRICEPS C6-8 ' ;
  muscnam[2]:=' DELTOIDE C5,6 ' ;
  muscnam[3]:=' BICEPS C5,6 ' ;
  muscnam[4]:=' ILIOPSOAS L1,3 ' ;
  muscnam[5]:=' EXTENSOR L4,5,S1 ' ;
  muscnam[6]:=' FLEXOR L5,S1,2 ' ;
  fin:=false;
  si:=false;
  while not fin do
    begin
      fin:=false;
      clrscr;
      gotoxy(24,1);writel('OPRINA LA TECLA CORRESPONDIENTE A');
      gotoxy(25,3);writel('LA OPERACION QUE DESEA EFECTUAR');
      gotoxy(29,5);writel('TECLA MUSCULO');
      gotoxy(29,7);textcolor(14);writel('F1');
      textcolor(2);writel(' TRICEPS C6-8 ');
      gotoxy(29,9);textcolor(14);writel('F2');
      textcolor(2);writel(' DELTOIDE C5,6 ');
      gotoxy(29,11);textcolor(14);writel('F3');
      textcolor(2);writel(' BICEPS C5,6 ');
      gotoxy(29,13);textcolor(14);writel('F4');
      textcolor(2);writel(' ILIOPSOAS L1,3 ');
      gotoxy(29,15);textcolor(14);writel('F5');
      textcolor(2);writel(' EXTENSOR L4,5,S1 ');
      gotoxy(29,17);textcolor(14);writel('F6');
      textcolor(2);writel(' FLEXOR L5,S1,2 ');
      gotoxy(29,19);textcolor(14);writel('END');
      textcolor(2);writel(' FINALIZA PRUEBAS');
      repeat
        read(kbd,car1)
      until car1=chr(27);
      read(kbd,car2);
      case ord(car2) of
        59,60,61,62,63,64: begin
          i:=ord(car2)-59+1;
          factor:=1.0;
          si:=true;

```

```

end;

79: fin:=true;
else ;
end (end case);
if (not fin) and (si) then
begin
contpru:=contpru+1;
adogra:=uscnaafi div 2+1, factor, tara, maxval1, maxval2;
si:=false;
regcap.result[i]:=maxval1;
regcap.result[i+1]:=maxval2;
if (contpru mod 3)=0 then
writeIn(1st,chr(12))
(endif)
end
(endif)
end
(endif)
(while)
i:=ptabla(1);
textcolor(14);
gotoxy(17,24);write('DESEAS REPETIR ALGUNA DE LAS PRUEBAS? (S/N): < >');
gotoxy(63,24);
repeat
read(1bd,car1);car1:=upcase(car1);
if (car1<'S' ) and (car1<'N') then
suena(50)
(endif)
until (car1='S') or (car1='N');
gotoxy(17,24);write(' :50');
if car1='S' then
pruebas
else
begin
escpan1st(regcap);
leecom;
writeIn(1st);
writeIn(1st);
writeIn(1st);
writeIn(1st,' COMENTARIOS ACERCA DE LA CONSULTA');
for i:=1 to 4 do
writeIn(1st,regcap.coment[i]);
(endif)
end
(endif)
end
(end procedure pruebas);

```

```

procedure edit(caa:caapo:integer);
var
caapoaux:integer;
begin
caapoaux:=caapo;
repeat
leecaa(cadena,caapo,arreglo,caapo);

```

```

if cadena="" then
  campo:=campoaux
endif)
until cadena="" ;
case campo of
  2: regcap.paciente:=cadena;
  3: regcap.nuamiss:=cadena;
  4: regcap.edad:=entero(cadena);
  5: regcap.peso:=entero(cadena);
  7: regcap.domicil:=cadena
end
endif)
end (procedure editcaa);

```

```

procedure editreg;
var
  car:char;
  i:integer;
begin
  for i:=14 to 24 do
    begin
      gotoxy(1,i);
      writel' :B0)
    end
  endif);
  gotoxy(31,16);writel' CAMPO A EDITAR?(< >');
  textcolor(14);gotoxy(34,19);writel' 1)';textcolor(2);writel' NOMBRE';
  textcolor(14);gotoxy(34,20);writel' 2)';textcolor(2);writel' MUN. ISES';
  textcolor(14);gotoxy(34,21);writel' 3)';textcolor(2);writel' EDAD';
  textcolor(14);gotoxy(34,22);writel' 4)';textcolor(2);writel' PESO';
  textcolor(14);gotoxy(34,23);writel' 5)';textcolor(2);writel' DOMICILIO';
  textcolor(14);gotoxy(34,24);writel' 6)';textcolor(2);writel' FIN EDICION';
  repeat
    gotoxy(48,16);writel' ';gotoxy(48,16);
    car:= ' ';
    readkbd(car);car:=upcase(car);
    if (car<'1') or (car>'6') then
      suena(500)
    else
      begin
        writel(car);
        case car of
          '1': editcaa(2);
          '2': editcaa(3);
          '3': editcaa(4);
          '4': editcaa(5);
          '5': editcaa(7)
        end (case)
      end
    endif)
  until (car='6');
  for i:=16 to 24 do
    begin gotoxy(1,i);writel' :B0) end
  endif)

```

```

end
{procedure editreg;

procedure leeapan(posini:integer;consini:integer);
begin
regcap.numcont:=numero;
posic:=posini;
while (posic<=0) do
begin
posant:=posic;
if posic<>6 then
begin
if posant=2 then
repeat
leecaa(cadena,posic,arreglo,2);
if cadena="" then
posic:=posant
until cadena()''
else
leecaa(cadena,posic,arreglo,7)
endif;
if (cadena='') and (posant=2) then
posic:=posant
else
case posant of
2: regcap.pacient:=cadena;
3: regcap.numass:=cadena;
4: regcap.edad:=entero(cadena);
5: regcap.peso:=entero(cadena);
7: regcap.domicil:=cadena;
6: if cadena='M' then
editreg;
else
if cadena('S' then
posic:=posic-1
endif)
endif;
end (end case)
endif)
end
else
begin
regcap.consult:=consini;
gotory(69,11);
write(regcap.consult:2)
end
endif;
posic:=posic+1
end
end while)
end
{procedure leeapan;

```

```

procedure altas;
var
  ok:char; i:integer;
begin
  clrscr;
  incrcr;
  gotoxy(20,1);write('SISTEMA DE DIAGNOSTICO');
  gotoxy(33,2);write('MODULO DE ALTAS');
  gotoxy(25,5);write('NUM. DE CONTROL DEL PACIENTE:(  )');
  gotoxy(10,7);write('NOMBRE DEL PACIENTE:(',':40,')');
  gotoxy(10,9);write('NUMERO DEL INSS:(  )');
  gotoxy(15,9);write('EDAD:(  )');
  gotoxy(10,11);write('PESO:(  )');
  gotoxy(22,11);write('CONSULTA NUMERO:(  )');
  gotoxy(15,13);write('DOMICILIO:(',':40,')');
  gotoxy(25,20);write('ESTAN CORRECTOS LOS DATOS? (Y/N)');
  assign(archdat, archdat);
  if existe(archdat) then (se verifica que exista el archivo de datos)
  begin
    numero:=0;
    reset(archdat);
    (numero:=filesize(archdat));
    regcap.numcont:=0;
    regajr.numcont:=0;
    while not eof(archdat) do
      begin
        read(archdat,regcap);
        if regcap.numcont<numero then
          numero:=regcap.numcont;
        (end if)
      end
    (end while);
    numero:=numero+1; (se genera el numero del nuevo paciente)
  end
  else (si el archivo de datos no existe, entonces es el primer paciente)
  begin
    numero:=1;
    rewrite(archdat);
  end
  (end if);
  gotoxy(25,5);write(numero);
  incrcr;
  leer:=0;
  regcap.fecha:=date;
  for i:=1 to 12 do
    regcap.result[i]:=0;
  (end for);
  prcrtas;
  seek(archdat, filesize(archdat));
  write(archdat,regcap);
  flush(archdat);
  close(archdat);
  end
(end procedure altas);

```

