

UNIVERSIDAD DON BOSCO

ESCUELA DE ESTUDIOS TECNOLOGICOS



**“ PROTOTIPO LECTOR/ESCRITOR DE CINTAS MAGNETICAS
CONTROLADO POR COMPUTADORA ”**

TRABAJO DE GRADUACION PRESENTADO POR:

JOSE ORLANDO ALAS RODRIGUEZ
EDWIN ANTONIO GAMERO BONILLA
ROBERTO CARLOS ALVARENGA RIVAS

PARA OPTAR AL GRADO DE:

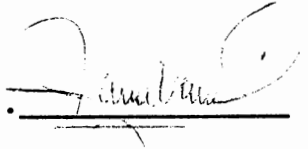
“TECNICO EN INGENIERIA ELECTRONICA/COMPUTACION”

FEBRERO, 1997

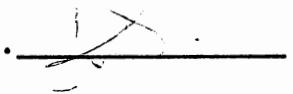
SAN SALVADOR, EL SALVADOR, CENTROAMERICA

COMITE EVALUADOR

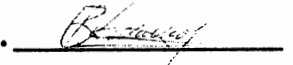
Presidente: Ing. Jaime Varela.

f. 

Miembro : Ing. Wilfredo Santamaria

f. 

Miembro : Tec. Oscar Villalta (asesor)

f. 

AUTORIDADES DE LA UNIVERSIDAD DON BOSCO

Rector: Ing. Federico Huguet.

Decano: Ing Carlos Gutiérrez Peña.

Encargado de Tecnológico: Ing. Angel Soriano.

Roberto Alvarenga.

Es un gran placer poder compartir mi júbilo con todas aquellas personas que de alguna forma colaboraron en la culminación de este trabajo, en especial:

A Dios Todopoderoso que me iluminó en todo momento para la realización de este trabajo de graduación.

A mis padres Juan Ramón Alvarenga y Francisca Elvia Rivas de Alvarenga, por su gran apoyo moral y económico.

A mis Hermanos Mirna Elizabeth Alvarenga Rivas y Juan Ramon Alvarenga Rivas por su comprensión y además por soportar todos mis problemas para realizar este trabajo.

Orlando Alas.

Cuando era pequeño siempre soñaba con llegar a ser grande, más no sabía cuan difícil era crecer y superarse. Pero agradezco a Dios Todopoderoso quien puso en mi vida a personas que me han ayudado a crecer y superarme. Hoy quiero dejar plasmado un humilde agradecimiento a todas esas lindas personas que me ofrecieron su ayuda de una manera incondicional.

En primerísimo lugar agradezco a mi mamá Sra. Antonia Margarita, pues me ha apoyado en todo momento, me ha animado a seguir luchando y este triunfo es dedicado a ella... **!Muchas gracias mamá!**. También es muy importante para mí agradecer a mi tío Sr. Pablo Gallardo ya que siempre ha confiado en mí y no ha dudado en apoyarme. **!Tío lo logré...gracias por todo!**. En tercer lugar (pero no menos importante), agradezco a mi novia Suyapa Marroquin por brindarme su gran positivismo y alegría. **!Gracias amor!**.

Estoy consiente que hay más personas a quienes les debo un agradecimiento. Lastimosamente no puedo mencionarlos a todos pero quiero que sepan que les estoy muy agradecido. **!Gracias a todos!**.

Edwin Gamero.

Agradezco primeramente a Dios por la oportunidad de vivir que me ha brindado, oportunidad de la cual depende todo cuanto hago. En la realización de este proyecto agradezco a mis padres y a mi pequeño hermano Walter por la paciencia y comprensión que tuvieron en los días de desvelo, así como su apoyo moral en toda circunstancia.

Gracias a los amigos que siempre me dieron ánimos para continuar, a pesar de las dificultades, y que siempre creyeron en mi.

INDICE

	Pags.
FINALIDAD DEL PROYECTO	1
INTRODUCCION	2
 CAPITULO I	
DESCRIPCION DEL PROYECTO	
1. SOFTWARE.....	4
1.1 Generalidades	4
1.2 Descripción del Software	5
2. HARDWARE	22
2.1 Descripción del Sistema Mecánico	22
2.2 Diseño Gráfico de la Estructura Mecánica	24
2.3 Descripción del Hardware	28
 CAPITULO II	
DISEÑO DEL PROYECTO	
1. DISEÑO DEL SOFTWARE	40
1.1 Metodología Utilizada	40
1.1.1 Estándares de Documentación	41
1.1.2 Programación	41
1.1.3 Modularidad	44
1.1.4 Convención de Nombres	45
1.1.5 Programación de Teclas	46
1.2 Diseño de Base de Datos	46
1.3 Programas en Pascal para Lectura y Escritura	51
1.4 Gestión de Datos	55

	Pags.
2. DISEÑO DE HARDWARE	58
2.1 Circuito de Control	58
2.2 Circuito de Lectura	60
2.3 Circuito de Escritura	64
2.4 Diseño de Aspectos Varios	68

ANEXOS

FINALIDAD DEL PROYECTO

GENERAL:

Presentar un prototipo lector/escritor de tarjetas magnéticas que permita a otros proyectos futuros agilizar el proceso de reservación y uso de horas máquina de un centro de cómputo.

ESPECIFICOS:

- Aplicar uno de los métodos (FSK), usados para la lectura/escritura de datos binarios sobre tarjetas magnéticas.

- Mostrar la emisión y recepción de datos por medio del puerto serie RS-232C de una PC compatible a IBM.

- Utilizar un sistema manejador de base de datos y el lenguaje de programación Pascal para gestionar información.

INTRODUCCION

En la era actual se han tratado de automatizar procesos manuales, siendo ésta una de las principales preocupaciones en las áreas de electrónica y computación. Actualmente en algunos procesos se ha logrado ésta automatización, como por ejemplo: control de personal, control de calidad, cajeros automáticos por medio de tarjetas de crédito, manejo de inventarios, terminales de venta, ...etc. En muchos de estos casos se utiliza el principio de la lectura y escritura en cintas magnéticas, similar a las tarjetas de crédito; el cual agiliza el control de gran cantidad de personal y reduce el trabajo de este control.

El uso de la cinta magnética no es algo nuevo, puesto que en la década del 60 se ocupó ampliamente para guardar información de computadora, lo que es nuevo es la aplicación a las tarjetas de identificación.

En la última década, nuestro país ha recibido mucho equipo electrónico moderno y es por eso que avanza rápidamente en el uso de lectores de tarjetas magnéticas computarizados. Es posible construir un prototipo de acuerdo a la necesidad concreta. Estos lectores electrónicos son especialmente útiles cuando se quiere agilizar un sistema de colas, el manejo de mucho personal o reducir el papeleo de identificación personal.

En resumen facilitar el trabajo al usuario y al encargado del control. El desarrollo de este trabajo se basa en la descripción de un prototipo lector-escritor.

El presente documento se ha estructurado de la siguiente manera:

CAPITULO I:

Se establecen generalidades del proyecto (Software), definiendo el ambiente en el que se desarrollará la aplicación; y describiendo así el software (Funcionamiento del sistema) y hardware aplicado (Sistema Mecánico y Electrónico).

CAPITULO II:

En este capítulo se establece el diseño del Software y Hardware. En el aspecto de SOFTWARE, se establece la metodología utilizada, y el diseño de las bases de datos con la función de cada campo dentro ellas. En el ámbito de HARDWARE se establece su diseño a través de diagramas, cálculos, tablas y funcionamiento de componentes utilizados.

Para mayor información se han incluido algunos conceptos básicos que ayudarán a la comprensión del diseño.

CAPITULO I

DESCRIPCION DEL PROYECTO

1.S SOFTWARE

1.S.1 Generalidades

Para el desarrollo del sistema administrador de horas en Centro de Cómputo, utilizamos un Gestor de Base de Datos como lo es Fox-Pro 2.6; ya que este gestor permite la interacción amigable entre la computadora y el usuario. Fox-Pro se caracteriza por ser al mismo tiempo un lenguaje muy interactivo, potente, sencillo y compatible con otros gestores de base de datos tales como: dBase III plus, dBase IV, ... etc.

La metodología desarrollada por Fox-Pro se apoya de una nueva tecnología que permite crear de una manera rápida una base de datos, acceder un registro con notable rapidez; resultando verdaderamente efectiva dicha gestión cuando el número de datos almacenados es elevado.

Las aplicaciones que se pueden desarrollar en Fox-Pro son grandiosas, ya que podemos crear ventanas capaces de establecer diálogos interactivos y generar informes relacionales, etiquetas, tablas Browse y todo tipo de aplicaciones por muy complejas que sean.

En Fox-Pro se dedica especial atención al manejo de objetos (partes de una pantalla que permiten ejecutar ciertas tareas), que son de gran utilidad para crear sofisticadas entradas de datos que controlan la pantalla con herramientas muy eficaces como son los radios botones, botones invisibles, etc.

Otra característica muy importante de Fox-Pro, es que nos permite manipular los datos de un archivo de texto que ha sido creado en otro lenguaje de programación.

También se ha trabajado con el lenguaje de programación PASCAL, debido a que nuestra aplicación requiere de manipular datos a bajo nivel (salida y entrada a los puertos), y sólo este lenguaje proporciona las herramientas necesario para hacerlo. Por ejemplo, manejar los registros del microprocesador, para poder enviar y traer datos del puerto serie, es una de las ventajas que éste lenguaje de alto nivel posee.

1.8.2 Descripción del Software

Para la elaboración del software administrador de horas en centro de cómputo, se han tomado en cuenta las siguientes opciones: (Ver Fig S1)

- Usar tarjeta
- Ver estado general
- Administrar Bases
- Formatear tarjeta.

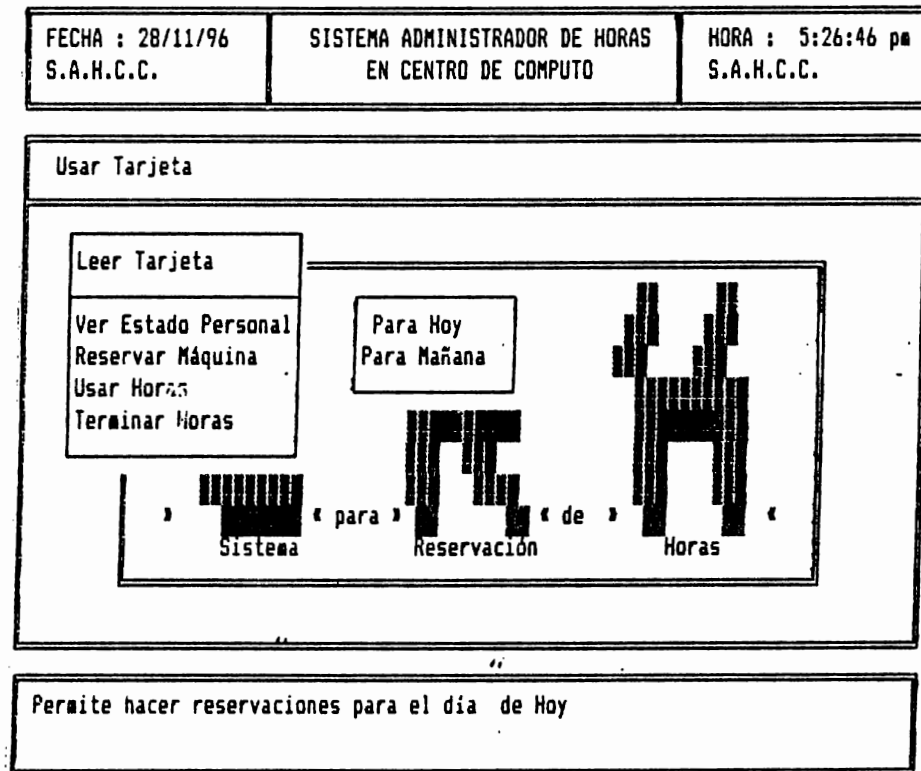


Fig S1: Pantalla de menú principal

Usar tarjeta:

Dentro de esta opción se encuentra: Leer Tarjeta, Ver estado Personal, Reservar Máquina, Usar Horas y Terminar horas (Ver Fig S1). Si selecciona leer tarjeta (Ver Anexo C), el programa llamará una rutina elaborada en Pascal que permite verificar si realmente existe la tarjeta en el lector/escritor; si existe, los datos escritos en la tarjeta son leídos, dentro de estos datos tenemos: carnet, horas disponibles, hora1, hora2, máquina1, máquina2, multas y fecha de uso; de tal forma que esos datos sean almacenados en una

variable como cadena de caracteres y queden disponibles para ser manipulados después por otras aplicaciones que el usuario realizará. Luego de leídos los datos el programa habilita las demás opciones, para que el usuario pueda hacer uso adecuado de ellas.

```

----- ESTADO PERSONAL -----
CARNET : 94-04002          MULTA : ¢ 10          5:21:09 pm
APELLIDO : Alvarez Martinez
NOMBRE : Dora Alicia
HORAS DISPONIBLES : 242

Reservación para hoy
Horas Reservadas : 2
Primer Hora : 17
Máquina 1 : 1
Segunda Hora : 18
Máquina 2 : 1
Fecha de uso : 28/11/96

Reservación para mañana
Horas Reservadas : 0
Primer Hora : 0
Máquina 1 : 0
Segunda Hora : 0
Máquina 2 : 0
Fecha de uso : Mañana

Presione Cualquier Tecla Para Retornar

```

Fig. S2: Pantalla de ESTADO PERSONAL del alumno

Una vez hecho lo anterior, podemos seleccionar la opción de ver estado personal , la cual tiene como función la de presentar los datos leídos de la tarjeta, tales como: carnet, horas disponibles, reservaciones, fecha de uso y multa; el carnet es buscado en una base de datos, donde se encuentran registrados todos los alumnos que pueden hacer uso del centro de cómputo; si existe, se presentan los apellidos y nombres. Con estos datos como base el programa verifica si el usuario tiene reservaciones hechas para el día de hoy o mañana; en las cuales se presentan dos cajas, las cuales indican la cantidad de horas reservadas, primer hora, segunda hora, máquina 1, máquina2 y la fecha de uso. (ver figura S2).

En la opción de reservar máquina, aparecerá un pequeño menú, indicando si la reservación será para hoy o mañana (Ver Fig S1), si selecciona cualquiera de las dos, el programa permitirá reservar solamente dos horas si el usuario tiene horas disponibles, si se han reservado 2 horas, no se podrá reservar. Si solamente se reservó una hora el programa permite reservar la otra hora; dentro de esto aparecerá una distribución física de las máquinas con las horas, luego pedirá la cantidad de horas a reservar; si el dato es correcto, se podrá posicionar en la hora que se desea reservar (Ver Fig S3). Una vez reservada las horas, se genera un informe donde se presenta el carnet, nombre, apellido y las horas que se han reservado (te indica el número de la máquina asignada).

DISTRIBUCION DE MAQUINAS Y HORAS														
■ = Indica Hora Reservada † = Indica Hora Disponible										HORA : 5:34:42 pm FECHA: 28/11/96				
MAÑANA							TARDE							
Máquina	7 am	8 am	9 am	10am	11am	12m	1 pm	2 pm	3 pm	4 pm	5 pm	6 pm	7 pm	8 pm
1	†	†	†	†	†	†	†	†	†	†	■	■	†	†
2	†	†	†	†	†	†	†	†	†	†	■	■	†	†
3	†	†	†	†	†	†	†	†	†	†	†	†	†	†
4	†	†	†	†	†	†	†	†	†	†	†	†	†	†
5	†	†	†	†	†	†	†	†	†	†	†	†	†	†
6	†	†	†	†	†	†	†	†	†	†	†	†	†	†
7	†	†	†	†	†	†	†	†	†	†	†	†	†	†
8	†	†	†	†	†	†	†	†	†	†	†	†	†	†
9	†	†	†	†	†	†	†	†	†	†	†	†	†	†
10	†	†	†	†	†	†	†	†	†	†	†	†	†	†
11	†	†	†	†	†	†	†	†	†	†	†	†	†	†
12	†	†	†	†	†	†	†	†	†	†	†	†	†	†
13	†	†	†	†	†	†	†	†	†	†	†	†	†	†

AREA DE MENSAJES
 Presione Cualquier Tecla para Retornar

Fig. S3: Pantalla de Reservación de horas.

Cuando se selecciona la opción de usar horas, el programa compara la hora reservada con la hora del sistema, si concuerdan se marca la entrada; y a partir de allí empiezan a contar las horas.

La última opción de terminar horas, finaliza las horas si se ha marcado la entrada, si el usuario se pasa varias horas sin marcar la salida, el sistema le descontará dichas horas de las horas disponibles, si el usuario no cuenta con horas, el sistema pondrá una multa que equivale a \$10.00 por cada hora pasada.

Ver Estado General:

Esta opción esta estructura de la siguiente manera: Reporte General de alumnos, Personas con multas, ver reservaciones de hoy y ver reservaciones de mañana. (Ver Fig S4)

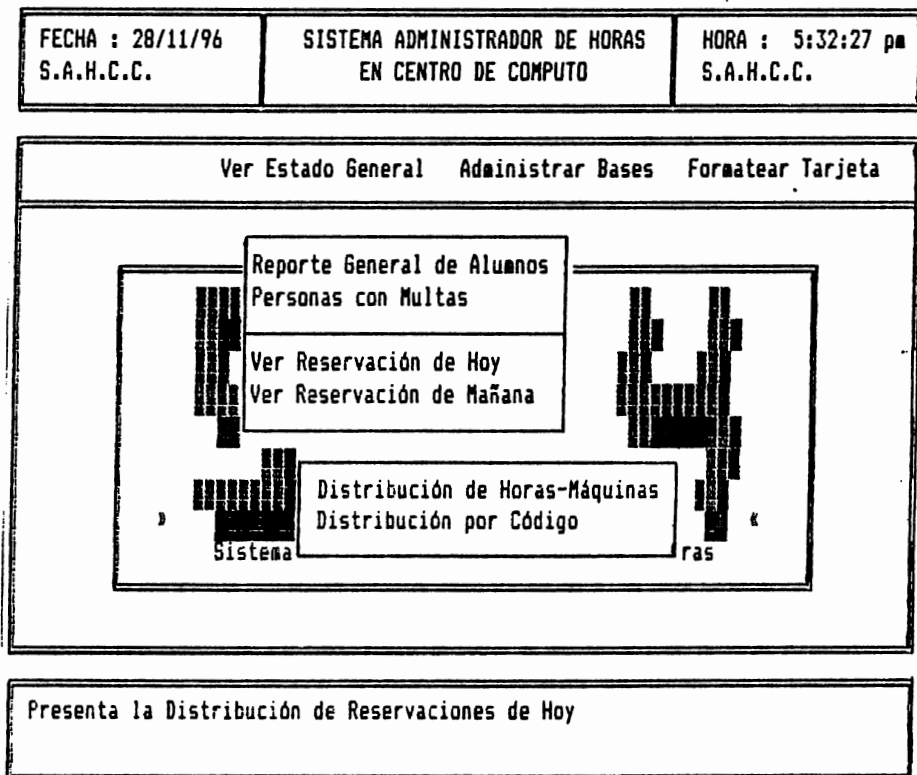


Fig. S4: Pantalla de Estado General.

Una característica principal del sistema, es que el menú presentado anteriormente sólo puede ser accedido por la persona encargada del centro de cómputo, ya que a través de esta puede verificar el estado de las reservaciones hechas por los usuarios.

Si se selecciona la opción de Reporte General de Alumnos, aparecerá un listado de todos los alumnos que pueden hacer uso del centro de computo, es decir alumnos de cualquier carrera que tienen en sus materias prácticas de laboratorio. En dicho listado aparecerá el carnet, apellidos, nombre y multas. (Ver Fig S5)

28/11/96		REPORTE GENERAL DE ALUMNOS		5:30:08 pm	
LISTA DE ALUMNOS					
Codigo	Apellidos	Nombres	Horas_dis	Multas	
94-04002	Alvarez Martinez	Dora Alicia	240	¢	10
94-04003	Aquino	Alejandro de Paul	250	¢	0
94-04004	Cerros Hernandez	Merlin Edibel	250	¢	0
94-04005	Cordova de la O	Gissela Guadalupe	0	¢	0
94-04007	Elias Hernandez	Jaime Yasir	250	¢	60
94-04008	Flores Piche	Karla Marisol	240	¢	0
94-04009	Flores Nolasco	Sandra Elizabeth	200	¢	0
94-04010	García Cortez	Maria Santos Patrici	250	¢	0
94-04011	Gochez Juarez	Maria de los Angeles	300	¢	0
94-04012	Gomez Acosta	Elida Luz	250	¢	0
94-04013	Henriquez Ferman	Carlos Heriberto	250	¢	0
94-04014	Hernandez Rivera	Lorena Veralice	250	¢	70

<Imprimir> <Ver Reporte> <Retornar>

Fig S5 : Pantalla de Reporte General.

La opción de personas con multas, aparecerá un listado similar al anterior, con la diferencia que solo aparecerán aquellas personas que han tenido recargos; o sea personas que de alguna manera ocuparon más horas que las reservadas sin tener disponible ninguna. (Ver Fig S6)

28/11/96		REPORTE DE ALUMNOS CON MULTAS		5:31:12 pm	
LISTA DE ALUMNOS					
Codigo	Apellidos	Nombres	Horas_dis	Multas	
94-04002	Alvarez Martinez	Dora Alicia	240	€ 10	
94-04007	Elias Hernandez	Jaime Yasir	250	€ 60	
94-04014	Hernandez Rivera	Lorena Veralice	250	€ 70	
94-34032	Alvarenga Rivas	Roberto Carlos	0	€ 50	

Fig S6: Pantalla de personas con multas.

Las opciones de Ver reservaciones de hoy y mañana presentan dos opciones:

- Distribución de Horas-Máquinas
- Distribución por Código.

En la distribución de horas-máquinas, aparecerá la distribución gráfica de las Máquinas vrs horas, de tal manera que pueda tener una panorámica de como esta siendo ocupado el centro de cómputo en ese momento. (Ver Fig S3)

En la distribución por código, aparece un reporte de todas las personas que han reservado ya sea para hoy o para mañana. En dicho reporte aparecerá el carnet, apellidos, nombres, horas reservadas, hora 1, hora 2, máquina 1, máquina 2. Al final del reporte se muestra la cantidad de alumnos que han reservado. (Ver Fig S7)

28/11/96		PERSONAS QUE HAN RESERVADO EL DIA DE HOY		5:35:54 pm		
RESERVACIONES						
Codigo	Nombres	Apellidos	Horas_res	hora1	Hora2	Mi
9404002	Dora Alicia	Alvarez Martinez	2	17	18	
9404008	Karla Marisol	Flores Piche	2	17	18	
Cantidad de Alumnos 2						

Fig 97: Pantalla de Personas que han reservado.

Administrar bases:

Aparecen dos opciones:

- ▣ Mantenimiento de Alumnos
- ▣ Actualizar Reservas.

Mantenimiento de alumnos, nos sirve para darle mantenimiento adecuado a la base de datos donde se encuentran todos los alumnos inscritos. Dentro de ésta aparecerán 4 opciones:

- Adicionar
- Consultar
- Modificar
- Eliminar

(Ver Fig. S8).

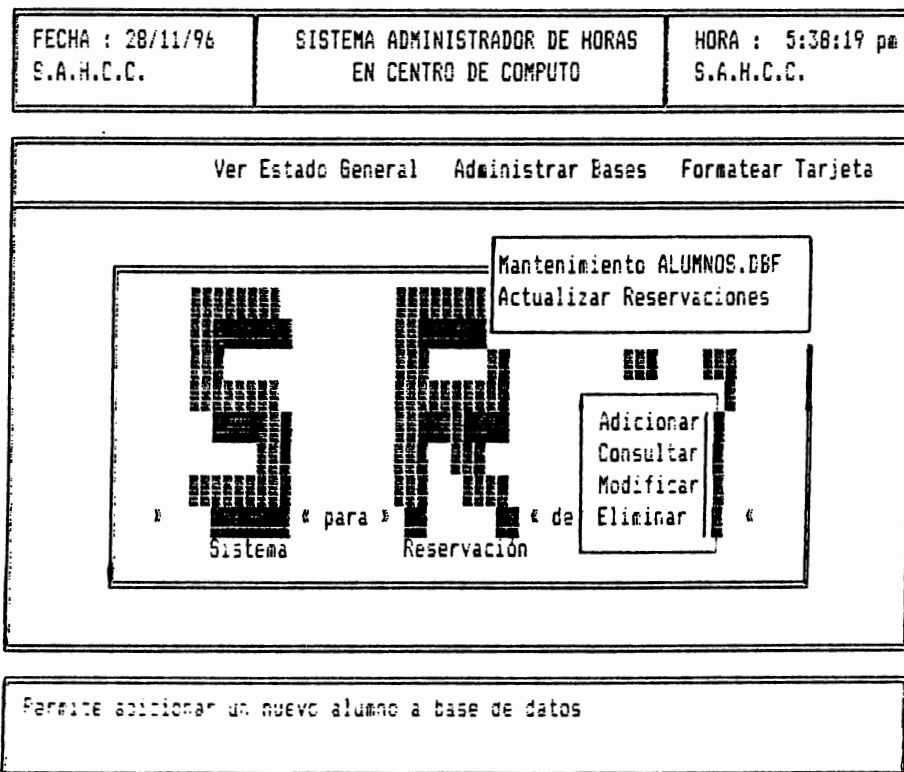


Fig S8: Menú de Mantenimiento.

