

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA

“Diseño y construcción de una interfaz para control de acceso
utilizando lectores magnéticos de proximidad mediante el uso del
microcontrolador Dallas 5000T.”

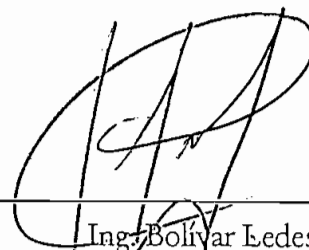
IVÁN PATRICIO FREIRE MOREIRA

SONIA BOLIVIA MAILA LOACHAMÍN

TESIS PREVIA A LA OBTENCIÓN DEL TÍTULO DE INGENIERO
EN LA ESPECIALIZACIÓN ELECTRÓNICA Y CONTROL.

Quito, Octubre de 1998

Certifico que el presente trabajo ha sido realizado en su totalidad por el Sr. Iván P. Freire M. y la Srta. Sonia B. Maila L.

A handwritten signature in black ink, appearing to be 'B. Ledesma', written over a horizontal line.

Ing. Bolívar Ledesma

DIRECTOR DE TESIS

Dedicado

A Dios, creador y esencia de nuestra vida.

A nuestros padres, por su apoyo y confianza.

A nuestros hermanos, motivadores de la culminación de nuestros estudios.

Agradecimiento

Al Ingeniero Bolívar Ledesma por su guía, ayuda y colaboración en la realización de la presente tesis.

Al Ingeniero Juan Carlos Romero por su ayuda en la elaboración de los circuitos impresos.

CONTENIDO

CONTENIDO	i
INTRODUCCIÓN.....	vi

CAPÍTULO I: CONCEPTOS GENERALES

I.1. Tecnología de Sistemas de Control de Accesos	1
I.1.1 Sistema de Control de Accesos	1
I.1.1.1. Terminología utilizada en Sistemas de Control de Accesos.....	1
I.1.2. Tarjetas de Control de Accesos.....	3
I.1.3. Lectoras Wiegand y Lectoras de Salida Wiegand.....	5
I.2. Automatización OLE.	8
I.2.1. Principios OLE	9
I.2.2. Automatización OLE.....	10
I.3. Principios de Desarrollo de las Bases de Datos Relacionales	12
I.3.1. ¿Qué es una base de datos relacional?.....	12
I.3.2. Funciones Principales de una base de datos.....	13
I.3.3. Breve Descripción de Microsoft Access	14
I.3.3.1. Arquitectura de Microsoft Access	14
I.3.3.2. Arquitectura de una aplicación de Microsoft Access.....	18
I.3.3.3. Introducción al lenguaje SQL.....	21

I.4. Breve Descripción de Labview	23
I.4.1. Labview como cliente de Automatización OLE. Subvis incorporados en el programa	24
I.4.2 Un ejemplo del uso de los Vis de Labview para una comunicación OLE con Excel.....	25

CAPÍTULO II: DESARROLLO DEL HARDWARE.

II.1. Requerimientos	29
II.2. Interfaz de Comunicación Serial.....	31
II.2.1. Interfaz del Computador	31
II.2.2. Interfaz del Microcontrolador	33
II.3. Interfaz de Manejo de ocho entradas de alarma.....	34
II.4. Interfaz de Manejo de Display	34
II.5. Interfaz de Lectoras de Tarjetas, Manejo de Salidas de Relé y Lectura de Dipswitches	35
II.5.1. Circuito de Multiplexación	35
II.5.2. Pórtico Multiplexado 0: Interfaz de Lectura de Dipswitches y Lectura de Pulsantes	37
II.5.3. Pórtico Multiplexado 1: Interfaz de Lectoras Magnéticas de Proximidad y Salidas de Relé	37
II.6. Diagrama Circuitual Completo del Sistema Propuesto	39

CAPÍTULO III: DESARROLLO DEL SOFTWARE.

III.1. Requerimientos de software del Sistema de Control de Accesos	43
III.2. Desarrollo de software en el microcontrolador Dallas DS5000T	44
III.2.1. Descripción del Programa en el Microcontrolador	44
III.2.2. Administración de la comunicación serial.....	44
III.2.2.1. Protocolo.....	45
III.2.2.2. Administración de memoria.....	47
III.2.2.3. Administración de recepción.....	50
III.2.2.4. Decodificación y validación	53
III.2.2.5. Transmisión.....	56
III.2.3 Administración de Lectoras de Proximidad	56
III.2.3.1. Interrupciones.....	57
III.2.3.2. Reconocimiento y validación	58
III.2.4 Administración del ECC, LCD y Monitoreo de las entradas de alarma	60
III.2.4.1. Administración del Reloj Calendario	60
III.2.4.2. Manejo del Display.....	62
III.2.4.3. Monitoreo de las Entradas de Alarma.....	64
III.3. Desarrollo del software en el computador personal	64
III.3.1. Diagrama de Bloques del Programa del PC	64
III.3.2. Interfaz de Labview	66

III.3.2.1.	Programa Principal	66
III.3.2.2.	Rutinas de Configuración.....	74
III.3.2.3.	Rutinas de Generación de Reportes	87
III.3.3.	Desarrollo de Programa de Comunicación Serial	92
III.3.4.	Desarrollo de Rutinas de Automatización Ole	93
III.3.4.1.	Rutinas de apertura de la comunicación con Access	94
III.3.4.2.	Rutinas de manejos de registros.....	98
III.3.4.3.	Filosofía de manejo de la base de datos.....	102
III.3.5.	Desarrollo de la Base de Datos	102
III.3.5.1.	Tabla Contraseñas	104
III.3.5.2.	Tabla Equipo	104
III.3.5.3.	Tabla Usuarios	104
III.3.5.4.	Tabla Puertas	104
III.3.5.5.	Tabla Registro de Usuarios por Equipo	105
III.3.5.6.	Tabla Control de Accesos	106
III.3.5.7.	Tabla Zonas de Tiempo.....	106
III.3.5.8.	Tabla Horario Timezones	107

CAPÍTULO IV: PRUEBAS Y RESULTADOS.

IV.1.	Pruebas en Stand Alone.	108
IV.1.1.	Detección de Tarjeta	109
IV.1.2.	Funcionamiento de Salidas de Relé	109

IV.1.3.	Apertura Manual.....	110
IV.2.	Pruebas del Sistema de Control de Accesos.....	111
IV.2.1.	Comunicación entre el equipo y el computador personal	112
IV.2.2.	Configuración de Equipo	113
IV.2.3.	Base de datos.....	114
IV.2.4.	Reportes.....	115

CAPÍTULO IV: CONCLUSIONES Y RECOMENDACIONES.

V.1.	Conclusiones.....	118
V.2.	Recomendaciones.....	120

ANEXOS.

Anexo A:	Hojas de Datos Dallas DS5000T.....	A-1
Anexo B:	Manual de Usuario.....	B-1

INTRODUCCIÓN

El presente trabajo tiene como objetivo desarrollar un “sistema de control de accesos utilizando lectoras de proximidad”. El sistema está constituido de dos partes: un equipo y un software en el computador.

El equipo está comandado por el microcontrolador Dallas DS50000T, que manipula el control de acceso de dos puertas a través de lectura de un código por medio de lectoras de proximidad. El ingreso de una persona es permitido si el equipo reconoce el código de tarjeta y se encuentra dentro del horario configurado para dicho código de tarjeta.

El equipo tiene salidas de relé que sirven para conectar a las cerraduras de las puertas respectivas.

Además el sistema administra ocho entradas de alarma y genera un protocolo de comunicación con el computador personal.

El computador personal tiene un software que es la interfaz entre el operador del sistema y la actividad del sistema.

El operador a través del software puede:

- Configurar los parámetros del equipo: códigos de tarjeta autorizados, horarios permitidos para los códigos de tarjeta, sincronización del reloj interno del equipo.
- Visualizar cambios en las entradas de alarma del equipo.
- Visualizar la detección de código de tarjeta en las lectoras y la acción tomada por el equipo

- Ver la historia de actividad del control de accesos del sistema en pantalla utilizando los datos almacenados dentro de una base de datos.

El sistema está diseñado de manera que una vez configurado el equipo, este pueda realizar el control de accesos sin la intervención del computador, es decir la decisión del ingreso de un código de usuario la toma el microcontrolador, el computador es el encargado de recibir la decisión que tomó el microcontrolador.

En esta tesis se presenta el uso de una tecnología de control de accesos que se encuentra en auge como son las lectoras de proximidad mediante el uso de un microcontrolador que ofrece facilidades para este trabajo como es el microcontrolador Dallas DS5000T, el manejo de un software moderno y sencillo como es LABView y la facilidad de uso de la base de datos Microsoft Access.

El desarrollo del presente trabajo se encuentra organizado por capítulos. El capítulo I describe los conceptos generales referentes a las tecnologías y terminología que se utiliza en los sistemas de control de accesos, además se realiza una breve descripción de las ayudas computacionales utilizadas y una ligera introducción a la vinculación e incrustación de objetos. En el capítulo II se desarrolla los requerimientos de hardware del sistema propuesto. El capítulo III contiene un desarrollo completo del software requerido para realizar tanto el interfaz del sistema a través del computador como el control de accesos a través del microcontrolador. Mientras que el capítulo IV trata sobre las pruebas y resultados del sistema. Por último en el capítulo V se presenta las conclusiones y recomendaciones a las que se llegó luego de elaborar este trabajo de investigación.

CAPÍTULO I:

CONCEPTOS GENERALES

I.1. TECNOLOGIAS DE SISTEMAS DE CONTROL DE ACCESOS.

I.1.1. SISTEMA DE CONTROL DE ACCESOS.

Control de accesos es un control sobre la entrada a un área, que puede ser de seguridad. La entrada solamente se permite a personas autorizadas y/o en tiempos determinados.

Para realizar un control de acceso, cada persona recibe un identificador que puede ser: una tarjeta, una clave de acceso e inclusive la huella digital de la persona, la cual restringe el acceso en áreas determinadas y en tiempos autorizados.

I.1.1.1. Terminología utilizada en Sistemas de Control de Accesos.

Usuario: Es la persona que tiene asignado un identificador y hace uso del sistema.

Operador: Es la persona que administra el sistema y que generalmente se le asigna una contraseña. Controla los privilegios de acceso de los usuarios, tiene acceso al historial de datos del sistema y puede cambiar la configuración general del sistema.

Zonas de Tiempo: Las zonas de tiempo (Timezones) son asignaciones de tiempo utilizadas en el Sistema de Control de Accesos para definir horarios para el control de acceso a los usuarios o para realizar una acción particular.

Nivel de Acceso: Es un elemento que define los lugares y horarios de acceso permitidos a un usuario.

Reporte: Presenta los datos almacenados en la base de datos hacia el operador sobre las diferentes actividades realizadas por el sistema como por ejemplo: reporte de accesos al sistema, reportes de actividades realizadas por el operador, reporte de actividad de alarmas, entre otros. Los reportes se pueden restringir a datos en tiempos determinados por el operador del sistema de control de accesos, por equipos, por usuarios o cualquier otro parámetro que se considere necesario. Básicamente, constituye el manejo de la base de datos, el seguimiento de todos los eventos que se dan en el sistema.

Tipo de Accesos

Acceso Denegado: El acceso es denegado cuando un usuario intenta ingresar dentro de un horario no definido para el mismo, o por un sitio no permitido para él.

Acceso Aceptado: Cuando el sistema permite el ingreso del usuario al área que desea acceder.

Tarjeta no reconocida: Evento que ocurre cuando el identificador del usuario no pertenece al área donde intenta ingresar.

Tarjeta expirada: Evento que ocurre cuando un usuario tiene un identificador que el sistema considera que esta fuera del tiempo de uso para el que fue asignado.

Código de acceso: Es el identificador que ubica al usuario dentro del sistema.

Los identificadores (elementos que permiten el acceso) comúnmente utilizados se basan en TARJETAS DE CONTROL DE ACCESOS. En la actualidad las tecnologías con tarjetas de control de accesos más usadas son: WIEGAND, PROXIMIDAD, CÓDIGO DE BARRAS, CHIP.

I.1.2. TARJETAS DE CONTROL DE ACCESOS.^[1]

A continuación se presenta una breve descripción de las diferentes tecnologías que son utilizadas en las tarjetas de control de accesos.

Código de barra

La tecnología de código de barra es muy utilizada en aplicaciones que no requieren seguridad, pocas veces esta tecnología se escoge para seguridad y control de accesos. La lectura del código de barras se realiza de un sistema de lentes. Esta tecnología tiene una visible desventaja en cuanto a seguridad, esta puede reproducirse utilizando una fotocopidora, sin embargo es posible enmascarar el código de barra con un filtro haciendo difícil su duplicación. Otra desventaja que presenta esta tecnología es la limpieza que requieren los lentes para su correcto funcionamiento.

Banda magnética

Esta tecnología continua siendo utilizada en aplicaciones de control de accesos. Para muchas personas esta tecnología es conocida porque se utiliza en la operación de tarjetas de crédito. El principio de funcionamiento consiste en el contacto de la cinta magnética de la tarjeta con la cabeza lectora. Cuando la tarjeta toma contacto con la cabeza lectora se recogen los datos de la tarjeta almacenado en la cinta magnética. Esta operación de contacto crea un desgaste tanto de la tarjeta como de la cabeza lectora. La pérdida de datos entre otras razones hace que en la actualidad no se utilice esta tecnología para el control de accesos. Provee un mediano nivel de seguridad porque es posible, con un buen equipo y apropiado conocimiento, duplicar las tarjetas.

Wiegand

Originalmente creada para proveer de una tarjeta insensible a los campos magnéticos externos. La *tarjeta de efecto Wiegand* se basa en el “Efecto Wiegand” (ver ítem **Lectoras Wiegand y Lectoras de Salida Wiegand**, para conocer en que consiste este efecto). La tarjeta al interactuar magnéticamente con la lectora genera un flujo de bits que es utilizado para identificar al usuario. La tecnología Wiegand es comúnmente utilizada en aplicaciones de alta seguridad.

Proximidad

Es la tecnología de más rápido desarrollo para aplicaciones de control de acceso. En la tecnología de proximidad la lectora transmite de manera continua una señal fija RF de bajo nivel que provee de energía a la tarjeta. Cuando la tarjeta se coloca a una cierta distancia de la lectora, la señal de RF es absorbida por una pequeña bobina dentro de la tarjeta que activa un microchip que contiene el código de identificación. Una vez activada, la tarjeta transmite el código a la lectora. El proceso se completa en microsegundos.