```

procedure verifica;
var contreg:integer;
begin
  clrscr;
  gotoxy(30,1);write('SISTEMA DE DIAGNOSTICO');
  gotoxy(21,3);write('MODULO DE VERIFICACION DE INFORMACION. ');
  gotoxy(21,5);write('INTRODUZCA EL NUM. DE CONTROL :( > ) (0-FINI)');
  assign(archdat,'archdat');
  repeat
    leecan(cadena,posit,arreglo,7);
    posit:=1
  until cadena('<<');
  numero:=enterocadena;
  if (existel'archdat') and (numero<0) then {se verifica que exista el archivo de datos}
  begin
    posit:=1;
    reset(archdat);
    regcap.numcont:=0;
    regaux.numcont:=0;
    continua:='S';
    contreg:=0;
    while (not eof(archdat)) and (continua='S') do
      begin
        readf(archdat,regcap);
        if regcap.numcont=numero then {si hay registro del paciente}
        begin
          contreg:=contreg+1;
          escan(regcap);
          gotoxy(25,15);write('FECHA DE LA CONSULTA:( > )');
          gotoxy(47,15);write(regcap.fecha:8);
          repeat
            inptabla(2);
            posit:=9;
            textcolor(14);
            gotoxy(25,24);write('DESEAS VER LOS COMENTARIOS?(< >)');
            textcolor(2);
          repeat
            leecan(cadena,posit,arreglo,7);
            posit:=9
          until (cadena='S') or (cadena='N');
          if cadena='S' then
            begin
              gotoxy(25,24);write(' :35);
              escan;
              textcolor(14);
              gotoxy(25,24);write('DESEAS VER LOS RESULTADOS? (< >)');
              textcolor(2);
              posit:=9;
              repeat
                leecan(cadena,posit,arreglo,7);
                posit:=9
              until (cadena='S') or (cadena='N')
            end
          end
        end
      end
    end
  end

```

```

endif)
until cadena='N';
if not eof(archdat) then
begin
posic:=9;
gotoxy(25,24);write('CONTINUA CON OTRO REGISTRO?:( )');
repeat
leecaa(cadena,posic,arreglo,7);
posic:=9;
until (cadena='S') or (cadena='N');
if cadena()='S' then
continua:=cadena;
endif)
end
else
begin
textcolor(12+16);suena(500);
gotoxy(25,24);write('ES EL ULTIMO REGISTRO (RETURNO) ');
while not keypressed do;
continua='N';
textcolor(12)
end
endif)
end
else
if eof(archdat) and (contreg)0 then
begin
textcolor(12+16);suena(500);
gotoxy(25,24);write('FIN DEL ARCHIVO (RETURNO) ');
while not keypressed do;
textcolor(12)
end
endif)
else
if contreg=0 then
begin
clrscr;
textcolor(28);gotoxy(21,12);
write('IMPOSIBLE VERIFICAR INFORMACION (RETURNO)');
suena(500);
textcolor(12);
while not keypressed do;
continua='N'
end
endif)
endif)
end
end while);
end
else
if (numera)0 then
begin
clrscr;
textcolor(12+16);suena(500);
gotoxy(22,10);write('NO ES POSIBLE VERIFICAR INFORMACION');

```

```

        while not keypressed do;
            textcolor(2)
        end
    (endif)
endif);
close(archdat)
end
(end procedure verifica);

procedure consultas;
var i:integer;
begin
    clrscr;
    gotoxy(10,1);write('SISTEMA DE DIAGNOSTICO');
    gotoxy(19,3);write('VISITA PERIODICA DE PACIENTES REGISTRADOS.');
```

gotoxy(21,5);write('INTRODUZCA EL NUM. DE CONTROL :(< > (0=FIN)');

```

    assign(archdat,'archdat');
    posic:=1;
    repeat
        lee(ca:cadena,posic,arreglo,7);
        posic:=1
    until cadena('<'');
    numero:=entero(cadena);
    if existe('archdat') and (numero>0) then (se verifica que exista el archivo de datos)
    begin
        reset(archdat);
        regcap.numcon:=0;
        regaux.numcon:=0;
        continua:='S';
        while (not eof(archdat)) do
            begin
                read(archdat,regcap);
                if regcap.numcon=numero then (si hay registro del paciente)
                    regaux:=regcap
                end
            (end while);
            if regaux.numcon=numero then
                begin
                    regaux.consult:=regaux.consult+1;
                    escan(regaux);
                    gotoxy(25,15);write('FECHA DE LA CONSULTA:(',regaux.fecha:8,')');
```

regcap:=regaux;

```

                    gotoxy(22,20);write('OPRIMA CUALQUIER TECLA PARA PROSEGUIR.');
```

while not keypressed do;

```

                    for i:=1 to 12 do
                        regcap.result[i]:=0
                    (end for);
                    pruebas;
                    seek(archdat,filesize(archdat));
                    writet(archdat,regcap);
                    close(archdat)
                end
            else
                begin
```

```

    clrscr;
    textcolor(12+16);suena(500);
    gotoxy(25,10);writeln('NO HAY INFORMACION DISPONIBLE. ');
    while not keypressed do;
    textcolor(2)
    end
  (endif)
end
else
  if (numero>0) then
    begin
      clrscr;
      gotoxy(25,10);writeln('NO HAY INFORMACION DISPONIBLE. ');
      while not keypressed do;
      end
    (endif)
  (endif);
end
(end procedure consultas);

procedure respaldo;
var archaux:archivo;
begin
  clrscr;
  gotoxy(30,5);writeln('SISTEMA DE DIAGNOSTICO ');
  gotoxy(23,7);writeln('MODULO DE RESPALDO DE INFORMACION. ');
  gotoxy(21,9);writeln('A CONTINUACION PODRA SALVAR EL ARCHIVO ');
  gotoxy(24,11);writeln('DE DATOS COMPLETO A UN DISKETTE. ');
  gotoxy(18,15);writeln('COLOQUE EL DISKETTE EN LA UNIDAD DE FLOPPYS. ');
  gotoxy(23,17);writeln('OPRIMA UNA TECLA CUANDO ESTE LISTA ');
  while (not keypressed) do;
  assign(archdat,'archdat');
  if existe('archdat') then (se verifica que exista el archivo de datos)
  begin
    reset(archdat);
    regcap.numcont:=0;
    assign(archaux,'a:archaux');
    rewrite(archaux);
    clrscr;
    gotoxy(29,10);writeln('REGISTROS TRANSFERIDOS ');
    while (not eof(archdat)) do
      begin
        read(archdat,regcap);
        write(archaux,regcap);
        gotoxy(37,12);writeln(filepos(archdat):6);
        delay(2)
      end
    (end while);
  end
else
  begin
    clrscr;
    textcolor(12+16);suena(500);
    gotoxy(22,10);writeln('NO ES POSIBLE TRANSFERIR INFORMACION ');

```

```

    while not keypressed do
      textcolor(2)
    end
  (endif);
  close(archaux);
  close(archdat);
end
(end procedure respaldo);

procedure checpas(var cuenterr:integer;pasok:cad6;var paschec:cad6);
var
  i:integer;
begin
  cuenterr:=0;
  repeat
    read(kb,paschec);
    for i:=1 to 6 do
      paschec[i]:=upcase(paschec[i])
    (end for);
    if paschec()=pasok then
      begin
        cuenterr:=cuenterr+1;
        gotoxy(24,24);
        textcolor(12+16);suena(500);
        write('CLAVE ERRONEA, LE QUEDAN ',3-cuenterr;2,' INTENTOS')
      end
    (endif)
  until (cuenterr=3) or (paschec=pasok);
  textcolor(2);
  gotoxy(16,9);write(' ':50);
end
(procedure checpas);

```

```

procedure selybaj;
var
  archpos:array[1..100] of real;
  consin,regist,intcons,varaux:integer;
  consucad00;
  carac:char;
  stpass,stpasarcad5;
  archaux:archivo;
begin
  numero:=1;
  for regist:=1 to 100 do
    archpos[regist]:=0
  (endif);
  assign(archdat,'archdat');
  if existe('archdat') then (se verifica que exista el archivo de datos)
    while numero<>0 do
      begin
        clrscr;
        carac:=' ';
        gotoxy(30,11);write('SISTEMA DE DIAGNOSTICO');

```

```

gotoxy(19,3);write('MODULO DE SELECCION Y BAJA DE INFORMACION. ');
gotoxy(21,5);write('NUMERO DE CONTROL. (0-FIN)  :(< >');
posic:=1;
repeat
  leeaa(cadena,posic,arreglo,?);
  posic:=1
until entero(cadena)=0;
numero:=entero(cadena);
if numero>0 then
  begin
    reset(archdat);
    regcap.nuacnt:=0;
    regaux.nuacnt:=0;
    regist:=0;
    while not eof(archdat) do
      begin
        read(archdat,regcap);
        if regcap.nuacnt=numero then
          begin
            regaux:=regcap;
            regist:=regist+1;
            archpos(regist):=filepos(archdat)-1
          end
        (endif)
      end
    (end while);
    intcons:=0;
    if regaux.nuacnt=numero then
      while intcons=0 do
        begin
          regcap:=regaux;
          escpan(regcap);
          gotoxy(14,15);
          write('ESTA ES LA INFORMACION REQUERIDA, HAY ',regist:3,' REGISTROS');
          gotoxy(12,16);
          write('CON ESTA INFORMACION, ES LA INFORMACION QUE DESEAS BORRAR? ');
          gotoxy(30,17);write(' [ S/N ] :(< > ');gotoxy(wherex-3,wherey);
          intcons:=0;
          consu:='';
          intcons:=0;
          carac:='';
          repeat
            read(hbd,carac);carac:=upcase(carac);
            if ((carac='S') or (carac='N')) then
              begin
                consu:=consu+carac;
                intcons:=intcons+1;
                write(carac)
              end
            else
              begin
                intcons:=0;
                consu:='';
                carac:='';
                gotoxy(40,17);

```