Adicionar : Nos permite agregar un nuevo alumno a la base de datos; para ello nos pide el carnet, apellidos, nombres y horas disponibles, si el alumno ya existe, el sistema rechazará los datos de entrada. (Ver Fig S9)

FECHA : 28/11/96 5:39:24 pm

ADICION DE DATOS DEL ALUMNO

CARNET : 94-04100
APELLIDOS : Enriquez Torres
NOMBRES : Mario Antonio
HORAS DISPONIBLES : 250 hrs.

Mensajes
<Adicionar> <Corregir> <Otro >

Cantidad Actual de Alumnos : 33

Fig S9 : Pantalla de adición de datos del alumno.

Consultar : Consulta un determinado alumno, para ello nos pide el carnet de la persona que se desea consultar, si existe se presentarán los apellidos, nombres, horas disponibles y multa. (Ver Fig S10)

FECHA : 28/11/96 5:48:47 pm

CONSULTA DE DATOS DEL ALUMNO

CARNET : 94-34032	MULTAS : ¢ 50.00
APELLIDOS : Alvarenga Rivas	
NOMBRES : Roberto Carlos	
HORAS DISPONIBLES : 0 hrs.	

Mensajes

<Otro > <Salir >

Cantidad Actual de Registros : 03

Fig S10: Pantalla de Consulta de datos del alumno.

Modificar : Permite modificar los datos de un determinado alumno; para lo cual pide el carnet del alumno, si existe presentará los apellidos, nombres, horas disponibles y multas, luego permitirá modificar cualquiera de ellos y almacenarlos nuevamente a la base de datos. (Ver Fig S11)

FECHA : 29/11/96

5:41:53 pm

MODIFICAR DATOS DEL ALUMNO	
CARNET : 94-34032 MULTAS : ¢ 50.00	
APELLIDOS : Aivarenga Rivas	
NOMBRES : Roberto Carlos	
HORAS DISPONIBLES : 200 hrs.	
----- Mensajes -----	
<Modificar Otravez> <Cancelar>	

Cantidad Actual de Alumnos : 33

Fig S11: Pantalla de Modificación de datos del alumno.

Eliminar : Permite eliminar un determinado alumno, introduciéndole el carnet, es buscado en la base de datos, si existe es presentado, y luego se pide si desea ser eliminado; si se confirma la eliminación, desaparecerá físicamente de la base de datos. (Ver Fig S12)

FECHA : 28/11/96 5:43:07 pm

ELIMINACION DE DATOS DEL ALUMNO

CARNET : 94-04005	MULTAS : ¢ 0.00
APELLIDOS : Cordova de la O	
NOMBRES : Gissela Guadalupe	
HORAS DISPONIBLES : 0 hrs.	

Mensajes

<Eliminar> <Cancelar>

Cantidad Actual de Alumnos 37

Fig S12: Pantalla de eliminación de datos del alumno.

En la opción de actualizar reservaciones, se actualizan todas la reservaciones hechas del día de hoy con las del día de mañana, en la cual presentará las bases de datos que se está actualizando. Este proceso de actualización consiste en pasar todos los datos (reservaciones) de la base de datos BASE2.DBF a BASE1.DBF, que son las distribuciones gráficas de máquinas-horas; y pasar los datos de BASEMAÑ.DBF a BASEHOY.DBF, que son las distribuciones por código. (Ver Fig 13)

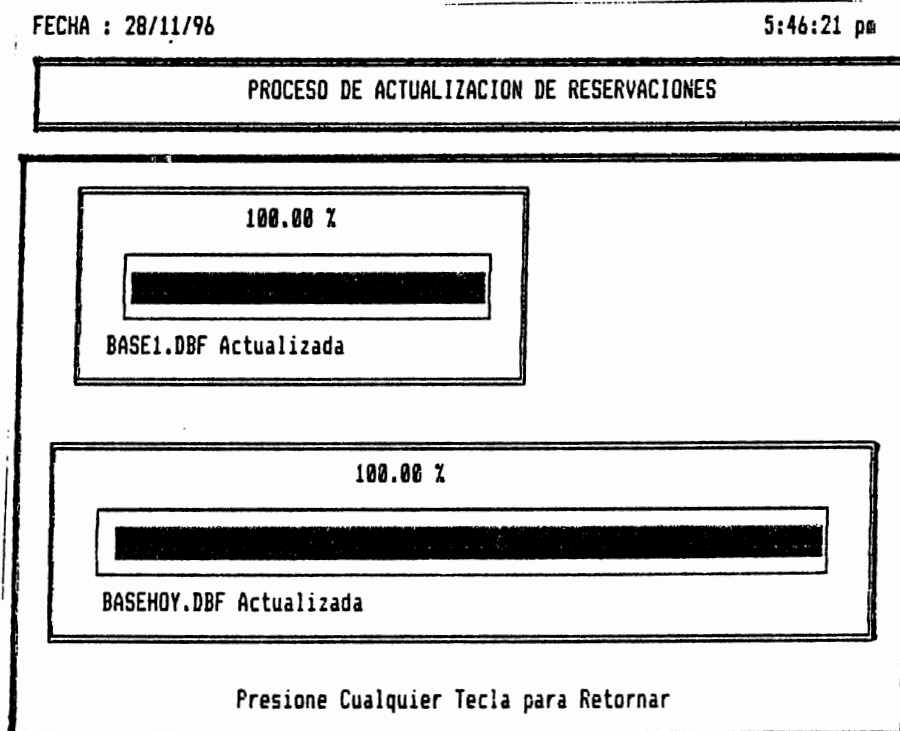


Fig. S13: Pantalla de actualización de Reservaciones

Formatear tarjeta:

En ésta opción se llama una rutina hecha en pascal, que tiene como función la de enviar los datos (carnet, horas disponibles, horal, hora2, maquina1, maquina2, fecha de uso y multa) al puerto serie para que el lector/escritor los escriba en la cinta, la cual permite darle formato inicial a la tarjeta que el usuario poseerá. Esta rutina tiene la función de escribir todos los datos que al sistema le servirán para identificar al usuario, a la hora que haga sus reservaciones. (Ver fig S14)

FECHA : 28/11/96	FORMATEO DE TARJETA	5:44:39 pm
------------------	---------------------	------------

CARNET : 94-04015		MULTAS : ¢ 0.00
APELLIDOS : Jovel Melendez		
NOMBRES : Erlinda Yessenia		
HORAS DISPONIBLES : 200 hrs.		

Mensajes		
<Formatear>	<Otro >	<Salir>

Fig. S14: Pantalla Para Formatear Tarjeta

1.H HARDWARE

1.H.1 Descripción del Sistema Mecánico.

El presente prototipo no ha sido diseñado basandose en los sistemas lectores/escritores ya existentes en el mercado. Por tal razón se hace necesario describir el sistema mecanico diseñado especialmente para este proyecto.

La descripción y criterios tomados en consideración para su fabricación fueron:

- Uso de aluminio como materia prima en la construcción, por ser blando y ligero.
- Los cabezales se desplazan estando la tarjeta estática, debido a que se requería rapidez constante porque el manejo del flujo de datos es asincrónico.

Se diseñó un mecanismo de sustentación de la tarjeta de altura ajustable, esto se realizó usando tornillos sin fin. La razón de lo anterior se debe a que en el proceso de lectura/escritura sobre la tarjeta se necesita evitar los siguientes errores: Azimut, Altura, Angulo, Tangencia y contacto.(Ver tema "Alineación de las cabezas magnéticas", anexo).

- Se utiliza un motor paso a paso por la exactitud que se requiere en el movimiento de los cabezales y la facilidad que presenta de invertir el giro.
- El tipo de faja usada es dentada ya que así se evita el deslizamiento al girar el motor.
- Para evitar que la faja soportara todo el peso de los cabezales fue necesario construir un riel de guía de desplazamiento, tomando en cuenta que es una barra rectangular que no da lugar a "juego mecánico".

A continuación se presentan los dibujos a escala de la estructura mecánica ocupada en el proyecto.

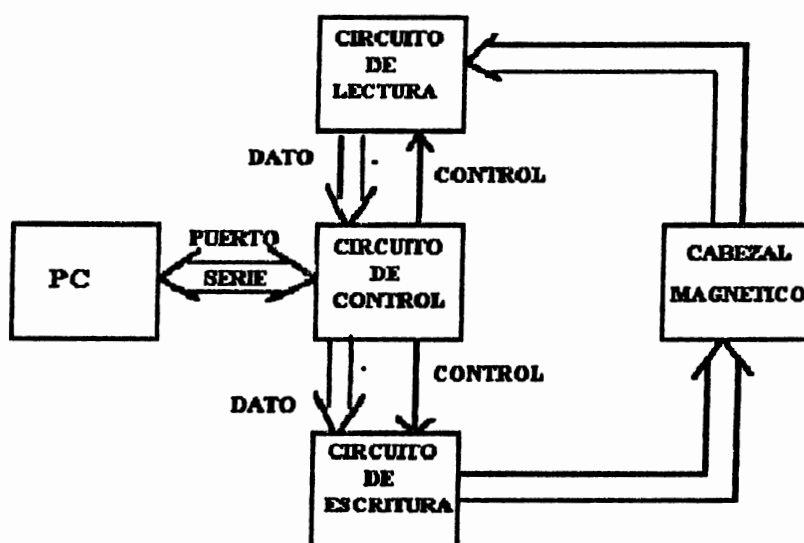
1.H.2 Descripción del Hardware

Diagrama a Bloques

El presente prototipo es una idea original que no ha tomado en consideración los diseños de los lectores escritores comerciales. Los criterios y diseños son propios, siendo el campo de aplicación del prototipo un centro de computo que necesita un sistema de reservación de horas.

En el proyecto se utilizan dos sectores o pistas de la cinta magnética de una tarjeta, esto como una demostración de la capacidad de almacenamiento de datos que se puede tener en dicho dispositivo.

DIAGRAMA BLOQUE



Se sabe que toda computadora personal contiene una tarjeta manejadora de puerto serie (primer bloque del diagrama). A través de este puerto el prototipo diseñado recibe y envía los datos desde/hacia la tarjeta magnética. El propósito de utilizar una computadora personal es la gran capacidad de manejo de datos y la flexibilidad desde el punto de vista de software.

El bloque de control es el encargado de coordinar las operaciones de escritura lectura, movimiento de motor y censo de presencia de tarjeta. Este bloque está compuesto totalmente de circuitería digital y posee dos tipos de líneas:

- Línea de datos.
- Línea de control o de mando, en base a su función en secuencia de eventos, necesitando de la computadora un pulso por el puerto serie como start. Todo el resto de la operación se controla internamente.

El bloque lector se encarga de recibir los datos de la tarjeta a través del cabezal magnético. Estos datos son convertidos de señales analógicas a señales digitales y luego transmitidas por el bloque de control hacia la computadora. El centro de este circuito es un oscilador controlado por voltaje (VCO) que es el responsable directo de la FSK.

El bloque de escritura recibe las señales procedentes de la computadora (previamente capturados por el bloque de control),

transformandolas en señales analógicas que son colocadas en el cabezal magnético y luego grabadas a la tarjeta. El centro de este bloque es un PLL que genera una variación de voltaje dc conforme la frecuencia cambia. A través de este bloque se realiza una demodulación FSK.

CIRCUITO DE CONTROL

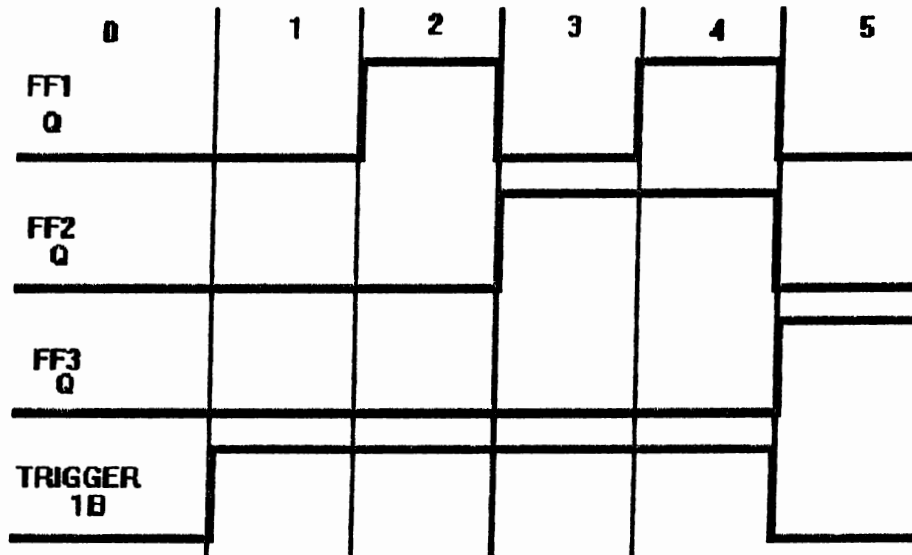
Este circuito tiene su centro en el monoestable 1B que es el encargado de la ventana de trabajo y en tres flip-flops que en conjunto realizan las siguientes funciones:

- Hacer el cambio del sentido de giro del motor.
- Habilitar los interruptores electrónicos de lectura o escritura según la función que se esté haciendo.
- Hacer el cambio de pista.
- Hacer el cambio de modo de operación.

Los demás dispositivos se utilizan para el control de velocidad del motor, reinicio de este y habilitación de los einterruptores analógicos.

La siguiente explicación se basa en el diagrama de tiempos presentado a continuación.

DIAGRAMA DE TIEMPOS



TIEMPO 0: El circuito esta en modo de lectura y el motor no se mueve debido a que no estan habilitados el contador y demux que lo controlan. El circuito espera una transición negativa desde el puerto serie (start) para el inicio de su operación.

TIEMPO 1: El monoestable recibe una transición desde el puerto serie y se abre la ventana de trabajo. Al activarse este pulso tanto el demux (74156) como el contador up/down (74191) son habilitados. El arreglo interno and/or del 74H50 esta utilizado como un mux de dos entradas y una salida. Al estar Q en 1 del FF1 el arreglo and/or deja pasar la frecuencia del 555 dividida entre 2 (200Hz) proveniente de FF4.

El contador tiene un cero en su entrada U/D por lo que la cuenta es ascendente. Esto hace que el demux de un desplazamiento de bits correspondiente al avance del motor.

TIEMPO 2: Este tiempo inicia cuando ya los cabezales recorrieron completamente una vez la tarjeta y llegaron hasta el microinterruptor de fin de carrera. Este evento habilita el conteo descendente del 74191 y la and/or deja pasar la frecuencia de 400Hz directamente del 555. Este 1 en el Q del FF1 deshabilita todas las compuertas ands (7411) por lo que ningun interruptor electrónico (4066) esta habilitado.

TIEMPO 3: La pieza que soporta los cabezales activa el microinterruptor del inicio de carrera por lo que el Q del FF1 pasa a ser 0. Esta transición negativa pone en 1 el Q del FF2 por lo que se activa la and correspondiente a la lectura de la pista 2. De nuevo la and/or deja pasar la frecuencia de 200Hz y el conteo es ascendente por lo que el movimiento es en el mismo sentido del tiempo 1.

TIEMPO 4: Al llegar al microinterruptor de fin de carrera el FF1 coloca Q=1 con lo que se deshabilitan todos los interruptores electrónicos. El contador va en sentido descendente y el motor regresa a su posición originl. El FF3 tiene Q=0 por lo que el

sistema esta todavía en modo de lectura. La ventana aun permanece en alto.

TIEMPO 5: Inicia cuando los cabezales llegan a su posición original. En este instante debe caer la ventana, lo cual hace critica su duración exacta. De no ser así el motor seguirá moviendose, siendo lo correcto el hecho de que se quede en su posición original. La salida Q del FF3 pasa a 1 por lo que se habilitarán sólo las ands que controlan los interruptores de escritura (cambio de modo). El FF1 y FF2 regresan a los estados que tenían en el tiempo cero y de nuevo la interfase espera un pulso de start del puerto serie para inicial la siguiente operación.

En el instante en que cae la ventana un arreglo de compuertas dispara el monoestable 1A cuya función es limpiar las salidas del contador para hacer que el motor quede con las bobinas apagadas.

Si la tarjeta no estuviese presente el monoestable 1B no esta habilitado y aunque llegue un pulso de start el motor no se moverá puesto que no habrá ventana.

La red RC en el clear de los flip-flop hace que todos los Q se inicien en 0. Por lo que el estado original siempre será el mismo al encender el equipo. Se incluye un interruptor (WRITE/READ) para hacer un cambio de modo manual.

CIRCUITO DE LECTURA

La extracción de la señal desde el cabezal presenta la dificultad de que el voltaje inducido es de 1mVp-p , por lo que es necesario amplificarlo grandemente, además de filtrar las señales de alta frecuencia (ruido). En el presente prototipo fue sumamente problemático el diseño de este amplificador, pues se tenía la dificultad de oscilaciones por la alta ganancia y el ruido. Es por ello que en el diagrama esquemático se presenta como un bloque (Ver sugerencias en sección de diseño).

Después de que la señal ha sido amplificada a unos 200mVpp se alimenta el circuito PLL con ella. El LM565 (PLL) presenta la característica de producir un voltaje DC en el pin 7 que varia linealmente en el rango de cierre con la frecuencia. Al estar la frecuencia de entrada arriba de la central, el pin 7 reduce su voltaje con respecto al pin 6 (voltaje de referencia) y viceversa. Es así como se puede sintetizar la función del PLL en que es un demodulador FSK que produce un voltaje a la salida correspondiente a la frecuencia de entrada. Este voltaje DC es el resultado de un filtro pasobajas en el pin 7, para botar las componentes de suma de frecuencia.

El pin 6 presenta un voltaje esencialmente constante a la frecuencia y el pin 7 varia linealmente e inversamente con ella.

La salida a decodificar es :

FRECUENCIA DE ENTRADA	NIVEL LOGICO
2300Hz	0
1700Hz	1

Así, cuando se presenta una frecuencia de 1700Hz el pin 7 tiene un voltaje mayor que el pin 6, por lo que el comparador tiene la entrada no inversora mayor que la inversora. Esto abre el colector abierto de la salida del LM311 y pone en el pin 7 de esté el nivel de la resistencia de pull-up, para el caso de +5Vdc. Al estar presente la frecuencia de 2300Hz la terminal inversora es menor que la inversora por lo que la salida va a tierra.

Al estar trabajando con FSK a 110 baudios, las frecuencia cambian en el tiempo rapidamente por lo que el pin 7 (LM565) presenta ciertos amortiguamiento que hacen que la salida tenga algunos transitorios en el pulso en alto. esto no es deseable pues la computadora puede recibir un dato erroneo. Para evitar esto se colocó una red RC con un τ (tao) equivalente al tiempo de duración del transitorio. Es así como el arreglo RC, con la descarga del capacitor, cubre la caída temporal del pulso dando un 1. En transmisión muy lenta de datos (un dato por segundo) este problema no esta presente (para mayor detalle vea la sección de diseño).

La salida de este arreglo RC y de las salidas directas del comparador se suman en una OR, para dar un valor reconstruido de la señal.

Esta señal se pasa al convertidor 75188 para que sea leída en el puerto serie.

CIRCUITO DE ESCRITURA

Una vez que el dato en formato RS-232C es convertido a TTL por el 75189, este 1 o 0 lógico se utiliza para controlar la frecuencia del VCO. Debido a que el LM566 (VCO), no puede tener en su pin 5 voltajes menores que $3/4$ de VCC (con un máximo de VCC), es necesario construir un circuito para hacer compatibles los ceros y unos lógicos a esta restricción. Esto se logra por medio de un sumado directo, en el que se commuta un transistor PNP para detectar la variación de niveles lógicos. Y esto se suma con un voltaje fijo de DC para que la diferencia este en el rango de $[(3/4)VCC] < V_5 < VCC$, como especifican las hojas técnicas. Al haber un 1 lógico el transistor no se satura por lo que el pin 5 del LM566 tiene un valor alto produciendo la frecuencia de 1700Hz. Al ser un cero lógico el transistor se satura, el pin 5 baja de voltaje y la frecuencia aumenta a 2300Hz.

La salida del VCO se toma de la señal triangular (que tiene superpuesto un nivel de DC), y se pasa a un integrado para cambiarla a una onda semejante a la senoidal. El integrador debe

compensarse por medio de una resistencia en paralelo con el capacitor de retroalimentación.

A la salida del integrador se incluye un atenuador para reducir la señal a 200mVpp, que es el nivel con el que se alimenta a la cabeza grabadora.

Debido al efecto de remanencia magnética es necesario agregar a la señal del atenuador un nivel DC para evitar la inversión del campo magnético que distorsione la señal. Este método se denomina "POLARIZACION CD", siendo el voltaje offset óptimo de 100mV. Esta suma se logra por medio de un amp-op conectado como sumador directo.

La salida de este sumador se alimenta a los interruptores electrónicos para grabar en la pista correcta. Antes de escribir es necesario borrar la información previa por lo que se utiliza un 555 conectado en astable para generar una señal de 13kHz. Esta señal se atenúa por medio de una resistencia de 680K para que tenga un nivel de 200mVpp, dando un "achataamiento" de las aristas de la onda cuadrada (pin 3, LM555) a causa de la red RL formada por el cabezal y la resistencia antes mencionada.

ASIGNACION Y DESCRIPCION DE PINES DEL PUERTO SERIE DB25

Pin	Function
1	NC
2	Transmit Data
3	Receive Data
4	Request to Send
5	Clear to Send
6	Data Set Ready
7	Signal Ground

8	Data Carrier Detect
9	NC
10	NC
11	NC
12	NC
13	NC
14	NC
15	NC
16	NC
17	NC
18	NC
19	NC
20	Data Terminal Ready
21	NC
22	Ring Indicator
23	NC
24	NC
25	NC

Algunas de las líneas NC se utilizaron para obtener las alimentaciones de la fuente de la computadora hacia la interfase.

En la descripción se menciona un computador como dispositivo transmisor (DTE) y un MODEM como DCE.

- PIN 1: Protección de tierra, debe de conectarse al chasis del MODEM y del computador con la tierra del circuito.
- PIN 2: Transmisión de datos, va de la computadora o terminal hasta el modem. Este PIN es la entrada para el PIN llamado transmit en el modem.
- PIN 3: Receptor de datos, lleva los datos a la computadora desde el modem y es un pin de salida para este último.

- PIN 4: RTS (Request to send, solicitud de envío), es una entrada para el MODEM que debe estar en un cero lógico para habilitar al modem a transmitir datos.
- PIN 5:CTS (clear to send, limpiar para enviar), es una señal desde el modem que indica que éste está en la condición de transmisor. Un cero lógico indica que el modem esta listo para transmitir.
- PIN 6: DSR(Data Set Ready, Listo para colocar dato), indica que el modem está conectado al canal de comunicaciones y que está listo para transmitir o recibir datos cuando un cero lógico esté presente.
- PIN 7: Tierra del circuito.
- PIN 9-19,21,23-25: No tienen conexión.
- PIN 20: DTR (Data Terminal Ready, terminal de datos lista), es una entrada para el modem que éste usa para conectarse o desconectarse a si mismo desde el canal de comunicaciones. Si es un "0" lógico la comunicación esta permitida, si es un "1" lógico el modem está desconectado.
- PIN 22: Ring, indica que el modem está recibiendo una señal de tono de la línea telefónica. Este habilita automáticamente la circuitería de contestado en algunos modems.

CAPITULO II

DISEÑO DEL PROYECTO

2.3 DISEÑO DEL SOFTWARE

2.3.1 Metodología Utilizada

El software desarrollado, cubre una serie de procedimientos y funciones bastante complejas; ya que el hecho de administrar resulta bastante complicado, más si se trata de organizar y distribuir máquinas en determinadas horas, dependiendo de lo que el usuario desea.

El Gestor de Base de Datos Fox-Pro nos permite desarrollar de mejor manera esta aplicación, ya que la forma de tratar los datos es bastante fácil y dinámica.

Para este tipo de aplicación se requiere del lenguaje de programación Pascal; ya que se necesita manejar datos a bajo nivel para poder interactuar con el puerto serie.

2.9.1.1 Estandares de Documentación

La función de los estandares de documentación, es presentar la forma uniforme y consistente en la que se han codificado los programas y diseñado pantallas. Esto se presenta de esta manera para que el usuario pueda entender cual ha sido la lógica con la que se desarrollaron los programas.

2.9.1.2 Programación

Para la elaboración de los programas, se tomó en cuenta la programación estructurada (Técnica Top-Down). A continuación se presenta de una manera más definida este tipo de programación utilizada.

2.9.1.3 Instrucciones Programas

La codificación del programa está organizada de tal forma que las instrucciones de mayor precedencia se encuentran dos o tres columnas antes de ejecutar la siguiente; es decir, consideremos un bucle como ejemplo, de la siguiente manera:

```
DO' WHILE condicion
    instrucciones
ENDDO
```

Considerando una instrucción de decisión:

```
IF found()
    instruccion1
        instrucción 2
ELSE
    instrucción 1
        instrucción 2
ENDIF
```

Como podemos observar en los ejemplos anteriores las instrucciones que se van a ejecutar dentro de un lazo o una decisión se encuentran alineadas tres columnas después de la instrucción principal.

2.8.1.4 Comentarios

Los comentarios son parte de la documentación de todo programa, ya que nos ayuda a entenderlo de mejor manera. Los comentarios son colocados de la siguiente manera:

* comentario *

pueden ser colocados en cualquier parte del programa.