Las ventajas de la tecnología de proximidad son numerosas: no hay movimiento de las partes, no hay desgaste mecánico, no existen slots, no existen cabezas lectoras que requieran mantenimiento, pueden ser colocadas en cualquier ambiente y sobre cualquier superficie, puede estar a la intemperie o recubiertas. La tarjeta es prácticamente imposible de duplicar. La lectura de las tarjetas puede realizarse a través de una cartera, bolso, o cualquier otro material no metálico. La orientación de la tarjeta y la lectora no es crítica y el contacto por ejemplo con elementos metálicos no altera el código de la misma o provoca una lectura inexacta.

Los rangos de lectura dependen principalmente de la lectora. Grandes distancias de lectura

requieren lectoras de mayor tamaño, muchas veces con antena incorporada. Los rangos de lectura se determinan con tarjetas estándares. Si la bobina de la tarjeta es más pequeña como en tarjetas delgadas de fotoidentificación, la distancia de lectura disminuye hasta en un 40%.

I.1.3. LECTORAS WIEGAND Y LECTORAS DE SALIDA WIEGAND.^[2]

Wiegand fue introducido en 1970, e inmediatamente tuvo un rotundo éxito por la necesidad de durabilidad, tarjetas de seguridad y tecnología de lectora. Hay una diferencia entre una lectora de salida Wiegand y una lectora Wiegand. Las lectoras Wiegand utilizan el efecto Wiegand para convertir la información en datos digitales. Las lectoras de salida Wiegand utilizan para la comunicación el protocolo Wiegand.

Las lectoras Wiegand son desarrolladas bajo el efecto Wiegand en el cual un segmento de un alambre especial genera un pulso electrónico cuando esta sujeto a un campo magnético específico. Este pulso puede ser detectado por un circuito electrónico. Poniendo varias filas de alambres y pasando alrededor de una bobina se genera una serie de pulsos. Poniendo dos filas de alambres, una llamada fila "BIT CERO" y la otra "BIT UNO" y pasando estas por dos bobinas diferentes se puede generar dos series de pulsos, o bits de datos. Estos bits de datos pueden ser interpretados como datos binarios y utilizados para el control o identificación de equipos. Las bobinas, selladas y recubiertas por un material magnético, un LED y algunos circuitos conforman la lectora Wiegand. Los alambres, son laminados en vinilo con algunos trabajos artísticos para comprimir en la tarjeta Wiegand.

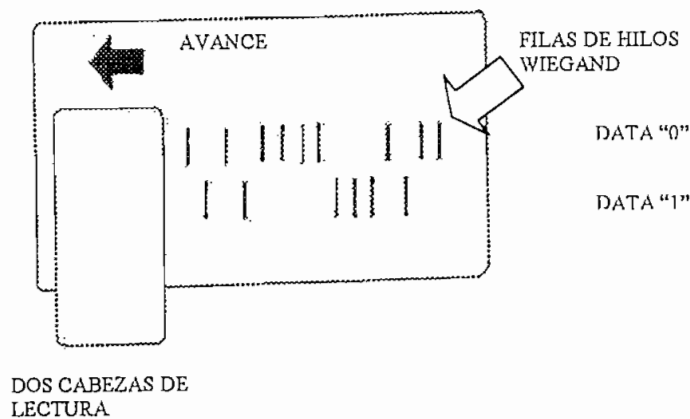


FIGURA I-1. ESQUEMA DE LA FORMA DE LECTURA DE TARJETAS WIEGAND.^[3]

Una propiedad importante de estas tarjetas, es que los alambres Wiegand no pueden ser alterados. Los alambres no hacen contacto con la lectora, así reducen el desgaste y aumenta el tiempo de vida de la tarjeta. Las tarjetas nunca tocan la cabeza lectora, por lo que no se requiere mantenimiento. Los datos llegan con un nivel de salida y velocidad consistente. Esto se ha hecho un estandard industrial.

Hoy en día, casi todos los productos en la industria de control de accesos pueden comunicarse con *equipos de salida Wiegand*, así se pueden conseguir en el mercado lectoras de proximidad, banda magnética, keypads y código de barras con salidas Wiegand. La tecnología Wiegand es popular debido a su simplicidad. La flexibilidad de Wiegand ha hecho de ella un modelo para el desarrollo de normas ISO, tecnología de tarjetas inteligentes entre otras.

Los *equipos de salida Wiegand* se caracterizan por el formato de salida de datos realizados a través de dos líneas conocidas como "DATA0" y "DATA1", como ocurre con las lectoras de tarjetas de efecto Wiegand.

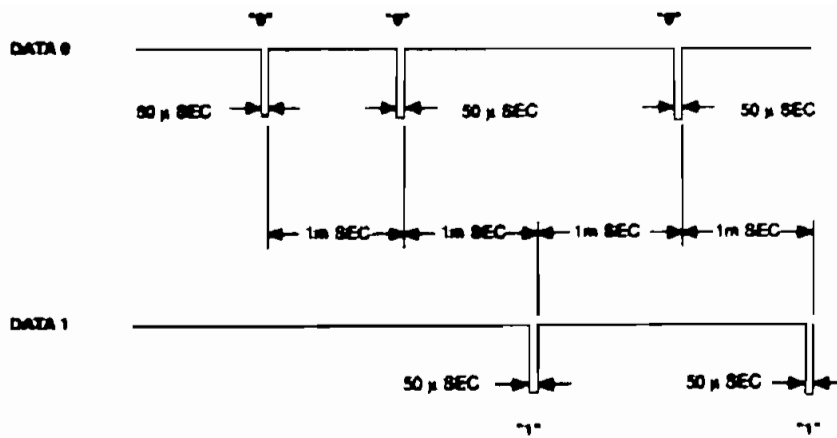


FIGURA I-2. FLUJO DE BITS EN EQUIPOS DE SALIDA WIEGAND.^[4]

Hasta el momento el presente capítulo ha tratado de manera sencilla los elementos periféricos que conforman el sistema de control de acceso basados en lectoras de tarjetas y sus respectivas tecnologías. Sin embargo un sistema de control de acceso no consta solamente de estos elementos de lectura de código de usuario, es necesario un enlace entre el equipo que realiza el control y el operador que administra el sistema. Este enlace se realiza a través de un computador personal. Un programa en el computador realiza el enlace entre los equipos periféricos y el usuario, teniendo entre sus actividades: la comunicación con los equipos periféricos, la configuración por parte del operador y el almacenamiento de la actividad del sistema (base de datos). Es decir el software del computador involucra la programación de un sistema de interfaz y el requerimiento de una base de datos que almacene la actividad del sistema.

Las herramientas de programación: bases de datos y otros programas de ayuda para la conformación de un sistema de control de accesos se venden como herramientas separadas. Esto hace necesario que exista algún medio que permita unir y trabajar como un solo conjunto estas diferentes herramientas. Existen varios protocolos que permiten la integración entre aplicaciones (herramientas de programación), los más conocidos y

difundidos son: DDE y OLE. La presente tesis se desarrolla con ayuda del protocolo OLE más conocido como Automatización OLE.

El resto de este capítulo trata dos temas muy importantes para el desarrollo de la presente tesis:

- Una descripción de los aspectos básicos sobre las herramientas que se utilizan en la programación de esta tesis: Microsoft Access y LABVIEW 4.1.
- Una explicación de lo que constituye la automatización OLE, que se utilizan para la comunicación entre las dos aplicaciones.

1.2. AUTOMATIZACIÓN OLE.

La forma de elaborar documentos electrónicos utilizando los últimos esquemas de edición, graficación, vinculación de objetos y extracción de datos de otros archivos, para producir un único documento final, ha cambiado significativamente. La composición integrada de documentos provee acceso a poderosas herramientas para cada una de las partes que compone un archivo sean tratadas como objetos independientes que se trabajan por separado y se unifican posteriormente mediante enlaces lógicos, haciendo más rápida y eficiente la forma de editar, dibujar, esquematizar y actualizar cualquier archivo soportado por estas técnicas.

En el mercado de software para composición de documentos existen docenas de productos "todo en uno", que permiten entrelazar todas las aplicaciones especializadas como editores de imágenes, hoja electrónica, paquetes de comunicación y visores de datos, para que la conformación del documento final se haga desde el mismo procesador de palabras, intercambiando datos desde y hacia las demás aplicaciones, y ensamblando todas las partes

en un archivo final.

1.2.1. PRINCIPIOS OLE.

OLE son las siglas en inglés de Incrustación y enlace de objetos (Object Linking and Embedded) y constituye todo un estándar en la industria del software para la comunicación entre diferentes aplicaciones corriendo en un sistema computacional.

Utilizando OLE se puede crear o editar documentos que contienen datos u objetos creados en diferentes formatos y aplicaciones. Estos documentos son llamados **documentos compuestos**. Los documentos compuestos contienen diferentes tipos de datos, como bitmaps, hojas de datos y sonidos. Una **aplicación contenedor** es una aplicación que puede contener datos de objetos enlazados o incrustados en sus documentos compuestos. En una aplicación contenedora los usuarios pueden insertar nuevos datos o editarlos. Una **aplicación servidor** crea objetos OLE para ser usados por la aplicación contenedora. Un objeto OLE que es parte de un documento compuesto pertenece al **Modelo de Objetos Componentes**. Este modelo define el comportamiento e interacción de los objetos OLE. Los objetos OLE consisten de datos y métodos para la manipulación de datos. Un objeto también provee una interfaz a través de la cual otros objetos se pueden comunicar entre sí. Un objeto participa en diferentes acciones dependiendo del contexto en el cual vaya a interactuar. En el contexto de documentos compuestos, los objetos toman las siguientes acciones:

- Incrustación y Enlace - Dos métodos para almacenamiento de objetos dentro de la composición de documentos.
- Activación Dentro de Lugar - Principalmente por la manipulación de objetos.

- Automatización - La habilidad de crear programas de aplicación por exposición de objetos.
- Archivos compuestos - Un modelo de almacenamiento estructurado para objetos.
- Transferencia Uniforme de datos - Poderosos mecanismos de transferencia de datos para objetos.

Una aplicación contenedor puede tener componentes o controles personalizados OLE (OLE custom controls – OCX). Estos controles conocidos como arquitectura Active X es un conjunto de extensiones a las interfaces OLE existentes que transforman los contenedores y objetos OLE en un poderoso “contenedor de controles”. Un control OLE (OCX) es un objeto que utiliza un conjunto de tres atributos:

Propiedades .- Nombres de características del control como son: color, texto, número y fondo.

Eventos .- Activación de acciones del control en respuesta a una acción externa sobre el control como por ejemplo: un click del ratón, enfoque del control con el ratón, entre otros.

Métodos .- Un conjunto de funciones implementadas en el control que pueden manipularse a través de un código externo y cambiar el comportamiento, apariencia del control.

Las propiedades, métodos y eventos de un control utilizado son almacenadas en archivo de extensión .OCX.

I.2.2. AUTOMATIZACIÓN OLE.^[6]

La automatización OLE es utilizada para control de aplicaciones e intercambio de datos.

Un objeto es parte de una aplicación, ésta puede ser programada. Los objetos mostrados

en una aplicación son llamados objetos de la Automatización OLE. Por ejemplo, una aplicación de hoja de cálculo, puede mostrar una hoja, un diagrama, celdas o un rango de celdas con diferentes tipos de objetos. Un procesador de palabras puede exponer objetos como documentos, párrafos, u oraciones. Cada objeto de la Automatización OLE va construyendo un bloque que se puede utilizar en la colección de datos, envío de datos. Usando automatización OLE se puede:

- Crear aplicación que muestren objetos.
- Crear y manipular objetos en una aplicación y que se muestre en otra aplicación.

La aplicación que expone objetos OLE se llama **Aplicación Objeto** o **Servidor de Automatización OLE**. La aplicación que accesa a los objetos OLE se denomina un **Cliente de Automatización OLE**.

La automatización OLE define dos tipos de miembros para mostrar objetos - **métodos y propiedades**.

Métodos son funciones que ejecutan la acción sobre un objeto. *Propiedades* son funciones que ponen o retoman información acerca del estado de un objeto.

Los **objetos de automatización OLE** son diferentes que los **objetos OLE** por que se puede acceder solamente por medio de lenguajes de programación. Estos objeto no son visibles al usuario y son usados por tareas automáticas en las que no se requiere interacción del usuario.

Los clientes deben tener información acerca de los servidores de objetos, propiedades y métodos si los clientes van a manipular el servidor de objetos. Las propiedades tienen tipos de datos mientras que los métodos retornan valores y parámetros de aceptación. La

información de los objetos expuestos, métodos y propiedades de una aplicación se la encuentra en la documentación de dicha aplicación, sin embargo se puede encontrar esta lista de objetos en la librería de tipos de la aplicación (si dispone de la misma).

Una **librería de tipos** contiene una lista completa de todos los métodos y propiedades de los objetos expuestos por una aplicación servidor de automatización. Estas librerías no almacenan objetos, pero ellas almacenan el tipo de información que necesitan los objetos. Mediante librerías de tipos, las aplicaciones cliente pueden determinar las características de los objetos en determinada aplicación, como por ejemplo su interfaz de aplicación. Una librería de tipos puede tener las siguientes extensiones: .olb, .tlb o .dll.

I.3. PRINCIPIOS DE DESARROLLO DE LAS BASES DE DATOS RELACIONALES.

I.3.1. ¿QUÉ ES UNA BASE DE DATOS RELACIONAL?

En sentido general una base de datos es un conjunto de registros y archivos que están organizados para un uso determinado. Los documentos de tratamiento de textos que se organizan por tópicos son un tipo de base de datos. Los archivos de hojas electrónicas que se organizan de acuerdo con sus usos son otro tipo de base de datos. En aplicaciones de mayor alcance se podrá gestionar cientos de hojas electrónicas que se organizarán utilizando directorios y subdirectorios, cuando se realiza esta acción la aplicación se convierte en el gestor de base de datos.

El nombre relacional procede del hecho de que cada registro de la base de datos contiene información relacionada con el tema y sólo con ese tema. Además, los datos de dos clases de información pueden ser manipulados como una única entidad basada en los valores de

los datos relacionados. Por ejemplo, sería una redundancia almacenar información sobre el nombre y la dirección de un cliente con todos los pedidos que efectúe ese cliente.