```

        writel' ');
        gotoxy(40,17);
        suena(500)
    end
    (endif)
    until (intcons=1) and ((consu='S') or (consu='N'));
    end
    (end while)
else
    begin
    clrscr;
    textcolor(12+16);suena(500);
    gotoxy(22,10);writel'NO EXISTE INFORMACION DISPONIBLE...';
    while not keypressed do;
    textcolor(2)
    end
    (endif)
    end
    (endif);
if caract='S' then
    begin
    clrscr;
    gotoxy(30,5);writel'SISTEMA DE DIAGNOSTICO';
    gotoxy(23,7);writel'MODULO DE RESPALDO DE INFORMACION.';
    gotoxy(16,9);writel'ANTES DE PROSEGUIR INTRODUCA LA CLAVE PARA BOGAR';
    assign(archaux, archpwa);
    if existe(archpwa) then
        begin
        reset(archaux);
        seek(archaux,5);
        readf(archaux,regaux);
        closef(archaux);
        stpasa:=regaux.pacient;
        end
    else
        stpasa:='PASBOR'
    (endif);
    varaux:=0;
    chequea(varaux,stpasa,stpasa);
    if (varaux=1) and (stpasa=stpasa) then
        begin
        gotoxy(24,24);writel' :36:;
        gotoxy(19,9);writel'A CONTINUACION PODRA SALVAR LOS REGISTROS';
        gotoxy(20,11);writel'SELECCIONADOS PARA POSTERIORES CONSULTAS';
        gotoxy(18,15);writel'COLOQUE EL DISQUETE EN LA UNIDAD DE FLOPPYS.';
        gotoxy(23,17);writel'OPRIMA UNA TECLA CUANDO ESTE LISTO';
        while (not keypressed) do;
        assign(archaux,'asarches');
        if not existe('asarches') then
            rewrite(archaux)
        else
            begin
            reset(archaux);
            seek(archaux,filesize(archaux))
            end
        end
    end

```

```

endif);
for intcons:=1 to regist do
begin
seek(archdat,archpos[regist]);
read(archdat,regaux);
write(archaux,regaux)
end
(end for);
close(archaux);
assign(archaux, 'archleop');
rewrite(archaux);
reset(archdat);
varaux:=1;
for intcons:=0 to filesize(archdat)-1 do
begin
read(archdat,regaux);
if filepos(archdat)-1=archpos[varaux] then
varaux:=varaux+1
else
write(archaux,regaux)
endif)
end
endif);
close(archdat);
close(archaux);
erase(archdat);
rename(archaux, 'archdat')
end
(end if)
end
endif)
end
(end while)
else
begin
clrscr;
textcolor(12+16);suena(500);
gotoxy(22,10);write('NO ES POSIBLE VERIFICAR INFORMACION');
while not keypressed do;
textcolor(2)
end
endif);
close(archdat)
end
(procedure selyba);

```

```

procedure escop(var opcion:integer);
var aux1:char; i:integer;
begin
clrscr;
gotoxy(29,3);write('ESCOGE UNA OPCION: < >');
gotoxy(23,6);textcolor(14);write('1');
textcolor(2);write(' ALTA DE UN NUEVO PACIENTE');
gotoxy(23,8);textcolor(14);write('2');

```

```

textcolor(2);write(' VERIFICACION DE INFORMACION');
gotoxy(23,10);textcolor(14);write('3');
textcolor(2);write(' CONSULTAS POSTERIORES A LA PRIMERA');
gotoxy(23,12);textcolor(14);write('4');
textcolor(2);write(' RESPALDO DE LA INFORMACION A DISQUETE');
gotoxy(23,14);textcolor(14);write('5');
textcolor(2);write(' ELIMINACION SELECTIVA DE INFORMACION');
gotoxy(23,16);textcolor(14);write('6');
textcolor(2);write(' EDICION Y ACTUALIZACION DE INFORMACION');
gotoxy(23,18);textcolor(14);write('7');
textcolor(2);write(' BÚSQUEDA ALFABETICA DE UN PACIENTE');
gotoxy(23,20);textcolor(14);write('8');
textcolor(2);write(' IMPRESION MANUAL DE USUARIO');
gotoxy(23,22);textcolor(14);write('9');
textcolor(2);write(' SALIDA DEL SISTEMA');
opcion:=0;
repeat
  gotoxy(49,3);read(hd,auxil);
  if (auxil='1') and (auxil='9') then
    begin
      write(auxil);
      opcion:=entero(auxil)
    end
  else
    suena(500)
  (end);
until opcion=0
end
(procedure escapc);

procedure edicion;
var
  archpos:array[1..100] of real;
  consin,regist,intcons,sondi:integer;
  consucad99;
  caract:char;
begin
  numero:=1;
  for regist:=1 to 100 do
    archpos[regist]:=0
  (end);
  assign(archdat,'archdat');
  if existe(archdat) then (se verifica que exista el archivo de datos)
    while numero<>0 do
      begin
        clrscr;
        gotoxy(30,1);write(' SISTEMA DE DIAGNOSTICO ');
        gotoxy(24,3);write(' MÓDULO DE EDICION DE INFORMACION ');
        gotoxy(21,5);write(' NÚMERO DE CONTROL (0=FIN) :(' ' ');
        posic:=1;
        repeat
          leeaa(cadena,posic,arreglo,7);
          posic:=1
        until entero(cadena)=0;

```

```

numero:=entero(cadena);
if numero<0 then
begin
reset(archdat);
regcap.nuacont:=0;
regaux.nuacont:=0;
regist:=0;
while not eof(archdat) do
begin
read(archdat,regcap);
if regcap.nuacont=numero then (si hay registro del paciente)
begin
regaux:=regcap;
regist:=regist+1;
archpos[regist]:=filepos(archdat)-1
end
(endif)
end
(end while);
intcons:=1;
if regaux.nuacont=numero then
while intcons<0 do
begin
regcap:=regaux;
escan(regcap);
gotoxy(14,15);
write('ESTA ES LA INFORMACION REQUERIDA, HAY ',REGIST,' REGISTROS');
gotoxy(12,16);
write('CON ESTA INFORMACION, INDICA LA CONSULTA QUE DESEAS EDITAR');
gotoxy(30,17);write('TO=FIN? ( )');gotoxy(wherex-4,wherey);
intcons:=0;
consu:='';
intcons:=0;
carac:=' ';
repeat
read(kbd,carac);
if ((carac='9') or (carac='0')) and (carac<chr(13)) then
suena(500)
else
if carac<chr(13) then
begin
consu:=consu+carac;
intcons:=intcons+1;
write(carac)
end
else
intcons:=3
(endif)
(endif);
if (entero(consu)>regist) then
begin
intcons:=0;
consu:='';
carac:=' ';
gotoxy(39,17);

```

```

        writel' ');
        gotoxy(39,17)
    end
    (endif)
until ((intcons=3) and ((enterolconsu)=0) and (enterolconsu)<=regist));
intcons:=enterolconsu);
if intcons<=0 then
    begin
        gotoxy(14,15);writel' ':60);
        gotoxy(12,16);writel' ':60);
        gotoxy(20,17);writel' ':15);
        reset(archdat);
        seek(archdat,archpos(intcons));
        read(archdat,regcap);
        escpan(regcap);
        consin:=regcap.consult;
        gotoxy(25,20);writel' ESTAN CORRECTOS LOS DATOS? :< >');
        editreg;
        gotoxy(25,20);writel' ':40);
        seek(archdat,archpos(intcons));
        writet(archdat,regcap);
        regaux:=regcap
    end
    (endif)
end
(end while)
else
    begin
        clrscr;
        textcolor(12+16);suena(500);
        gotoxy(22,10);writel' NO ES POSIBLE EDITAR INFORMACION...');
        while not keypressed do;
            textcolor(2)
        end
    (endif)
end
(endif)
end
(end while)
else
    begin
        clrscr;
        textcolor(12+16);suena(500);
        gotoxy(22,10);writel' NO HAY INFORMACION DISPONIBLE...');
        while not keypressed do;
            textcolor(2)
        end
    (endif);
close(archdat)
end
(procedure edicion);

procedure busqueda;
var

```

```

var bol:boolean;
cadbus:cad80;
iaux:integer;

function compara(cadbus:cad80;cadhec:cad80):boolean;
var
  apl:integer;
  bolaux:boolean;
begin
  bolaux:=true;
  for apl:=1 to length(cadbus) do
    bolaux:=(bolaux and (cadbus[apl]=cadhec[apl]));
  end for;
  compara:=bolaux
end
(procedure compara);

begin
  cadbus:='';
  while cadbus<>'0' do
    begin
      clrscr;
      gotoxy(30,1);write('SISTEMA DE DIAGNOSTICO');
      gotoxy(10,3);write('MODULO DE BUSQUEDA ALFABETICA DE INFORMACION. ');
      gotoxy(21,5);write('INTRODUCA LA CADENA A BUSCAR ID=fin1 : ');
      gotoxy(10,7);write('NOMBRE DEL PACIENTE:<' : ' :40,')';
      cadbus:='';
      repeat
        posic:=2;
        leecan(cadbus,posic,arreglo,7)
      until (cadbus<'') or (cadbus='0');
      if cadbus<>'0' then
        begin
          for iaux:=1 to length(cadbus) do
            cadbus[iaux]:=upcase(cadbus[iaux])
          end for;
          assigniarchdat,'archdat';
          if existe('archdat') then
            begin
              reset(archdat);
              varbol:=false;
              iaux:=0;
              while ((not eof(archdat)) and (not varbol)) do
                begin
                  seek(archdat,iaux);
                  iaux:=iaux+1;
                  read(archdat,regcap);
                  regaux:=regcap;
                  varbol:=compara(cadbus,regaux.pacient);
                  if varbol then
                    begin
                      escpan(regaux);
                      while not keypressed do
                        end
                    end
                end
              end
            end
          else
            begin
              write('No se encuentra el paciente');
              readln;
            end
          end
        end
      end
    end
  end
end

```