También se utilizan comentarios al final de alguna instrucción, para lograr esto se coloca && comentario.

ejemplo:

```
@12,30 SAY "Nombre:" GET nombre && Introducción del nombre.
```

2.9.1.5 Uso de letras Mayúsculas/Minúsculas

Se utilizaron letras mayúsculas en las palabras reservadas de Fox-Pro; y letras minúsculas utilizados para variables, comentarios, campos. La combinación de este tipo de letras contribuye a distinguir o diferenciar lo que es una variable de una instrucción, ya que de esta manera el programa toma una buena estética, y permite una mayor comprensión de lo que hace.

Ejemplo:

```
IF opcion=1      && adición de datos
  APPEND BLANK
  REPLACE nombre WITH vnombre
  REPLACE codigo WITH vcodigo
  .....
ENDIF
```

2.9.1.6 Modularidad

Para la elaboración de todo el sistema, se crearon rutinas de forma independiente. Cuando se diseñó la opción de "Usar tarjeta", se particionó el problema en 5 partes: Leer Tarjeta, Ver estado Personal, Reservar Máquina, Usar horas y Terminar horas. A partir de aquí se creó otra rutina que las enlazara y llamara. De la misma manera se hizo para las otras opciones del sistema (Ver estado general, Administrar Bases y Formatear Tarjeta).

En resumen modular un programa significa crear subrutinas conocidas como: procedimientos y funciones, que son definidas en lugares especiales dentro de los programas, y son llamados dentro de un programa principal. Por ejemplo el sistema que administra horas en centro de computo posee algunas las siguientes subrutinas:

- Captura de datos
 - Reservar horas
 - Reporte General
 - Formato de pantallas
 - Rutinas para enviar mensajes.
- ... etc.

Todas estas subrutinas son controladas dentro de un programa principal, que se encarga de llamarlas de acuerdo a la elección del usuario.

Entrando en la programación estructurada, podemos observar que la modularidad es una característica esencial en el desarrollo de programas eficientes.

El ambiente de Fox-Pro y Pascal; permite modular un programa. Las ventajas que produce la modularidad son las siguientes:

- Ayuda al entendimiento de la función de un programa.
- Ayuda a corregir errores.
- Facilita la depuración de un programa.
- Hace más eficiente un programa.
- Facilidad de Codificación.

2.8.1.7 Convención de Nombres

Los nombres de los programas, reportes, tablas, campos, funciones, procedimientos y variables están identificados por palabras nemónicas para dar una idea de su función. Por ejemplos la base de datos donde se encuentran registrados los alumnos que haran uso de sus horas, se llama DATOSPER.DBF; esto nos da la idea que son datos personales.

2.8.1.8 Programación de teclas

Se ha utilizado esta técnica para desactivar todas las teclas que el usuario no utilizará, y dejar activa solo aquellas teclas que al usuario le servirán para manejar el sistema y entrar datos al sistema. Se hizo uso de esta herramienta para evitar que el usuario se salga del ambiente en el que trabaja.

2.8.2 Diseño de Base de Datos

Se han diseñado 5 base de datos, las cuales están en constante relación con el sistema. Fox-Pro nos proporciona una herramienta necesaria para crear base de datos, la cual es: CREATE <nombre de base de datos>.

La primera base de datos tiene el nombre de DATOSPER.DBF, en esta base se encuentran almacenados todos los alumnos que pueden hacer uso del centro de cómputo. La estructura de esta base de datos es la siguiente :

□DATOSPER.DBF

CAMPO	NOMBRE DEL CAMPO	TIPO	ANCHO
1	CODIGO	NUMERICO	7
2	NOMBRES	CARACTER	20
3	APELLIDOS	CARACTER	20
4	HORAS DIS	NUMERICO	3
5	MULTAS	NUMERICO	10

En el campo CODIGO se almacena el carnet del alumno, en el segundo campo van los nombres, en el tercero los apellidos, en HORAS_DIS se colocan las horas disponibles del alumno; es decir las horas a las que tiene derecho para entrar en el centro de cómputo. En el campo MULTAS se colocan los recargos que el alumno tenga por haber ocupado más de sus horas reservadas sin tener ninguna hora disponible. Los datos de esta base de datos solamente son manipulados por un programa que le da el mantenimiento.

La segunda base de datos BASEHOY.DBF, se almacenan todas las reservaciones hechas para el día de hoy.

BASEHOY.DBF

CAMPO	NOMBRE DEL CAMPO	TIPO	ANCHO
1	CODIGO	NUMERICO	7
2	HORAS_RES	NUMERICO	1
3	HORA1	NUMERICO	2
4	HORA2	NUMERICO	2
5	MAQUINA1	NUMERICO	2
6	MAQUINA2	NUMERICO	2
7	FECHA_USO	FECHA	8
8	MARCA	CARACTER	1

CAMPO 1: CODIGO

Representa el carnet del alumno.

CAMPO 2: HORAS_RES

Representa cantidad de horas reservadas

CAMPO 3: HORA1

Indica la primer hora reservada.

CAMPO 4: HORA2

Indica la segunda hora reservada.

CAMPO 5: MAQUINA1

Representa el número de máquina donde se hará uso de la primer hora.

CAMPO 6: MAQUINA2

Representa el número de máquina donde se hará uso de la segunda hora.

CAMPO 7: FECHA_USO

Representa la fecha en la que se reservaron las horas y la fecha en la que se usarán.

CAMPO 8: MARCA

Este campo tiene una función muy importante dentro del proceso de reservación de horas, ya que através de este se controla cuando el usuario podrá entrar al centro para hacer uso de sus horas, y sirve como marco de referencia para controlar la salida. En este campo queda almacenado una letra "E", indicando que el usuario ha entrado.

Una tercera base de datos es BASEMAÑ.DBF, en la cual se almacenan todas las reservaciones hechas para día de mañana. La estructura de esta base de datos es igual a la estructura de BASEHOY.DBF.

BASEMAÑ.DBF

CAMPO	NOMBRE DEL CAMPO	TIPO	ANCHO
1	CODIGO	NUMERICO	7
2	HORAS_RES	NUMERICO	1
3	HORA1	NUMERICO	2
4	HORA2	NUMERICO	2
5	MAQUINA1	NUMERICO	2
6	MAQUINA2	NUMERICO	2
7	FECHA_USO	FECHA	8
8	MARCA	CARACTER	1

BASE1.DBF Y BASE2.DBF

En estas bases de datos se encuentran almacenados caracteres que indican las reservaciones hechas; representan la distribución gráfico de máquina Vrs horas. En BASE1.DBF se encuentran las reservaciones hechas del día de hoy, y en BASE2.DBF se encuentran las reservaciones del día de mañana.

La utilidad de estas base de datos es que permite interactuar con el usuario, para que la persona que va hacer sus reservaciones pueda darse cuenta como estan distribuidas las máquinas, si hay disponibles en determinadas horas.

La estructura de estas bases de datos es la siguiente :

BASE1.DBF

CAMPO	NOMBRE DEL CAMPO	TIPO	ANCHO
1	SIETEAM	CARACTER	1
2	OCHOAM	CARACTER	1
3	NUEVEAM	CARACTER	1
4	DIEZAM	CARACTER	1
5	ONCEAM	CARACTER	1
6	DOCE	CARACTER	1
7	UNAPM	CARACTER	1
8	DOSPM	CARACTER	1
9	TRESM	CARACTER	1
10	CUATROM	CARACTER	1
11	CINCOPM	CARACTER	1
12	SEISPM	CARACTER	1
13	SIETEPM	CARACTER	1
14	OCHOPM	CARACTER	1

BASE2.DBF

CAMPO	NOMBRE DEL CAMPO	TIPO	ANCHO
1	SIETEAM	CARACTER	1
2	OCHOAM	CARACTER	1
3	NUEVEAM	CARACTER	1
4	DIEZAM	CARACTER	1
5	ONCEAM	CARACTER	1
6	DOCE	CARACTER	1
7	UNAPM	CARACTER	1
8	DOSPM	CARACTER	1
9	TRESM	CARACTER	1
10	CUATROM	CARACTER	1
11	CINCOPM	CARACTER	1
12	SEISPM	CARACTER	1
13	SIETEPM	CARACTER	1
14	OCHOPM	CARACTER	1

La función de cada campo es bastante obvia, ya que como su nombre lo indica, representan las horas en las que el usuario puede reservar una máquina.

En cada campo va un carácter de control que indica que dicha hora está reservada o disponible.

2.5.3 Programas en Pascal para lectura y escritura al puerto serie

(Ver flujogramas en anexo A y B)

Antes de comenzar de establecer el funcionamiento de los programas en Pascal, es necesario conocer el funcionamiento de la interrupción de 14 que es utilizada dentro de ellos.

Funcionamiento de la INT 14 de la BIOS

Función 00h, Inicializar el puerto serie.

Es una interrupción que se utiliza para inicializar la configuración del puerto serie especificado en DX. Se puede utilizar la función 00h para inicializar el puerto serie 1 ó 2 (COM 1 o COM 2) en una computadora personal. En el registro AL, se especifica como la comunicación del puerto podrá ser inicializada. Use los parámetros presentados en la tabla BIOS, que a continuación se presenta:

Tabla BIOS: Parámetros para inicializar puerto serie.

7,6,5 Velocidad	4,3 Paridad	2 Bits de parada	1,0 Longitud de palabra.
000=110 baud	00=none	0=1 bits	10=7 bits
001=150 baud	01=odd	1=2 bits	11=8 bits
010=300 baud	10=none		
011=600 baud	11=even		
100=1200 baud			
101=2400 baud			
110=4800 baud			
111=9600 baud			

Función 01h, Escribir un caracter para puerto de comunicación.

Sacar un caracter al puerto serie

Llamando registros: AH 01h

 AL Caracter a escribir

 DX Puerto de comunicación

 (0=COM1, 1=COM2)

 (2=COM3, 3=COM4)

Retorno de registros: AH bit 7 0 (función exitosa)

 AH bit 7 1 (función falló)

Comentarios: Esta función escribe un caracter y retorna el estado del puerto de comunicación especificado. Ordinariamente, escribir un caracter al puerto serie no es una tarea difícil. Esto puede ser hecho cuando el caracter esté listo. Esta función puede escribir al puerto serie 1 y 2 (COM1 y COM2). Antes de llamar esta función, asegúrese de usar Int 14h, Función 00h para inicializar el puerto.

Función 02h, Leer un caracter del puerto de comunicación.

Leer un caracter del puerto serie

Llamando registros: AH 02h
 DX puerto de comunicación
 (0=COM1, 1=COM2)
 (2=COM3, 3=COM4)

Retorno de registros: AH bit 7 0 (función exitosa)
 AL Caracter
 AH bit 7 1 (función falló)

Comentarios: Esta función lee un caracter del puerto de comunicación especificado, retorna el estado del puerto, y puede leer del puerto serie 1 y 2 (COM1 y COM2).

Para el proceso de escritura y lectura a la cinta magnética se han elaborado dos rutinas independientes en el lenguaje de programación PASCAL. El primer programa es llamado READ.PAS, el cual tiene la función de leer los datos que se encuentran en la cinta magnética. Para lograr esta función, en el programa se utiliza la Interrupción 14 definida anteriormente, que nos permite inicializar la comunicación con el puerto serie; para ello, utilizamos la librería DOS y el tipo de dato REGISTERS. Como consecuencia de esto, en AX asignamos la cantidad de 3 (00000011),

indicando que la longitud de palabra será de 8 bits, que se utilizará un Stop Bits, que no habrá paridad y con una velocidad de 110 baudios. En DX asignamos la cantidad de 1, indicándole que se utilizará el puerto serie 2 (COM2). Luego aplicamos la interrupción 14, y la configuración del puerto queda establecida. A partir de aquí, el puerto serie está listo para poder enviar una señal y así establecer el movimiento del motor sobre la cinta magnética, y para poder leer los datos es necesario aplicar la función 02h de la Int 14. La captura de los datos es por bloques de 6 y 5, hasta completar 11 posiciones de 1 byte cada una en un arreglo, es decir que nos representa 22 caracteres, luego cada posición del arreglo es convertido a una cadena de caracteres y concatenado con una variable de caracteres de 22 posiciones, al completar dichas posiciones la variable de caracteres es almacenada permanentemente en un archivo de texto; con el objetivo que el texto quede disponible para que Fox-Pro pueda manipularlos. Los datos que se encuentran en el archivo de texto son números que representan el carnet, horas disponibles, hora1, hora2, máquina1, máquina2, fecha de uso y multa.

El segundo programa es llamado WRITE.PAS, para inicializar la comunicación con el puerto es de la misma forma que READ.PAS. Cuando el proceso de escritura se da, Fox-Pro manipula un archivo de texto, por el cual se envía la cadena de caracteres a escribir, luego el programa de escritura actúa de tal forma que a través de una variable de caracteres de 22 posiciones, captura los datos que

se encuentran en el archivo de texto; una vez capturados los datos, son convertidos a byte dentro de una lista de 11 posiciones. Una vez hecho esto, se utiliza la función 01h de int 14, para escribir los datos. El programa envía la señal para el movimiento del motor (start), y luego escribe tres datos para identificar el inicio de la cinta magnética, a partir de aquí se empieza a escribir la cadena de caracteres (datos del alumno) usando la interrupción 14 antes mencionada.

2.S.4 Gestión de Datos

Dentro de las bases de datos creadas tenemos: DATOSPER.DBF, BASEHOY.DBF, BASEMAÑ.DBF, BASE1.DBF Y BASE2.DBF.

El uso de DATOSPER.DBF se da cuando el usuario ve su estado personal, es aquí donde se presenta el nombres, apellidos, horas disponibles y multa del alumno. Otro uso especial es cuando el administrador del centro de cómputo desea generar un reporte general de los alumnos o solo desea personas con multas; si se desea un reporte de personas con multas, se crean filtros con aquellos cuyas multas que sean diferente de 0. También si se desea formatear la tarjeta, de esta base de datos se obtienen los datos principales (carnet, horas disponibles y multas) para escribirlos. Lo más importante en esta base de datos es el mantenimiento que se

le da, a través de la Adición, Consulta, Modificación y Eliminación, ya que es lo que da veracidad a los datos que en ella se almacenan.

Las bases de datos BASEHOY.DBF Y BASEMAÑ.DBF son utilizadas cuando el usuario ve su estado personal, el carnet es buscado en ambas bases, para obtener las reservaciones hechas, si las tiene. Dentro de los datos que son extraídos tenemos: horas reservadas, hora1, hora2, máquina1, máquina2 y fecha de uso en ambas bases de datos.

Cuando se da el proceso de reservación de máquina para hoy, en BASEHOY.DBF se almacena el carnet, horas reservadas, hora1, hora2, máquina1, máquina2, lo mismo sucede con la reservación para mañana (BASEMAÑ.DBF). Una vez hecha la reservación se hace uso de DATOSPER.DBF para descontarle las horas reservadas de sus horas disponibles.

En el proceso de usar horas, se verifica en BASEHOY.DBF las horas (hora1 y hora2) para darle acceso al centro de cómputo, esta verificación consiste en comparar la hora del sistema con las horas reservadas para determinar si puede entrar o todavía no es hora. Si el acceso es permitido, la BASEHOY.DBF posee un campo llamado MARCA, en el cual se coloca la letra "E", que indica que el usuario está haciendo uso de sus horas en el centro de cómputo.

En el proceso de terminar horas, se verifica la hora del sistema y el campo MARCA (el campo debe tener la letra "E") para poder cancelar las horas, y de esta manera se controla que el

usuario haya terminado adecuadamente sus horas. Si utilizó más horas de las que había reservado, el sistema se las descuenta de las horas disponibles, y si no posee horas disponibles es recargado con una multa, que equivale a \$10.00 por hora pasada. Otra utilidad de estas bases de datos es cuando el administrador desea observar las reservaciones hechas por código. Para lograr una mayor comprensión de las personas que han reservado, se relacionan estas bases de datos con DATOSPER.DBF para obtener de ahí los nombres y apellidos, esta relación se establece a través del CODIGO, ya que es común en ambas bases de datos.

Las base de datos BASE1.DBF y BASE2.DBF son utilizadas para presentar la distribución gráfica de máquinas vrs horas. Cuando se reservan máquinas para hoy se presenta la BASE1.DBF y si se reservan máquinas para mañana se presenta BASE2.DBF. Para lograr la reservación se almacenan en ella caracteres que indican que dicha máquina está reservada para determinada hora. Si el administrador del centro de cómputo desea observar las reservaciones por distribución gráfica de las máquinas vrs horas, son estas bases de datos las que se presentan.

Al final del día es necesario que el administrador del centro de cómputo actualice las reservaciones hechas para el día de mañana, por lo tanto BASEMAÑ.DBF es copiada a BASEHOY.DBF y BASE2.DBF es copiada a BASE1.DBF. BASEMAÑ.DBF y BASE2.DBF son limpiadas para que esten disponibles para el día de mañana.

2.H DISEÑO DEL HARDWARE

2.H.1 Circuito de Control

Debido a que es necesario iniciar los estados de los flip-flops en Q=0 se colocó un red RC en los CLEAR de estos, para que recibieran un bajo momentaneo al encender el equipo. Asi se asegura que los flip-flop den la secuencia correctamente. El criterio es: $5\tau_1 \geq 10\text{ms}$ para asegurar que el bajo es recibido.

Con $R=1\text{k}\Omega$ y $C=4.7\mu\text{F}$ se tiene:

$5\tau_1 = 23.5\text{ms}$ que es mayor a 10 ms.

El monoestable 2A tiene una red RC en el habilitador B para asegurar que el motor se dispare al encender el equipo. Su tiempo de subida debe ser mayor que el tiempo en que se activa el tiempo de los flip-flops, para que el habilitador A este ya activo. Con el criterio de que: $\tau_2 \approx 2\tau_1$, se tiene que si $R=1\text{k}\Omega$:

$$C = 2 * 23.5\text{ms} / 1\text{k} = 47\mu\text{F}$$

Para el movimiento del motor se utilizó un oscilador con 555 conectado en astable. La frecuencia máxima requerida es de 400Hz por lo que: si $R_A=100\Omega$ y $C=1\mu\text{F}$,

$$F = [1.44 / ((R_A + 2R_B) * C)] \text{ se tiene } R_B = 1.75\text{k}\Omega.$$

Para efecto de calibración exacta se puso R_B como trimmer. Todos los monoestables son 74221 y responden a la ecuación :

$$T = RC * \ln 2 \approx 0.7RC$$

A continuación se listan los tiempos de cada uno de los monoestables:

MONOESTABLE 1A (ANTIRREBOTE)	T=4mS
MONOESTABLE 1B (VENTANA)	T=6.5S
MONOESTABLE 2A (CLEAR DEL CONTADOR)	T=50uS

NOTA: A tiempos mayores de 4 seg. se observó experimentalmente que la fórmula se desvía y para exactitud de ajuste se utilizó un trimer.

En cuanto a selección de componentes se utilizaron los siguientes CHIPS por estas razones:

74221: Monoestable dual no retriguable, necesario para que sólo el primer pulso del puerto serie sea tomado como start.

74191: Contador ascendente/descendente, tipo binario necesario para la inversión de giro del motor.

7473: Flip-Flop JK disparado por transiciones negativas necesaria esta última condición según el diagrama de tiempos.

2.H.2 Circuito de Lectura

La mayor dificultad esta en el diseño adecuado del PLL. Primeramente se seleccionó el LM565 por su fácil adquisición en el mercado nacional, la bibliografía abundante sobre él y su versatilidad en aplicaciones (de propósito general).

En los libros técnicos se recomienda el uso de un factor de amortiguamiento (ξ) igual a 0.7. Esto debido a que mayor sea la atenuación se tendrán más oscilaciones en la respuesta de la red RC. Un factor de amortiguamiento muy alto $\xi=5$ da un filtrado pobre pero estabilidad en la respuesta; Un $\xi=0.3$ da una buena atenuación pero tendencia a oscilar.

Las frecuencias de detección son 1.7kHz y 2.3kHz, por lo que la frecuencia central es de 2kHz. Se elige una resistencia tal que $2k\Omega \leq R_0 \leq 20k\Omega$. Si $C_0=0.033\mu F$ y $F_0=[1/(3.7R_0C_0)]$ se tiene que $R_0=4.1k\Omega$.

Se uso en R_0 un trimer de 5k para mejor calibración.

El semiancho de banda es $B=300Hz$ debido a que $F_0=2kHz$ y las frecuencias de trabajo se desvian 300Hz a partir de ellas.

Existe una frecuencia natural de la red RC de filtrado que debe cumplir con la siguiente ecuación:

$$\omega_n = [2B / (\xi + 1/4\xi)] = 567.6 \text{ rad/S.}$$

$$\tau_2 = R_2 C_1 = 2\xi / \omega_n = 2 * 0.7 / 567.6 = 2.4667 \text{ mS.}$$

Se necesita la ganancia de lazo cerrado que es función de la frecuencia central (F_o) y el voltaje total de alimentación V_T , así:
 $K=33.6F_o/V_T = 6109.1\text{Hz/V}$.

Para despejar R_2 y C se necesita otra ecuación que es a partir de: $\tau_1+\tau_2 = R_1C_1+R_2C_1 = K/\omega_n^2 = 18.96\text{mS}$.

Utilizando como R_1 la resistencia de salida de este PLL que es de $3.6\text{k}\Omega$, junto con el resultado de τ_2 se obtiene:

$$R_1C_1 = 18.96 \text{ mS} - 2.467\text{mS} = 16.5\text{mS}$$

por lo tanto $C_1 = 4.7\mu\text{F}$, con $R_1 = 3.6\text{k}\Omega$.

Para R_2 : $\tau_2 = R_2C_1$ por lo que $R_2 = 538.6\Omega$ aproximadamente igual al valor comercial de 560Ω .

Para elegir C_2 se toma el criterio:

$$C_2 \leq 10\% C_1$$

$$C_2 < 10 \cdot 4.7\mu / 100 = 0.47\mu\text{F}$$

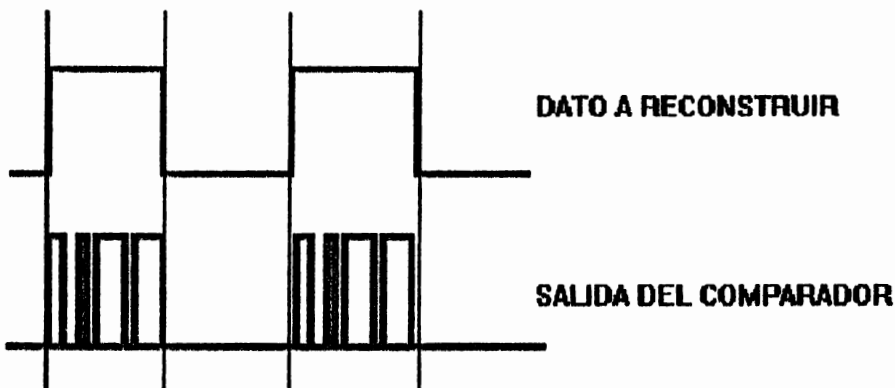
$$C_2 = 0.33\mu\text{F}.$$

Para filtrar la salida (pin 7) con respecto al voltaje de referencia (pin 6) es necesario botar las componentes de alta frecuencia y dejar en el pin 7 un DC (o aproximadamente) para ello se colocó un doble filtro pasobajas RC, cuyos valores se obtuvieron experimentalmente, siendo los resultados: $R= 10\text{k}\Omega$ y $C=0.02\mu\text{F}$.

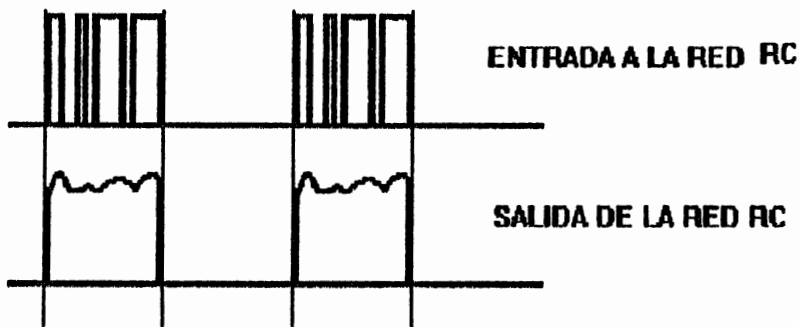
puesto que la diferencia de voltajes con respecto al pin 6 de la salida (pin 7) del LM565 es cuando mucho de 300mV se requirió un comparador de precisión. Se seleccionó el LM311 por las siguientes razones:

- Alta impedancia de entrada.
- Salida compatible a niveles TTL.
- Velocidad alta de respuesta (200 nS).
- Respuesta a voltajes muy pequeños.

Cómo se mencionó en la descripción del proyecto la salida del comparador no reconstruía adecuadamente la señal, gráficamente:



Para solucionar el problema se utilizó una red RC en la cual se produjese el siguiente efecto:



Para calcular los componentes de la red RC se hace que τ sea aproximadamente igual al periodo de los pulsos internos. Con osciloscopio se encontró que este tiempo es de 2mS.

$$\text{Si } \tau = 2\text{mS} = \tau = RC$$

con $R = 10\text{k}\Omega$ se obtiene un $C = 0.2\mu\text{f}$.

La impedancia de entrada del LM565 es baja ($5\text{k}\Omega$) por lo que se recomienda colocar a la entrada un seguidor de voltaje.

SUGERENCIAS PARA EL DISEÑO.

El amplificador que maneja la señal proveniente de los cabezales debe ser capaz de:

- * Tener una ganancia muy alta.
- * No presentar oscilaciones a pesar de la alta ganancia.
- * Botar las componentes de alta frecuencia y dejar pasara solo las componentes de audio.