En un sistema de gestión de base de datos relacional, el sistema trata todos los datos en tablas. Las tablas almacenan información sobre un tema y disponen de unas columnas que contienen los diferentes tipos de información sobre ese tema y de filas que describen todos los atributos de una única instancia del tema. Incluso cuando se utiliza una de las facilidades del sistema de gestión de base de datos relacional para buscar información procedente de una o más tablas a menudo denominado una consulta, el resultado es siempre algo semejante a otra tabla.

Además es posible unir la información de varias tablas o consultas por valores relacionados.

1.3.2. FUNCIONES PRINCIPALES DE UNA BASE DE DATOS.^[6]

Un sistema de gestión de base de datos proporciona un control completo sobre la forma de definir los datos, presenta tres tipos de funciones: definición de los datos, manipulación de los datos y control de los datos.

Definición de datos.- Es posible definir los datos que se almacenarán en una base de datos, el tipo de los mismos y la forma en que están relacionados. En algunos casos, también es posible definir los datos que deben estar con formato y los datos que deben ser validados.

Manipulación de datos.- Es posible trabajar con los datos de muchas formas. Se puede seleccionar los campos de datos que se desea, o se puede filtrar los datos y ordenarlos.

Control de los datos.- Se puede definir personas que están autorizadas a leer, actualizar, o insertar los datos.

Un sistema de gestión de base de datos (SGBD) permite definir el tipo de datos y la forma en que deben almacenarse. También se puede definir reglas que el SGBD debe utilizar para asegurar la integridad de los datos, reglas conocidas como *reglas de validación*. En su forma más sencilla, una regla de validación puede asegurar por ejemplo que no se almacenen accidentalmente caracteres alfabéticos en un campo que sólo debería contener números.

1.3.3. BREVE DESCRIPCIÓN DE MICROSOFT ACCESS.

Microsoft Access es una base de datos que aprovecha características que proporciona el entorno Windows en cuanto a facilidad de uso. Microsoft Access utiliza la interfaz múltiple (MDI) de Windows que le permite trabajar con varios objetos al mismo tiempo. Esto significa que se puede estar trabajando con varias tablas, formularios, informes, macros o módulos (ver siguiente ítem) a la vez.

1.3.3.1. Arquitectura de Microsoft Access.

Microsoft Access llama a todo lo que puede tener un nombre *objeto*. Dentro de una base de datos de Microsoft Access, los objetos principales son: tablas, consultas, formularios, informes, macros y módulos.

Tabla.- Objeto que se define y utiliza para almacenar datos. Una tabla contiene información sobre un tema o asunto particular. Las tablas contienen *campos* que almacenan los diferentes tipos de datos, y registros (también conocidos como *filas*) que recogen toda la información sobre una instancia determinada de un tema.

Empleados : Tabla						
Id. de empleado	Apellidos	Nombre	Cargo	Tratamiento	Fecha de nacimiento	Fe
1	Davolio	Nancy	Representante de ventas	Sra.	08-Dic-48	
2	Fuller	Andrew	Vicepresidente comercial	Dr.	19-Feb-52	
3	Leverling	Janet	Representante de ventas	Sra.	30-Ago-63	
4	Peacock	Margaret	Representante de ventas	Sra.	19-Sep-37	
5	Buchanan	Steven	Gerente de ventas	Sr.	04-Mar-55	
6	Suyama	Michael	Representante de ventas	Sr.	02-Jul-63	
7	King	Robert	Representante de ventas	Sr.	29-May-60	
8	Callahan	Laura	Coordinador ventas interno	Sra.	09-Ene-58	
9	Dodsworth	Anne	Representante de ventas	Sra.	27-Ene-66	
*	(Autonumérico)					

FIGURA I-3. TABLA DE MICROSOFT ACCESS.^[7]


Consulta.- Objeto que se usan para ver, cambiar y analizar datos de distintas maneras. Proporciona la visión personal de los datos es decir se puede escoger que tipos de datos se desea ver o cambiar de una tabla o de un conjunto de tablas. Por ejemplo si se tiene una tabla con los nombres de los alumnos y otra tabla con las calificaciones, se puede crear una consulta para ver los datos de los alumnos que empiezan con la letra A y cuyas calificaciones sean mayores a 8, estos datos se visualizan en forma de una tabla, es decir los campos como columnas y los registros como filas. En Microsoft Access, se puede crear consultas utilizando una facilidad gráfica que se dispone o escribiendo sentencias en SQL (lenguaje estructurado de consultas).

Formulario.- Es un objeto en el que se colocan controles para efectuar acciones o para introducir datos en los campos, mostrarlos y editarlos. Diseñado principalmente para convertirse en la interfaz amigable para la entrada datos. Se utiliza para personalizar completamente la presentación de datos que se extraen de las consultas o tablas (las tablas o consulta se conoce como *origen de datos*).

Empleados _ □ ×

Nancy Davolio

Información de compañía	Información personal
Id. de empleado:	174
Nombre:	Nancy
Apellidos:	Davolio
Cargo:	Representante de ventas
Jefe:	Fuller, Andrew
Fecha de contratación:	01-May-92
Extensión:	5467




Registro: 1 de 9

FIGURA I-4. EJEMPLO DE UN FORMULARIO EN MICROSOFT ACCESS.^[8]

Informe: Es un objeto que presenta información de datos almacenados en una tabla o consulta de manera ordenada según la necesidad del usuario. Ejemplos de informes son resúmenes, listas de teléfonos y etiquetas para el correo. Un informe es una forma efectiva de presentar los datos en formato impreso como por ejemplo: facturas de venta.

Catálogo _ □ ×

Bebidas
Gaseosas, café, té, cervezas y
malts



Nombre de producto:	Id. de producto:	Cantidad por unidad:	Precio por unidad:
Café de Malasia	43	16 - latas 500 g	\$ 46.00
Cerveza Klosterbräu	75	24 - bot. 0,5l	\$ 7.75
Cerveza Laughing Lumberjack	67	24 - bot. 12l	\$ 14.00
Cerveza negra Steekeye	35	24 - bot. 12l	\$ 18.00
Cerveza Outback	70	24 - bot. 355 ml	\$ 15.00

Página: 3 de 3

FIGURA I-5. INFORME EN MICROSOFT ACCESS.^[9]

Macro.- Objeto que define en forma estructurada la acción o acciones que el usuario desea que Access realice en respuesta a un evento determinado, tal como abrir un formulario o imprimir un informe. Las macros pueden ayudar a automatizar las tareas comunes, por ejemplo, puede ejecutar una macro que imprima un informe cuando el usuario haga clic en un botón de comando.

Módulo.- Objeto que contiene procedimientos personales que se codifican utilizando Access Basic (versiones 7.0 o anteriores de Microsoft Access) o Visual Basic para Aplicaciones (versión 8.0). Los módulos ayudan a construir aplicaciones de base de datos sofisticadas.

La Figura I-6 muestra una visión general conceptual de la relación entre los objetos de Microsoft Access (ver también Figura I-7).

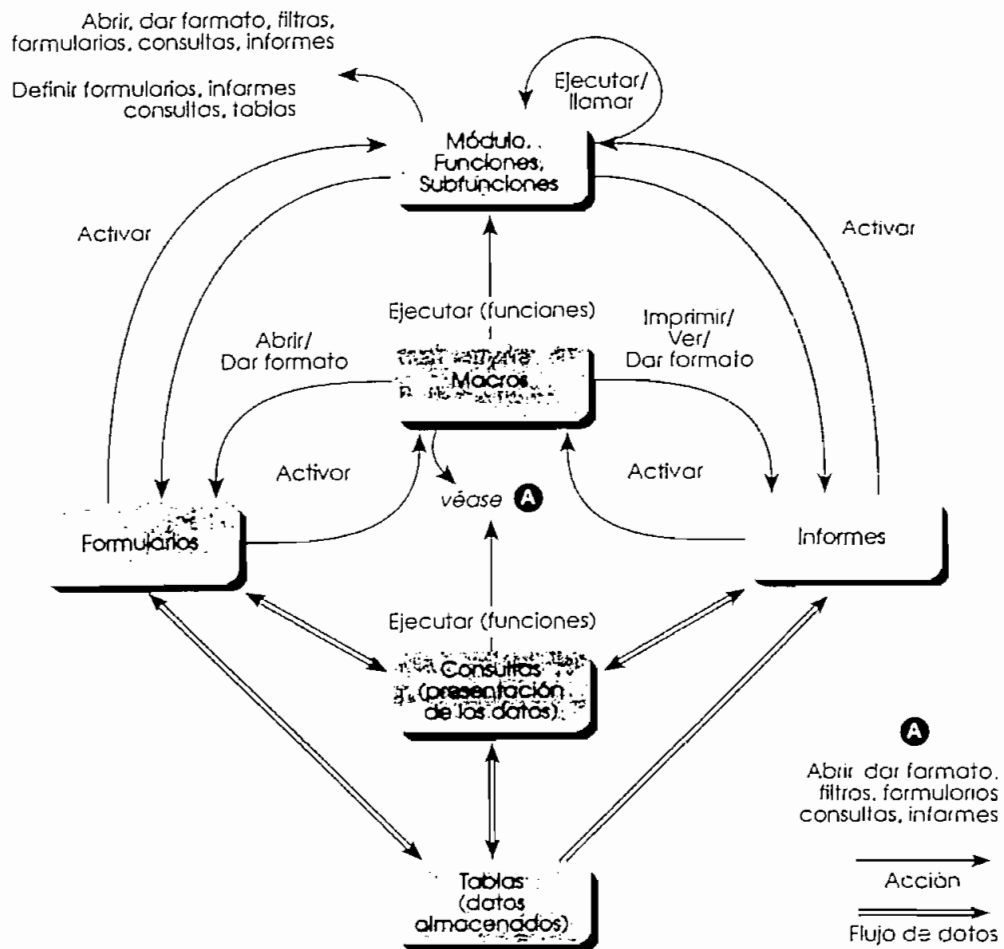


FIGURA I-6. RELACIÓN ENTRE LOS OBJETOS DE MICROSOFT ACCESS.^[10]

1.3.3.2. Arquitectura de una aplicación de Microsoft Access.

Microsoft Access posee dos componentes principales: el motor o máquina de aplicación que controla la programación y la interfaz de usuario final con el producto, y el JET DBEngine, que controla el almacenamiento de los datos y la definición de todos los objetos de la base de datos. (En la Figura I-7 se muestra la arquitectura de Microsoft Access). Cuando abrimos una base de datos, la máquina de aplicación utiliza el DBEngine para determinar el nombre de todas las tablas, consultas, formularios, informes, macro y módulos a visualizarse en la base de datos.

El DBEngine controla todos los objetos de su base de datos a través de una jerarquía de colecciones, objetos y propiedades.

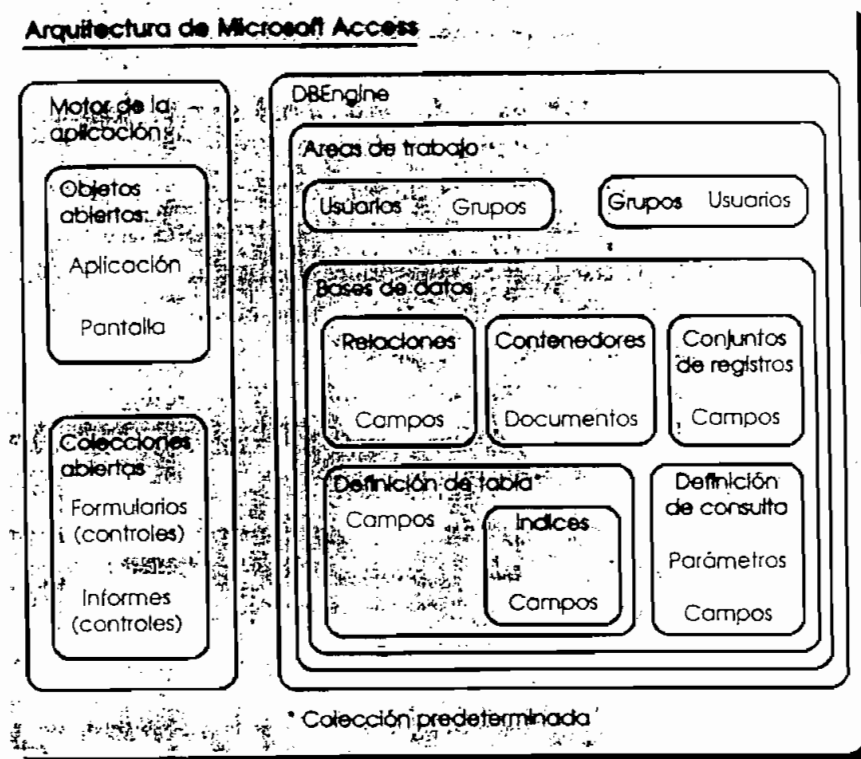


FIGURA I-7. ARQUITECTURA DE MICROSOFT ACCESS.^[11]

Métodos y propiedades de Microsoft Access.

Como se ha explicado Access posee una Arquitectura basada en objetos, así por ejemplo: tablas, consultas, registros son objetos (ver Figura I-7). Cada objeto tiene métodos y propiedades que pueden ser utilizados mediante Automatización OLE. Además la relación de objetos, métodos y propiedades responde a una jerarquía, por ejemplo no puede ver un registro de datos de una tabla sin antes abrir la base de datos y la tabla. La jerarquía de los objetos indica un orden para manipular los datos de una base de datos.