```

else
  if (eof(archdat) and (not varbol) then
    begin
      clrscr;
      textcolor(12+16);suena(500);
      gotoxy(16,12);
      write('NO EXISTEN REGISTROS QUE CUMPLAN CON LA CONDICION');
      while not keypressed do;
      textcolor(2)
      end
    (endif)
  (endif)
  end
  (end while)
end
else
  begin
    clrscr;
    textcolor(12+16);suena(500);
    gotoxy(20,12);
    write('ERROR, EL ARCHIVO NO CONTIENE INFORMACION');
    while not keypressed do;
    textcolor(2)
    end
  (endif);
  close(archdat)
  end
(endif)
end
(end while)
end
(procedure busqueda);

```

```

procedure mapear;
begin
end;

```

```

procedure cotfechivar datecad@;
begin
  date:= '87/12/25'
  end
(procedure cotfech);

```

```

begin
  writeln(list,chr(12));
  inicializarregio;
  obtfech(date);
  clrscr;
  textmode(c40);
  gotoxy(9,7);write('SISTEMA DE DIAGNOSTICO');
  gotoxy(11,21);write('OPRIMA CUALQUIER TECLA PARA CONTINUAR...');
  gotoxy(10,25);write('ALL RIGHTS RESERVED');

```

```
while not keypressed do;
  textmode(80);
  textcolor(2);
  opcion:=0;
  while opcion<9 do
    begin
      clrscr;
      escop(opcion);
      case opcion of
        1: altas;
        2: verifica;
        3: consultas;
        4: respaldo;
        5: selybaj;
        6: edicion;
        7: busqueda;
        8: lapan;
        else ;
      end (end case)
    end
  (end while);
  close(archdat);
  clrscr
end.
```

**APENDICE B**

**DATOS TECNICOS GENERALES**

# Installation

## BACKING UP THE DISKETTE

The software supplied with the AIO8 is on a standard 5 1/4" diskette (double sided, double density, 360K). You should make a backup copy of the diskette before you proceed further. The diskette is not copy-protected. Therefore you may use either the COPY or DISKCOPY commands found in DOS.

You should copy the AIO8 software routines to the same diskette you run BASIC from, or to the subdirectory which contains BASIC if you are using a hard disk drive.

## HARDWARE INSTALLATION

The AIO8 requires eight consecutive address locations in I/O address space. The base address of the AIO8 is restricted by the following conditions:

1. The base address must be within the range 200 to 3FF hex.
2. The base address (and the following seven consecutive addresses) should not conflict with any IBM reserved I/O address.
3. The base address (and the following seven addresses) must not conflict with any address assigned to another active device.
4. The base address will always automatically fall on an 8-byte boundary. This means that the base address will end in a 0 or an 8 hex.

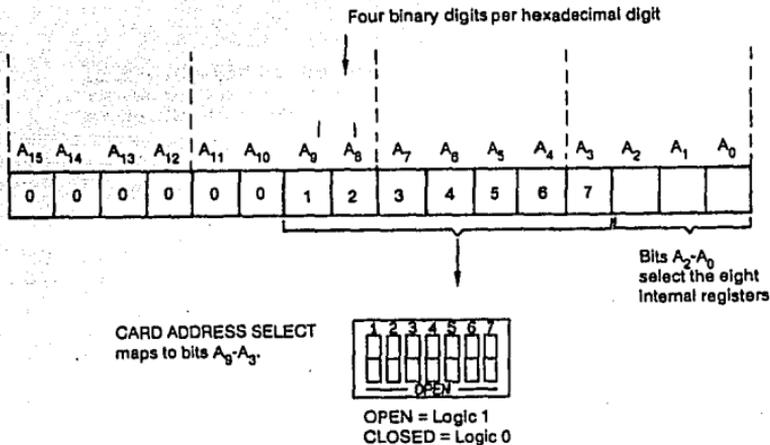
## IBM Reserved I/O Addresses

ADDRESS	DEVICE
000-01F	DMA controller 1
020-03F	Interrupt controller 1
040-05F	Timer
060-06F	Keyboard
070-07F	Real-time clock
080-09F	DMA page register
0A0-0BF	Interrupt controller 2
0C0-0DF	DMA controller 2
0F0-0FF	Math coprocessor
100-1EF	Not usable
1F0-1F8	Fixed disk
200-207	Game I/O
278-27F	Parallel printer port 2 (LPT2:)
2F8-2FF	Serial Port 2 (COM2:)
300-31F	Prototype card
360-36F	Reserved
378-37F	Parallel printer port 1 (LPT1:)
380-38F	Monochrome display
3C0-3CF	Reserved
3D0-3DF	Color graphics display
3F0-3F7	Diskette controller
3F8-3FF	Serial port 1 (COM1:)

This covers the standard I/O options, but if you have other I/O peripherals (such as hard disk drives, special graphics boards, prototype cards etc.) they will also be sharing I/O address space. Memory addressing is separate from I/O addressing so there is no possible conflict with any add-on memory.

Usually, a good choice is to put the AIO8 at base address 300, 308 or 310 hex. (Note if you are using an IBM prototype board, it uses the Hex 300-31F address space and would conflict; 330 hex or 340 hex would be a good choice in this case).

There is a BASIC routine called INSTALLBAS on the diskette supplied with the AIO8. This routine will give you a visual display of the correct settings for the base address DIP-switch, for any address desired. The program is very easy to use, and prompts you at each step.



#### Base I/O Address Switch Setting

INSTALL.BAS performs one further optional function. You can generate a file named AIO8.ADR which contains the base I/O address that you have selected. If your application programs read this file instead of declaring the address in each program (list LOADCALL.BAS to see how it's done), then should you wish to change this address in the future, all you have to do is alter the AIO8.ADR file instead of altering dozens of application programs. Your AIO8 disk comes with a AIO8.ADR file loaded with decimal 768 (Hex 300). This file will, of course, be overwritten if you choose to generate another address file when you run INSTALL.

The next step is to remove the AIO8 board from its protective electrostatic packaging and set the DIP-switch. It is a good precaution to discharge any electrostatic charge you may have accumulated by touching the metal frame of your computer (assuming you have it grounded as it should be for safety). Also, at this stage, check that the interrupt level selection jumper on header J2 is in the rightmost (X = inactive) position. We don't want to start off by hanging your computer up in some non-existent interrupt routine!

TURN OFF THE POWER on your computer and remove the case (See the instruction manual for the computer you own). The AIO8 will fit in any of the regular full depth slots or "half" slots, with one important exception:

## NOTE

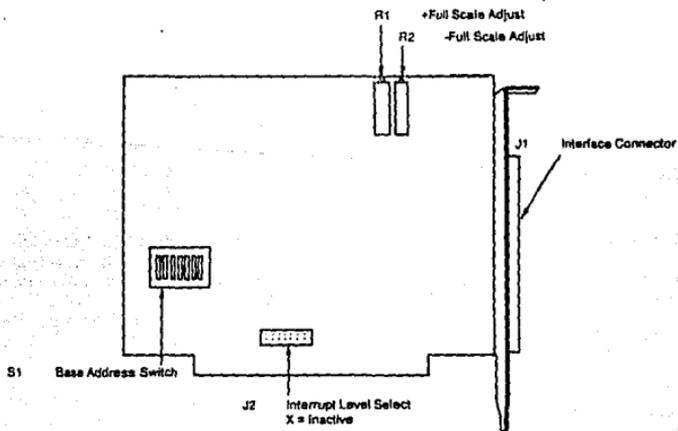
If you own an IBM PC-XT or an Action 5531, do not install the AICP-AIO8 in slot J8, the slot next to the power supply. This slot has separate bus driver circuitry from the other slots, because it was intended for connection to an expansion chassis. The AIO8 will not work properly in this slot.

Installation is now complete. You may plug any of the AIO8's accessories or your own cable into the 37-pin D connector on the rear.