Es interesante notar que los cabezales sacan un voltaje típico de 1mVpp, con ruido ambiental debido al bajo nivel.

Utilizar cable blindado para la señal de los cabezales es obligatorio.

No utilizar filtros activos a la entrada de un PLL que ocupa un detector de fase tipo multiplicador analógico (como el LM565 y 567) pues los filtros pueden producir cambios bruscos en la fase de la señal de entrada, lo cual puede hacer que el PLL tenga un "enganche" inestable.

BIBLIOGRAFIA PARA EL CIRCUITO DE LECTURA.

"DISPOSITIVOS PLL DE FUENTES REGULADAS, TEMPORIZADORES Y DE TELECOMUNICACIONES", Arthur Williams, Mc Graw/Hill, 1990.

"AMPLIFICADORES OPERACIONALES Y CI LINEALES", Coughlin/Driscoll, Prentice Hall, 1993.

2.H.3 Circuito de Escritura

El centro de este circuito es el VCO, que para el caso se seleccionó el 565, debido a la fácil adquisición en el mercado y la abundante bibliografía.

Se encontró la dificultad de que la frecuencia del oscilador es función del voltaje de alimentación y que la fórmula tiene cierta discrepancia con la realidad.

De tablas y gráficas de fabricante se encontró un par de valores que están en el rango de operación: $R_o = 10k\Omega$ y $C_o = 0.02\mu f$.

La frecuencia de operación en función del voltaje de entrada está dada por:

$$F_o = [2(V_{cc}-V_5)/(R_o C_o V_{cc})]$$

Es necesario despejar los voltajes de entrada, sabiendo que el voltaje en el pin5 debe de estar en el rango: $(3/4)V_{cc} < V_5 < V_{cc}$.

De allí que los voltajes, según fórmula, correspondientes a las frecuencias de 1700 y 2300 Hz son, respectivamente:

$$V_5 = 10.08v \text{ para un } 1 \text{ lógico.}$$

$$V_5 = 9.66 \text{ para un } 0 \text{ lógico.}$$

Los amplificadores operacionbales 1 y 2 tiene ganacia unitaria y son sumadores directos, cumpliendo con la ecuación:

$$V_o = V_{ref} + V_{in}$$

Para el diseño, se toma $R_f = R_f' = 100k\Omega$.

$$V_o = (R_f'/R_{in1}) V_{ref} + (R_f'/R_{in2}) V_{in}$$

Para obtener una ganancia unitaria se requiere que $R_{in1} = R_{in2} = R_f' = 100k\Omega$. La resistencia de balance R_b se calcula a partir de la igualdad de ganancias, tal y como sigue:

$$(R_f'/R_{in1}) + (R_f'/R_{in2}) = R_f/R_b ; \text{ despejando } R_b:$$

$$R_b = 50k\Omega.$$

Debido a que los cabezales distorsionan una entrada en escalón se utilizó la salida triangular del VCO. Esta salida se pasó a través de un integrador para obtener una señal "alisada" semejante a una seno.

Para reducir los efectos de desajuste por corriente se incluye una R_s y para limitar la ganancia CD, así como dar un camino de descarga al capacitor se incluyó una R_d en paralelo con C . Se seleccionó una ganancia en CD igual a la unidad, por lo que se hizo $R = R_d = 100k\Omega$. La siguiente consideración es la frecuencia de trabajo máximo que se calcula a partir de:

$$F(\text{de operación}) = 3/(2\pi R_d C);$$

De donde se despeja C, obteniéndose el valor de:

$C = 0.002\mu\text{f}$, siendo el valor comercial $C = 0.0022\mu\text{f}$.

El error por la corriente de polarización se reduce por medio de un AMP-OP de entrada JFET (como es el caso del ECG 589) y R_s , siendo esta igual a:

$$R_s = R_d // R = 50\text{k}\Omega.$$

SUGERENCIAS PARA EL CIRCUITO DE ESCRITURA.

Para un futuro diseño es recomendable utilizar un VCO cuya frecuencia no dependa del voltaje de alimentación, así como también responda directamente a niveles TTL en la entrada.

Es importante atenuar la señal que se va grabar en los cabezales, puesto que si es muy grande se puede dañar la bobina interna. En el mejor de los casos a señal grande se satura la bobina y el campo magnético ya no será proporcional a la entrada sino el máximo que da el cabezal. Un valor de 200mVpp superpuesto con un voltaje de DC de 100mVdc es lo más adecuado, tal como se utilizó en el presente proyecto.

BIBLIOGRAFIA PARA EL CIRCUITO DE ESCRITURA.

"MANUAL DE NATIONAL SEMICONDUCTORS"

"INTRODUCCION A LOS AMP-OP CON APLICACIONES A CI LINEALES",
Faulkenberry, Editorial LIMUSA, 1990.

2.H.4 Diseño de Aspectos Varios

Se consiguió un motor que tenía una capacidad de giro de 1.8 grados por pulso. Con 200 pulsos se obtenía una revolución completa.

Se decidió tomar una velocidad del avance de los cabezales (cuando leían o escribían) de la mitad del tiempo que cuando regresaban. Debido a la restricción en frecuencia que tiene todo motor paso a paso las frecuencias de pulso seleccionadas fueron:

- 200 Hz para avanzar en lectura o escritura.
- 400 Hz para retroceder.

Con 200Hz se tiene 1 revolución por segundo, $\omega = 2\pi$ rad/s la velocidad lineal de avance se obtiene de la expresión:

$$v = \omega r, \text{ con } r = 0.6\text{cm}$$

De aquí que: $v = 2\pi * 0.6\text{cm} = 3.77$ cm/s.

Para el regreso de los cabezales se utilizó 400Hz por lo que $\omega = 4\pi$ rad/s. La rapidez lineal al regreso es de $v = 7.54$ cm/s.

Al transmitir a 110 baudios se tiene un tiempo de bit a:

$$t_{\text{bit}} = 1/\text{baudios} = 1/110 = 9.09\text{ms}$$

La cantidad de ciclos de las frecuencias de trabajo que caben en este tiempo de bit se obtiene así:

de ciclos por dígito = tiempo de bit / periodo de la señal de t .

los resultados son:

Nivel Lógico	Número de ciclos
0	20.9
1	15.5

Es importante que el número de ciclos por dígito no sea menor que 5 puesto que los PLL podrían no tener la suficiente señal como para cerrar el lazo.

El espacio que ocupa cada bit en la cinta magnética se obtiene así: $\text{espacio de bit} = e_{\text{bit}} = v \cdot t_{\text{bit}} = 0.34\text{mm}$

Si el espacio disponible en tarjeta es de 8.6cm la cantidad de bits por pista es de:

$\text{Número de bits por pista} = \text{Esp. disp. máx.} / e_{\text{bit}} = 253 \text{ bits.}$

Teniendo dos pistas y ocupando 10 bits por byte (debido al bit de stop y marca) la cantidad máxima de bytes se obtiene así:

$\text{Cantidad máx de bytes} = 2 \cdot 253 / 10 = 50 \text{ bytes en total.}$

DIAGRAMA DE FLUJO PARA
LECTURA EN CINTA MAGNETICA

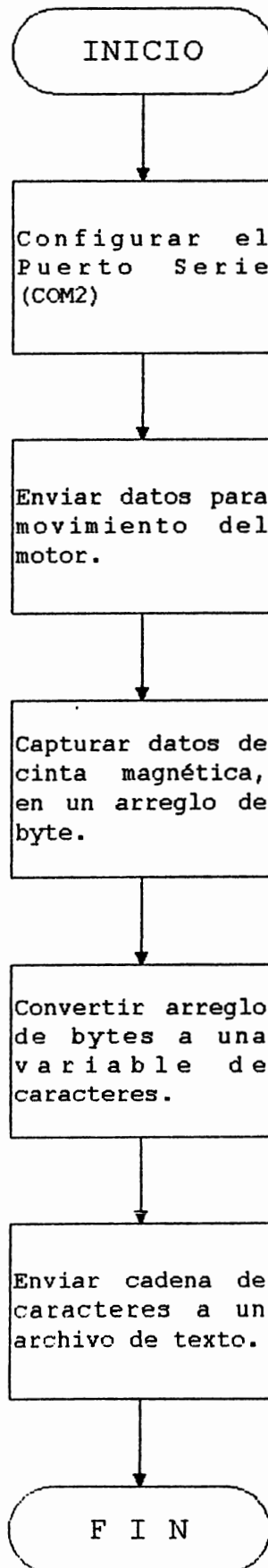


DIAGRAMA DE FLUJO PARA
ESCRITURA EN CINTA MAGNETICA

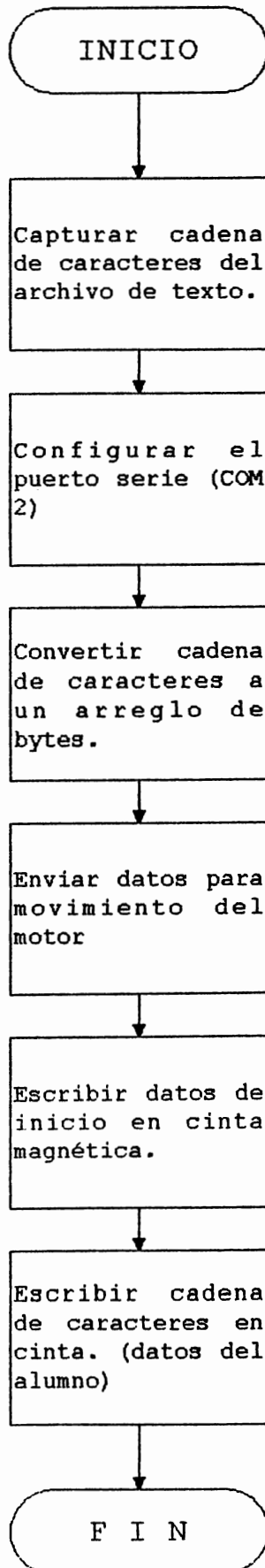
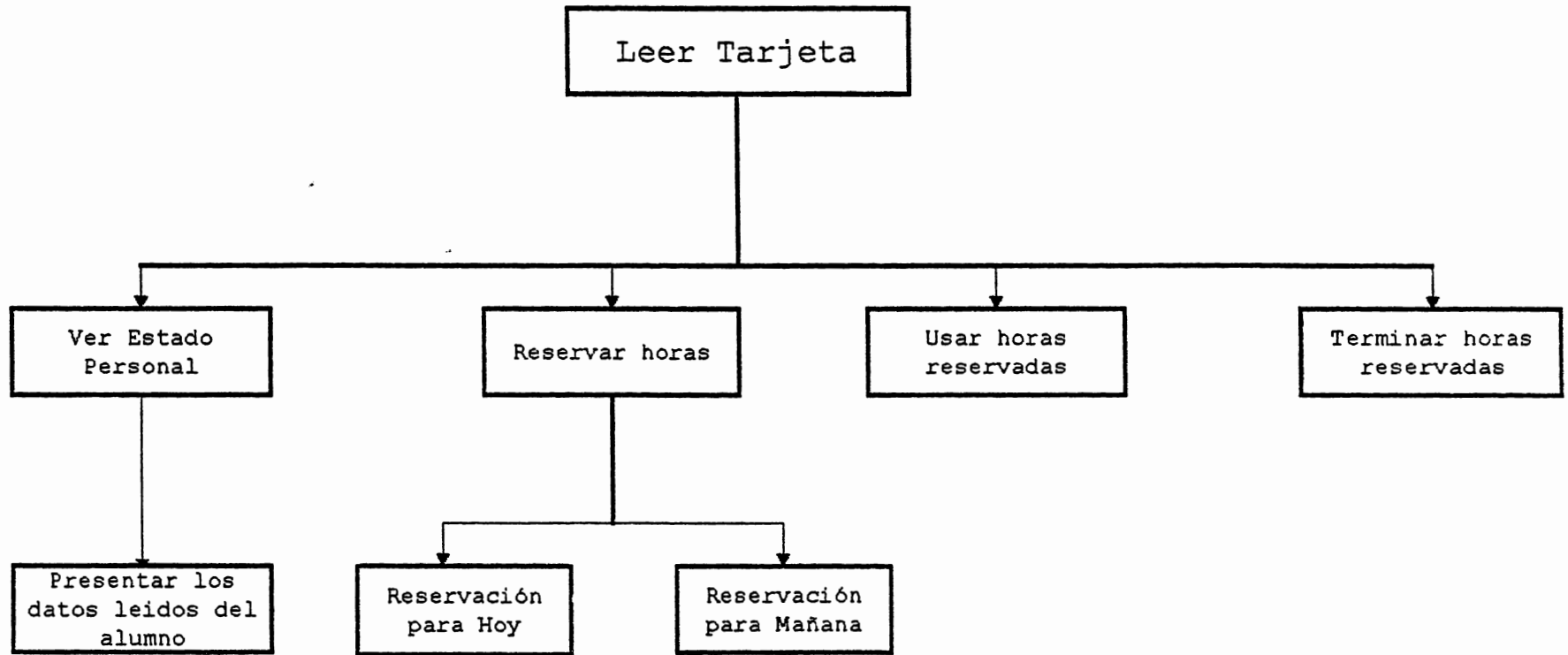


DIAGRAMA GENERAL

PROCESO DE RESERVACION DE HORAS



ANEXO D

Programas en Fox-Pro Versión 2.6

```

1
2 ANEXO D
3
4 * Programa que Genera el Menú Principal
5 * Para poder llamar las Aplicaciones del
6 * Proceso de Reservación De Horas en un
7 * Centro de Cómputo.
8
9 * Fecha de Creación : 24-Junio-96
10 * Hora de Creación : 10:21:08 am
11
12 * Desactivaciones Iniciales
13 SET TALK OFF
14 SET STATUS OFF
15 SET BELL OFF
16 SET DELETE ON
17 SET MESSAGE TO 22 LEFT
18 SET CLOCK TO 01,67
19 SET DATE TO FRENCH
20
21 * Desactivaciones de Las Funciones
22
23 CLEAR
24
25 DO ACTIVAR_FUNS
26 DO FORMATO
27 * Creación de Menú Principal
28
29 DEFINE MENU barra
30 DEFINE PAD uno OF barra PROMPT "\<Usar Tarjeta" AT 05,02
31 DEFINE PAD dos OF barra PROMPT "\<Ver Estado General" AT 05,17
32 DEFINE PAD tres OF barra PROMPT "\<Administrar Bases" AT 05,38
33 DEFINE PAD cuatro OF barra PROMPT "\<Formatear Tarjeta" AT 05,58
34 MESSAGE "|| Permite darle un formato inicial a la tarjeta;
35 ||"
36
37 ON PAD uno OF barra ACTIVATE POPUP tarjeta
38 ON PAD dos OF barra ACTIVATE POPUP estado
39 ON PAD tres OF barra ACTIVATE POPUP admon
40
41 DEFINE POPUP tarjeta SHADOW FROM 07,04
42 DEFINE BAR 1 OF tarjeta PROMPT "\<Leer Tarjeta ";
43 MESSAGE "|| Permite que el sistema pueda leer tu tarjeta;
44 ||"
45
46 DEFINE BAR 2 OF tarjeta PROMPT "\<Ver Estado Personal";
47 MESSAGE "|| Ver cantidad de horas disponibles;
48 ||"
49 DEFINE BAR 3 OF tarjeta PROMPT "\<Reservar Máquina";
50 MESSAGE "|| Presenta la distribución de Horas y Máquinas;
51 ||"
52 DEFINE BAR 4 OF tarjeta PROMPT "\<Usar Horas";
53 MESSAGE "|| Permite que el sistema pueda marcar la entrada;
54 para usar horas ||"
55
56 DEFINE BAR 5 OF tarjeta PROMPT "\<Terminar Horas";
57 MESSAGE "|| Permite Finalizar las horas reservadas;
58 ||"
59
60
61 ON BAR 3 OF tarjeta ACTIVATE POPUP tiempo

```

```

62
63 DEFINE POPUP tiempo SHADOW FROM 09,29
64   DEFINE BAR 1 OF tiempo PROMPT " Para \<Hoy";
65   MESSAGE "|| Permite hacer reservaciones para el día;
66   de Hoy
67   DEFINE BAR 2 OF tiempo PROMPT " Para \<Mañana";
68   MESSAGE "|| Permite hacer reservaciones para el día;
69   de Mañana
70
71 DEFINE POPUP estado SHADOW FROM 07,20
72   DEFINE BAR 1 OF estado PROMPT "Reporte General de Alumnos ";
73   MESSAGE "|| Genera el Listado General de Todos los alumnos;
74   inscritos
75   DEFINE BAR 2 OF estado PROMPT "Personas con \<Multas ";
76   MESSAGE "|| Genera una Lista de Personas con Multas;
77
78   DEFINE BAR 3 OF estado PROMPT "Ver Reservación de \<Hoy ";
79   DEFINE BAR 4 OF estado PROMPT "Ver Reservación de \<Mañana"
80
81 DEFINE POPUP admon SHADOW FROM 07,42
82   DEFINE BAR 1 OF admon PROMPT "\<Mantenimiento ALUMNOS.DBF ";
83   MESSAGE "|| Permite llevar el control de todos los alumnos;
84
85   DEFINE BAR 2 OF admon PROMPT "\<Actualizar Reservaciones ";
86   MESSAGE "|| Permite Actulizar todas las reservaciones de Hoy ;
87   Para Mañana
88
89 ON BAR 1 OF admon ACTIVATE POPUP alumnos
90
91 DEFINE POPUP reservar_hoy SHADOW FROM 14,25
92   DEFINE BAR 1 OF reservar_hoy PROMPT " \<Distribución de
93   Horas-Máquinas";
94   MESSAGE "|| Presenta la Distribución de Reservaciones de Hoy;
95
96   DEFINE BAR 2 OF reservar_hoy PROMPT " Distribución por \<Código
97   ";
98   MESSAGE "|| Preseta la Distribución por Código de las reservaciones;
99   de Hoy
100
101
102 ON BAR 3 OF estado ACTIVATE POPUP reservar_hoy
103
104 DEFINE POPUP reservar_mañ SHADOW FROM 14,25
105   DEFINE BAR 1 OF reservar_mañ PROMPT " \<Distribución de
106   Horas-Máquinas";
107   MESSAGE "|| Presenta la Distribución de Reservaciones de Mañana;
108
109   DEFINE BAR 2 OF reservar_mañ PROMPT " Distribución por \<Código
110   ";
111   MESSAGE "|| Preseta la Distribución por Código de las reservaciones d.
112   Mañana;
113
114
115 ON BAR 4 OF estado ACTIVATE POPUP reservar_mañ
116
117 DEFINE POPUP alumnos SHADOW FROM 12,50
118   DEFINE BAR 1 OF alumnos PROMPT " \<Adicionar";
119   MESSAGE "|| Permite adicionar un nuevo alumno a base de datos;
120
121   DEFINE BAR 2 OF alumnos PROMPT " \<Consultar";
122   MESSAGE "|| Permite Consultar determinado alumno en base de datos;
123

```

```

124 DEFINE BAR 3 OF alumnos PROMPT " \<Modificar";
125 MESSAGE "|| Permite Modificar determinado alumno en base de datos;
126 ||"
127 DEFINE BAR 4 OF alumnos PROMPT " \<Eliminar ";
128 MESSAGE "|| Permite Eliminar determinado alumno en base de datos;
129 ||"
130
131
132 ON SELECTION BAR 2 OF tarjeta DO ESTADO
133 ON SELECTION BAR 1 OF tiempo DO HOY
134 ON SELECTION BAR 2 OF tiempo DO MAÑANA
135 ON SELECTION BAR 4 OF tarjeta DO USAR_HORAS
136 ON SELECTION BAR 5 OF tarjeta DO FINALIZAR
137 ON SELECTION BAR 1 OF estado DO REPORTE_GENERAL
138 ON SELECTION BAR 2 OF estado DO REPORTE_DE_MULTAS
139 ON SELECTION BAR 1 OF reservar_hoy DO VER_HOY02
140 ON SELECTION BAR 2 OF reservar_hoy DO VER_HOY01
141 ON SELECTION BAR 1 OF reservar_mañ DO VER_MAÑ02
142 ON SELECTION BAR 2 OF reservar_mañ DO VER_MAÑ01
143 ON SELECTION BAR 1 OF alumnos DO PROC_ADICIONAR
144 ON SELECTION BAR 2 OF alumnos DO PROC_CONSULTAR
145 ON SELECTION BAR 3 OF alumnos DO PROC_MODIFICAR
146 ON SELECTION BAR 4 OF alumnos DO PROC_ELIMINAR
147 ON SELECTION BAR 2 OF admon DO ACTUALIZAR
148
149
150 * Activando Menú Principal
151
152 ACTIVATE MENU barra
153 DEACTIVATE MENU barra
154
155 DO DESACTIVAR_FUNS
156
157 RETURN
158
159 PROCEDURE nada
160
161 RETURN
162
163 PROCEDURE FORMATO
164 SET CLOCK TO 01,67
165 @04,00 FILL TO 20,79 COLOR N/RB
166 @01,02 SAY "FECHA : "
167 @01,10 SAY DATE()
168 @02,02 SAY "S.A.H.C.C."
169 @01,25 SAY "SISTEMA ADMINISTRADOR DE HORAS"
170 @02,25 SAY " EN CENTRO DE COMPUTO "
171 @01,60 SAY "HORA : "
172 @02,60 SAY "S.A.H.C.C."
173 @00,00 TO 03,79 DOUBLE COLOR R+
174 col=21
175 SET COLOR TO R+
176 FOR i=1 TO 2
177 @00,col SAY "||"
178 @01,col SAY "||"
179 @02,col SAY "||"
180 @03,col SAY "||"
181 col=col+36
182 ENDFOR
183 SET COLOR TO
184 DO bigfonts
185 @04,00 TO 20,79 DOUBLE COLOR BG+

```



```

186 @06,01 TO 06,79 DOUBLE COLOR BG+
187 SET COLOR TO BG+
188 @06,00 SAY "||"
189 @06,79 SAY "||"
190 SET COLOR TO
191 @21,00 FILL TO 24,79 COLOR N
192 @21,00 TO 24,79 DOUBLE COLOR W
193 RETURN
194

```

```

195 PROCEDURE bigfonts
196 SET COLOR TO BG+/RB+
197 TEXT

```

```

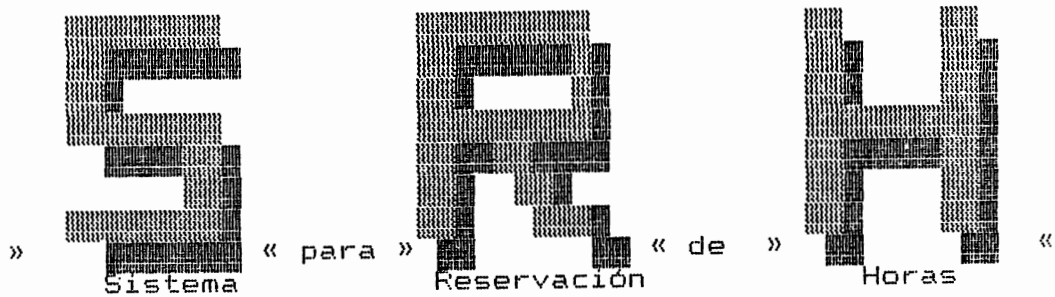
198
199
200
201
202

```

```

203
204
205
206
207
208
209
210
211
212
213

```



```

214 ENDTEXT
215 @08,09 FILL TO 18,70 COLOR N+/W+
216 SET COLOR TO
217 @08,09 TO 18,70 DOUBLE COLOR B+/W+
218
219 RETURN
220

```

```

221 PROCEDURE ACTIVAR_FUNS
222 ON KEY LABEL ESC do nada
223 ON KEY LABEL F1 do nada
224 ON KEY LABEL F2 do nada
225 ON KEY LABEL F3 do nada
226 ON KEY LABEL F4 do nada
227 ON KEY LABEL F5 do nada
228 ON KEY LABEL F6 do nada
229 ON KEY LABEL F7 do nada
230 ON KEY LABEL F8 do nada
231 ON KEY LABEL F9 do nada
232 ON KEY LABEL F10 do nada
233 ON KEY LABEL F11 do nada
234 ON KEY LABEL F12 do nada
235 ON KEY LABEL DEL do nada
236 ON KEY LABEL INS do nada
237 ON KEY LABEL END do nada
238 ON KEY LABEL BACKTAB do nada
239 ON KEY LABEL PGUP do nada
240 ON KEY LABEL PGDN do nada
241 ON KEY LABEL SPACEBAR do nada
242 ON KEY LABEL HOME do nada
243 ON KEY LABEL TAB do nada