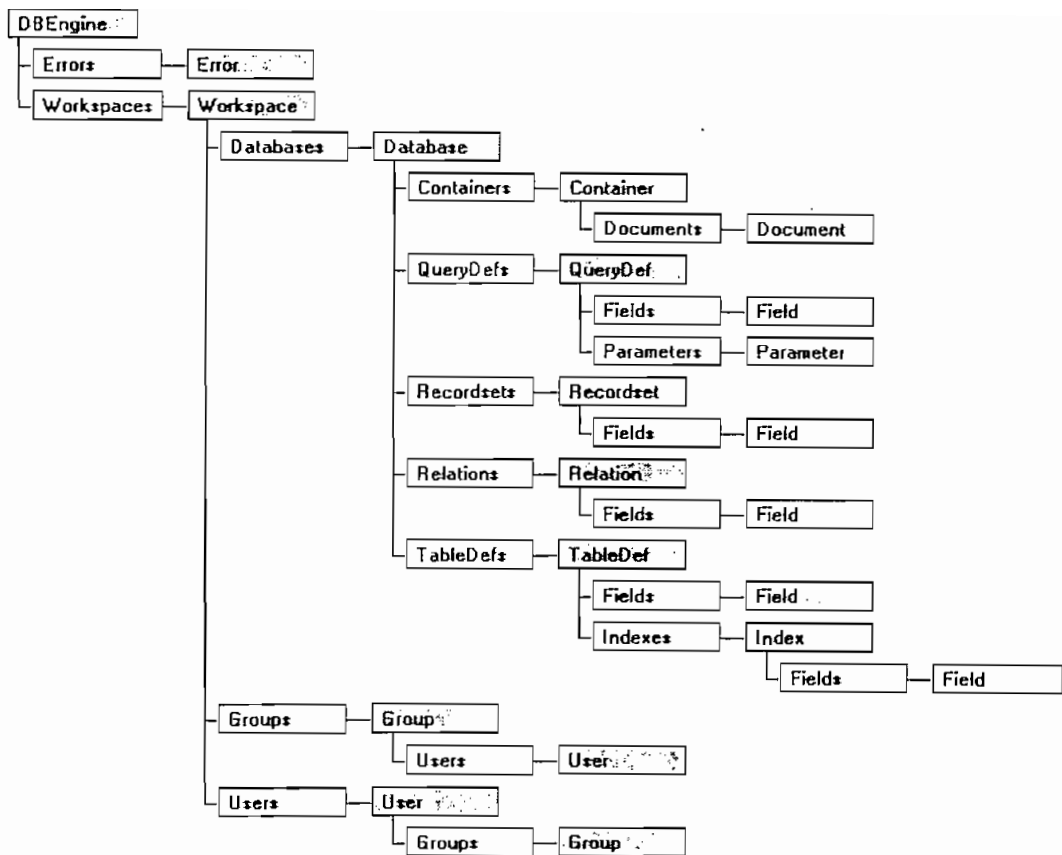


FIGURA I-8. JERARQUÍA DBENGINE.^[12]

Métodos comúnmente utilizados para la manipulación de datos.

Cuando se desea aplicar una acción para el caso de un objeto de una base de datos (tal como abrir una consulta como un conjunto de registros o ir a la siguiente fila de un conjunto de registros), aplicamos un *método* para el objeto o para una variable objeto que se ha asignado para indicar a ese objeto. La mayor parte de los métodos aceptan parámetros.

Microsoft Access soporta muchos métodos de objetos diferentes, tal vez uno de los grupos de métodos más útil es el grupo que podemos utilizar para crear un conjunto de registros y luego leer, actualizar, insertar y eliminar filas del mismo.

Para crear un conjunto de registros, en primer lugar se debe declarar una variable objeto Recordset. A continuación, abrir el conjunto de registros utilizando el método

OpenRecordSet frente a la base de datos actual (especificando un nombre de tabla, un nombre de consulta, o una cadena SQL para crear un conjunto de registros). Después de abrir un conjunto de registros, podemos utilizar uno de los métodos Move para desplazarnos a un registro determinado. Para eliminar una fila en un conjunto de registros actualizables, simplemente se desplaza a esa fila del conjunto de registros que desea eliminar y, a continuación se utiliza el método Delete.

Si se desea actualizar las filas de un conjunto de registros, se desplaza a la fila que desea actualizar y, a continuación se utiliza el método Edit para bloquear la fila y hacerla actualizable. Se utiliza el método Update en el conjunto de registros para almacenar los cambios antes de desplazarse a otra fila.

Finalmente, para insertar una nueva fila en un conjunto de registros, se inicia una nueva fila utilizando el método AddNew. Se establece los valores de todos los campos requeridos en la fila, y luego se utiliza el método Update para almacenar la nueva fila.

1.3.3.3. Introducción al lenguaje SQL.

Una consulta SQL es una consulta creada mediante una instrucción SQL. Una instrucción SQL es una **expresión** que define una acción a tomarse con la consulta.

Este tema trata sobre cómo crear una instrucción sencilla de Lenguaje de consultas estructurado (SQL) para manipular los datos de la base de datos. Las instrucciones SQL pueden ser utilizadas para recuperar datos de una o más tablas, añadir, actualizar o eliminar registros, ejecutar procedimientos almacenados o instrucciones SQL en bases de datos externas (o remotas).

Generalidades

La estructura básica de una instrucción SQL es la siguiente:

- En primer lugar se tiene una sentencia que es el núcleo de la instrucción SQL, esta sentencia indica la acción que tomará la consulta.
- Luego se ingresa los campos (separados por comas) que se desea que aparezca en la consulta.
- Se utiliza la cláusula FROM y se ingresa las tablas que contienen los datos de origen para la consulta.
- Luego se tiene una serie de comandos para organizar los datos de acuerdo a lo que se necesite, por ejemplo: se puede organizar los datos en orden alfabético descendente.

Un ejemplo de una instrucción sencilla sería:

```
SELECT Nombre, Apellido, Ciudad  
FROM Empleados  
WHERE Ciudad="Quito"  
ORDER BY Apellido
```

La instrucción SQL indicada realiza una consulta que selecciona los datos (nombre, apellido y ciudad) de la Tabla Empleados cuyos datos del campo ciudad sea Quito (cláusula WHERE). Los datos se ordenan en forma ascendente (cláusula ORDER BY) respecto al campo Apellido.

A continuación se indica una instrucción que modifica datos:

```
INSERT INTO Empleados  
VALUES (Nombre = "Juan", Apellido = "Guzmán", Ciudad = "Tulcán")
```

Esta instrucción SQL añade un registro a la tabla Empleados en los campos: Nombre, Apellido y Ciudad con los valores Juan, Guzmán y Tulcán respectivamente (si la tabla tiene más campos, estos se llenan con valores por defecto que se diseñan con la tabla).

Se ha presentado estos dos ejemplos debido a que son las dos instrucciones SQL que se utilizan en el desarrollo del programa del trabajo de investigación que se realiza (ver Documentación del Software).

1.4. BREVE DESCRIPCIÓN DE LABVIEW.

Labview es un programa desarrollado para crear aplicaciones, diseñado con un lenguaje de programación gráfica (lenguaje G). Los programas se realizan en forma de diagramas de bloques y utilizan una terminología de íconos e ideas familiares con las ciencias y la ingeniería, es decir se relacionan con símbolos gráficos. Cada bloque (ícono) puede tener una descripción del programa que ejecuta y representa lo que en un lenguaje de programación escrito es una función o una subrutina.

Labview tiene una extensa librería de funciones y subrutinas para la mayoría de las tareas de programación, además de librerías especializadas para funciones de adquisición de datos, VXI, GPIB (instrumentación computarizada y protocolos de comunicación industrial) y control de instrumentos por protocolo serial. Este programa tiene un uso preferente en sistemas industriales donde se requiere control de procesos a través de señales procedentes de varios equipos y representa una de las razones para el desarrollo de la presente tesis en este ambiente de programación.

A continuación se presenta las rutinas que Labview presenta para realizar una comunicación a través de automatización OLE.

1.4.1. LABVIEW COMO CLIENTE DE AUTOMATIZACIÓN OLE. SUBVIS INCORPORADOS EN EL PROGRAMA.

Labview es un Cliente de la automatización OLE. Este puede acceder a métodos y propiedades de objetos expuestos en diferentes aplicaciones en un sistema. Las aplicaciones como Labview utilizan números de referencia para acceder a la programación por objetos. Cada vez que se desea acceder a un nuevo objeto, método o propiedad el servidor de automatización OLE genera un identificador explícito para cada uno.

La figura I-9 demuestra como Labview se relaciona con aplicaciones, estas demuestran los Objetos OLE. Labview, empieza como Cliente de la Automatización OLE, puede ejecutar acciones que involucren a métodos, llamadas o poniendo propiedades, mientras el servidor define y demuestra los objetos de automatización. Labview también accesa a las librerías que dan información sobre el servidor de objetos.

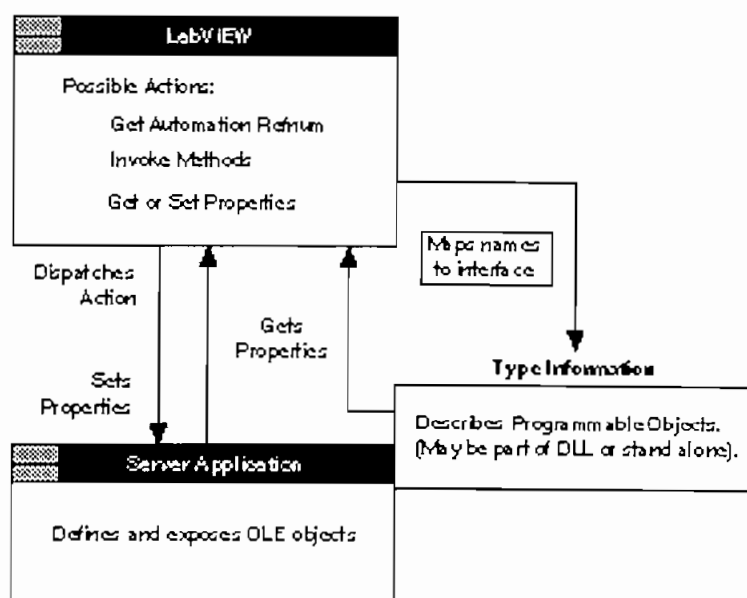


FIGURA I-9. RELACIÓN DE LABVIEW CON OTRAS APLICACIONES. [13]

La comunicación con automatización OLE es semejante con Word o Excel para Labview, usando los Vis que se demuestran en la figura I-10.

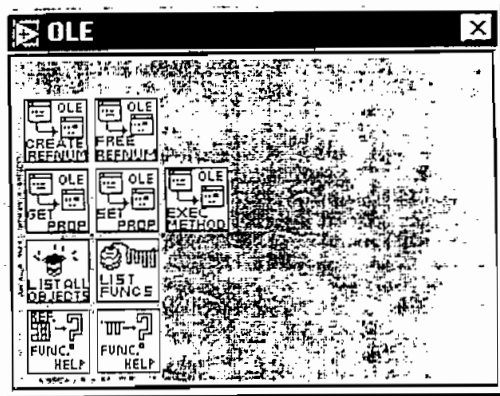


FIGURA I-10. SUBVIS DE PARA AUTOMATIZACIÓN OLE.^[14]

Create Automation Refnum llama a referencia numérica de la automatización para definir la interfaz Idispatch.

Execute Method VI ejecuta un método o función.

Set Property VI Pone el valor a una propiedad.

Get Property VI se utiliza par leer el valor presente de una propiedad.

Release Refnum VI libera la automatización.

I.4.2. UN EJEMPLO DEL USO DE LOS VIS DE LABVIEW PARA UNA COMUNICACIÓN OLE CON EXCEL.

Para poder comunicar Labview con otra aplicación se debe conocer el modelo de objetos de la aplicación con la que se trabaja. Para el desarrollo del ejemplo con Excel, la aplicación define un modelo conformado de 128 objetos, entre los que se dispone por ejemplo una gráfica (objeto *Chart*) o una hoja de datos (objeto *Worksheet*). Antes de escribir una rutina en Labview para acceder a los objetos de Excel es esencial entender la estructura y funcionalidad de cada uno de los objetos.

Los objetos en Excel tienen jerarquía (para conseguir una lista de todos los objetos y

propiedades que tiene Excel se puede referir a la ayuda en línea de Excel, *Ayuda>Contenido>Programación con Visual Basic*). Para llamar a los métodos y propiedades de un objeto en Excel se debe establecer un enlace primero con todos los objetos superiores dentro de la jerarquía. Por ejemplo para acceder al objeto *Worksheet* primero se debe acceder al objeto *Application*, luego al objeto *Workbook* y luego al objeto *Worksheet*. Labview usa el concepto de números de referencia (RefNums) para acceder a los objetos dentro de la jerarquía del modelo de automatización. Cuando se empieza una comunicación utilizando automatización OLE se utiliza el VI Create Automation RefNum para generar un numero de referencia de automatización de la aplicación, una vez que se tiene este numero de referencia se puede acceder a los demás objetos a través de sus métodos y propiedades.

Como un ejemplo considere que se desea suministrar datos de un array desde Labview a Excel y luego se desea mostrar los datos en un gráfico en Excel utilizando automatización OLE. A continuación se muestra el diagrama de programación en Labview que realiza esta acción.

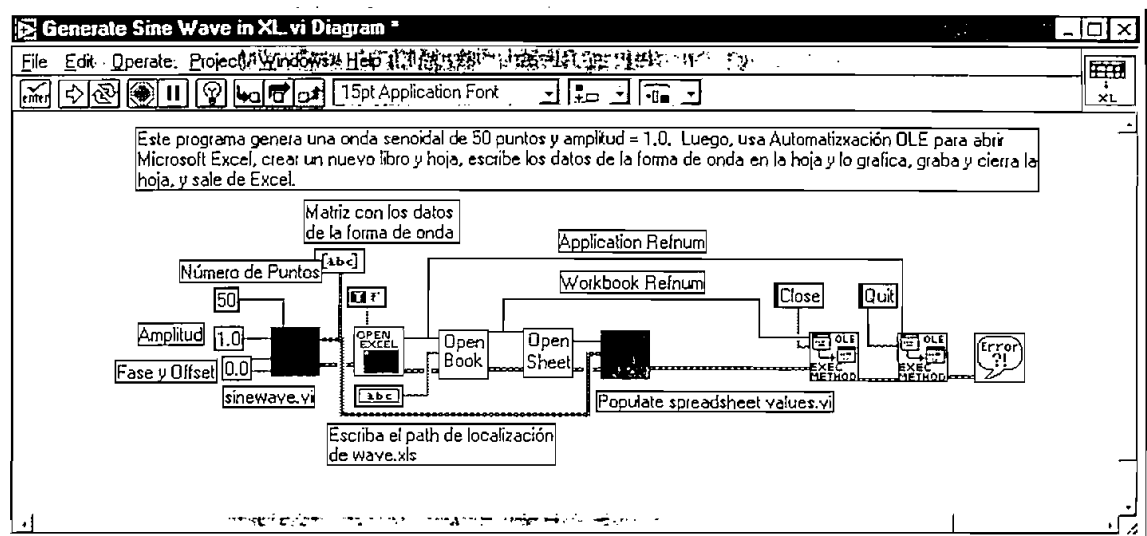


FIGURA I-11. VI QUE GENERA UNA ONDA SENOIDAL EN EXCEL.^[15]

El subvi Sine Wave.Vi genera una matriz con los datos de la onda senoidal. De acuerdo al modelo jerárquico de objetos de Excel, el objeto *Application* es el primero en llamarse, a continuación el objeto *Workbook* y luego el objeto *Worksheet*. Labview abre la aplicación Excel y luego el libro de trabajo "Wave.xls" y luego traza la onda senoidal con los datos que se envían desde la Fila 3 a la 52 en la Columna 1. Después de acceder al objeto worksheet, el "populate spreadsheet values.vi" (Figura I-12) usa el método *Range* para direccionar a celdas específicas en la hoja de datos y la propiedad *Value* para escribir valores en dicha celda.