Remember the golden rule whenever removing or re-installing any peripheral board, including the AIO8: always TURN OFF THE POWER. Failure to observe this precaution may cause costly damage to your computer or the AIO8 board.

One more precaution concerning storage and handling of the AIO8 board. If for any reason you wish to remove the AIO8 board, retain the special electrostatically shielded package and use it for storage of AIO8 when not in your computer.

Figure 2.2 shows the locations of switches and functions on the AIO8 board.



Layout of the AICP-AIO8

## Storage of Integer Variables

Data is stored in integer variables (% type) in 2's complement form. Each integer variable uses 16 bits or 2 bytes of memory. 16 bits of data is equivalent to values from 0 to 65,535 decimal, but the 2's complement convention interprets the most significant bit as a sign bit so the actual range becomes -32,768 to +32,767 (a span of 65,535). Numbers are represented as follows:

	HIGH BYTE								LOW BYTE							
	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
+32,767	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
+10,000	0	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0
+1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
-10,000	1	1	0	1	1	0	0	0	1	1	1	1	0	0	0	0
-32,768	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Sign bit

1 if negative, 0 if positive

Integer variables are the most compact form of storage for the 12-bit data from the A/D converter and 16-bit data of the 8253 interval timer. So, to conserve memory and disk space and optimize execution speed, all data exchange via the CALL is through integer type variables. This poses a programming problem when handling unsigned numbers in the range 32,768 to 65,535.

If you wish to input or output an unsigned integer greater than 32,767 then it is necessary to work out what its 2's complement signed equivalent is. As an example, assume we want to load a 16 bit counter with 50,000 decimal. An easy way of turning this to binary is to enter BASIC and execute PRINT HEX\$(50000). This returns C350 or binary:

50,000 (Hex C350) Binary 1100 0011 0101 0000

Since the most significant bit is 1 this would be stored as a negative integer and in fact the correct integer variable value would be 50,000 - 65,536 = -15,536. The programming steps for switching between integer and real variables for representation of unsigned numbers between 0 and 65,535 is therefore:

PIN	FUNCTION
1.	+ 12 V
2.	CTR. 0 CLOCK
3.	CTR. 0 OUT
4	CTR. 1 CLOCK
5.	CTR. 1 OUT
6	CTR. 2 OUT
7.	OP 1
8.	OP 2
9.	OP 3
10	OP 4
11	DIG. COM.
12	L.L. GND.
13	L. L. GND.
14	L.L. GND.
15	L.L. GND.
16	L.L. GND.
17	L.L. GND.
18	L.L. GND
19	V ref. (10VDC)
20	- 12 V
21	CTR 0 GATE
22	CTR 1 GATE
23	CTR. 2 GATE
24	INTERRUPT INPUT
25	IP 1
26	IP 2
27	IP 3
28	DIG. COM.
29	+ 5 V
30	ANALOG IN 7
31	ANALOG IN 6
32	ANALOG IN 5
33	ANALOG IN 4
34	ANALOG IN 3
35	ANALOG IN 2
36	ANALOG IN 1
37	ANALOG IN 0

Digital outputs

Digital Inputs

A108  
PIN ASSIGNATION

Analog signal inputs  
( use L.L. GND'S for return)

PIN	FUNCTION
1	+5V
2	CTR 0 OUT
3	OP 3
4	OP 1
5	IP 3
6	IP 1
7	POWER GND.
8	V ref(-5V)
9	D/A 0 OUT
10	D/A 0 REF IN
11	CH7 LO IN/ *CH15 HI IN
12	CH6 LO IN/ *CH14 HI IN
13	CH5 LO IN/ *CH13 HI IN
14	CH4 LO IN/ *CH12 HI IN
15	CH3 LO IN/ *CH11 HI IN
16	CH2 LO IN/ *CH10 HI IN
17	CH1 LO IN *CH9 HI IN
18	CH0 LO IN *CH8 HI IN
19	L. L. GND.
20	CTR 2 OUT
21	CTR 0 CLOCK IN
22	OP 2
23	OP 0
24	IP2. CTR 0 GATE
25	IPO TRIG 0
26	D A 1 REF IN
27	D A 1 OUT
28	L.L. GND
29	L. L. GND.
30	CH7 HI IN
31	CH6 HI IN
32	CH5 HI IN
33	CH4 HI IN
34	CH3 HI IN
35	CH2 HI IN
36	CH1 HI IN
37	CH0 HI IN

AID16  
PIN ASSIGNATION

Alternative connections apply in 16 channel single Ended (S.E.) input configuration ( set by 8 16 switch)  
I O connector (37 pin male "D")

PIN	FUNCTION
1	+ EXCITATION
2	+ EXCITATION
3	
4	
5	-EXCITATION
6	- EXCITATION
7	+ EXCITATION SENSE
8	-EXCITATION SENSE
9	
10	CHL +
11	CHL -
12	GND
13	GND
14	CH2 +
15	CH2 -
16	GND
17	GND
18	CH3+
19	CH3 -
20	GND
21	GND
22	CH4+
23	CH 4-
24	GND
25	GND
26	DIGITAL OUTPUTS
27	DIGITAL OUTPUTS
28	DIGITAL OUTPUTS
29	DIGITAL OUTPUTS
30	DIGITAL OUTPUTS
31	DIGITAL OUTPUTS
32	DIGITAL OUTPUTS
33	DIGITAL OUTPUTS
34	GND
35	GND
36	+ 5
37	+ 5

SG04

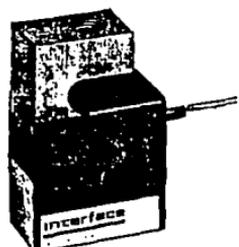
PIN ASSIGNATION

**Interface**  
ADVANCED FORCE MEASUREMENT

## SUPER-MINI LOAD CELLS



Model SM-25



Model SM-1000

Designed  
for  
Precision  
Electronic  
Force Measurement

### FEATURES

- Ultra Precision
- Excellent Linearity
- High Repeatability
- Thermally Compensated
- Low Moment Sensitivity
- Low Cost
- Easily Installed
- NBS Handbook 44 Sealable

RATED CAPACITIES: 10, 25, 50, 100, 250, 500 and 1000 pounds  
(44N, 111N, 222N, 445N, 1112N, 2224N, 4448N)

These ultra precision strain gage load cells are designed expressly for all controlled environment, tension and compression force and weight measuring requirements. Interface's application of proprietary advanced materials technology, in strain gage and flexure design, produces load cells with the highest accuracy in the industry yet priced competitively with lower performance units.

These rugged cells have no moving parts to wear out or get out of adjustment. The specifications listed below illustrate the superior performance of Interface SM Series load cells and are a major factor in their worldwide acceptance in applications such as structural force testing, thrust measurement, steelyard rod conversions (to H-44, PTB and SIM requirements), conveyor scales, check weighers, counting and whole scales, tensile testing and engine dynamometers.

The Interface optional MR (Moisture Resistant) Super-Mini is now available as a cost effective method of protecting 25 thru 250 lb load cells against the effects of exposure to high humidity (up to 95% RH) and periodic condensation.

For metric applications see Metric Super-Mini Series offering 200N, 500N, 1000N, 2000N and 5000N capacities and metric mounting threads.