```

```

244
245 RETURN
246
247 PROCEDURE DESACTIVAR_FUNS

```

```
248 ON KEY LABEL ESC
249 ON KEY LABEL F1
250 ON KEY LABEL F2
251 ON KEY LABEL F3
252 ON KEY LABEL F4
253 ON KEY LABEL F5
254 ON KEY LABEL F6
255 ON KEY LABEL F7
256 ON KEY LABEL F8
257 ON KEY LABEL F9
258 ON KEY LABEL F10
259 ON KEY LABEL F11
260 ON KEY LABEL F12
261 ON KEY LABEL DEL
262 ON KEY LABEL INS
263 ON KEY LABEL END
264 ON KEY LABEL BACKTAB
265 ON KEY LABEL PGUP
266 ON KEY LABEL PGDN
267 ON KEY LABEL SPACEBAR
268 ON KEY LABEL HOME
269 ON KEY LABEL TAB
270
271 RETURN
272
273 PROCEDURE ESTADO
274 HIDE MENU barra
275 HIDE POPUP ALL
276 DO ESTA_FER
277 CLEAR
278 DO FORMATO
279 RETURN
280
281 PROCEDURE USAR_HORAS
282 HIDE MENU barra
283 HIDE POPUP ALL
284 DO USAHORAS
285 CLEAR
286 DO FORMATO
287 RETURN
288
289 PROCEDURE HOY
290 HIDE MENU barra
291 HIDE POPUP ALL
292 DO TODAY
293 CLEAR
294 DO FORMATO
295 RETURN
296
297 PROCEDURE MAÑANA
298 HIDE MENU barra
299 HIDE POPUP ALL
300 DO TOMORROW
301 CLEAR
302 DO FORMATO
303 RETURN
304
305 PROCEDURE FINALIZAR
306 HIDE MENU barra
307 HIDE POPUP ALL
308 DO FINHORAS
309 CLEAR
```

```
310 DO FORMATO
311 RETURN
312
313 PROCEDURE REPORTE_GENERAL
314 HIDE MENU barra
315 HIDE POPUP ALL
316 DO REP_GEN
317 CLEAR
318 DO FORMATO
319 RETURN
320
321 PROCEDURE REPORTE_DE_MULTAS
322 HIDE MENU barra
323 HIDE POPUP ALL
324 DO REP_MULT
325 CLEAR
326 DO FORMATO
327 RETURN
328
329 PROCEDURE VER_HOY01
330 HIDE MENU barra
331 HIDE POPUP ALL
332 DO REP_HOY1
333 ON KEY LABEL ESC DO nada
334 CLEAR
335 DO FORMATO
336 RETURN
337
338 PROCEDURE VER_MAR01
339 HIDE MENU barra
340 HIDE POPUP ALL
341 DO REP_MAR1
342 ON KEY LABEL ESC DO nada
343 CLEAR
344 DO FORMATO
345 RETURN
346
347 PROCEDURE VER_HOY02
348 HIDE MENU barra
349 HIDE POPUP ALL
350 DO REP_HOY2
351 ON KEY LABEL ESC DO nada
352 CLEAR
353 DO FORMATO
354 RETURN
355
356 PROCEDURE VER_MAR02
357 HIDE MENU barra
358 HIDE POPUP ALL
359 DO REP_MAR2
360 ON KEY LABEL ESC DO nada
361 CLEAR
362 DO FORMATO
363 RETURN
364
365
366 PROCEDURE PROC_ADICIONAR
367 HIDE MENU barra
368 HIDE POPUP ALL
369 ON KEY LABEL SPACEBAR
370 DO ADICION
371 ON KEY LABEL SPACEBAR DO nada
```

```
372 ON KEY LABEL ESC DO nada
373 CLEAR
374 DO FORMATO
375 RETURN
376
377 PROCEDURE PROC_CONSULTAR
378 HIDE MENU barra
379 HIDE POPUP ALL
380 DO CONSULTA
381 ON KEY LABEL ESC DO nada
382 CLEAR
383 DO FORMATO
384 RETURN
385
386 PROCEDURE PROC_MODIFICAR
387 HIDE MENU barra
388 HIDE POPUP ALL
389 ON KEY LABEL SPACEBAR
390 DO MODIFICA
391 ON KEY LABEL SPACEBAR DO nada
392 ON KEY LABEL ESC DO nada
393 CLEAR
394 DO FORMATO
395 RETURN
396
397 PROCEDURE PROC_ELIMINAR
398 HIDE MENU barra
399 HIDE POPUP ALL
400 DO ELIMINAR
401 ON KEY LABEL ESC DO nada
402 CLEAR
403 DO FORMATO
404 RETURN
405
406 PROCEDURE ACTUALIZAR
407 HIDE MENU barra
408 HIDE POPUP ALL
409 DO ACT_RES
410 ON KEY LABEL ESC DO nada
411 CLEAR
412 DO FORMATO
413 RETURN
414
```

```

1 * Programa que Lista el Estado Personal
2 * del que hace uso del Centro de Cómputo,
3 * el cual lee la tarjeta del alumno.
4
5 SET TALK OFF
6 SET STATUS OFF
7 SET BELL OFF
8
9 CLEAR
10
11 CLOSE ALL
12 SELECT 1
13 USE DATOSPER
14 SELECT 2
15 USE BASEHOY
16 SELECT 3
17 USE BASEMAÑ
18
19 carnet=0
20 vnombre=space(40)
21 vapellido=space(40)
22 vhoras_dis=0
23 vmulta=0
24 vhoras_res=0
25 vhora1=0
26 vhora2=0
27 vmaq1=0
28 vmaq2=0
29 vfecha=space(8)
30 vfecha="Ninguna"
31
32 @12,25 SAY "Introduzca Carnet :" GET carnet PICTURE "99-99999"
33 READ
34 CLEAR
35 SELECT 1
36 ERASE CARNET.IDX
37 INDEX ON CODIGO TO CARNET
38 SET INDEX TO CARNET
39
40 SEEK(carnet)
41 IF found()
42     vnombre=nombres
43     vapellido=apellidos
44     vhoras_dis=horas_dis
45     vmulta=multas
46 ENDIF
47 DO FORMATO
48 SELECT 2
49 IF NOT FILE("CARNET02.IDX")
50     INDEX ON CODIGO TO CARNET02
51     SET INDEX TO CARNET02
52 ELSE
53     SET INDEX TO CARNET02
54     REINDEX
55 ENDIF
56 SEEK(carnet)
57 IF found()
58     vhoras_res=horas_res
59     vhora1=hora1
60     vhora2=hora2
61     vmaq1=maquina1

```

```

62         vmaq2=maquina2
63         vfecha=fecha_uso
64     ENDIF
65     DO VENTANA01
66     vhoras_res=0
67     vhora1=0
68     vhora2=0
69     vmaq1=0
70     vmaq2=0
71     vfecha="Ninguna"
72
73     SELECT 3
74     ERASE CARNET01.IDX
75     INDEX ON CODIGO TO CARNET01
76     SET INDEX TO CARNET01
77
78     SEEK(carnet)
79     IF found()
80         vhoras_res=horas_res
81         vhora1=hora1
82         vhora2=hora2
83         vmaq1=maquina1
84         vmaq2=maquina2
85         vfecha=fecha_uso
86     ENDIF
87     DO VENTANA02
88
89     CLOSE ALL
90     RETURN
91
92     PROCEDURE FORMATO
93     @00,00 TO 24,79 DOUBLE COLOR R+
94     @06,00 SAY "||" COLOR R+
95     @06,79 SAY "||" COLOR R+
96     @06,01 TO 06,78 DOUBLE COLOR R+
97     @00,25 SAY " E S T A D O   P E R S O N A L " COLOR BG+
98     @02,05 SAY "CARNET  :"
99     @02,14 SAY carnet PICTURE "99-99999"
100    @03,05 SAY "APELLIDO  :"
101    @03,16 SAY vapellido
102    @04,05 SAY "NOMBRE  : "
103    @04,14 SAY vnombre
104    @05,05 SAY "HORAS DISPONIBLES  :"
105    @05,25 SAY vhoras_dis PICTURE "999"
106    RETURN
107
108    PROCEDURE VENTANA01
109    @09,07 FILL TO 20,36 COLOR W/W+
110    @08,05 FILL TO 19,34 COLOR N
111    @08,05 TO 19,34 DOUBLE COLOR RB+
112    @08,08 SAY " Reservación para hoy "
113    SET COLOR TO BG+
114    @10,07 SAY "Horas Reservadas  : "
115    @10,26 SAY vhoras_res PICTURE "9"
116    @12,07 SAY "Pimer Hora  :"
117    @12,20 SAY vhora1 PICTURE "99"
118    @13,07 SAY "Máquina 1  :"
119    @13,20 SAY vmaq1 PICTURE "99"
120    @15,07 SAY "Segunda Hora  :"
121    @15,22 SAY vhora2 PICTURE "99"
122    @16,07 SAY "Máquina 2  :"
123    @16,20 SAY vmaq2 PICTURE "99"

```

```
124     @18,07 SAY "Fecha de uso :"  
125     @18,22 SAY vfecha  
126     SET COLOR TO  
127 RETURN  
128  
129 PROCEDURE VENTANA02  
130     @09,44 FILL TO 20,74 COLOR W/W+  
131     @08,42 FILL TO 19,72 COLOR N  
132     @08,42 TO 19,72 DOUBLE COLOR RB+  
133     @08,45 SAY " Reservación para mañana "  
134     SET COLOR TO BG+  
135     @10,44 SAY "Horas Reservadas : "  
136     @10,63 SAY vhoras_res PICTURE "9"  
137     @12,44 SAY "Pimer Hora :"  
138     @12,57 SAY vhora1 PICTURE "99"  
139     @13,44 SAY "Máquina 1 :"  
140     @13,57 SAY vmaq1 PICTURE "99"  
141     @15,44 SAY "Segunda Hora :"  
142     @15,59 SAY vhora2 PICTURE "99"  
143     @16,44 SAY "Máquina 2 :"  
144     @16,57 SAY vmaq2 PICTURE "99"  
145     @18,44 SAY "Fecha de uso :"  
146     @18,59 SAY "Mañana"  
147     read  
148     SET COLOR TO  
149 RETURN  
150
```

```

1 *
2 * OBJETIVO DEL PROGRAMA: Reservar Horas
3 * en el Centro de Cómputo y Optimizar el
4 * uso de las horas del usuario.
5 *
6 ||
7 * Fecha de Creación : 20-Junio-1996
8 * Hora de Creación : 10:42 a.m.
9 *
10
11
12 * Desactivaciones Iniciales
13
14 SET STATUS OFF
15 SET TALK OFF
16 SET BELL OFF
17 SET CLOCK TO 01,60
18 SET DATE TO FRENCH
19
20 * Fin de las Desactivaciones Iniciales
21
22 CLEAR
23 CLOSE ALL
24
25 ***** Area para Inicializar Variables *****
26 ***** Globales del Programa *****
27
28 * { Variable para seleccionar las horas a reservar }
29 amopm=0
30 * { Variable para incrementar fila en procedimiento FORMATO }
31 fila=08
32 * { Variable para incrementar columnas en FORMATO }
33 control=0
34 * { Variable para obtener el número de registro de BASE1.DBF }
35 numreg=0
36 * { Primer hora a reservar }
37 vhora1=0
38 * { Segunda hora a reservar }
39 vhora2=0
40 * { Máquina Número 1 de la primera hora reservada }
41 nummaq1=0
42 * { Máquina Número 2 de la segunda hora reservada }
43 nummaq2=0
44 * { Cantidad de horas a reservar (1 ó 2) }
45 cant=0
46 * { Controla la cantidad de horas que puedes reservar }
47 h_res=0
48 * { Variable para determinar el Número de máquina en }
49 * { Procedimiento LOCALIDAD }
50 loc=0
51 * { Variable para controlar la cantidad de horas reservadas, }
52 * { y la selección de horas }
53 band=0
54 * { Contiene el carnet del alumno o usuario }
55 vcarnet=0
56 * { Contiene el nombre del alumno }
57 vnombres=space(40)
58 * { Contiene el apellido del alumno }
59 vapellidos=space(40)
60 * { Sirve para controlar si el alumno está seguro de reservar }
61 acceso=0

```



```

62 * { Contiene la hora del sistema }
63 sishora=0
64 * { Apunta al registro, donde el usuario ha reservado }
65 registro=0
66 * { Controla cuando el usuario solo ha reservado una hora, }
67 * { y desea reservar otra hora }
68 otrahora=0
69
70
71 * Inicio del Programa Principal
72
73 sishora=val(substr(time(),1,2))
74 DO CONTINUAR
75 IF acceso=1 and h_res<>2
76     CLEAR
77     DO FORMATO
78     DO ACTUALIZAR
79     IF sishora<7 or sishora>19
80         @23,02 SAY space(77)
81         DO MENS00 WITH " Ya no se pueden hacer Reservaciones, Estas fuer.
82         de Hora "
83     ELSE
84         DO RESERVACION
85         IF band=0
86             DO LEER
87             DO INFORME
88         ENDIF
89     ENDIF
90 ENDIF
91 CLOSE ALL
92 RETURN
93
94 * Fin del Programa Principal
95
96 * Area de Definición de Procedimientos
97
98
99 PROCEDURE LEER
100     * { Mantiene las horas disponibles del alumno }
101     vhorasdis=0
102     * { Sirve para cambiar el orden de las horas al momento de }
103     * { reservarlas }
104     vtemp=0
105     @23,05 SAY "CARNET : " GET vcarnet PICTURE "99-99999"
106     READ
107     SELECT 1
108         USE DATOSPER
109     SELECT 2
110         USE BASEHOY
111
112     * Abrimos base de datos DATOSPER.DBF *
113     SELECT 1
114
115         IF NOT FILE("CARNET.IDX")
116             INDEX ON CODIGO TO CARNET
117             SET INDEX TO CARNET
118         ELSE
119             SET INDEX TO CARNET
120             REINDEX
121     ENDIF
122     SEEK vcarnet
123     IF found()

```

```

124         vhorasdis=horas_dis-cant
125         vnombres=nombres
126         vapellidos=apellidos
127         REPLACE horas_dis WITH vhorasdis
128     ENDIF
129
130     * Abrimos Base de Datos BASEHOY.DBF *
131     SELECT 2
132         IF h_res=1
133             GO registro
134             IF hora1=vhora1
135                 DO MENSAJE WITH " Ya Reservaste la misma Hora "
136             ELSE
137                 cant=cant+1
138                 vhora2=vhora1
139                 vhora1=hora1
140                 nummaq2=nummaq1
141                 nummaq1=maquina1
142                 DELETE
143                 PACK
144             ENDIF
145         ENDIF
146         IF band<>2
147             IF (vhora1>vhora2) and vhora2<>0
148                 vtemp=vhora1
149                 vhora1=vhora2
150                 vhora2=vtemp
151                 vtemp=nummaq1
152                 nummaq1=nummaq2
153                 nummaq2=vtemp
154             ENDIF
155             APPEND BLANK
156             REPLACE codigo      WITH vcarnet
157             REPLACE horas_res  WITH cant
158             REPLACE hora1      WITH vhora1
159             REPLACE hora2      WITH vhora2
160             REPLACE maquina1   WITH nummaq1
161             REPLACE maquina2   WITH nummaq2
162             REPLACE fecha_uso  WITH date()
163         ENDIF
164     CLOSE ALL
165
166     RETURN
167
168     PROCEDURE RESERVACION
169         @23,02 SAY space(77)
170         do flechas with 13,61
171         IF h_res=1
172             @23,16 SAY "Introduzca Cantidad de Horas a Reservar : " GET cant;
173             PICTURE "9";
174             VALID(cant=1);
175             ERROR "Solo Puede Reservar 1 Hora"
176             READ
177         ENDIF
178         IF h_res=0
179             @23,16 SAY "Introduzca Cantidad de Horas a Reservar : " GET cant;
180             PICTURE "9";
181             VALID(cant=1 or cant=2);
182             ERROR "Solo Puede Reservar 1 ó 2 Horas"
183             READ
184         ENDIF
185     FOR i=1 TO cant

```

```

186     band=2
187     DO WHILE band=2
188         @23,02 SAY space(77)
189         do flechas with 18,58
190         @23,20 SAY " MENSAJE : Seleccione Hora a Reservar "
191         @06,09 PROMPT "7 am"
192         @06,14 PROMPT "8 am"
193         @06,19 PROMPT "9 am"
194         @06,24 PROMPT "10am"
195         @06,29 PROMPT "11am"
196         @06,34 PROMPT "12 m"
197         @06,39 PROMPT "1 pm"
198         @06,44 PROMPT "2 pm"
199         @06,49 PROMPT "3 pm"
200         @06,54 PROMPT "4 pm"
201         @06,59 PROMPT "5 pm"
202         @06,64 PROMPT "6 pm"
203         @06,69 PROMPT "7 pm"
204         @06,74 PROMPT "8 pm"
205     MENU TO amopm
206     IF i=1
207         vhora1 = amopm + 6
208         DO ACEPTAR_HORA WITH vhora1
209     ELSE
210         loc=1
211         vhora2 = amopm + 6
212         DO ACEPTAR_HORA WITH vhora2
213     ENDIF
214     IF vhora1=vhora2 or otrahora=vhora1
215         @23,02 SAY space(77)
216         DO MENS00 WITH " No Puedes Reservar la misma hora "
217         band=2
218     ELSE
219         IF band=0
220             DO RESER_HORA WITH amopm
221         ENDIF
222     ENDIF
223     ENDDO
224     ENDFOR
225     RETURN
226
227     PROCEDURE ACEPTAR_HORA
228     PARAMETER hora
229     vmin=0
230     vhour=0
231     vhour=val(substr(time(),1,2))
232     vmin=val(substr(time(),4,2))
233     IF hora>=vhour
234         IF hora=vhour
235             IF vmin>40
236                 band=1
237                 DO MENS00 WITH " Te pasaste 40 minutos de la Hora "
238             ENDIF
239         ENDIF
240     ELSE
241         band=1
242         DO MENS00 WITH " Hora no Disponible "
243     ENDIF
244     IF band=1
245         band=2
246     ELSE
247         band=0

```

```

1 * Programa para Reservación
2 * de Horas-Máquina
3 * Fecha de Creación : 20-Junio-1996
4 * Hora de Creación : 10:42 a.m.
5
6 * Desactivaciones Iniciales
7 SET STATUS OFF
8 SET TALK OFF
9 SET BELL OFF
10 SET CLOCK TO 01,60
11 SET DATE TO FRENCH
12
13 CLEAR
14 CLOSE ALL
15
16 * Area para Inicializar Variables
17 * Globales
18 amop=0
19 fila=00
20 control=0
21 nuareg=0
22 vhora1=0
23 vhora2=0
24 nummaq1=0
25 nummaq2=0
26 cant=0
27 loc=0
28 band=0
29 vcarnet=0
30 vnombres=space(40)
31 vapellidos=space(40)
32 acceso=0
33 h_res=0
34 otrahora=0
35 registro=0
36 vhoras_dis=0
37
38 * Inicio del programa
39
40 DO CONTINUAR
41 IF acceso=1
42 CLEAR
43 DO FORMATO
44 DO ACTUALIZAR
45 DO RESERVACION
46 DO LEER
47 DO INFORME
48 ENDIF
49 CLOSE ALL
50 RETURN
51
52 * fin del Programa
53
54 * Area de Definición de Procedimientos
55
56
57 PROCEDURE LEER
58 @23,05 SAY "CARNET : " GET vcarnet PICTURE "99-99999"
59 READ
60 SELECT 1
61 USE DATOSPER

```

```

62 SELECT 2
63 USE BASEMAÑ
64
65 * Abrimos Base de Datos DATOSPER.DBF *
66 SELECT 1
67
68 IF NOT FILE("CARNET.IDX")
69 INDEX ON CODIGO TO CARNET
70 SET INDEX TO CARNET
71 ELSE
72 SET INDEX TO CARNET
73 REINDEX
74 ENDIF
75 SEEK vcarnet
76 IF found()
77 vhorasdis=horas_dis-cant
78 vnombres=nombres
79 vapellidos=apellidos
80 REPLACE horas_dis WITH vhorasdis
81 ENDIF
82
83 * Abrimos Base de Datos BASEMAÑ.DBF *
84 SELECT 2
85
86 IF h_res=1
87 GO registro
88 cant=cant+1
89 vhora2=vhora1
90 vhora1=hora1
91 nummaq2=nummaq1
92 nummaq1=maquina1
93 DELETE
94 PACK
95 ENDIF
96 IF band<>2
97 IF (vhora1>vhora2) and vhora2<>0
98 vtemp=vhora1
99 vhora1=vhora2
100 vhora2=vtemp
101 vtemp=nummaq1
102 nummaq1=nummaq2
103 nummaq2=vtemp
104 ENDIF
105 APPEND BLANK
106 REPLACE codigo WITH vcarnet
107 REPLACE horas_res WITH cant
108 REPLACE hora1 WITH vhora1
109 REPLACE hora2 WITH vhora2
110 REPLACE maquina1 WITH nummaq1
111 REPLACE maquina2 WITH nummaq2
112 REPLACE fecha_uso WITH date()
113 ENDIF
114
115 CLOSE ALL
116 RETURN
117
118 PROCEDURE RESERVACION
119
120 @23,02 SAY space(77)
121 do flechas with 13,61
122 IF h_res=0
123 @23,16 SAY "Introduzca Cantidad de Horas a Reservar : " GET cant;

```

```

124 PICTURE "9";
125 VALID(cant=1 or cant=2);
126 ERROR " Solo Puede Reservar 1 ó 2 Horas "
127 READ
128 ENDIF
129 IF h_res=1
130 @23,16 SAY "Introduzca Cantidad de Horas a Reservar : " GET cant;
131 PICTURE "9";
132 VALID(cant=1);
133 ERROR " Solo Puede Reservar 1 Hora "
134 READ
135 ENDIF
136 FOR i=1 TO cant
137 band=2
138 DO WHILE band=2
139 @23,02 SAY space(77)
140 do flechas with 8,48
141 @23,10 SAY " MENSAJE : Seleccione Hora a Reservar "
142 @06,09 PROMPT "7 am"
143 @06,14 PROMPT "8 am"
144 @06,19 PROMPT "9 am"
145 @06,24 PROMPT "10am"
146 @06,29 PROMPT "11am"
147 @06,34 PROMPT "12 m"
148 @06,39 PROMPT "1 pm"
149 @06,44 PROMPT "2 pm"
150 @06,49 PROMPT "3 pm"
151 @06,54 PROMPT "4 pm"
152 @06,59 PROMPT "5 pm"
153 @06,64 PROMPT "6 pm"
154 @06,69 PROMPT "7 pm"
155 @06,74 PROMPT "8 pm"
156 MENU TO amopm
157 IF i=1
158 vhora1 = amopm + 6
159 DO ACEPTAR_HORA WITH vhora1
160 ELSE
161 loc=1
162 vhora2 = amopm + 6
163 DO ACEPTAR_HORA WITH vhora2
164 ENDIF
165 IF vhora1=vhora2 or otrahora=vhora1
166 @23,02 SAY space(77)
167 DO MENS00 WITH " No Puedes Reservar la misma hora "
168 band=2
169 ELSE
170 IF band=0
171 DO RESER_HORA WITH amopm
172 ENDIF
173 ENDIF
174 ENDDO
175 ENDFOR
176 RETURN
177
178 PROCEDURE ACEPTAR_HORA
179 PARAMETER hora
180 vmin=0
181 vhour=0
182 vhour=val(substr(time(),1,2))
183 vmin=val(substr(time(),4,2))
184 IF vhour<7 or vhour>19
185 band=1

```

```

124 PICTURE "9";
125 VALID(cant=1 or cant=2);
126 ERROR " Solo Puede Reservar 1 ó 2 Horas "
127 READ
128 ENDIF
129 IF h_res=1
130 @23,16 SAY "Introduzca Cantidad de Horas a Reservar : " GET cant;
131 PICTURE "9";
132 VALID(cant=1);
133 ERROR " Solo Puede Reservar 1 Hora "
134 READ
135 ENDIF
136 FOR i=1 TO cant
137 band=2
138 DO WHILE band=2
139 @23,02 SAY space(77)
140 do flechas with 8,48
141 @23,10 SAY " MENSAJE : Seleccione Hora a Reservar "
142 @06,09 PROMPT "7 am"
143 @06,14 PROMPT "8 am"
144 @06,19 PROMPT "9 am"
145 @06,24 PROMPT "10am"
146 @06,29 PROMPT "11am"
147 @06,34 PROMPT "12 m"
148 @06,39 PROMPT "1 pm"
149 @06,44 PROMPT "2 pm"
150 @06,49 PROMPT "3 pm"
151 @06,54 PROMPT "4 pm"
152 @06,59 PROMPT "5 pm"
153 @06,64 PROMPT "6 pm"
154 @06,69 PROMPT "7 pm"
155 @06,74 PROMPT "8 pm"
156 MENU TO amopm
157 IF i=1
158 vhora1 = amopm + 6
159 DO ACEPTAR_HORA WITH vhora1
160 ELSE
161 loc=1
162 vhora2 = amopm + 6
163 DO ACEPTAR_HORA WITH vhora2
164 ENDIF
165 IF vhora1=vhora2 or otrahora=vhora1
166 @23,02 SAY space(77)
167 DO MENS00 WITH " No Puedes Reservar la misma hora "
168 band=2
169 ELSE
170 IF band=0
171 DO RESER_HORA WITH amopm
172 ENDIF
173 ENDIF
174 ENDDO
175 ENDFOR
176 RETURN
177
178 PROCEDURE ACEPTAR_HORA
179 PARAMETER hora
180 vmin=0
181 vhour=0
182 vhour=val(substr(time(),1,2))
183 vmin=val(substr(time(),4,2))
184 IF vhour<7 or vhour>19
185 band=1