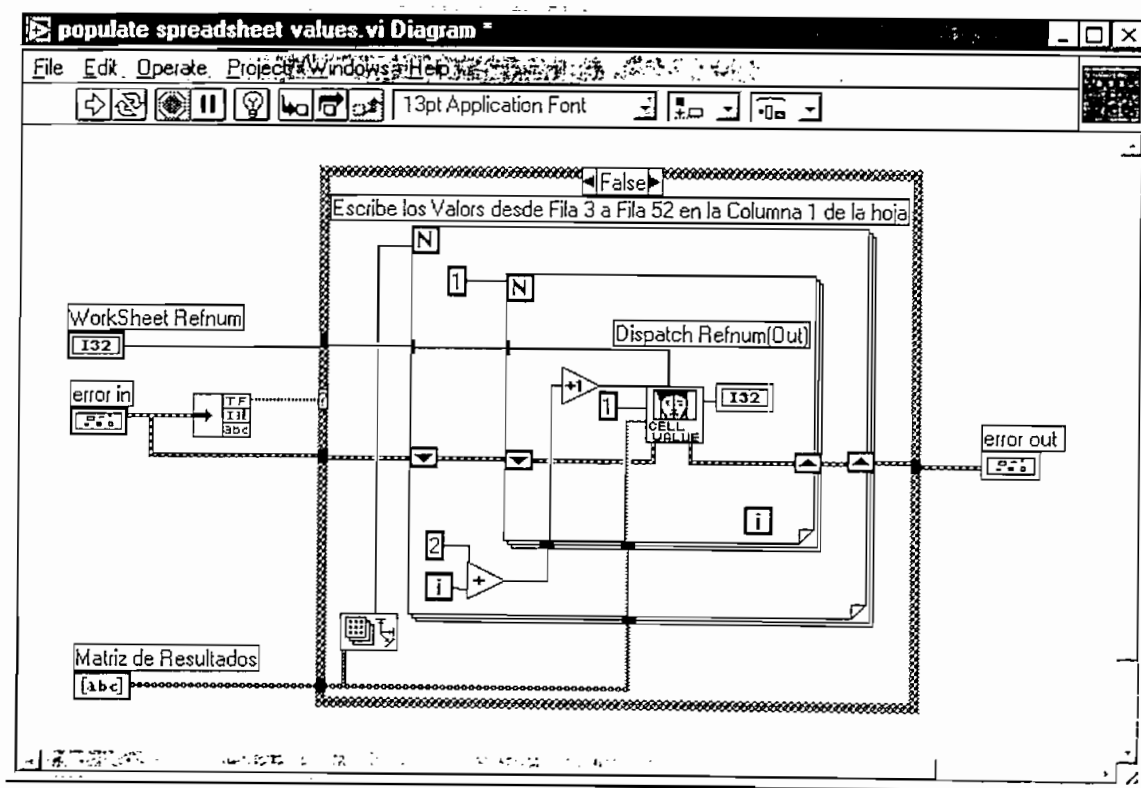


FIGURA I-12. VI POPULATE SPREADSHEET VALUES

Por último Labview utiliza el método *Close* para cerrar la comunicación (números de referencia) con el objeto *Workbook* y luego el método *Quit* para salir de Excel.

A continuación se muestra la figura que se obtiene en Excel.

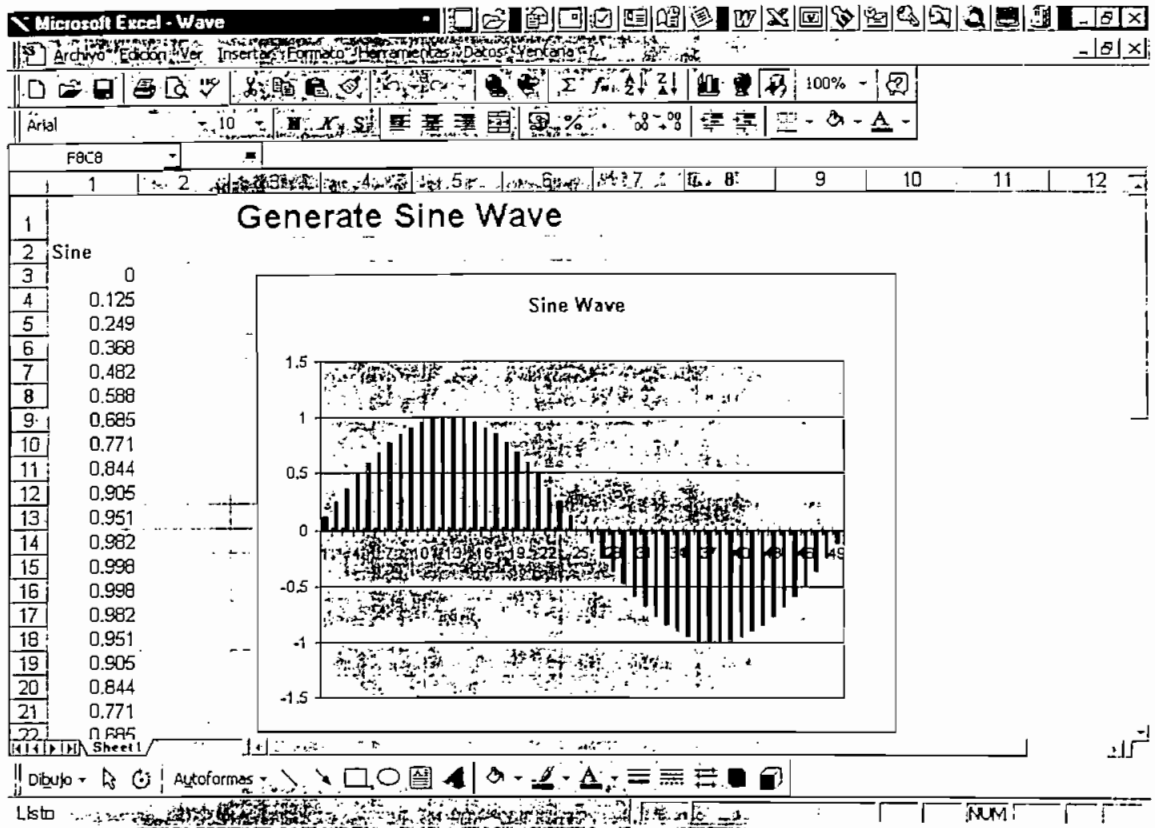


FIGURA I-13. ONDA SENOIDAL EN EXCEL.^[16]

CAPÍTULO II:

DESARROLLO DEL HARDWARE.

II.1. REQUERIMIENTOS.

El sistema a desarrollarse en la presente investigación debe contener las siguientes características en cuanto al hardware del sistema:

1. Interfaz de comunicación serial: uno de RS-232 a RS-422 que se acoplará al puerto serial del computador personal y otro de comunicación con niveles TTL a RS-422 que se conectará con el microcontrolador.
2. Manejo de 8 entradas de alarmas con niveles TTL.
3. Interfaz de control para dos lectoras magnéticas de proximidad.
4. Interfaz de lectura de dipswitches.
5. Salidas de relé para el manejo de las cerraduras.
6. Salidas de mensajes a través de un dispositivo de visualización (display).
7. Manejo de pulsantes que activen las salidas de relés.

El equipo para su funcionamiento requiere un microcontrolador que manejará el hardware antes indicado. El microcontrolador debe cumplir con características especiales para esta aplicación, que son: preservar la información almacenada en ausencia de sistema de alimentación (memoria RAM no volátil), programación a través del pórtilo serial sin necesidad de desarrollar un sistema adicional para esta acción, reloj-calendario en tiempo real. El microcontrolador que cumple con estos requerimientos es el Dallas5000T 32/16

que además se ajusta al estándar 8051 en cuanto a la arquitectura y al set de instrucciones, son compatibles de pin a pin. El Dallas5000T 32/16 tiene las siguientes características: batería interna de litio para preservar el reloj, 32K de memoria no volátil, programación a través del pòrtico serial, 16Mhz de velocidad.

El desarrollo de los requerimientos del sistema propuesto se lo puede sintetizar a través del siguiente diagrama de bloques.

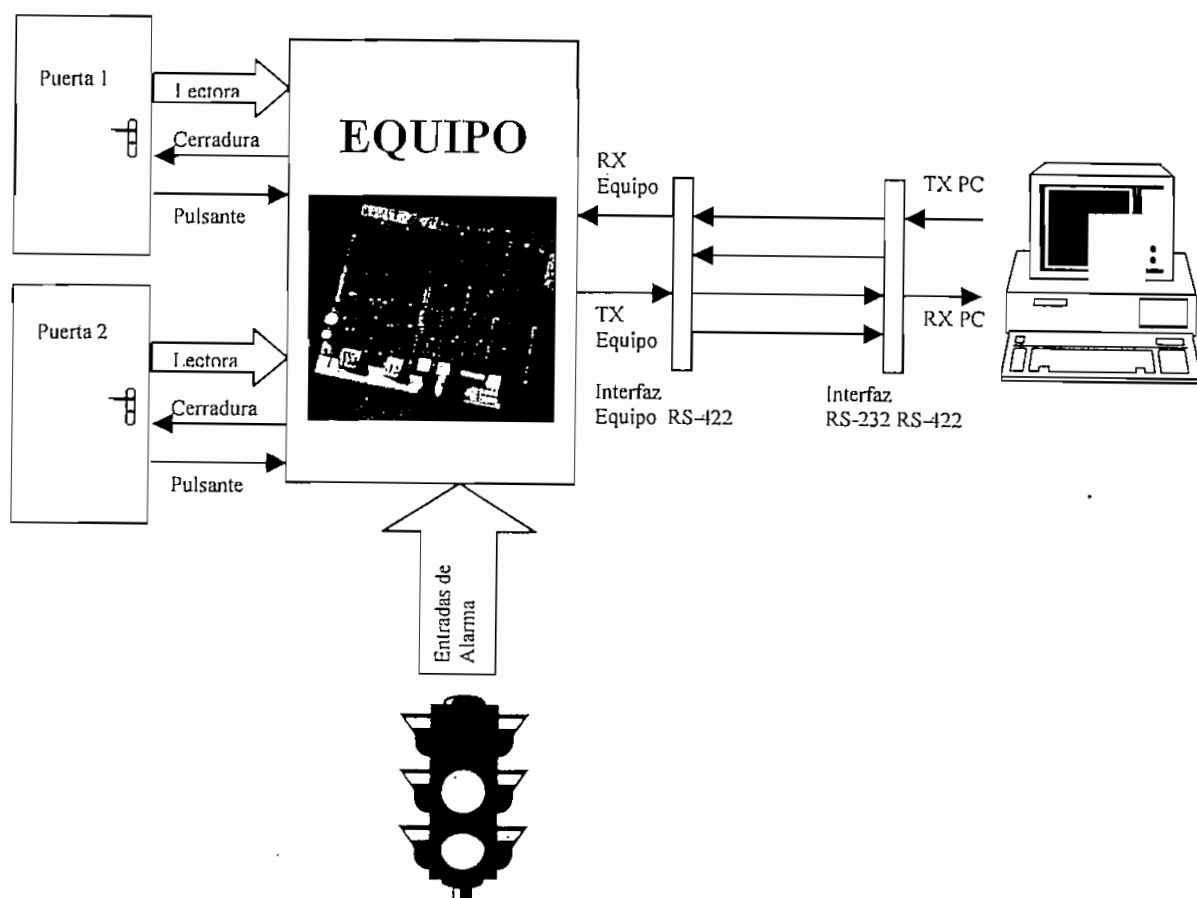


FIGURA II-1. DIAGRAMA DE BLOQUES DEL SISTEMA PROPUESTO.

En los siguientes ítems del presente capítulo se detalla los diseños constructivos de cada uno de los bloques indicados en la figura II-1.

II.2. INTERFAZ DE COMUNICACIÓN SERIAL.

Para el desarrollo de la interfaz de comunicación serial se plantea como filosofía de interconexión de los diferentes equipos una topología de red tipo anillo (figura II-2) que sea administrada conforme los estándares del protocolo RS-422.

Como se observa para la comunicación se necesitará una interfaz que irá conectada al computador y otra interfaz que se incluirá en la tarjeta del microcontrolador.

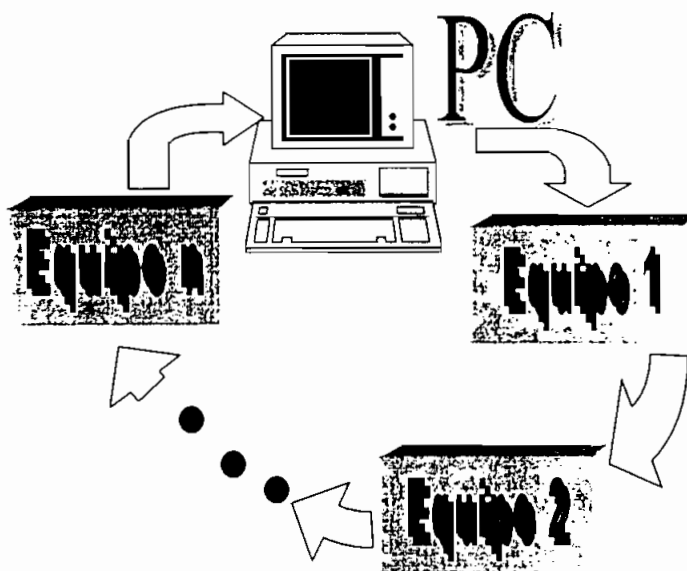


FIGURA II-2. TOPOLOGÍA EN ANILLO.

II.2.1. INTERFAZ DEL COMPUTADOR.