### SPECIFICATIONS\*

Non-Linearity—% Rated Output	±0.03
Hysteresis—% Rated Output	±0.02
Non-Repeatability—% Rated Output	±0.01
Temperature Range, Compensated—°F	{ -18 to 66°C }
Temperature Range, Operating—°F	{ -54 to 93°C }
Temperature Effect on Rated Output—% of Reading/100°F (% of Reading/55.6°C)	±0.08
Temperature Effect on Zero—% Rated Output/100°F (% Rated Output/55.6°C)	±0.08
Temperature Effect on Zero—% Rated Output/100°F (% Rated Output/55.6°C)	±0.12
For Moisture Resistant Models	
**Creep, After 20 Min.—% Rated Output	±0.03
Overload Ratings—% Rated Capacity	
Safe	±150
Ultimate	±500
Nominal Output—mV/V	2
Zero Balance—% Rated Output	±1
Input Resistance—Ohms	350 ± 3.5
Output Resistance—Ohms	350 ± 3.5
Excitation Voltage	
Recommended—VDC	10
Insulation Resistance, Bridge to Case—Megohms	5000

\*The Basic Load Cell Terminology and Definitions

\*\*Creep effect, shown as determined at rated capacity. Creep performance

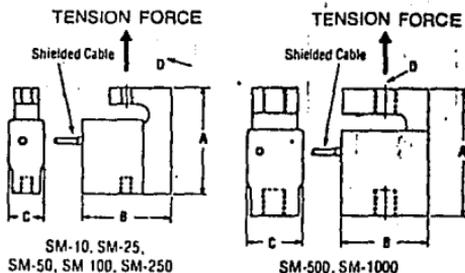
and load is proportional to the rated capacity.

**Interface**  
**LOAD CELLS**  
**SUPER-MINI**

# SUPER-MINI LOAD CELLS

## INSTALLATION DIMENSIONS

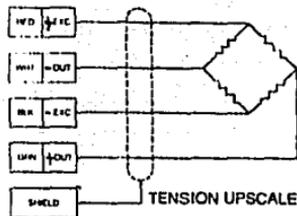
Model	A	B	C	D
SM 10, 25, 50	1/2" (12.7mm)	1/2" (12.7mm)	3/8" (9.5mm)	1/2-20 UNF-2B 1/2" Deep, top & bottom
SM 100, 250	3/4" (19.0mm)	1/2" (12.7mm)	1/2" (12.7mm)	1/2-20 UNF-2B 1/2" Deep, top & bottom
SM 500, 1000	1" (25.4mm)	3/4" (19.0mm)	1/2" (12.7mm)	1/2-20 UNF-2B 1/2" Deep, top & bottom



## ELECTRICAL INFORMATION

SM Series load cells are provided with a 4-conductor shielded cable (AWG 24) 1 (1.5m) long.

Wiring color code complies with ISA S37.6-1975 "Specifications and Tests for Strain Gauge Force Transducers."



## APPLICATION NOTES

- Super-Mini load cells are designed for controlled environmental applications. In general, they can be used anywhere a readout instrument is used.
- One diameter thread engagement is desirable—approximately 1/8" (3mm) on the 10 through 250 pound ranges and 1/2" (12mm) on the 500 and 1000 pound units.
- Multimedia Resistant Super-Mini load cells are marked -40. These units are resistant to high humidity conditions up to and including 95% Relative Humidity and periodic exposure to condensation. The design capability is not intended for submerged operation.

- Jam nuts may be used, however care should be exercised to not apply excessive torque across the load cell. Torque should be reacted against the load cell structure immediately adjacent to the jam nut.

Model	Jam nut torque/inch-pounds
SM-10	5
SM-25	10
SM-50	20
SM-100, 250	40
SM-500, 1000	200

## TERMS AND CONDITIONS

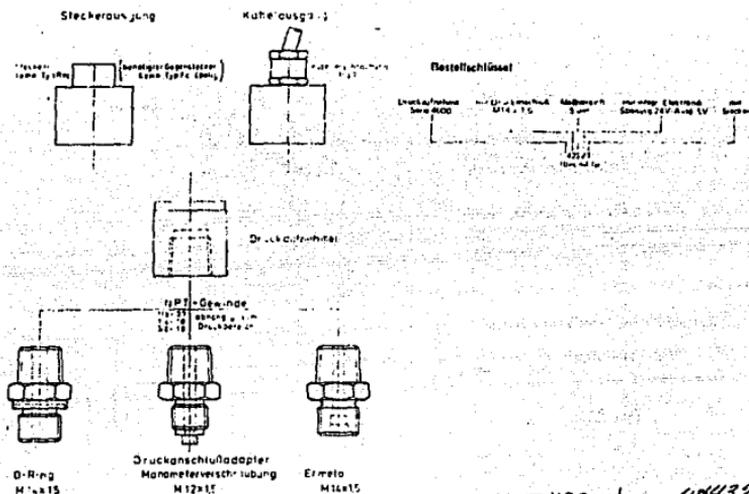
- Ordering Information:** Super-Minis are ordered by specifying Series (SM) and capacity (10, 25, 50, 100, 250, 500, 1000 pounds). Example: SM-100.
- Pricing:** Consult your local Interface Representative or the factory for price and delivery.
- Terms:** Net 30 days in U.S. dollars, FOB Scottsdale, Arizona U.S.A.
- Warranty:** Interface, Inc.'s standard two-year warranty is applicable to the Super-Mini Series load cell. Interface, Inc. certifies that its calibration measurements are traceable to the U.S. National Bureau of Standards (NBS).

Prices and specifications subject to change without notice

**interface**  
ADVANCED FORCE MEASUREMENT

INTERFACE, INC. 7401 E. BUTHERUS DR. SCOTTSDALE, ARIZONA 85260 USA (602) 948-5555 TELEX: 608-394

15-228-181 2.5K  
Printed in U.S.A.



TYPE Nr. 40432.

Serie	Druckanschluß	Meßbereich	Elektronik	Elektrischer Anschluß
4 ALU	0 Innengewinde *	0 Sonderbereich	0 ohne	0 ohne Endgehäuse
3 Stahl	1 Ermetoverschraubung	1 2 bar	1 ± 15V / 5V	1 Stecker
1/4 PH	2 M 14 x 1,5 DIN 6	2 5 bar	2 24V / 5V	2 Kabel
	3 Manometerverschraubung	3 15 bar	3 24V / 4 - 20 mA	
	* abhängig vom Druckbereich	4 100 bar	4 ± 15V / 10V	
	2, 5 und 10 bar - 3/8" NPT	5 300 bar		
	100 bar - 1/4" NPT	6 700 bar		
	300 und 700 bar - 1/8" NPT			

Druckaufnehmer Serie 4000 Standard, Meßbereich 2 bar - 700 bar mit Innengewinde ohne Endgehäuse ohne Elektronik		ALU-Ausführung DM 168,-	Edelstahlausführung DM 258,-
Option:	Druckanschluß	Ermeto M14 x 1,5 DIN 6 Manometerverschraubung M12 x 1,5	DM 48,- DM 48,- DM 48,-
	Elektronik	± 15V/5V + 24V/5V + 24V/4-20 mA + 15V/10V	DM 90,- DM 95,- DM 95,- DM 105,-
	Endgehäuse mit	Stecker Ausgang	DM 48,-
	Endgehäuse mit	Kabel 1 m	DM 38,-
		Gegenstecker	DM 35,-
		Kabel Hd. Meter	DM 12,-
			Mengenrabatt 1 - 24 Stk. = netto 25 - 40 Stk. = 5% 50 - 79 Stk. = 8% 1' - 19 Stk. = 10% r J Stk. = auf Anfrage

**Druckaufnehmer Serie 4000**

Die Druckaufnehmer der Serie 4000 eignen sich aufgrund ihrer mechanischen und elektrischen Spezifikation bestens zum industriellen Einsatz. Eine Metallmembrane macht den Geber unempfindlich gegen aggressive Medien. Die verschiedenen lieferbaren Druckanschlüsse ermöglichen einen problemlosen mechanischen Anschluß an jedes pneumatische oder hydraulische System.

Die Geber der Serie 4000 DC beinhalten eine komplette Elektronik. Der Ausgang von 5 V oder 4 - 20 mA erlaubt den Anschluß an alle gängigen Rechner und Maschinensteuerungssysteme.

Der Druckaufnehmer ist aus Aluminium gefertigt, sämtliche Versionen und Druckbereiche sind jedoch auch in einer Edelstahlausführung lieferbar.

**Spezifikationen**

**Meßbereiche:**  
2, 5, 15, 100, 300 und 700 bar (andere auf Kundenwunsch)

**Überlastsicherheit:** 2-facher Nenndruck

**Meßmedien:** Gase, Flüssigkeiten

**Ausgangssignal:**  $100 \text{ mV} \pm 1\%$   
mit Elektronik 0 - 5 V, 4 - 20 mA  $\pm 1\%$

**Linearität:** besser 1% v.E.

**Hysterese:** 0,1% v.E.

**Temperaturdrift:** 0,02 %/°C

**Arbeitstemperaturbereich kompensiert:** 0 bis 70 °C

**Lagertemperaturbereich:** -20 °C bis +100 °C

**Versorgungsspannung (Geber ohne Elektronik):** 5 V

**Versorgungsspannung (Geber mit Elektronik):**  
+ 15 V  $\pm 10\%$  oder 24 V  $\pm 10\%$

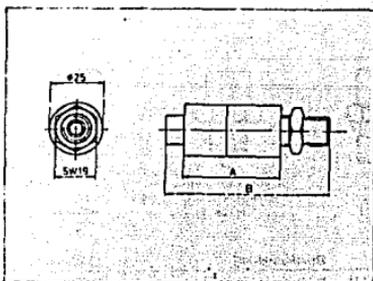
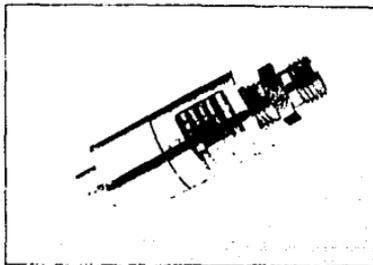