```

```
'6      @23,@2 SAY space(77)
'7      DO MENS@ WITH " Ya no se pueden hacer Reservasiones, Estas fuera
'8      de Hora "
'9      ENDIF
'0      IF band=1
'1          band=2
'2      ELSE
'3          band=@
'4      ENDIF
'5      RETURN
'6
'7
'8
```



```

1 SET TALK OFF
2 SET STATUS OFF
3 SET BELL OFF
4
5 CLEAR
6
7 vcarnet=0
8 sishora=0
9 sismin=0
10 vhora1=0
11 vhora2=0
12 maq1=0
13 maq2=0
14 vcant=0
15
16 CLOSE ALL
17 @12,20 SAY "Carnet : " GET vcarnet PICTURE "99-99999"
18 READ
19 USE BASEHOY
20 IF not file("CARNET01.IDX")
21     INDEX ON codigo TO CARNET01
22     SET INDEX TO CARNET01
23 ELSE
24     SET INDEX TO CARNET01
25     REINDEX
26 ENDIF
27 SEEK vcarnet
28 IF found()
29     sishora = val(substr(time(),1,2))
30     sismin = val(substr(time(),4,2))
31     vhora1 = hora1
32     vhora2 = hora2
33     maq1 = maquina1
34     maq2 = maquina2
35     vcant = horas_res
36     IF vcant=1
37         DO acceso WITH vhora1
38     ELSE
39         IF vhora1<vhora2
40             DO acceso WITH vhora1
41         ELSE
42             DO acceso WITH vhora2
43         ENDIF
44     ENDIF
45 ELSE
46     @22,20 SAY "No has Reservado Horas"
47     READ
48 ENDIF
49 CLOSE DATABASE
50 CLOSE ALL
51 RETURN
52
53 PROCEDURE acceso
54 PARAMETER hora
55 IF sishora=hora
56     IF sismin<=30
57         REPLACE MARCA WITH "E"
58         @12,20 SAY "Puedes Entrar"
59         READ
60     ELSE
61         @12,20 SAY "Finaliza tu hora, porque la has perdido"

```

```
62     READ
63     ENDF
64 ENDF
65 IF sishora<hora
66     @12,20 SAY "Espere hasta que llegue su hora"
67     READ
68 ENDF
69 IF sishora>hora
70     @12,20 SAY "Finaliza tu hora"
71     READ
72 ENDF
73 RETURN
```

```

1 * Programa que termina las horas
2 * que el usuario ha reservado
3
4 SET TALK OFF
5 SET STATUS OFF
6 SET BELL OFF
7 SET DELETED ON
8
9 CLEAR
10
11 vcarnet=0
12 vcant=0
13 vhora1=0
14 vhora2=0
15 sishora=0
16 sismin=0
17 descon=0
18 vtemp=0
19
20 CLOSE ALL
21 @12,25 SAY "Carnet : " GET vcarnet PICTURE "99-99999"
22 READ
23 SELECT 1
24 USE BASEHOY
25 SELECT 2
26 USE DATOSPER
27
28 SELECT 1
29 IF not file("carnet02.idx")
30 INDEX ON codigo TO carnet02
31 SET INDEX TO carnet02
32 ELSE
33 SET INDEX TO carnet02
34 REINDEX
35 ENDIF
36 SEEK vcarnet
37 IF found()
38 sishora=val(substr(time(),1,2))
39 sismin=val(substr(time(),4,2))
40 vcant=horas_res
41 vhora1=hora1
42 vhora2=hora2
43 IF vcant=1
44 DO finhora WITH vhora1
45 ELSE
46 DO finhoras WITH vhora1,vhora2
47 ENDIF
48 ELSE
49 @23,20 SAY " No has Reservado Horas "
50 READ
51 ENDIF
52
53 CLOSE ALL
54 RETURN
55 * Fin del Programa Principal *
56
57 * Area de Definición de Procedimientos *
58
59 PROCEDURE finhora
60 PARAMETER hora
61 IF hora<>0

```

```

62 IF (sishora-hora)=0
63   IF MARCA="E"
64     DO terminar_horas WITH 0,1
65   ELSE
66     IF sismin<=30
67       @23,02 SAY "No has marcado el inicio de tu hora"
68       READ
69     ELSE
70       IF vcant=1
71         DO terminar_horas WITH 0,1
72         @23,20 SAY "Perdiste la hora"
73         READ
74       ENDIF
75     ENDIF
76   ENDIF
77 ENDIF
78 IF (sishora-hora)>=1
79   IF sismin<=15
80     IF MARCA="E"
81       descon=sishora-hora-1
82       DO terminar_horas WITH descon,1
83     ELSE
84       descon=sishora-hora-1
85       DO terminar_horas WITH (descon+5),1
86     ENDIF
87   ELSE
88     IF MARCA="E"
89       descon=sishora-hora
90       DO terminar_horas WITH descon,1
91     ELSE
92       descon=sishora-hora
93       DO terminar_horas WITH (descon+5),1
94     ENDIF
95   ENDIF
96 ENDIF
97 IF (sishora-hora)<0
98   @23,20 SAY "Aviso : todavia no"
99   READ
100  ENDIF
101 ENDIF
102 RETURN
103
104 PROCEDURE finhoras
105 PARAMETER hora01,hora02
106 control=0
107 IF (hora02-hora01)=1
108   IF (sishora-hora01)=0 OR (sishora-hora01)=1
109     IF MARCA="E"
110       DO terminar_horas WITH 0,2
111     ELSE
112       IF (sishora-hora01)=0
113         IF sismin<=30
114           @23,02 SAY "No has marcado el inicio de tus horas"
115           READ
116         ELSE
117           . DO terminar_horas WITH 5,1
118         ENDIF
119       ENDIF
120     IF (sishora-hora01)=1
121       DO terminar_horas WITH 5,2
122     ENDIF
123   ENDIF

```

```

124   ENDIF
125   IF (sishora-hora01)>=2
126     IF sismin<=30
127       IF MARCA="E"
128         descon=sishora-hora01-2
129         DO terminar_horas WITH descon,2
130       ELSE
131         descon=sishora-hora01-2
132         DO terminar_horas WITH (descon+5),2
133       ENDIF
134     ELSE
135       IF MARCA="E"
136         descon=sishora-hora01-1
137         DO terminar_horas WITH descon,2
138       ELSE
139         descon=sishora-hora01-1
140         DO terminar_horas WITH (descon+5),2
141       ENDIF
142     ENDIF
143   ENDIF
144   IF (sishora-hora01)<0
145     @23,20 SAY "Aviso : Todavia no puedes entrar"
146     READ
147   ENDIF
148 ELSE
149   IF sishora=hora01
150     IF MARCA="E"
151       DO finhora WITH hora01
152     ELSE
153       IF sismin<=30
154         @23,20 SAY "No has marcado el inicio de tus horas"
155         READ
156       ELSE
157         DO terminar_horas WITH 5,1
158       ENDIF
159     ENDIF
160   ENDIF
161   IF sishora>hora01 and sishora<hora02
162     IF sismin<=30
163       IF MARCA="E"
164         control=sishora-hora01-1
165         DO terminar_horas WITH control,1
166         hora01=0
167       ELSE
168         control=sishora-hora01-1
169         DO terminar_horas WITH (control+5),1
170         hora01=0
171       ENDIF
172     ELSE
173       IF MARCA="E"
174         control=sishora-hora01
175         DO terminar_horas WITH control,1
176         hora01=0
177       ELSE
178         control=sishora-hora01
179         DO terminar_horas WITH (control+5),1
180         hora01=0
181       ENDIF
182     ENDIF
183   ENDIF
184   IF sishora=hora02 and hora01=0
185     IF MARCA="E"

```

```

186         DO finhora WITH hora02
187     ELSE
188         @23,20 SAY "no marcaste hora"
189         READ
190     ENDIF
191 ENDIF
192 IF sishora=hora02 and hora01<>0
193     IF sismin<=30
194         IF MARCA="E"
195             control=hora02-hora01-1
196             DO terminar_horas WITH control,1
197         ELSE
198             control=hora02-hora01-1
199             DO terminar_horas WITH (control+5),1
200         ENDIF
201     ELSE
202         IF MARCA="E"
203             control=hora02-hora01-1
204             DO terminar_horas WITH control,2
205         ELSE
206             control=hora02-hora01-1
207             DO terminar_horas WITH (control+5),2
208         ENDIF
209     ENDIF
210 ENDIF
211 IF sishora>hora02 and hora01<>0
212     IF sismin<=30
213         IF MARCA="E"
214             control=sishora-hora01-2
215             DO terminar_horas WITH control,2
216         ELSE
217             control=sishora-hora01-2
218             DO terminar_horas WITH (control+5),2
219         ENDIF
220     ELSE
221         IF MARCA="E"
222             control=sishora-hora01-1
223             DO terminar_horas WITH control,2
224         ELSE
225             control=sishora-hora01-1
226             DO terminar_horas WITH (control+5),2
227         ENDIF
228     ENDIF
229 ENDIF
230 IF (sishora-hora01)<0
231     @23,20 SAY " Aviso : Todavía no puedes entrar "
232     READ
233 ENDIF
234 ENDIF
235 RETURN
236
237 PROCEDURE terminar_horas
238 PARAMETER terdes,canthoras
239     vh_dis=0
240     mult_act=0
241     horactual=0
242     horactual=horas_res-canthoras
243     IF canthoras=1
244         IF vcant=1
245             REPLACE hora1     WITH 0
246             REPLACE hora2     WITH 0
247         ELSE

```

```

248     REPLACE hora1     WITH hora02
249     REPLACE hora2     WITH 0
250     REPLACE maquina1  WITH maquina2
251     REPLACE maquina2  WITH 0
252     REPLACE horas_res  WITH horactua1
253     REPLACE marca     WITH " "
254     ENDIF
255     ENDIF
256     IF canthoras=2
257         REPLACE hora1     WITH 0
258         REPLACE hora2     WITH 0
259         REPLACE maquina1  WITH maquina2
260         REPLACE maquina2  WITH 0
261         REPLACE marca     WITH " "
262     ENDIF
263     IF hora1=0 and hora2=0
264         DELETE
265     ENDIF
266
267     SELECT 2
268         IF not file("carnet.idx")
269             INDEX ON codigo TO carnet
270             SET INDEX TO carnet
271         ELSE
272             SET INDEX TO carnet
273             REINDEX
274         ENDIF
275         SEEK vcarnet
276         IF found()
277             IF terdes>0
278                 IF horas_dis=0
279                     mult_act=10*terdes
280                     @22,23 SAY "Tienes una multa de "
281                     @22,48 SAY mult_act PICTURE "$ 99.99"
282                     REPLACE multas WITH mult_act
283                     READ
284                 ELSE
285                     vh_dis=horas_dis
286                     vh_dis=vh_dis-terdes
287                     IF vh_dis<0
288                         REPLACE horas_dis WITH 0
289                         mult_act=ABS(vh_dis)*10
290                         REPLACE multas WITH mult_act
291                         @22,23 SAY "Tienes una multa de "
292                         @22,48 SAY mult_act PICTURE "$ 99.99"
293                         READ
294                     ELSE
295                         REPLACE horas_dis WITH vh_dis
296                     ENDIF
297                 ENDIF
298             ENDIF
299         ELSE
300             @12,50 SAY "No existe"
301             READ
302         ENDIF
303     CLOSE ALL
304
305     RETURN

```

ANEXO E

Programas en Pascal 6.0
Lectura y Escritura al Puerto Serie.


```

{ Programa en Pascal, que permite leer los datos }
{ del Puerto Serie }

PROGRAM Lectura;
USES crt,Dos;

TYPE
  tipoarray = ARRAY[1..11] OF byte;

VAR
  archivo:text;
  stringa:string[22];
  cadena:string[2];
  registros:Registers;
  lista:tipoarray;
  A,B,k:integer;
  deci1, deci2: real;

PROCEDURE rep(A1:integer; B1:integer);
var
  indice:integer;

  Begin
    while registros.ax<>$9B do
      intr($14, registros);
    while registros.ax=$9B do
      intr($14, registros);
    for indice:=A1 to B1 do
      begin
        lista[indice]:=LO(registros.AX);
        intr($14, registros);
      end;
  End;

BEGIN { inicio del programa principal }
  A:=1;
  B:=6;
  registros.AX:=3; { configuramos com2, 8 bits, 1 stop }
  registros.DX:=1;
  intr($14, registros);
  registros.AX:=$01FE;
  registros.DX:=1;
  intr($14, registros);
  while B<=11 do
    begin
      rep(A,B);
      B:=B+5;
      A:=A+6;
    end;
  stringa:='';
  for k:=1 to 11 do
    begin
      STR(lista[k], cadena);
      if lista[k]<10 then
        INSERT('0', cadena, 1);
      stringa:=stringa+cadena;
    end;

  Assign(archivo, 'cadena.txt'); { Escribiendo stringa en Cadena.txt }
  Rewrite(archivo);
  writeln(archivo,stringa);
  close(archivo);

```

```
{ Programa en Pascal, que permite enviar datos }  
{ al Puerto Serie }
```

```
PROGRAM Escritura;  
USES crt,Dos;
```

```
TYPE  
  tipoarray = ARRAY[1..11] OF byte;
```

```
VAR  
  archivo:text;  
  stringa:string[22];  
  registros:Registers;  
  lista:tipoarray;  
  A,B,H,N,D1,D2,R,cod:integer;
```

```
BEGIN { inicio del programa principal }  
  Assign(archivo, 'cadena.txt'); { leyendo archivo cadena.txt }  
  Reset(archivo);  
  readln(archivo, stringa);  
  close(archivo);  
  
  registros.AX:=3; { configuramos com2, 8 bits, 1 stop }  
  registros.DX:=1;  
  intr($14, registros);  
  A:=1;  
  B:=6;  
  H:=1;  
  for N:=1 to 11 do  
    begin  
      val(stringa[H],D1,cod);  
      val(stringa[H+1],D2,cod);  
      lista[N]:=D1*10+D2;  
      H:=H+2;  
    end;  
  registros.AX:=$01FE;  
  registros.DX:=$01;  
  intr($14, registros);  
  delay(5000);  
  for R:=1 to 3 do  
    begin  
      registros.AX:=$019B;  
      intr($14, registros);  
    end;  
  while B<=11 do  
    begin  
      for N:=A to B do  
        begin  
          registros.AX:=lista[N];  
          intr($14, registros);  
        end;  
      A:=A+6;  
      B:=B+5;  
    end;  
END. { fin del programa }
```

CONCEPTOS DE SOFTWARE

ACCESO DIRECTO: Es una forma de búsqueda de los registros de una base de datos, el cual consiste en el acceso más rápido y eficiente a dichos registros a través de una llave, es decir que los datos son buscados por la dirección de memoria que tienen.

ACCESO SECUENCIAL: Es una búsqueda, la cual consiste en ir recorriendo los registros de una base de datos uno a uno hasta encontrar el deseado (no es un acceso por llave).

ADMINISTRACION DE BASE DE DATOS: Consiste en mejorar de manera óptima el acceso a los registros de una base de datos y poder eliminarlos. También el administrar implica dar un ordenamiento lógico de los datos, y que dicho ordenamiento pueda ser usado y relacionado con otras entidades, o que sirvan para el adecuado procesamiento de los datos.

ARCHIVO DE DATOS : Es un conjunto de información de cualquier tipo, que se encuentra almacenada de modo permanente, para ser accedido o alterado cuando sea necesario.

ARCHIVO DE TEXTO : Es aquel tipo de archivo que contiene información en forma de caracteres consecutivos.

BASE DE DATOS: Es el conjunto de datos los cuales están organizados por filas y columnas, a las columnas se les conoce como "campos" y a las filas como "registros".

El objetivo de estar organizados de esta forma es el de generar información en un momento determinado y en base a dicha información poder tomar decisiones.

BASE DE DATOS RELACIONALES: El concepto de relacional es bastante nuevo en el uso y manejo de las bases de datos. Una base de datos relacional es la forma en que ésta interactura o se fusiona con otras bases de datos a la vez. Para hacer acceso a una base de datos relacional es necesario tener una llave principal para el control de las mismas.

DISEÑO DE BASE DE DATOS : Consiste en identificar todos los campos que serán incorporados a la base de datos. Cada campo esta caracterizado por algún tipo de dato y el nombre de cada campo deberá ser diferente a los demás.

INDEXAR: (indicar) Este término es muy utilizado en el manejo de base de datos, el cual produce una forma más directa de acceder los datos. Al hablar de indexar, nos estamos refiriendo a que vamos a indicar un campo (llave) o campos (llaves) específicos de una base de datos para hacer el acceso directo.

MANTENIMIENTO DE BASE DE DATOS: Es el proceso por el cual los registros de una o varias base de datos pueden ser consultados, modificados o eliminados. También se posee la capacidad de agregar nuevos registros a la base de datos.

CONCEPTOS DE HARDWARE

CABEZAS MAGNETICAS

Las cabezas magnéticas son dispositivos transductores capaces de transformar el campo magnético contenido en una cinta con base de plástico revestida con una emulsión magnética, en una señal eléctrica o viceversa.

• REGISTRO EN UNA CINTA MAGNETICA.

Cuando se aplica una tensión alterna a la bobina de una cabeza grabadora, se crea en el núcleo, de ésta un campo magnético también alterno. En el entrehierro, las líneas de fuerza se extienden en forma de arco más allá del núcleo. Con ellos algunas de estas líneas siguen el camino de la película magnética de la cinta, que corre frente a ellas, teniendo lugar la orientación de sus partículas, que puede asemejarse a pequeños imanes, en la dirección del campo magnético. De este modo se forman sobre la cinta, previamente desimantada, una cadena o serie de tramos magnéticos de intensidad y sentidos alternos, que constituyen la grabación del curso temporal de las oscilaciones eléctricas (información).

En la figura H1 se muestra la orientación de los imanes elementales en función de la corriente en la bobina, así como el correspondiente curso de la inducción en la cinta. Con una velocidad de la cinta de 9.5 cm/s y a una frecuencia de 1kHz, la longitud "a" de la oscilación completa representada en la figura es de:

$$a = (9.5 \text{ cm/s}) / 1000 \text{ kHz}$$

$$a = 95 \cdot \mu \text{ m}$$

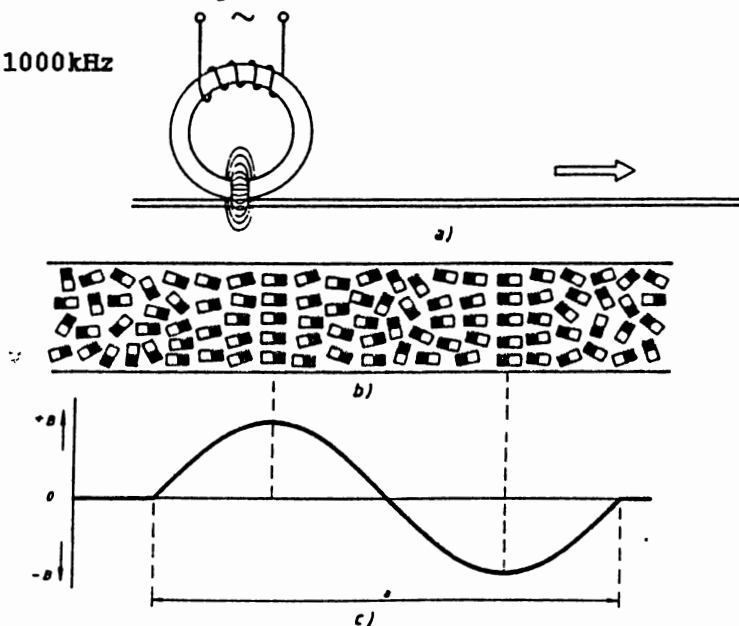


FIGURA H1: Ordenamiento de los fragmentos magnéticos de una cinta.

En este pequeño trozo de cinta encontraremos centenares de millones de minúsculos imanes elementales orientados como se muestra en la figura H1.

ESTANDAR RS-232C

Cuando se transmiten datos en serie con el estándar RS-232, se transmiten pulsos de voltaje a la velocidad seleccionada de baudios, las velocidades normalizadas son de 50, 75, 110, 134.5, 150, 300, 600, 1200, 1800, 2400, 3600, 4800 y 98000 baudios. Dichos pulsos se usan para representar los datos digitales (RS significa "recommended standard", estándar recomendado). El "1" lógico se representa por un nivel de voltaje transmitido en el rango de -3V a -15V, mientras que el "0" lógico se representa mediante un nivel de voltaje transmitido de +3V a + 15V. (Nótese que el RS-232C emplea una convención de lógica negativa, que tampoco es compatible con TTL, para representar sus señales de datos). Los datos digitales se pueden transmitir en serie a distancias hasta de 15 metros según el estándar RS-232C. Es el método más empleado en la transmisión en serie. También la norma RS-232C ha sido adoptada por la Electronic Industries Association (EIA) como un estándar oficial para comunicación en serie. Además de especificar los niveles de voltaje y la distancia máxima de transmisión de señales digitales de datos, el estándar define muchas otras características requeridas para conformar una interfase de datos , incluyendo la designación de pines del conector y demás consideraciones eléctricas para los dispositivos transmisores y receptores. Describiremos como el estándar RS-232C se desarrolló; esto ayudará a explicar la base de las especificaciones normativas que resultaron.

Cuando la transmisión de datos digitales asíncrona en serie estaba en su infancia, se consideraron las líneas telefónicas públicas como

eslabones de datos potenciales de longitud indefinida para estas transmisiones. Sin embargo, se notó que esas líneas telefónicas se establecieron originalmente para transportar señales de voz y no altos volúmenes de datos digitales en alta velocidad. Además, a las compañías telefónicas no les agradaba el prospecto de tener que manejar todo tipo de señales digitales extrañas en sus redes. Así, se hizo obvio que se debería crear una norma para hacer los datos digitales compatibles con la transmisión por las líneas telefónicas.

La norma resultante fue la EIA RS-232C más específicamente, el estándar fue formulado de modo que hubiese un conjunto uniforme de características eléctricas y mecánicas de interfase, al igual que una descripción funcional de los circuitos de intercambio. El equipo terminal de datos DTE incluye dispositivos tales como terminales de computadora, instrumentos digitales, teleimpresoras y computadoras. El equipo de comunicación de datos DCE incluye la categoría de dispositivos que codifican los datos digitales en forma de señales semejantes a la voz, permisibles en el sistema telefónico. Así, es el equipo DCE el que realmente convierte los datos digitales en ondas de señal compatibles con la transmisión por línea telefónica. El estándar RS-232C especifica el modo en el que se debe presentar los datos del equipo DTE al DCE, de modo que este último puede llevar a cabo su función correctamente.

El sistema BELL introdujo el Modem BELL 103 (modulador/demodulador) como el primer equipo del DCE que respondería a cualquier equipo DTE que se apegara a la norma RS-232C .

Aunque este paraje histórico ayuda a explicar el origen de la norma RS-232C, no explica plenamente por qué se adoptó la norma para la transmisión asíncrona de datos en serie en muchos otros sistemas digitales. La respuesta es que RS-232C puede trabajar tan correctamente

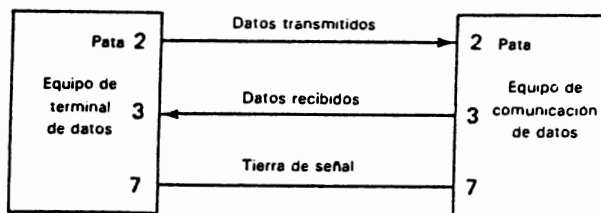
para transmitir datos digitales en forma en serie entre dos dispositivos digitales cualesquiera, sin importar si uno de ellos es un equipo DCE o no. Siempre que ambos equipos digitales sean aptos para transmitir y recibir de acuerdo con las especificaciones RS-232C, los datos digitales se pueden mandar en serie entre ellos obedeciendo al estándar.

Por lo tanto, podemos resumir que la RS-232C es un estándar que define los enlaces físicos y eléctricos de los sistemas para permitir transmisión de datos asíncronos en serie a través de :

1. Líneas telefónicas equipadas con modem
2. En sistemas digitales en los que la distancia de cableado sea menor de 15 metros.

Las especificaciones y características mas importantes que definen la RS-232C comprenden las siguientes:

1. Señales de voltaje que emplean una convención lógica negativa definen los niveles de señal de datos. Esto es, si se consideran las señales de datos como "1" lógico -MARCA cuando el voltaje, V_1 es de -3 a -15 V. Se consideran las señales de datos como "0" lógico = ESPACIO cuando el voltaje, V_1 es +3 a +15 V.
2. Por otra parte, los niveles de señal de control usan una lógica verdadera positiva (señal lógica de control "1" = +3 a +15 V y señal lógica de control "0" = -3 a -15 V . La mezcla de lógica positiva y negativa en la misma interconexión puede ser un punto de confusión a menos que el usuario este plenamente consciente.
3. Hay tres conductores que se usan para transmitir datos : dos son conductores portadores de datos (uno para los datos que se transmiten y uno para los que se reciben) y el tercero es un conductor de tierra de señal que sirve como camino de retorno de la corriente de la señal (fig. H2).



La RS-232C usa tres conductores para la transmisión de datos entre equipo de terminal de datos (DTE) y equipo de comunicación de datos (DCE). Los demás conectores asociados con la interconexión son líneas de control de señal.

FIGURA H2: La RS-232C usa tres conductores para la transmisión de datos.

También hay otros 22 conductores que sirven como conductores de control entre el DTE y el DCE. Los pines normalizados de las conexiones especificadas por la RS-232C se demuestran en la figura H2a. Los pines 2, 3 y 7 accesan las líneas que se usan para la transmisión de datos. Los pines restantes accesan las líneas de señales de control. La norma RS-232C especifica que la parte macho de un conector está en el lado del equipo de transmisión y la hembra del lado del equipo de comunicación.

FUNIONAMIENTO DE LAS CABEZAS MAGNETICAS .