Puesto que la interfaz se va a realizar entre el computador y el (los) microcontrolador(es), se debe tomar en cuenta que el pórtilo de comunicación serial del computador normalmente trabaja en modo Full-Duplex con protocolo RS-232 lo que significa que los niveles de voltaje del pórtilo están normalmente en ± 12 V y se necesita convertir a niveles TTL que maneja el microcontrolador, así como son los niveles de voltaje dados para la lógica de comunicación del protocolo RS-422, por lo cual se necesita un circuito de

conversión de RS-232 a RS-422 que será exclusivo para el computador.

En la figura II-3 se puede observar el diagrama esquemático RS-232 a RS-422; allí las líneas muestran la forma en que se desplazan las señales. De este diagrama conviene destacar los siguientes aspectos:

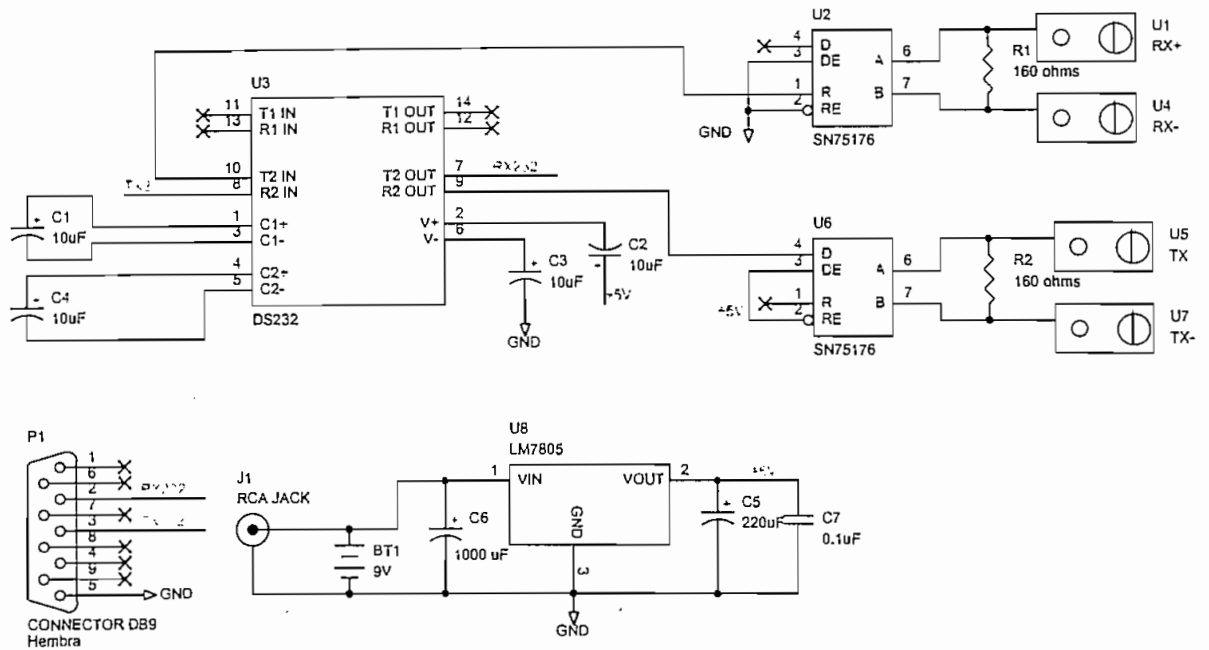


FIGURA II-3. CONVERSOR RS-232 A RS-422.

1. El circuito integrado U3 (MAX 232) es un convertidor RS-232 a niveles TTL. Este circuito hace el traslado de los niveles presentes en las salidas del computador PC (que difieren de niveles TTL) a niveles de voltaje que pueden interpretar los circuitos integrados U2 y U6 (SN75176), y viceversa (los niveles de salida TTL de los circuitos integrados 75176 son convertidos a niveles de entrada RS 232, para que ingresen al computador a través de la línea RX).
2. El circuito integrado U6 tiene su terminales RE (Receiver Enable, activo a un nivel lógico bajo) y DE (Driver Enable, activo a un nivel lógico alto) unidos a un potencial de 5 voltios, con lo cual éste queda habilitado para transmitir los datos e inhibido para

recibirlos. Este circuito integrado sólo puede trabajar en la tarjeta como transmisor de datos, los cuales toma de la entrada marcada como DI y los envía a través de los terminales marcados en el conector como T⁺ y T⁻.

3. Los terminales RE y DE del circuito integrado U2 se encontrarán en un estado lógico bajo, con lo cual el circuito integrado U2 actuará solamente como un receptor de datos.

La interfaz más sencilla es la RS-422, ya que es bidireccional, y con las líneas separadas para la transmisión y la recepción de datos

Para estas conexiones, se debe recordar que en la interfaz RS-422 sólo puede existir un transmisor y hasta 32 receptores. Un aspecto que se recalca en el caso de una red numerosa, es que solamente la interfaz de los extremos de la red debe llevar una resistencia entre las líneas de recepción.

II.2.2. INTERFAZ DEL MICROCONTROLADOR.

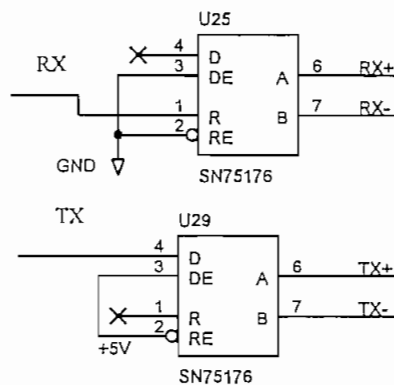


FIGURA II-4. CONVERTOR RS-422 A NIVELES TTL.

El microcontrolador necesita convertir de la estructura de dos líneas para transmisión y dos para recepción del protocolo RS-422 a una línea de recepción y otra de transmisión que admite el microcontrolador, para lo cual se utiliza el circuito convertidor RS-232 a RS-422 con la eliminación del circuito integrado MAX232 como se muestra en la figura II-4.

II.3. INTERFAZ DE MANEJO DE OCHO ENTRADAS DE ALARMAS.

En general las entradas de alarma que ingresan al equipo provienen de sensores que se encuentran a una distancia considerable del equipo de control. En muchas ocasiones por la naturaleza del origen de estas señales pueden llegar a niveles peligrosos para el microcontrolador, por esta razón se considera como una manera de protección al microcontrolador el realizar un aislamiento entre las señales que ingresan al equipo y los pines a los cuales se los conectará a través de un driver de tres estados 74LS244.

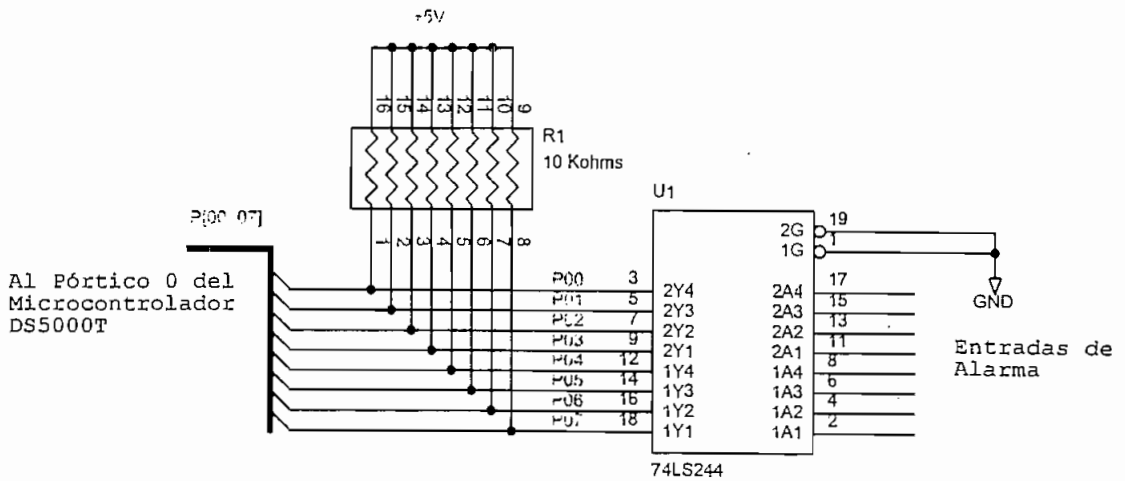


FIGURA II-5. CIRCUITO DE INTERFAZ PARA LAS ENTRADAS DE ALARMA.

II.4. INTERFAZ DE MANEJO DE DISPLAY.

Para el desarrollo de la interfaz se utiliza un display de cristal líquido con pines compatibles al LCD PHILIPS, utilizado en la Escuela Politécnica Nacional en el Laboratorio de Control con Microcontroladores, el manejo del display se realiza a través de un pòrtico del microcontrolador.

El display tiene dos modos de operación a 8 bits y a 4 bits, en este caso se utilizará el modo

a 4bits (4 líneas de datos) debido a la disponibilidad de solo 8 líneas en el pórtilo. De las cuatro líneas, tres son utilizadas para el control del display (R/W, E, RS como se indica en el manual de referencia del display).

El circuito se muestra a continuación.

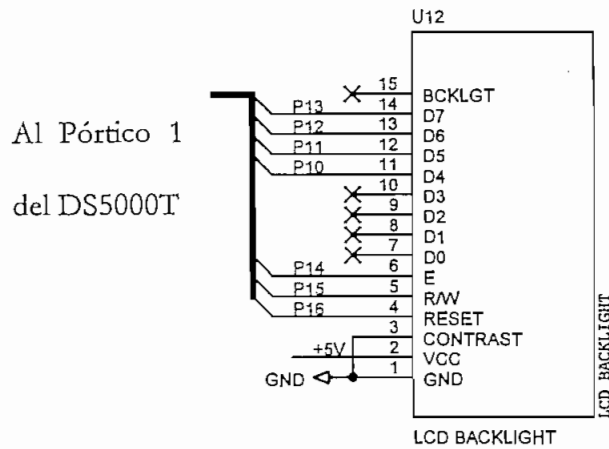


FIGURA II-6. CIRCUITO DE INTERFAZ VISUAL A TRAVÉS DE UN LCD.

II.5. INTERFAZ DE LECTORAS DE TARJETAS, MANEJO DE SALIDAS DE RELE Y LECTURA DE DIPSWITCHES.

Como se ha descrito hasta el momento el microcontrolador ha manejado directamente un solo dispositivo con cada pórtilo. Técnicamente tenemos dos pórtilos libres, sin embargo uno de ellos se utiliza de manera general para el manejo de interrupciones y otras funciones generales del microcontrolador, por lo que un solo pórtilo queda libre.

Debido a la necesidad de manejar tres dispositivos y contando con un solo pórtilo libre se requiere utilizar un método de multiplexación para el manejo de éstos dispositivos.

II.5.1. CIRCUITO DE MULTIPLEXACIÓN.

Para el desarrollo de este circuito se utiliza un multiplexor (74LS139) de dos entradas a 4

salidas, que es el mínimo que se encuentra en el mercado, de las cuales la entrada A será controlada por el pin T0 (P3.4) del microcontrolador y la entrada B se conectara a T1 (P3.5), como muestra la figura II-7.

Las salidas Y0 y Y1 serán utilizadas para controlar la habilitación de dos búffer de tres estados (74LS244) que son los encargados de activar al dispositivo respectivo con el pórtilo del microcontrolador. Es decir se simulan los dispositivos conectados como dispositivos compatibles con microcontroladores que disponen de una línea de habilitación que los sitúa en alta impedancia cuando no son seleccionados y los activa el instante que son habilitados.

Esta disposición de elementos como se muestra en la figura II-7 dan la base para la conexión de dispositivos tanto de escritura como de lectura.

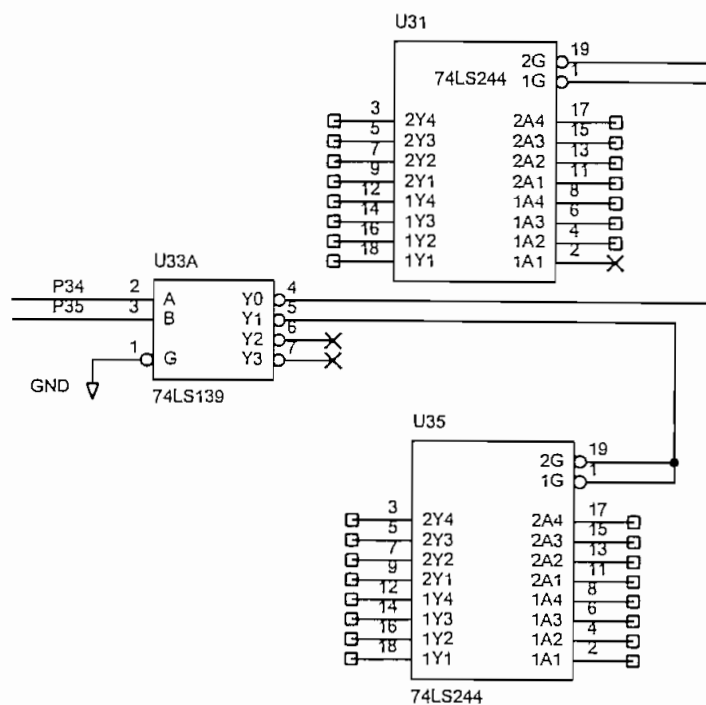


FIGURA II-7. CIRCUITO DE MULTIPLEXACIÓN PARA EL PÓRTICO 2.

II.5.2. PÓRTICO MULTIPLEXADO 0: INTERFAZ DE LECTURA DE DIPSWITCHES Y LECTURA DE PULSANTES.

Para conectar equipos en red se requiere definir diferentes identificadores para cada uno de estos. Los dipswitches permiten el cambio de identificador. El equipo lee los dipswitches durante la ejecución del programa principal y determina el número de identificador. Los dipswitches están conectados al pÓrtico multiplexado 1 como se muestra en la figura II-8.

Por otro lado el dispositivo debe leer dos pulsantes que permitan la activación manual de las salidas de relé. Estos pulsantes se conectan a dos pines libres del pÓrtico multiplexado al que se encuentran conectados los dipswitches.