**Mechanischer Anschluß:**

Ermeto oder M14 x 1,5 DIN Gsd. Manometerverschraub.

**Elektrischer Anschluß:**

4-adriges Kabel 1 m oder 4poliger Lemo-Stecker

**Mechanische Abmessung** siehe Zeichnung.



Druckbereich (bar)	A (mm)	B (mm)
2 - 5: 15	44,5	70
100	47	81
300	43	77
700	43,5	78

# LINEAR INTEGRATED CIRCUITS

# SERIES $\mu$ A7800 POSITIVE-VOLTAGE REGULATORS

BULLETIN NO. DL5 12386 MAY 1976—REVISED SEPTEMBER 1977

- 3-Terminal Regulators
- Output Current up to 1.5 A
- No External Components
- Internal Thermal Overload Protection
- Direct Replacements for Fairchild  $\mu$ A7800 Series
- High Power Dissipation Capability
- Internal Short-Circuit Current Limiting
- Output Transistor Safe-Area Compensation

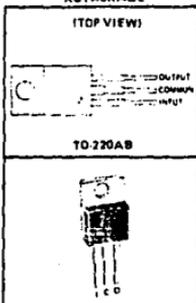
NOMINAL OUTPUT VOLTAGE	REGULATOR
5 V	$\mu$ A7805C
8 V	$\mu$ A7806C
9 V	$\mu$ A7808C
8.5 V	$\mu$ A7885C
10 V	$\mu$ A7810C
12 V	$\mu$ A7812C
15 V	$\mu$ A7815C
18 V	$\mu$ A7818C
22 V	$\mu$ A7822C
24 V	$\mu$ A7824C

## description

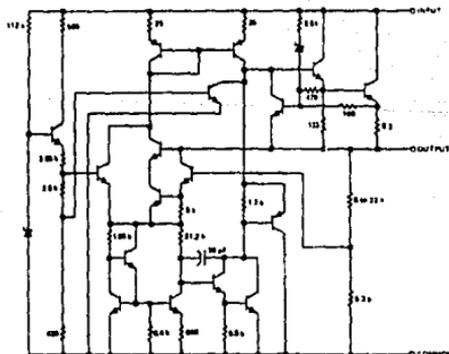
This series of fixed-voltage monolithic integrated-circuit voltage regulators is designed for a wide range of applications. These applications include on-card regulation for elimination of noise and distribution problems associated with single-point regulation. One of these regulators can deliver up to 1.5 amperes of output current. The internal current limiting and thermal shutdown features of these regulators make them essentially immune to overload. In addition to use as fixed-voltage regulators, these devices can be used with external components to obtain adjustable output voltages and currents and also as the power-pass element in precision regulators.

## KC PACKAGE

(TOP VIEW)



## schematic



Resistor values shown are nominal and in ohms.

## TYPES $\mu$ A7822C, $\mu$ A7824C POSITIVE-VOLTAGE REGULATORS

$\mu$ A7822C electrical characteristics at specified virtual junction temperature,  
 $V_I = 31$  V,  $I_O = 500$  mA (unless otherwise noted)

PARAMETER	TEST CONDITIONS <sup>1</sup>		$\mu$ A7822C			UNIT
			MIN	TYP	MAX	
Output voltage	$I_O = 5$ mA to 1 A, $P = 15$ W	$V_I = 25$ V to 35 V, 0°C to 125°C	25°C	21.1	22	22.9
			0°C to 125°C	20.9		23.1
Input regulation	$V_I = 25$ V to 35 V, $V_I = 26$ V to 34 V	25°C		17	440	mV
				6	220	
Ripple rejection	$V_I = 26$ V to 36 V, $f = 120$ Hz	0°C to 125°C	51	67		dB
Output regulation	$I_O = 5$ mA to 1.5 A, $I_O = 25$ mA to 750 mA	25°C		17	440	mV
				4	220	
Output resistance	$f = 1$ kHz	0°C to 125°C		0.028		$\Omega$
Temperature coefficient of output voltage	$I_O = 5$ mA	0°C to 125°C		-1.3		mV/°C
Output noise voltage	$f = 10$ Hz to 100 kHz	25°C		160		$\mu$ V
Dropout voltage	$I_O = 1$ A	25°C		2.0		V
Bus current		25°C		4.6	8	mA
Bus current change	$V_I = 25$ V to 36 V, $I_O = 5$ mA to 1 A	0°C to 125°C		1		mA
					0.5	
Short-circuit output current		25°C		175		mA
Peak output current		25°C		2.1		A

$\mu$ A7824C electrical characteristics at specified virtual junction temperature,  
 $V_I = 33$  V,  $I_O = 500$  mA (unless otherwise noted)

PARAMETER	TEST CONDITIONS <sup>1</sup>		$\mu$ A7824C			UNIT
			MIN	TYP	MAX	
Output voltage	$I_O = 5$ mA to 1 A, $P = 15$ W	$V_I = 27$ V to 38 V, 0°C to 125°C	25°C	23	24	25
			0°C to 125°C	22.8		25.2
Input regulation	$V_I = 27$ V to 38 V, $V_I = 28$ V to 36 V	25°C		18	480	mV
				6	240	
Ripple rejection	$V_I = 28$ V to 38 V, $f = 120$ Hz	0°C to 125°C	50	66		dB
Output regulation	$I_O = 5$ mA to 1.5 A, $I_O = 25$ mA to 750 mA	25°C		12	480	mV
				4	240	
Output resistance	$f = 1$ kHz	0°C to 125°C		0.028		$\Omega$
Temperature coefficient of output voltage	$I_O = 5$ mA	0°C to 125°C		-1.5		mV/°C
Output noise voltage	$f = 10$ Hz to 100 kHz	25°C		170		$\mu$ V
Dropout voltage	$I_O = 1$ A	25°C		2.0		V
Bus current		25°C		4.6	8	mA
Bus current change	$V_I = 27$ V to 38 V, $I_O = 5$ mA to 1 A	0°C to 125°C		1		mA
					0.5	
Short-circuit output current		25°C		150		mA
Peak output current		25°C		2.1		A

<sup>1</sup>All characteristics are measured with a capacitor across the input of 0.1  $\mu$ F and a capacitor across the output of 0.1  $\mu$ F. All characteristics except noise voltage and ripple rejection are measured using test techniques ( $t_{ON} < 10$  ms, duty cycles 4.5%). Output voltage changes due to changes in internal temperature must be taken into account separately.

## SERIES $\mu$ A7800 POSITIVE-VOLTAGE REGULATORS

absolute maximum ratings over operating temperature range (unless otherwise noted)

	$\mu$ A7822C, $\mu$ A7824C	$\mu$ A78...C	UNIT
Input voltage	4C All others	38	V
Continuous total dissipation at 25°C free-air temperature (see Note 1)		7	W
Continuous total dissipation at (or below) 25°C case temperature (see Note 1)		11	W
Operating free-air, case, or virtual junction temperature range		0 to 180	°C
Storage temperature range		-88 to 150	°C
Lead temperature 1/16 inch from case for 10 seconds		290	°C

Note 1: For operation above 25°C free-air or case temperature, refer to Dissipation Derating Curves, Figure 1 and Figure 2.

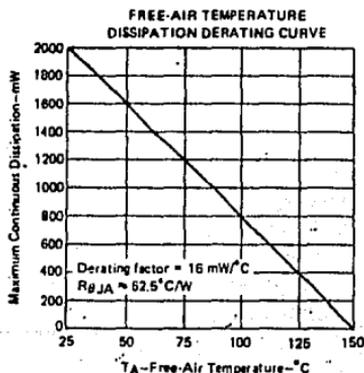


FIGURE 1

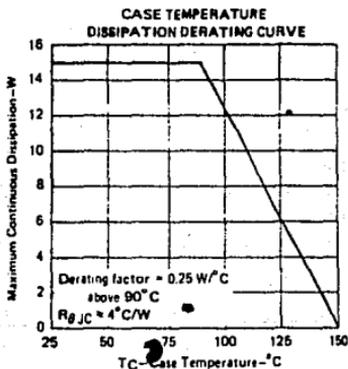


FIGURE 2

recommended operating conditions

	MIN	MAX	UNIT	
Input voltage, $V_I$	$\mu$ A7805C	7	25	V
	$\mu$ A7806C	8	25	
	$\mu$ A7808C	10.5	25	
	$\mu$ A7885C	10.5	25	
	$\mu$ A7810C	12.5	28	
	$\mu$ A7812C	14.5	30	
	$\mu$ A7815C	17.5	30	
	$\mu$ A7818C	21	33	
	$\mu$ A7822C	25	36	
$\mu$ A7824C	27	38		
Output current, $I_O$		1.5	A	
Operating virtual junction temperature, $T_J$	0	125	°C	

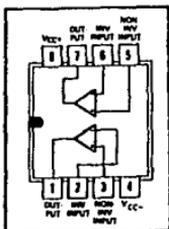
# LINEAR INTEGRATED CIRCUITS

# TYPES NE5532, NE5532A DUAL LOW-NOISE OPERATIONAL AMPLIFIERS

BULLETIN NO. DL-5 12732

- Equivalent Input Noise Voltage ... 5 nV/ $\sqrt{\text{Hz}}$  Typ at 1 kHz
- Unity-Gain Bandwidth ... 10 MHz Typ
- Common-Mode Rejection Ratio ... 100 dB Typ
- High DC Voltage Gain ... 100 V/mV Typ
- Peak-to-Peak Output Voltage Swing ... 32 V Typ with  $V_{CC1} = \pm 18$  V and  $R_L = 600 \Omega$