- Cabeza grabadora:

La bobina de la cabeza recibe variaciones de corriente eléctrica cuya amplitud y frecuencia dependen de la señal de audio. Dichas variaciones de corriente producen un campo magnético en el entrehierro, campo que varía igualmente en intensidad y polaridad de acuerdo con los cambios de dicha señal. Si estando éste campo magnético generado en la bobina se hace desplazar por delante del entrehierro una cinta en la que

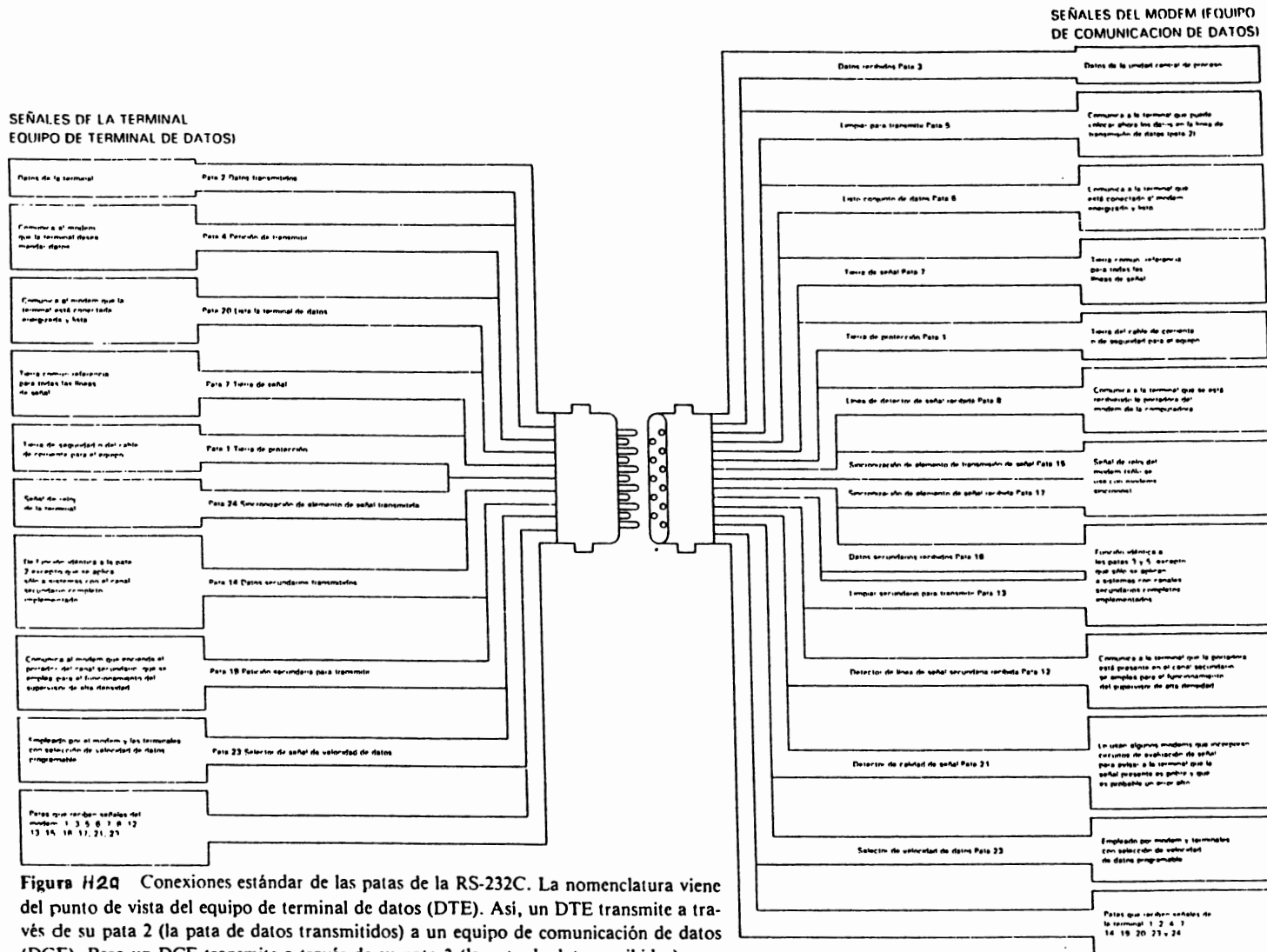


Figura H20 Conexiones estándar de las patas de la RS-232C. La nomenclatura viene del punto de vista del equipo de terminal de datos (DTE). Así, un DTE transmite a través de su pata 2 (la pata de datos transmitidos) a un equipo de comunicación de datos (DCE). Pero un DCE transmite a través de su pata 3 (la pata de datos recibidos) a un DTE. Los fabricantes de los instrumentos con norma RS-232C pueden ofrecer cables que permitan que sus instrumentos se vean como un DTE o como un DCE; sin embargo, generalmente no se tiene opción. Al emplear instrumentos semejantes, se puede tener que armar un cable de conductores cruzados para obtener las señales en los conductores correctos. (Cortesía de Tektronix, Inc.)

se han depositado uniformemente finísimas partículas de material magnético, dichas partículas irán adquiriendo un estado de magnetización impuesto por las líneas de flujo según sea la imanación del núcleo en cada instante. El resultado de todo ello es una variación magnética, impresa en la cinta en movimiento, y que corresponde a los cambios de la señal de audio.

Es muy importante que el campo magnético alcance el valor máximo en el centro del entrehierro y caiga de forma muy rápida en las proximidades de dicho entrehierro de lo contrario la magnetización de un sector interferirá con la del ya grabado.

Si se alimenta la cabeza grabadora exclusivamente con la tensión de la señal, se producen distorsiones debido a la relación no lineal entre intensidad de campo magnético H e inductancia magnética B , tal y como se muestra en la figura H3.

La curva de trabajo resulta ser la conocida curva de histéresis. En las flechas indicadas en la figura H3 se da una deformación de la señal, que da como resultado la desaparición de las señales débiles y el recorte de las crestas de onda de las señales fuertes.

Existen métodos para evitar este fenómeno de histéresis. El utilizado en el proyecto es el de polarización por corriente continua ya que no nos interesa la pérdida de fidelidad para señales de audio ya que solo utilizamos la frecuencia.

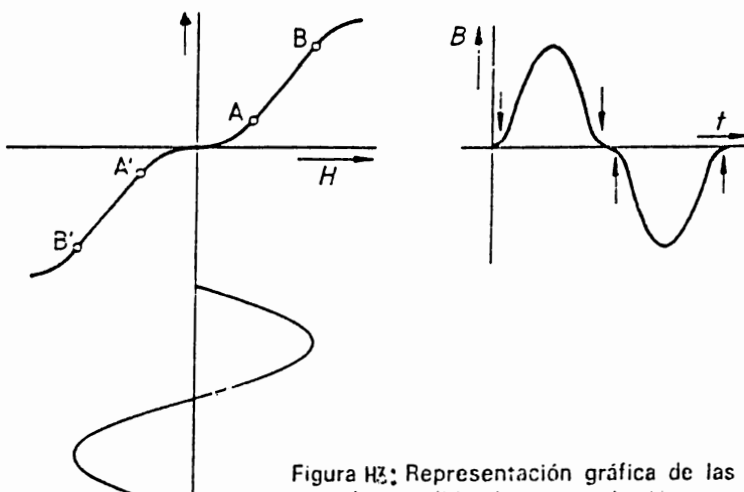


Figura H3: Representación gráfica de las señales de entrada y salida de una grabación magnética efec-

Este sistema de polarización consiste en añadir una corriente continua a la cabeza de registro, de esta forma se desplaza la línea de cero de la tensión de la señal a grabar más allá del tramo no lineal de la curva de imantación, lo cual da que la señal no se distorsione tal como se muestra en la siguiente figura (fig. H4).

• CABEZAS REPRODUCTORAS.

Las cabezas magnéticas lectoras están construidas de forma análoga a las cabezas grabadoras, de tal forma que en muchos equipos se utiliza una sola cabeza para la grabación y la reproducción (tal es nuestro caso).

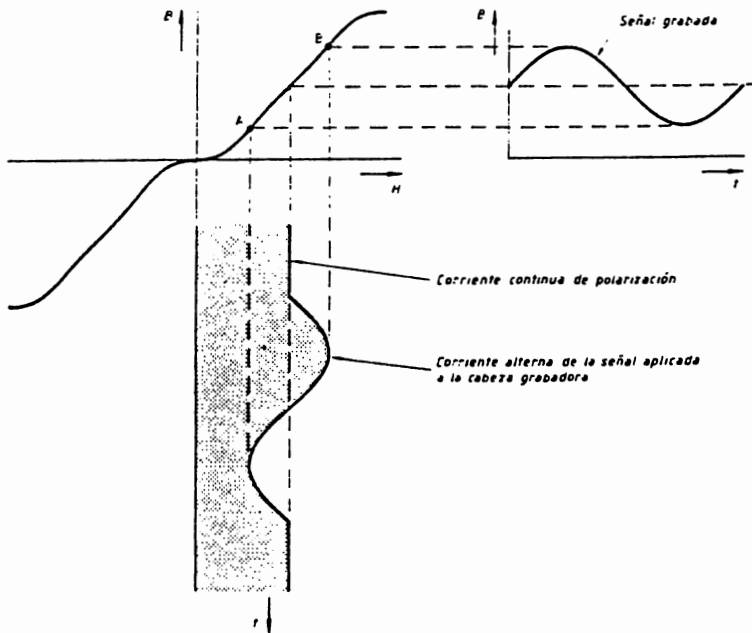


FIGURA H4: Grabación efectuada con una polarización por corriente continua.

Las cualidades que ha de tener una cabeza reproductora son:

1. El magnetismo remanente de la cabeza reproductora ha de ser el mínimo posible, con el fin de conferir una gran proporcionalidad entre el campo magnético de la cinta y la tensión inducida en la bobina.

2. La frecuencia propia de resonancia de la cabeza ha de ser lo más elevada posible y en todo caso superior a la frecuencia más elevada que deba reproducir.
3. La cabeza debe estar aterrizada al chasis con el fin de que quede protegida contra la acción de campos magnéticos parásitos.
4. La cabeza ha de ser resistente al desgaste producido por la fricción de la cinta.

La lectura de la cinta se realiza como sigue: la cinta, previamente grabada, contiene partículas de óxido de hierro más o menos magnetizadas por la cabeza grabadora. Al pasar por la cabeza reproductora las líneas de fuerza magnética de las partículas mencionadas encuentran un medio de conducción efectivo a través de la bobina de la cabeza y tienden a concentrarse ahí. El movimiento de la cinta provoca cambios en la dirección y fuerza del campo magnético en la cabeza, induciendo las tensiones correspondientes en la bobina, todo de acuerdo a la Ley de Faraday. tales tensiones son proporcionales al grado de magnetización y también a la velocidad de variación del campo a través de la cabeza, siendo por tanto proporcional a la velocidad a la que pasan por las partículas magnéticas contenidas en la cinta.

- **CABEZA DE BORRADO.**

El borrado de una cinta lo realiza una cabeza que es alimentada por una tensión continua o por una tensión alterna procedente de un oscilador.

La cabeza de borrado tiene un forma constructiva parecida a la de las cabezas de reproducción y grabación, si bien las exigencias técnicas de una cabeza de borrado son mucho menos críticas pues su función es grabar la cinta con una frecuencia fija elevada y por lo tanto no se producen cambios de impedancia.

La frecuencia de la tensión alterna de borrado está situado por encima del límite de audibilidad, por ejemplo 60 kHz.

La cabeza de borrado se dispone inmediatamente antes de la cabeza de grabación quedando conectada al oscilador de alta frecuencia con lo que se borra la antigua grabación antes de que se aplique la nueva.

- **ALINEACION DE LAS CABEZAS MAGNETICAS.**

Las cabezas magnéticas han de estar correctamente alineadas con respecto a la cinta, con el fin de obtener de ellas el máximo rendimiento, tanto en la grabación como en la reproducción.

Las alineaciones a considerar en toda cabeza magnética son las siguientes :

- Azimut.
- Altura.
- Angulo
- Tangencia.
- Contacto.

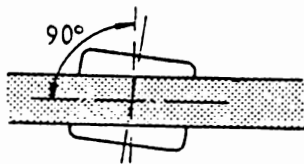
Un incorrecto ajuste del azimut origina una pérdida de las frecuencias altas, o una incorrecta decodificación en las cintas cuadrafónicas matriciales.

Si el entrehierro no está perfectamente ajustado en su altura con respecto a la cinta, no se aprovecha todo el campo magnético creado entre los polos de la cabeza o contenido de la cinta, por lo que la grabación y la reproducción se hacen deficientes.

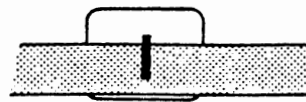
Lo mismo sucede si la cinta forma ángulo con respecto a la cabeza, con la particularidad de que esta última es la que se desgasta irregularmente por efecto de un roce de la cinta no uniforme sobre la superficie de la cabeza.

Otro desgaste irregular de la cabeza es causado por un deficiente ajuste del ángulo de tangencia.

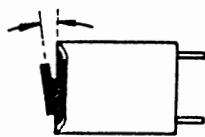
Finalmente un contacto deficiente entre cinta y cabeza se traduce en una reducción de la intensidad de campo que alcanza a la cinta y viceversa. Lo anterior se muestra en forma esquemática en la figura H5.



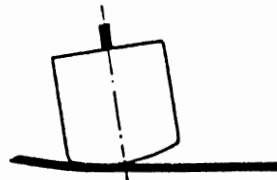
Error de azimut.



Error de altura.



Error de ángulo.



Error de tangencia.

Error de contacto.

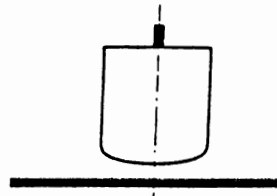


FIGURA H5: Errores en la instalación de una cabeza magnética.

TRANSMISION DE DATOS EN SERIE.

La transferencia de datos en paralelo en ocho líneas que conectan dos dispositivos para la comunicación es una manera rápida de movilizar datos. Sin embargo, no resulta práctico tener ocho o más líneas cuando dos dispositivos están alejados entre sí. A menudo es mejor reemplazar la transferencia de datos en paralelo por la transferencia de datos en serie en estas situaciones. De este modo se puede usar una sola línea para enlazar ambos dispositivos. Esto se conoce como un sistema semidúplex. Ambos dispositivos pueden enviar o recibir datos pero no al mismo tiempo. Un sistema dúplex total utiliza dos líneas, o el equivalente electrónico de dos líneas de modo que ambos dispositivos puedan enviar o recibir datos al mismo tiempo.

Una línea de datos en serie puede portar datos serializados sincrónicos o asincrónicos. La transferencia de datos en serie sincrónica consiste en una corriente constante de BITS. La transferencia asincrónica de datos en serie consiste en una transferencia de BITS solamente cuando un dispositivo transmisor tiene BITS para enviar. El grupo de BITS enviados se conoce como un caracter. El caracter es una serie de impulsos que son altos o bajos para un intervalo, o tiempo de BIT (BIT TIME). En la figura siguiente (figura H6) se muestra un caracter de datos en serie asincrónico típico.

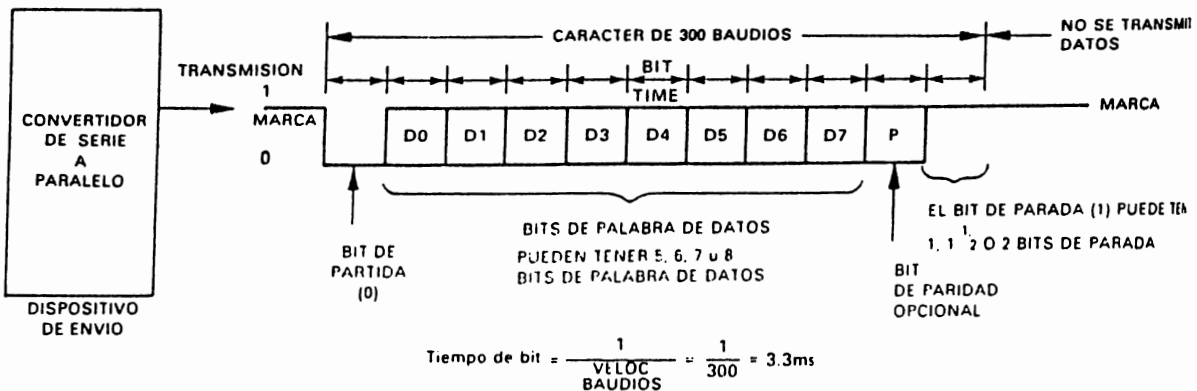
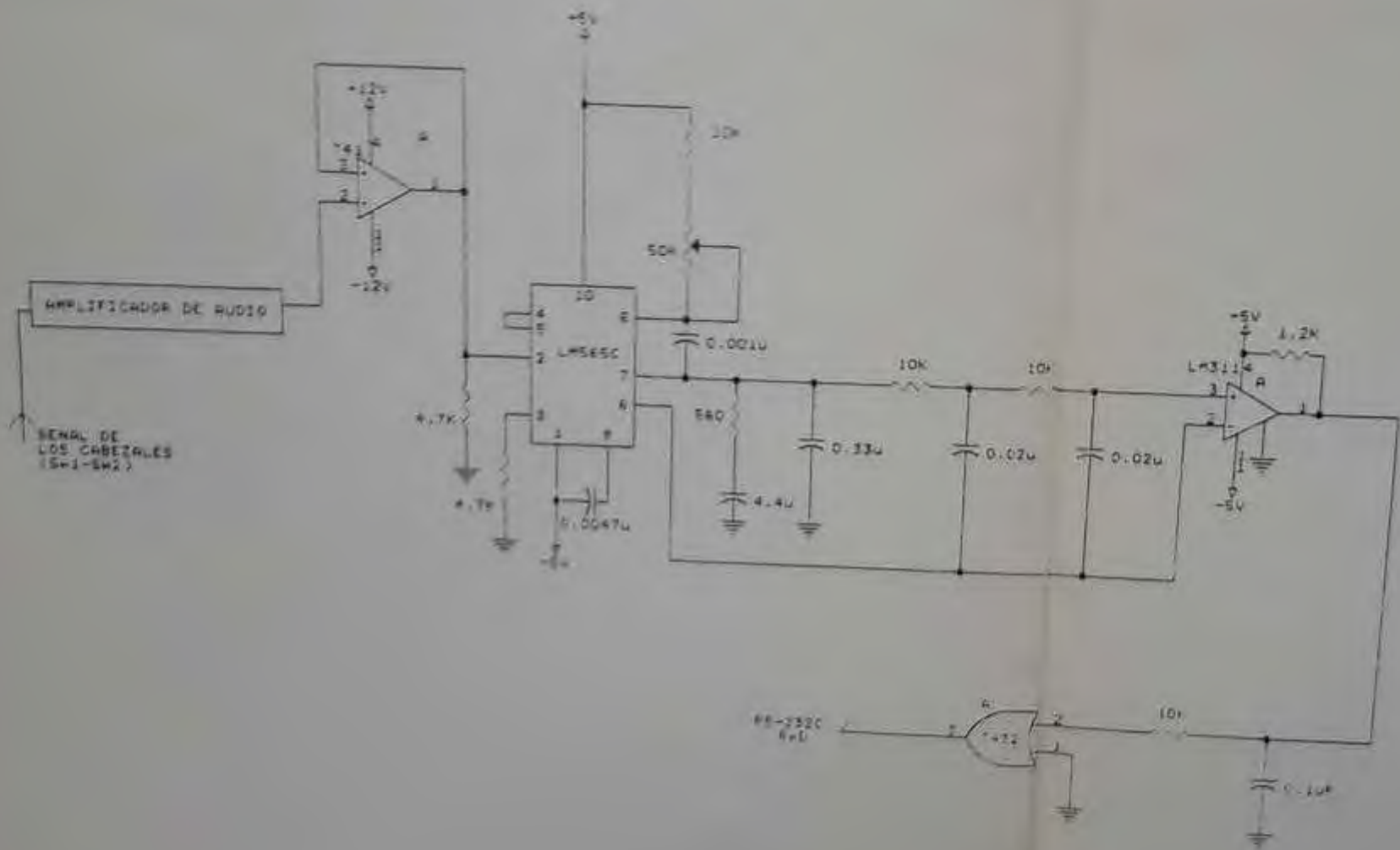
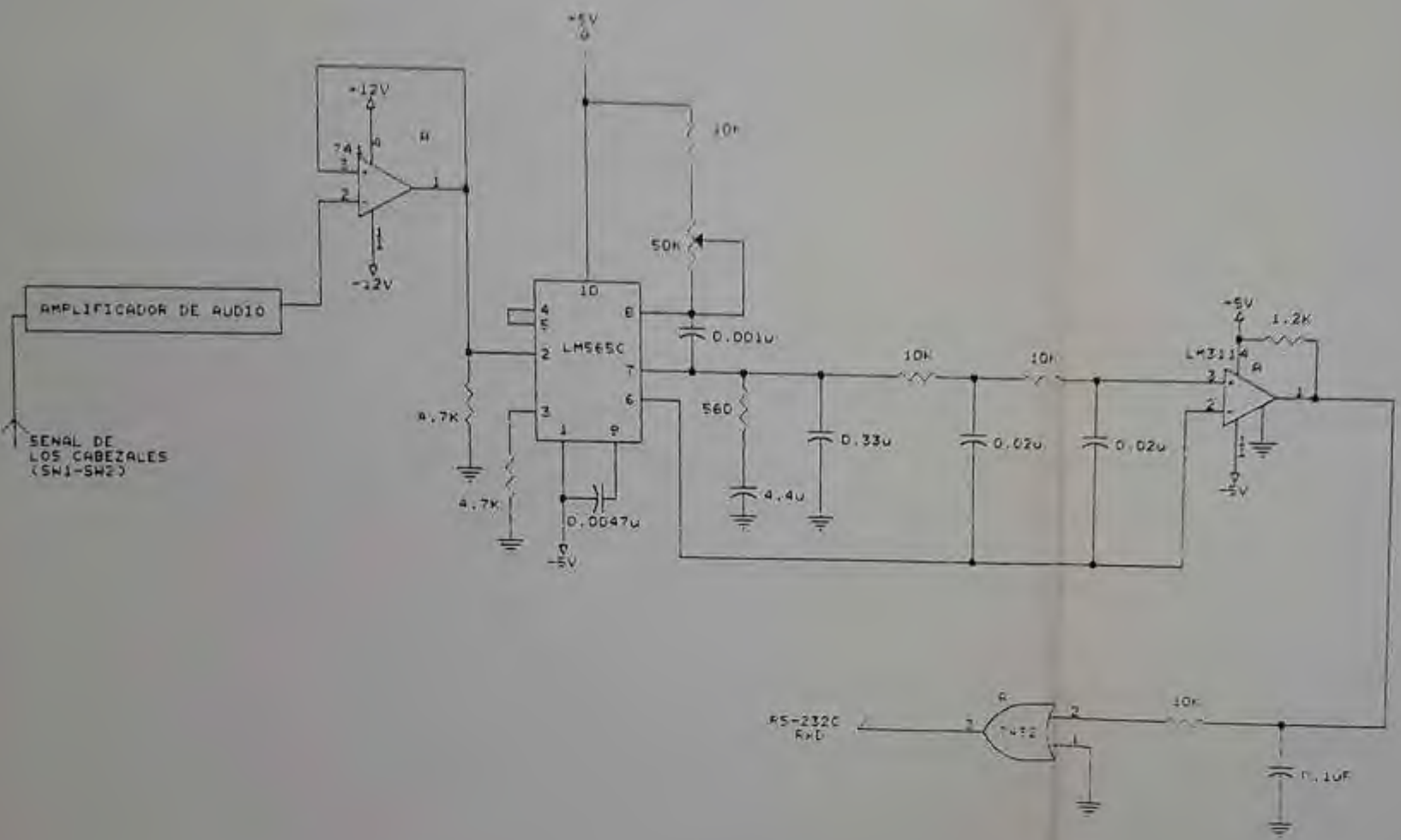


FIGURA H6: Esquema de un caracter en serie asincrono

Cuando un dispositivo emisor no tiene caracteres para enviar, la salida transmitida sería un 1. Eso se llama estado de marca. El caracter de datos en serie asincrónico comienza con un bit de partida, start bit(0 lógico). El caracter termina con un bit de parada, stop bit(1 lógico); algunos caracteres tienen 1, 2 bits de parada. Intercalados entre los bits de partida y los bits de parada están los bits de datos de la palabra de datos (D0 a D7) y un opcional de paridad (P). Hay 8 bits de datos mostrados (algunos caracteres tienen 5 a 8 bits de datos) y un bit de paridad. Si no se utiliza el bit de paridad , el bit de parada sigue al bit de datos más significativo. El bit de datos menos sigificativo en la palabra de datos es enviado o recibido primero. La velocidad a la cual los bits del caracter son enviados o recibidos es especificada por una velocidad en baudios (Bits por segundo, Bits/S).