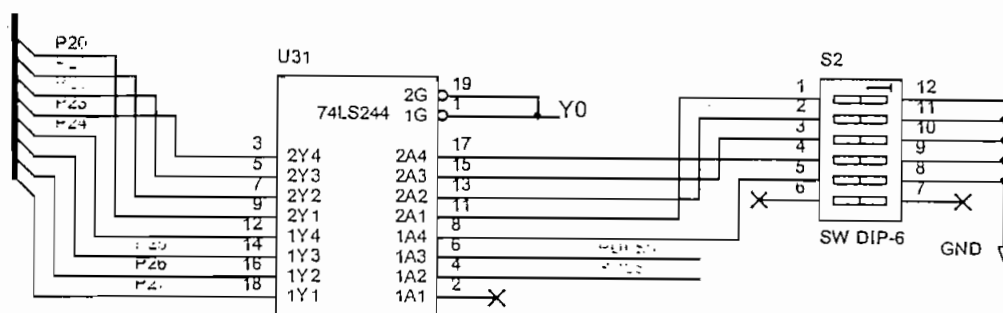


FIGURA II-8. CIRCUITO DE INTERFAZ PARA LECTURA DE DIPSWITCHES Y PULSANTES.

PUERTO MULTIPLEXADO 0

II.5.3. PÓRTICO MULTIPLEXADO 1: INTERFAZ DE LECTORAS MAGNETICAS DE PROXIMIDAD Y SALIDAS DE RELES.

En el interfaz de las lectoras magnéticas, el microcontrolador debe leer los datos a través de dos líneas de datos, DATA0 y DATA1 (como se explica en el capítulo I), proporcionadas por las lectoras. Estas líneas de datos normalmente se encuentran en estado alto (uno lógico), cada vez que una tarjeta interactúa existe un cambio de estado lógico en cualquiera de las dos líneas (según el código de la tarjeta).

Se plantea la lectura de datos por medio de interrupciones externas del microcontrolador aprovechando la transición lógica de las líneas de datos durante la lectura de una tarjeta. Cada lectora (dos en total) se maneja a través de una interrupción diferente que se activa por nivel bajo. Debido a que existen dos líneas de datos en cada lectora y una sola línea de interrupción, la compuerta AND facilita la activación de la interrupción cuando por una de las dos líneas se detecta un cambio de estado lógico.

Además los datos se deben leer a través del pórtico multiplexado del microcontrolador.

El diagrama circuital queda de la siguiente manera.

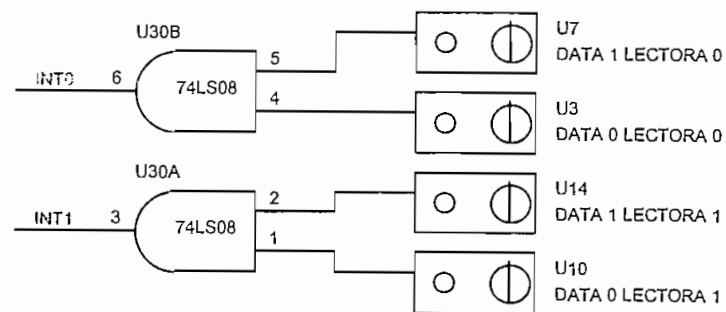


FIGURA II-9. CIRCUITO PARA GENERACIÓN DE INTERRUPCIONES CUANDO ACTUAN LAS LECTORAS.

Cada lectora se relaciona con un actuador que generalmente serán cerraduras eléctricas.

El equipo permite que se conecte el actuador adecuado a través de dos salidas de relés.

Estos relés tienen las siguientes características:

- Corriente: 1[A] de corriente alterna
- Voltaje: 120VAC

Debido a que los relés son elementos inductivos su activación requiere una corriente mayor de la salida que admite cualquiera de los pórticos del microcontrolador. Por esta razón se

necesita colocar entre la bobina de control y el pin correspondiente del microcontrolador un amplificador de corriente. Se ha utilizado para este diseño el buffer ULN2003, amplificador de corriente a 500mA.

Como se observa el pÓrtico multiplexado 0 tiene seis de sus ocho pines como entradas de datos, mientras que dos pines son de salida. Este factor debe tomarse en cuenta en las conexiones con el buffer tres estados.

La figura siguiente muestra el diagrama del pÓrtico multiplexado 1.

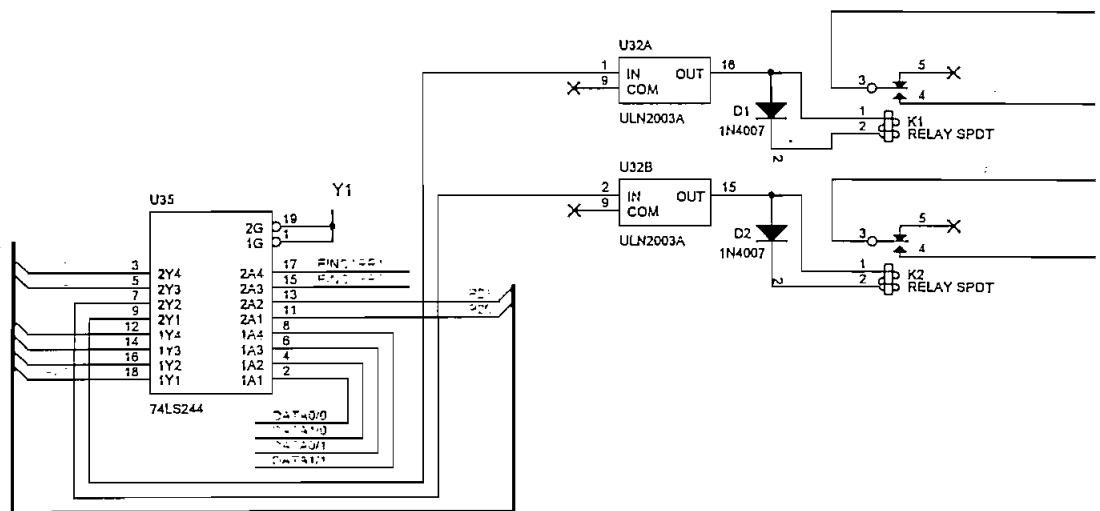


FIGURA II-10. CIRCUITO DE INTERFAZ PARA LAS LECTORAS, SALIDAS DE RELÉ, FINES DE CARRERA. PUERTO MULTIPLEXADO 1.

II.6. DIAGRAMA CIRCUITAL COMPLETO DEL SISTEMA PROPUESTO.

El desarrollo del hardware del sistema consta de dos tarjetas. Una tarjeta para el manejo de la comunicación serial del computador y otra tarjeta que es en sí el hardware del Equipo de Control de Accesos.

El diagrama del circuito impreso del equipo se muestra a continuación.

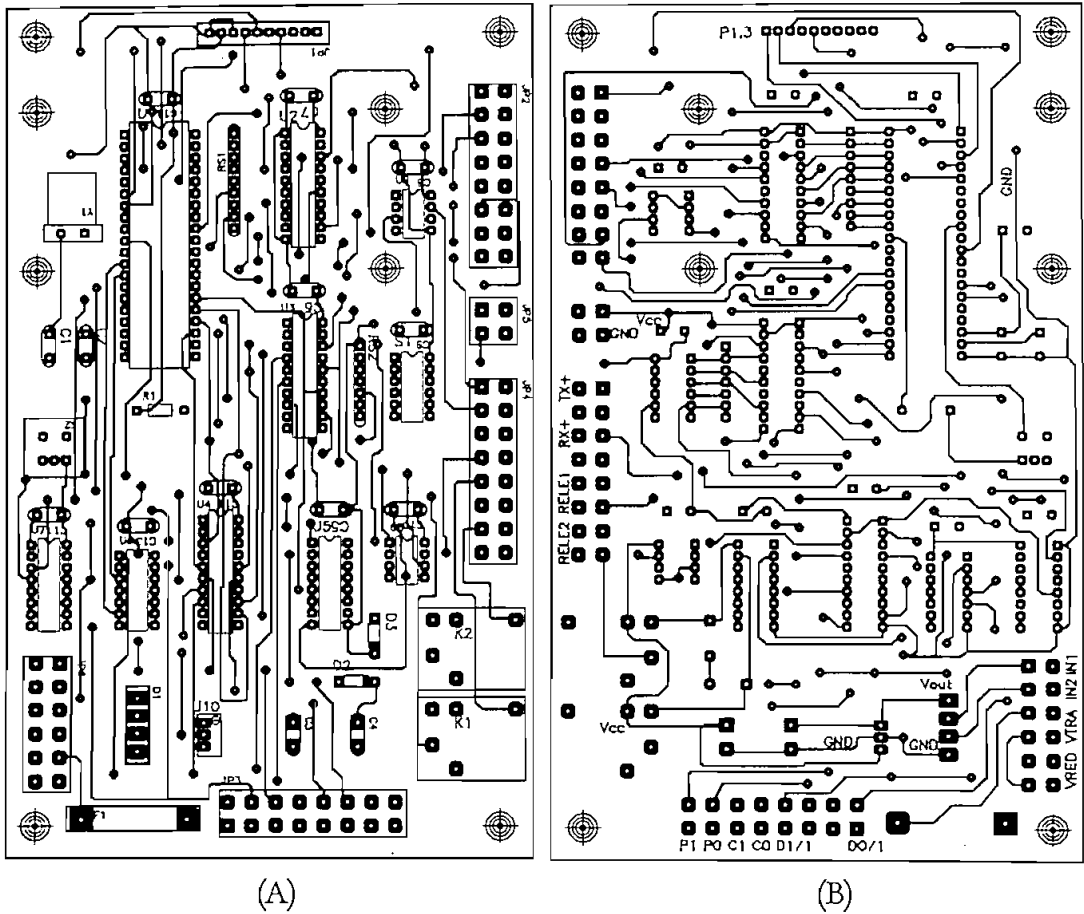


FIGURA II-11. CIRCUITO IMPRESO DEL EQUIPO.

(A) VISTA FRONTAL (B) VISTA POSTERIOR.

El diagrama del circuito impreso de la interfaz RS232-RS422 es:

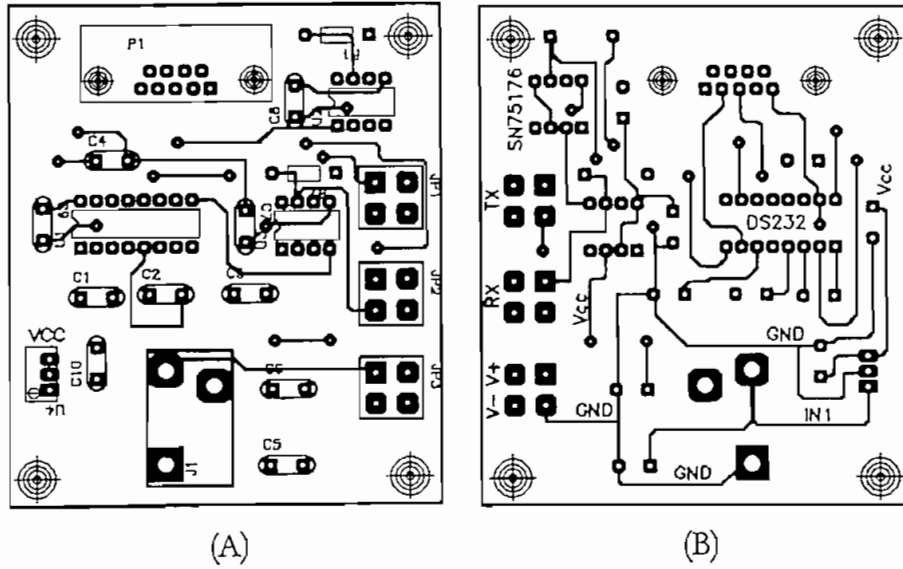


FIGURA II-12. CIRCUITO IMPRESO DE INTERFAZ HACIA EL COMPUTADOR PERSONAL.

(A) VISTA FRONTAL (B) VISTA POSTERIOR.

Como se observa los circuitos impresos disponen de borneras que facilitan las conexiones de los periféricos hacia el equipo.

A continuación se muestran los diagramas esquemáticos completos del equipo e interfaz de comunicación serial explicados en el presente capítulo.

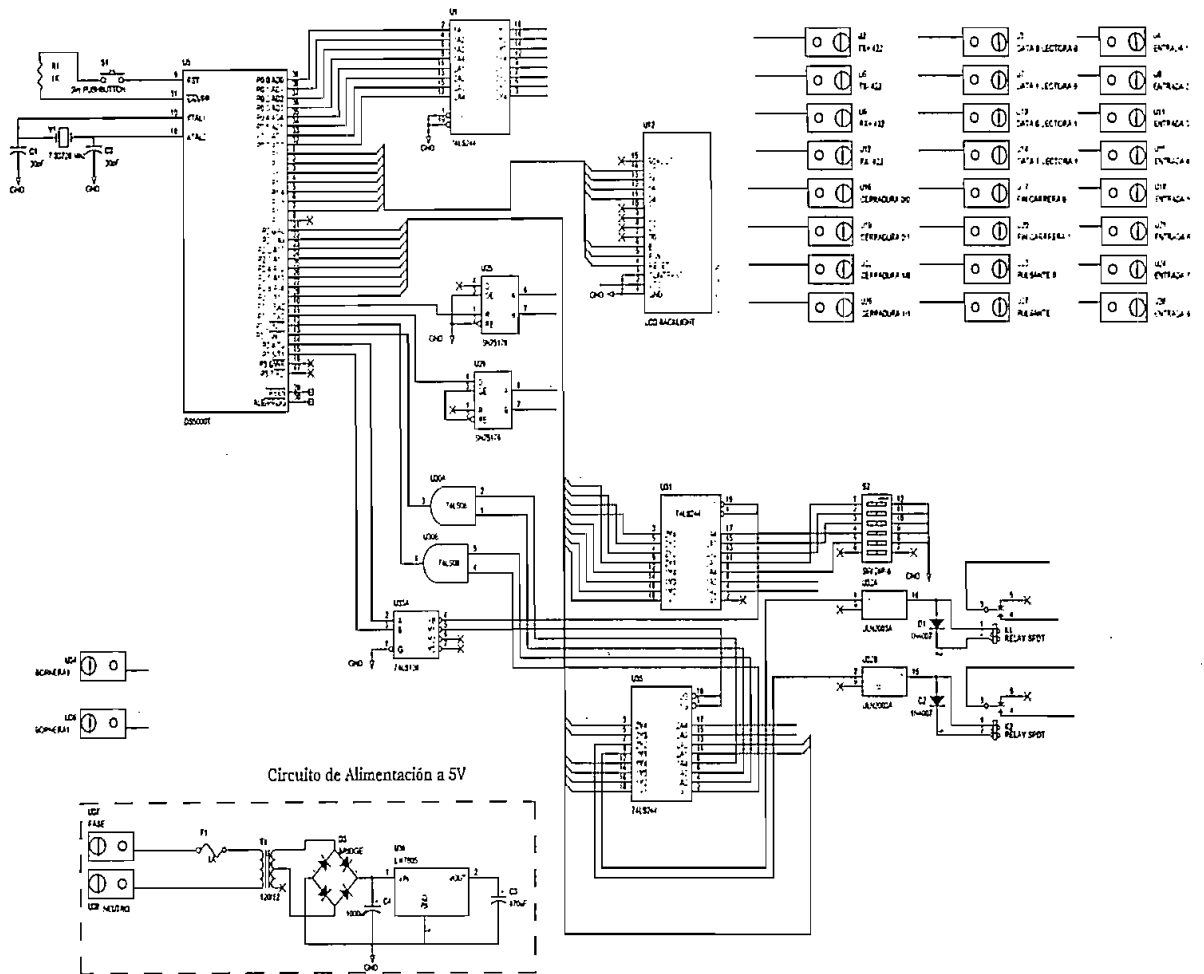


FIGURA II-13. DIAGRAMA ESQUEMÁTICO COMPLETO DEL EQUIPO.