- High Slew Rate ... 9 V/ $\mu\text{s}$  Typ
- Wide Supply Voltage Range ...  $\pm 3$  V to  $\pm 20$  V
- Designed to be Interchangeable with Signetics NE5532 and NE5532A

NE5532, NE5532A ... JG OMP  
DUAL IN-LINE PACKAGE  
(TOP VIEW)

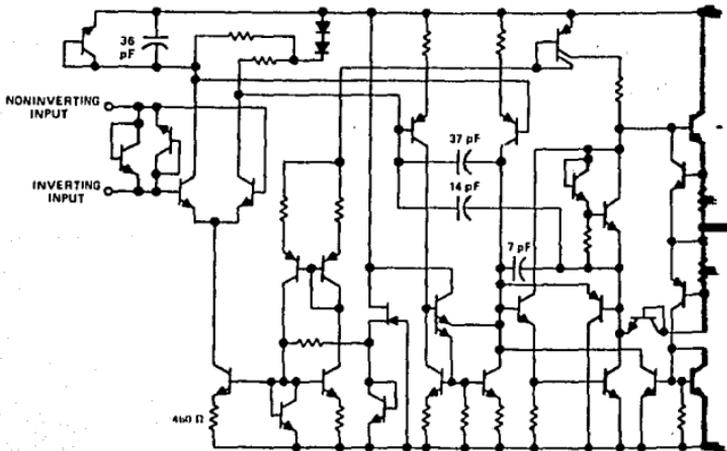


## description

The NE5532 and NE5532A are monolithic high-performance operational amplifiers combining excellent characteristics. They feature very low noise, high output drive capability, high unity-gain and maximum bandwidths, low distortion, high slew rate, input-protection diodes, and output short-circuit protection. Operational amplifiers are internally compensated for unity gain operation. The NE5532A has guaranteed limits for equivalent input noise voltage.

The NE5532 and NE5532A are characterized for operation from 0°C to 70°C.

## schematic (each amplifier)



All component values shown are nominal.

Copyright © 1979 by Texas Instruments

79

## ADVANCE INFORMATION

This document contains information on a new product. Specifications are subject to change without notice.

TEXAS INSTRUMENTS  
INCORPORATED

POST OFFICE BOX 225012 • DALLAS, TEXAS 75265

## TYPES NE5532, NE5532A DUAL LOW-NOISE OPERATIONAL AMPLIFIERS

electrical characteristics,  $V_{CC2} = \pm 15$  V,  $T_A = 25^\circ\text{C}$  (unless otherwise noted)

PARAMETER	TEST CONDITIONS	NE5532, NE5532A			UNIT
		MIN	TYP	MAX	
$V_{IO}$ Input offset voltage	$T_A = 25^\circ\text{C}$	0.5			mV
	$T_A = 0^\circ\text{C}$ to $70^\circ\text{C}$	5			
$I_{IO}$ Input offset current	$T_A = 25^\circ\text{C}$	10			nA
	$T_A = 0^\circ\text{C}$ to $70^\circ\text{C}$	200			
$I_{IB}$ Input bias current	$T_A = 25^\circ\text{C}$	200			nA
	$T_A = 0^\circ\text{C}$ to $70^\circ\text{C}$	800			
$V_{ICR}$ Common-mode input voltage range		$\pm 12$ $\pm 13$			V
$V_{OPP}$ Maximum peak-to-peak output voltage swing	$R_L > 600 \Omega$	$V_{CC2} = \pm 15$ V	24		V
		$V_{CC2} = \pm 18$ V	30		
$A_{VD}$ Large-signal differential voltage amplification	$R_L > 600 \Omega$ , $V_O = \pm 10$ V, $R_I > 2 \text{ k}\Omega$ , $V_O = \pm 10$ V	$T_A = 25^\circ\text{C}$	15		V/mV
		$T_A = 0^\circ\text{C}$ to $70^\circ\text{C}$	10		
		$T_A = 25^\circ\text{C}$	25		
		$T_A = 0^\circ\text{C}$ to $70^\circ\text{C}$	15		
$A_{vd}$ Small-signal differential voltage amplification	$f = 10$ kHz	2.2			V/mV
$B_{OH}$ Maximum output-swing bandwidth	$R_L = 600 \Omega$ , $V_O = \pm 10$ V	140			kHz
	$R_L = 600 \Omega$ , $V_{CC2} = \pm 18$ V, $V_O = \pm 14$ V	100			
$B_1$ Unity-gain bandwidth	$R_L = 600 \Omega$ , $C_L = 100$ pF	10			MHz
$r_i$ Input resistance		30			k $\Omega$
$Z_o$ Output impedance	$A_{VD} = 30$ dB, $R_L = 600 \Omega$ , $f = 10$ kHz	0.3			$\Omega$
$CMRR$ Common-mode rejection ratio		70			dB
$k_{SVR}$ Supply voltage rejection ratio ( $\Delta V_{CC2}/\Delta V_{IO}$ )		80			dB
$I_{OS}$ Output short-circuit current		38			mA
$I_{CC}$ Total supply current	No load	8			mA
$V_{O1}/V_{O2}$ Channel separation	$V_{O1} = 10$ V peak, $f = 1$ kHz	110			dB

operating characteristics,  $V_{CC2} = \pm 15$  V,  $T_A = 25^\circ\text{C}$

PARAMETER	TEST CONDITIONS	NE5532		NE5532A		UNIT
		MIN	TYP	MAX	MIN	
$SR$ Slew rate at unity gain		9				V/ $\mu$ s
Overshoot factor	$V_i = 100$ mV, $A_{VD} = 1$ , $R_L = 600 \Omega$ , $C_L = 100$ pF	10%		10%		
$V_n$ Equivalent input noise voltage	$f = 30$ Hz	8		8	10	nV/ $\sqrt{\text{Hz}}$
	$f = 1$ kHz	5		5	6	
$I_n$ Equivalent input noise current	$f = 30$ Hz	2.7		2.7		pA/ $\sqrt{\text{Hz}}$
	$f = 1$ kHz	0.7		0.7		

## TYPES NE5532, NE5532A DUAL LOW-NOISE OPERATIONAL AMPLIFIERS

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage, $V_{CC+}$ (see Note 1)	22 V
Supply voltage, $V_{CC-}$ (see Note 1)	-22 V
Input voltage, either input (see Notes 1 and 2)	$V_{CC\pm}$
Input current (see Note 3)	±10 mA
Duration of output short-circuit (see Note 4)	unlimited
Continuous total power dissipation at (or below) 25°C free air temperature (see Note 5):	
JG package	825 mW
P package	1000 mW
Operating free-air temperature range: NE5532, NE5532A	0°C to 70°C
Storage temperature range	-65°C to 150°C
Lead temperature 1/16 inch (1.6 mm) from case for 60 seconds: JG package	300°C
Lead temperature 1/16 inch (1.6 mm) from case for 10 seconds: P package	260°C

- NOTES: 1. All voltage values, except differential voltages, are with respect to the midpoint between  $V_{CC+}$  and  $V_{CC-}$ .
2. The magnitude of the input voltage must never exceed the magnitude of the supply voltage.
3. Excessive current will flow if a differential input voltage in excess of approximately 0.6 V is applied between the inputs unless some limiting resistance is used.
4. The output may be shorted to ground or either power supply. Temperature and/or supply voltages must be limited to ensure the maximum dissipation rating is not exceeded.
5. For operation above 25°C free-air temperature, refer to the Dissipation Derating Table. In the JG package, chips are glass-mounted.

DISSIPATION DERATING TABLE

PACKAGE	POWER RATING	DERATING FACTOR	ABOVE $T_A$
JG (Glass-Mounted chip)	825 mW	6.6 mW/°C	25°C
P	1000 mW	8.0 mW/°C	25°C

Also see Dissipation Derating Curves, Section 2.

## BIBLIOGRAFIA

*Circuitos electrónicos discretos e integrados*

Schilling y Belove. Editorial Marcombo

*Teoría de conmutación y diseño lógicos*

Hill Peterson. Editorial Limusa

*Manual de usuario del MS-DOS*

Microsoft

*Manual de operación de la tarjeta AICP-AIOB*

Action Instruments

*Manual de referencia turbo Pascal ver. 3.0*

Borland International

*Programación en Pascal con Pascal/ 100*

Peter Grogonio. Addison-Wesley Publishing Company

*Apuntes de electrónica I de la FES Cuautitlan*

Ing. Antonio Herrera Mejia.

*Apuntes de electrónica III de la FES Cuautitlan*

Ing. Antonio Herrera Mejia

*Apuntes de electrónica IV de la FES Cuautitlan*

*Ing. Antonio Herrera Mejia*

*Libro de datos de circuitos de control lineal*

*Texas Instruments*

*Libro de datos TTL para Ingenieros de diseño*

*Texas Instruments*