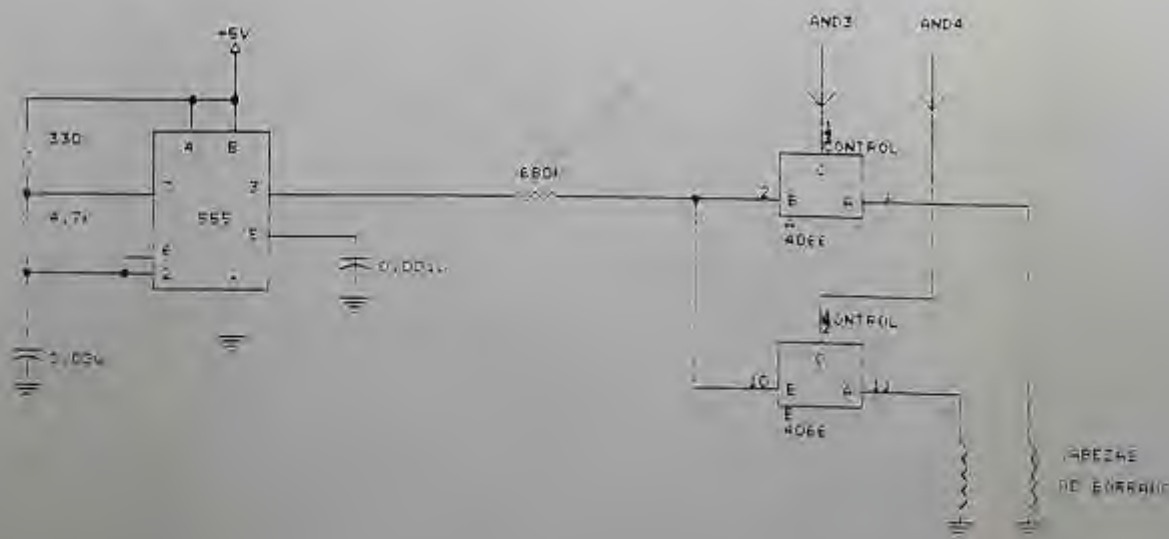
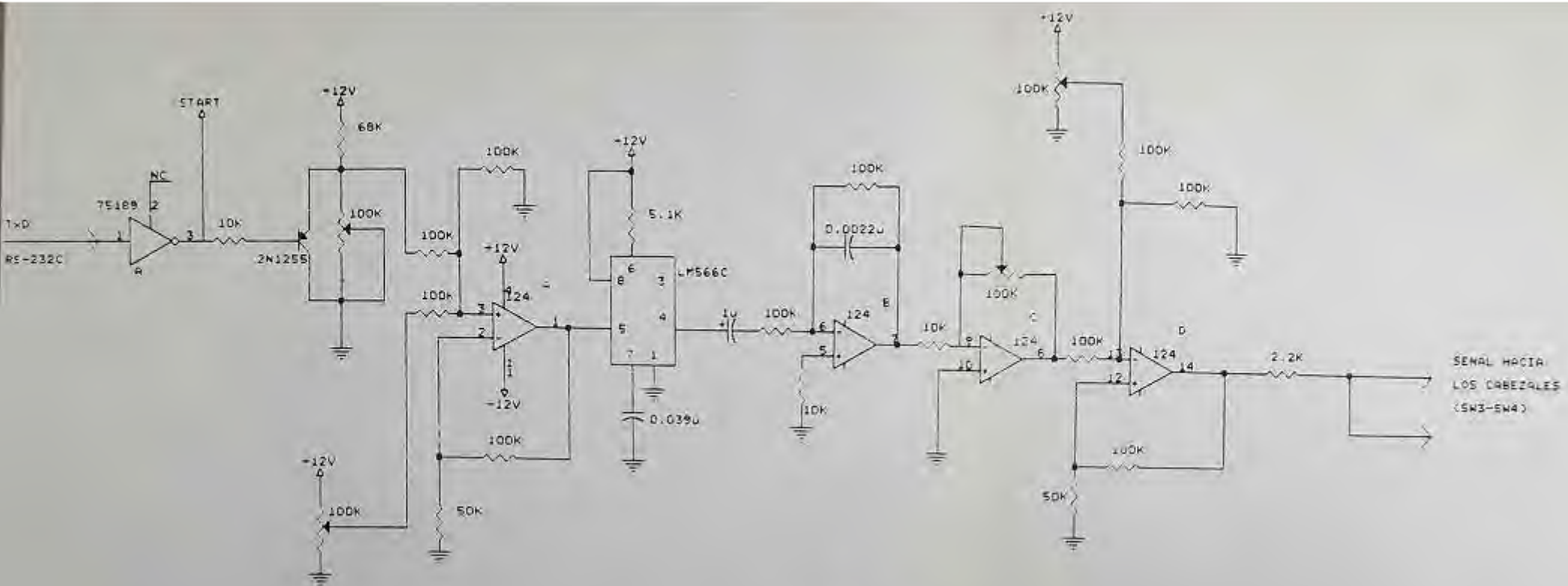
INTEGRANTES:
 JOSE WILSON ALAS RIVERA
 EDUARDO ANTONIO GARCIA BONILLA
 ROBERTO CARLOS ALFONSO
 FECHA: 14/02/2014
 LABORATORIO: 101
 TÍTULO: 101



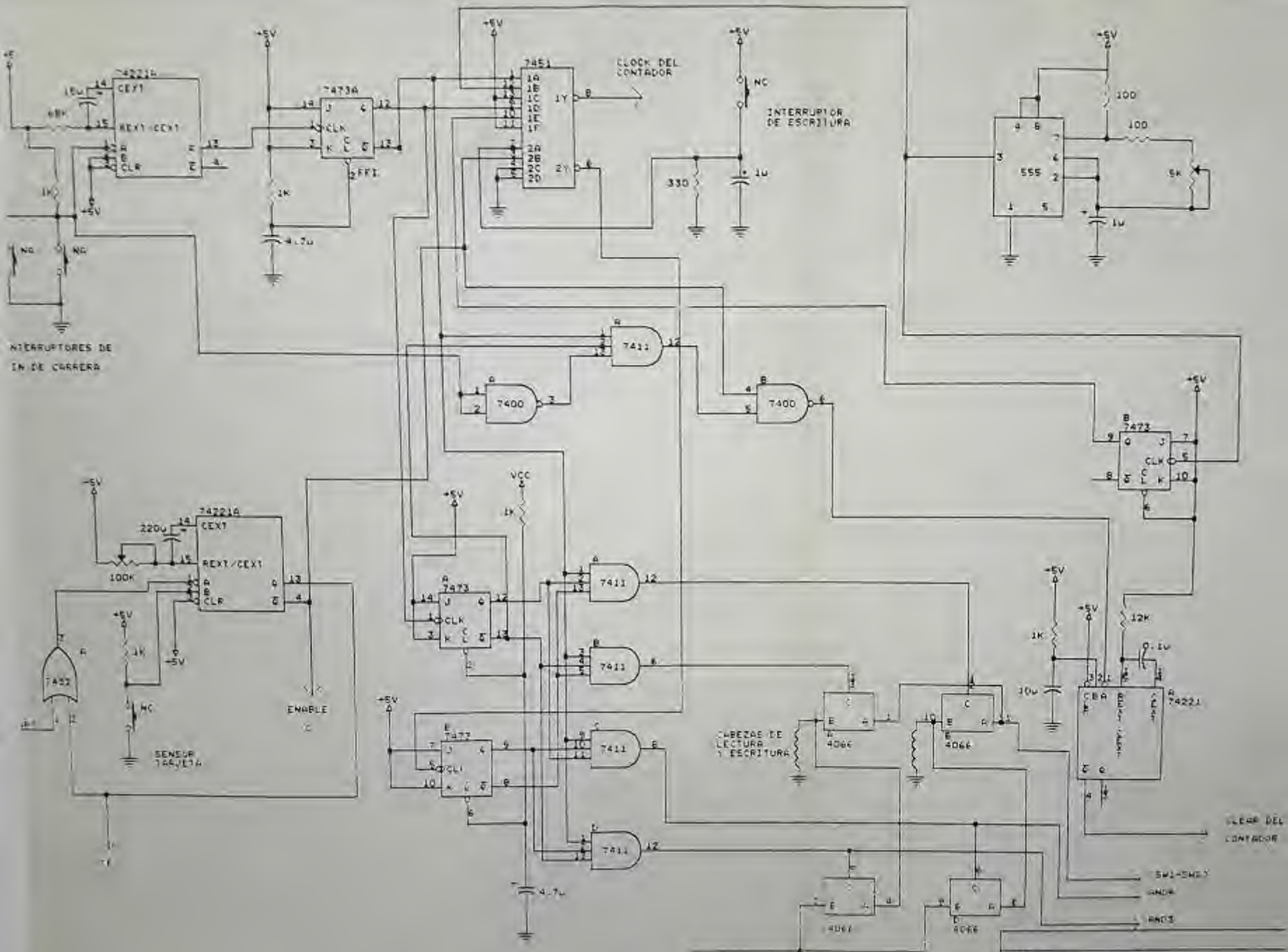
AMPLIFICADOR DE AUDIO

SEÑAL DE LOS CABEZALES (SM1-SM2)

INTEGRANTES:	
JOSE ORLANDO ALAS RODRIGUEZ	
EDWIN ANTONIO SANCHEZ BONILLA	
ROBERTO CARLOS ALVARENGA FIVAT	
Título	SEÑALES DE LECTURA
Número de Documento	2
Fecha	January 25, 1998

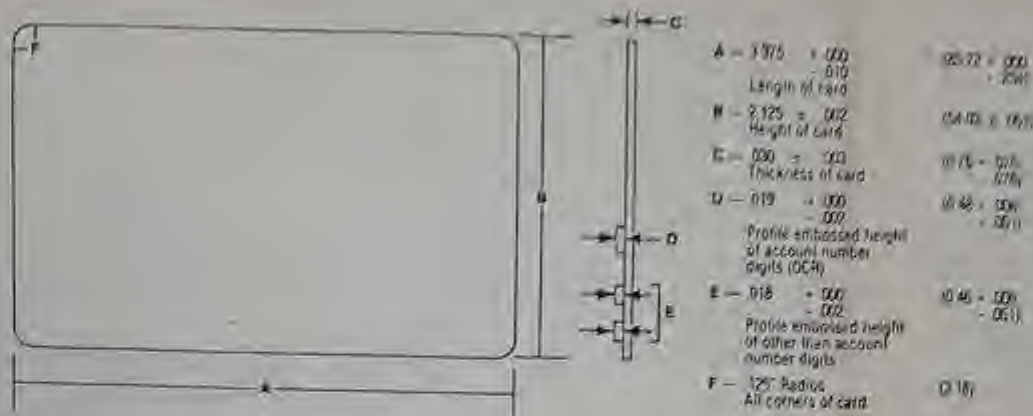


INTEGRANTES:	
JOSE ORLANDO ALAS RODRIGUEZ	
EDWIN ANTONIO GAMERO BONILLA	
ROBERTO CARLOS ALVARENGA RIVAS	
1177	CIRCUITO DE ESCRITURA
12	DocuNet Number
B	1
Catal: Tecnoloxía 21 (1996) 101	



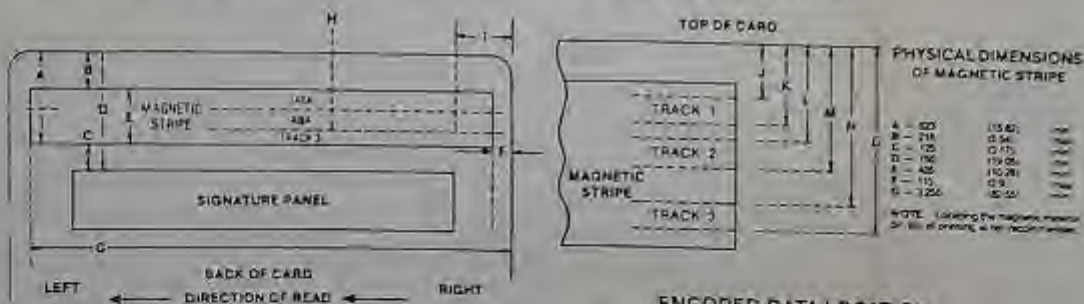
(SW1-SWR1)

FILE	CIRCUITO DE CONTROL
SIZE	Document Number
DATE	December 23, 1980



CARD DIMENSIONS

THREE TRACK MAGNETIC STRIPE SPECIFICATIONS



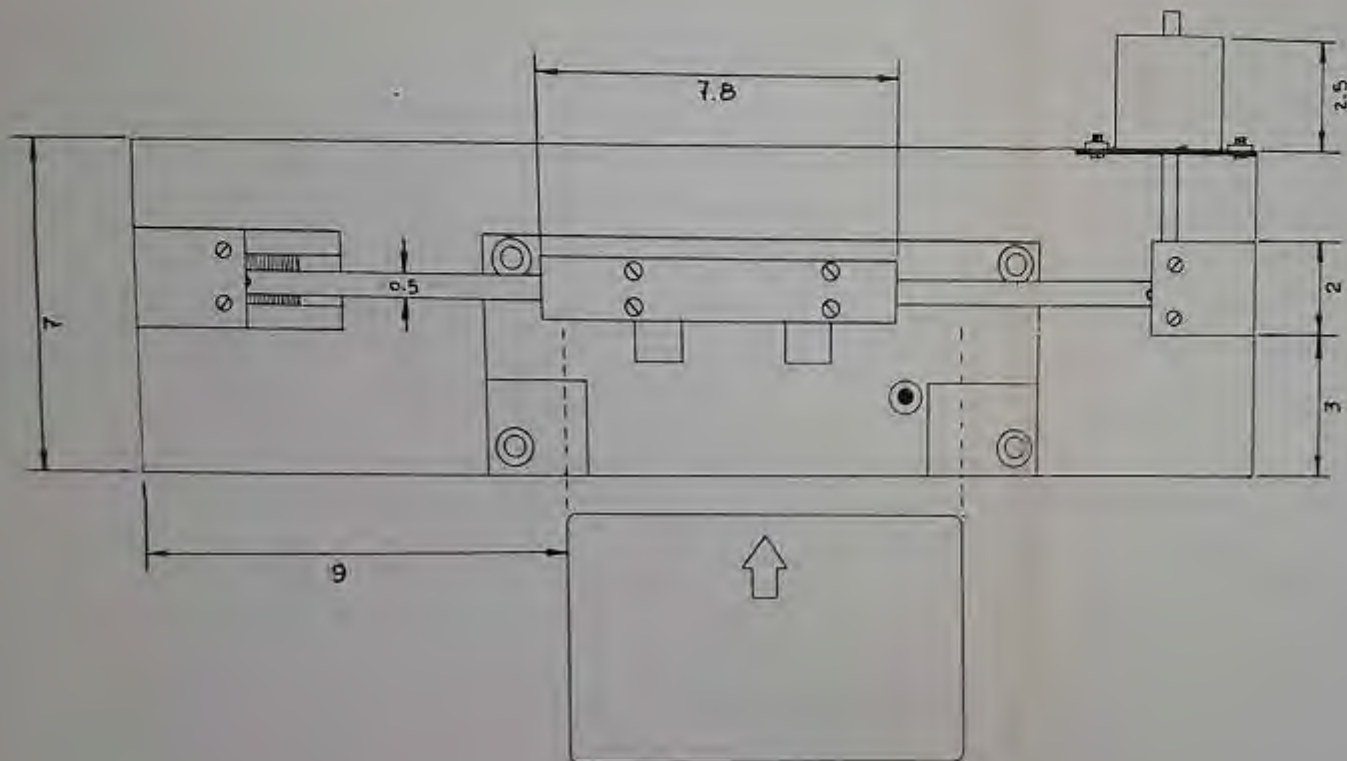
ENCODING SPECIFICATIONS

- Track 1 Encoding Specification (170 bits/21 spaces/5 Hz)
 Track 2 Encoding Specification (170 bits/26 spaces/5 Hz)
 Track 3 Encoding Specification (210 bits/21 spaces/5 Hz)
- D = Direction of recording
 R = Reverse
- 1 = 273 ± .025 (11.14 ± 0.01) Centes
 2 = 0.010 (0.39) Centes
- Note: All dimensions are in millimeters.

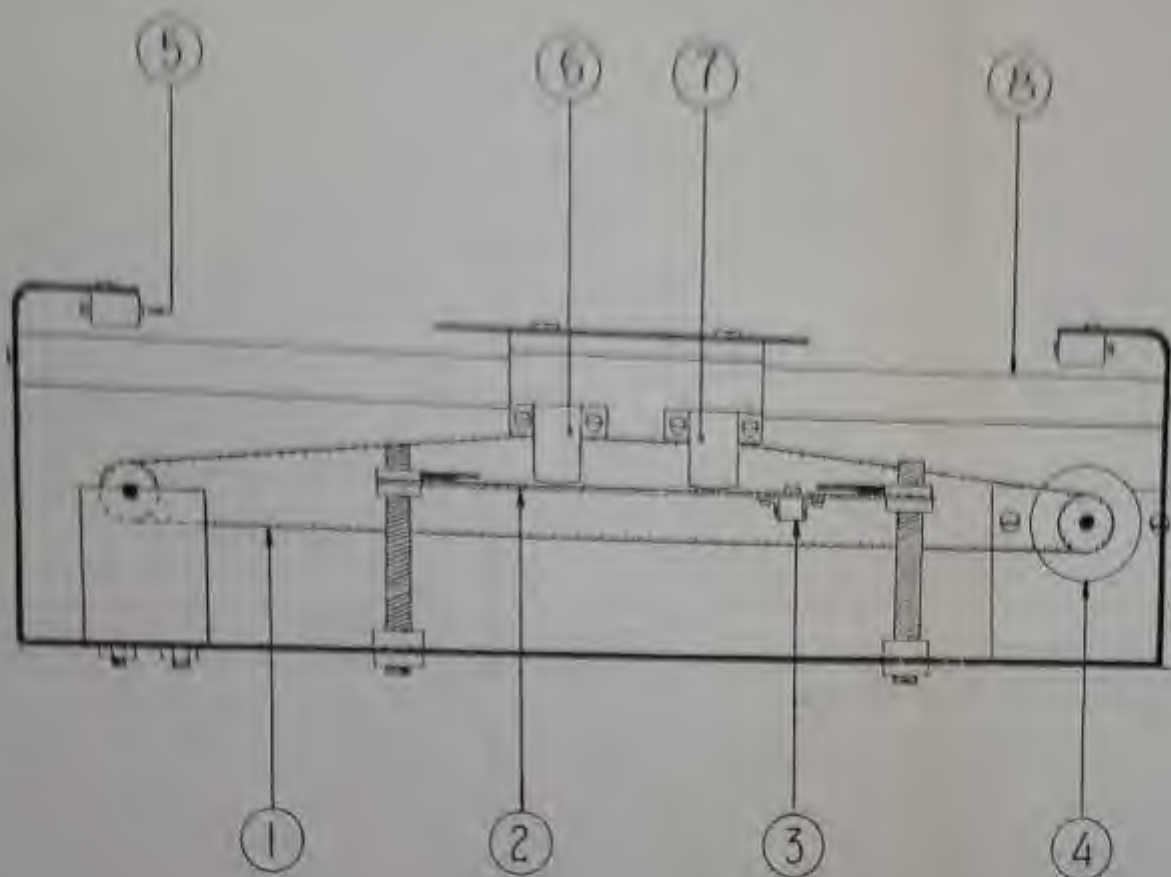
ENCODED DATA LOCATION

- The data must be recorded in a synchronous sequence of 21 characters without inter-character gaps. The upper track of recording must start at a minimum of .223 mm (0.0087 in) and a maximum of .48 mm (0.0189 in) from the top edge of the card, but not below a line 0.252 mm (0.0099 in) from the top edge of the card.
- The second track of encoded data must start at a minimum of .323 mm (0.0127 in) and a maximum of .443 mm (0.0174 in) from the top edge of the card, but not above a line 0.323 mm (0.0127 in) from the top edge of the card.
- Track number 3 of encoded data must extend between two lines 0.440 mm (0.0173 in) and 0.623 mm (0.0245 in) from the top edge of the card, but not above a line 0.443 mm (0.0174 in) from the top edge of the card.
- | | |
|---------|---------|
| J = 50 | (1.94) |
| K = 323 | (12.71) |
| L = 50 | (1.94) |
| M = 425 | (16.73) |
| N = 190 | (7.48) |
| O = 625 | (24.61) |

FECHA: 28/1/97	TITULO DEL PROYECTO: LECTOR/ESCRITOR
MATERIAL: PLASTICO	PARTE: DESCRIPCION DE LAS TARJETAS MAGNETICAS (CORTESIA DE CARNETSAL S.A.)
PROYECCION:	
ESCALA:	PRESENTADO POR: ALAS RODRIGUEZ GAMUÑO BONILLA ALVARENGA RIVAS
DIBUJO # 44	

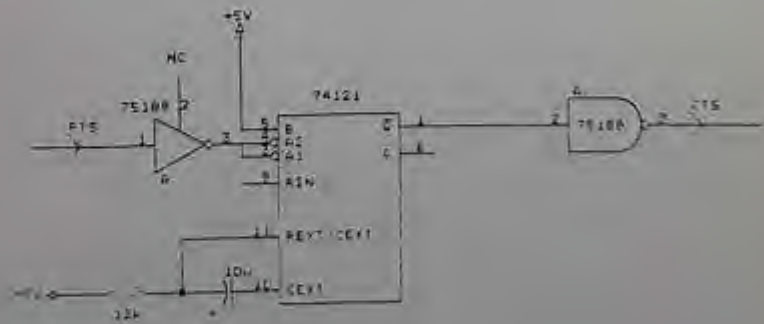
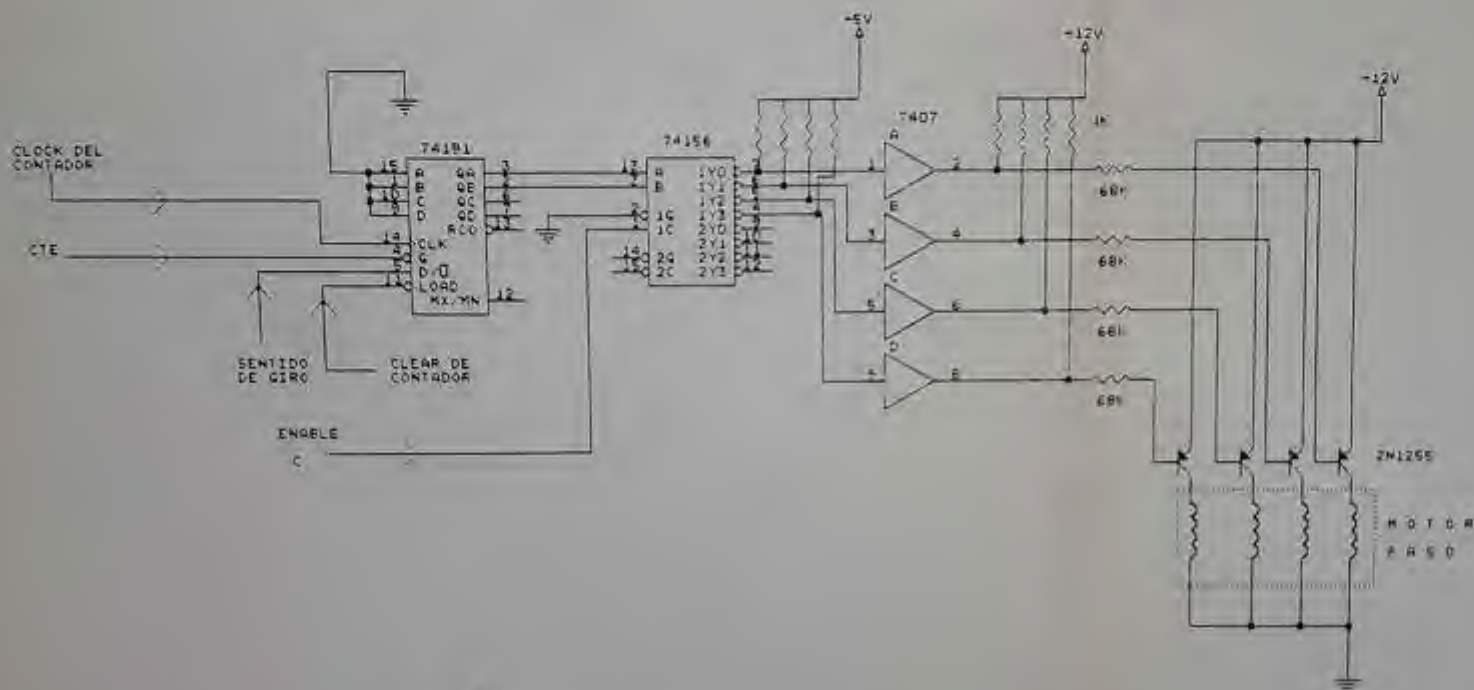


FECHA: 26/1/97	TITULO DEL PROYECTO: LECTOR/ESCRITOR
MATERIAL: ALUMINIO	PARTE: SISTEMA MECANICO (ACOTACIONES)
PROYECCION: PLANTA	PRESENTADO POR: ALAS RODRIGUEZ GAMERO BONILLA ALVARENGA RIVAS
ESCALA: 1:125	
DIBUJO # 34	



NUM.	NOMBRE DE LA PARTE
1	FAJA DENTADA
2	SOPORTE DE LA TARJETA
3	SENSOR DE TARJETA
4	MOTOR PASO A PASO
5	INTERRUPTOR DE INICIO DE CARRERA
6	CABEZAL ESCRITURA/LECTURA
7	CABEZAL BORRADO
8	RIEL

FECHA: 20/1/97	TITULO DEL PROYECTO: LECTOR/ESCRITOR
MATERIAL: ALUMINIO	PARTE: SISTEMA MECANICO (UBICACION DE PARTES)
PROYECCION: FRENTE	PRESENTADO POR: ALAS RODRIGUEZ GAMERO BONILLA ALVARENGA RIVAS
ESCALA: 1:125	
DIBUJO # 1/4	



INTEGRANTES:
 JOSE VALARDO ALBA RODRIGUEZ
 EDWIN ANTONIO GARCIA BARRILEA
 ROBERTO CARLOS ALVARENGA RIVERA

TITULO:
 CIRCUITO PARA CONTROL DE MOTOR PASO

COD. DOCUMENT NUMBER: 174
 DATE: December 20, 1990