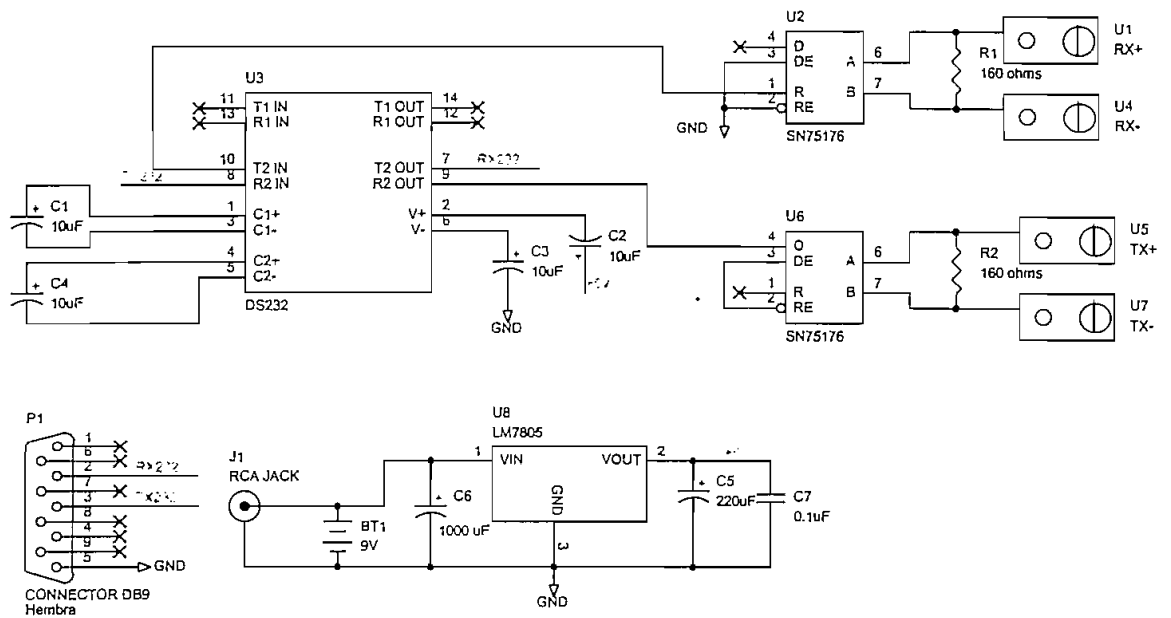


FIGURA II-14. DIAGRAMA ESQUEMÁTICO CIRCUITO INTERFAZ HACIA EL COMPUTADOR.

CAPÍTULO III:

DESARROLLO DEL SOFTWARE.

III.1. REQUERIMIENTOS DE SOFTWARE DEL SISTEMA DE CONTROL DE ACCESOS.

El Sistema de Control de Accesos propuesto requiere dos tipos de software: programa en el microcontrolador y programa en el computador personal.

En el microcontrolador se requiere desarrollar un programa que permita reconocer el código Wiegand enviado por las lectoras magnéticas de proximidad cuando detecta una tarjeta, además almacene y administre en memoria no volátil una base de códigos de usuarios y zonas de tiempo para el acceso, administre la comunicación con el computador utilizando protocolo RS-422, y realice el monitoreo de entradas de alarma.

El programa en el computador personal constituye la interfaz entre el microcontrolador y el operador autorizado para manejar el sistema de control de accesos, a través de este se definen los códigos de usuario autorizados, las zonas de tiempo de acceso, es decir horarios definidos para cada usuario, definir el tiempo de apertura de las cerraduras, monitorear el estado de las entradas de alarma, igualar el reloj calendario del microcontrolador con el del computador. Toda esta interfaz se desarrolla en el ambiente de programación LABVIEW 4.1.

Cada actividad descrita se guarda en una base de datos que permitirá llevar un registro de todas las acciones de control de accesos realizadas en el sistema. Toda la base de datos se desarrolla en Microsoft Access 8.0.

III.2. DESARROLLO DE SOFTWARE EN EL MICROCONTROLADOR DALLAS DS5000T.

III.2.1. DESCRIPCIÓN DEL PROGRAMA EN EL MICROCONTROLADOR.

La programación en el microcontrolador se realizó en lenguaje ASSEMBLER. Debido a que el microcontrolador DS5000T tiene el mismo conjunto de instrucciones que el microcontrolador 8751 se utilizó el Sistema Desarrollado para Microcontroladores 8751 (MICROLAB).

El microcontrolador realiza cuatro funciones principales que son:

- Administración de la comunicación serial entre el equipo y el computador personal.
- Administración de lectoras que consiste en leer el código wiegand enviado por las lectoras de proximidad y decodificarlo, es decir convertirlos en datos numéricos.
- Monitoreo de las entradas de alarma.
- Administración de memoria y manejo del ECC (reloj calendario del microcontrolador).

III.2.2. ADMINISTRACIÓN DE LA COMUNICACIÓN SERIAL.

El microcontrolador se comunica con el computador cada vez que ocurre uno de los siguientes eventos:

- Si una tarjeta es detectada por la lectora de proximidad.
- Si existe cambio en las entradas de alarma.
- Cuando el computador solicita la ejecución de alguna acción como por ejemplo: igualar

el reloj calendario, programar tiempo de apertura de cerradura, programar horarios, definir usuario.

III.2.2.1. **Protocolo**

El tipo de comunicación que se utiliza es comunicación serial asincrónica, (no se dispone de una señal de reloj para la sincronización) por lo que es necesario establecer un formato para la transmisión y para la recepción de datos. El modo de comunicación seleccionado es Full-Duplex, (simultáneamente cualquier equipo puede enviar y recibir datos). El protocolo de comunicación que se utiliza es el RS-422.

Puesto que la transmisión y la recepción pueden darse simultáneamente el software del microcontrolador administra *una pila de memoria (localidades de memoria) tanto de transmisión como de recepción*, esto permite recibir datos en medio de una transmisión o viceversa.

Formato

Para establecer comunicación entre el microcontrolador (equipo) y el computador se define un lenguaje o protocolo de comunicación que tiene el siguiente formato: cabecera (SOH Start of heading), dirección (identificador del equipo), longitud del mensaje, comando (instrucción de tres caracteres), datos (argumentos del comando máximo 12 caracteres) y carácter de control de errores (CRC Checksum). La Figura III-1 muestra el formato del protocolo de comunicación:

SOH	DIR	LEN	C1-C3	DATOS	CRC
-----	-----	-----	-------	-------	-----

FIGURA III-1. GRÁFICO DEL PROTOCOLO UTILIZADO PARA LA COMUNICACIÓN DEL SISTEMA.

Comandos utilizados

El microcontrolador realiza acciones específicas por solicitud del operador del sistema de control de accesos (el PC), acciones que son manejadas por el microcontrolador a través de comandos (instrucciones) que son de dos tipos: de escritura y de lectura. Los comandos de escritura al ser interpretados modifican registros de memoria que serán utilizados por el programa principal, y además transmite un mensaje de comando al operador (PC). Los comandos de lectura transmiten al computador datos almacenados en registros de memoria. La lista comandos/instrucciones es la siguiente:

TABLA III-1. TABLA DE COMANDOS/INSTRUCCIONES DEL PROTOCOLO SERIAL.

COMANDO	DESCRIPCIÓN
IDE	Comando que verifica si existe comunicación, no ejecuta ninguna acción.
HON	Programa horario de inicio de Timezone.
HOF	Programa horario final de Timezone.
FON	Habilitación 24 horas.
FOF	Negado acceso.
HOR	Consulta de horarios programados.
TIM	Igualación de reloj.
DAT	Igualación de fecha.
INP	Transmite al PC el estado de las entradas de alarma.
ECC	Consulta del reloj calendario.
NUM	Cambio de identificador de equipo.
WIE	Transmite al PC los cinco bytes de almacenamiento de códigos de tarjetas wiegand leídos.
MON	Programa el tiempo de apertura de la cerradura.
USR	Ingresa nuevo código de usuario.

	COMANDO	DESCRIPCIÓN
INI		Inicializa el equipo.
UTZ		Programa el número de Timezone para determinado usuario.

Además el microcontrolador envía comandos al computador en respuesta a dos eventos:

- Cuando se detecta el ingreso de una tarjeta. El microcontrolador transmite el comando TRJ con los datos del código de tarjeta y el mensaje de acceso.
- Cuando existe cambio en las entradas de alarma. El microcontrolador envía a través de la red el comando INP con los datos del estado de las entradas de alarma.

III.2.2.2. Administración de Memoria.

En el desarrollo del programa del microcontrolador se ha utilizado memoria interna y memoria externa.

La memoria externa de datos se ha dividido en varios bloques, cada uno de los cuales guarda información necesaria para la ejecución del programa. Así se tiene: buffer de transmisión de datos, buffer de recepción de datos, buffer para almacenamiento de códigos de usuarios, buffer para almacenar zonas de tiempo asignados a cada usuario, buffer para almacenar programación de horarios de cada zona de tiempo, asignación de localidades de memoria para el manejo de display.

En memoria interna se definen registros auxiliares para la administración de la comunicación serial, la validación de datos, y localidades donde se almacenan constantes del sistema.

Buffer de recepción de datos.

Este buffer se utiliza para almacenar cadena de datos conforme el protocolo establecido. Esta pila de memoria se encuentra dividida en cuatro páginas de memoria con 16 localidades cada una, es decir se pueden almacenar un total de cuatro mensajes (cadenas de protocolo) para ser decodificadas por el programa del microcontrolador (ver III.2.2.3 **Administración de recepción**). Para moverse de página a página se utiliza un puntero de recepción y puntero de validación.

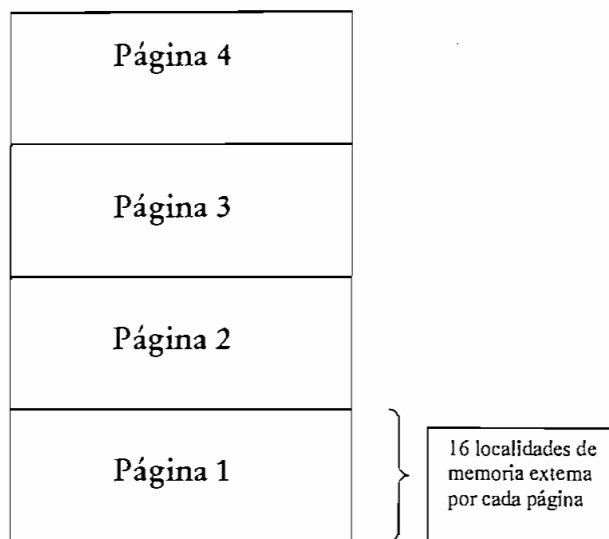


FIGURA III-2. PILA DE RECEPCIÓN EN EL MICROCONTROLADOR.

Buffer de transmisión de datos.

Este buffer almacena cadena de datos a transmitirse y su organización es exactamente igual a la del buffer de recepción como se observa en la Figura III-2.

Buffer de almacenamiento de códigos de usuario.

Este buffer utiliza 256 localidades de memoria para almacenar los códigos de tarjeta

correspondiente a los usuarios. Cada código se almacena en dos localidades de memoria.

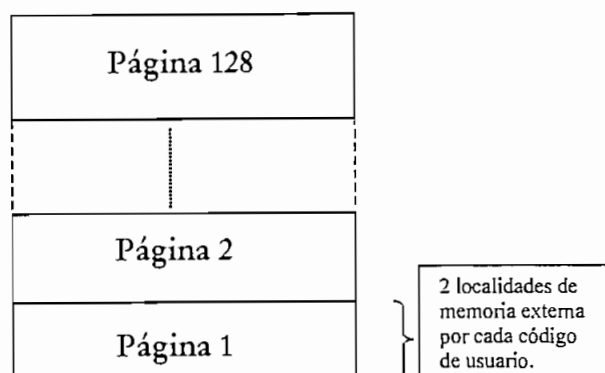


FIGURA III-3. BUFFER DE ALMACENAMIENTO DE CÓDIGOS DE USUARIOS

Buffer Zonas de Tiempo para los usuarios.

Este buffer se utiliza para almacenar a cuales zonas de tiempo (de 16 disponibles) tiene acceso el usuario (2 localidades de memoria por cada usuario) y por cual(es) puerta(s) puede ingresar el usuario (1 localidad por usuario). Para almacenar esta información se utiliza 3 localidades de memoria por usuario (en total son 384 localidades de memoria utilizadas para los 128 usuarios).

Este buffer es “paralelo” al buffer de códigos de usuarios, por ejemplo: el usuario de la página 1 del buffer de códigos de usuario tiene almacenado las zonas de tiempo autorizadas en la página 1 del buffer de zonas de tiempo.

Buffer Programación Horarios de Zonas de Tiempo.

Este buffer almacena los horarios “mínimo” y “máximo” de acceso de cada una de las 16 zonas de tiempo para cada día.

Los datos dentro de este buffer de memoria están organizados de la siguiente manera: se separan 64 bytes para los programas de un día (16 zonas de tiempo x 4bytes) y se almacena desde la localidad más baja: HorasOn, MinutosOn, HorasOff y MinutosOff (en formato BCD). En total se tiene entonces que este buffer ocupa 448 localidades de memoria para el almacenamiento de los horarios de las zonas de tiempo.

III.2.2.3. Administración de recepción.

La recepción de datos se realiza a través de la interrupción de comunicación serial del microcontrolador. La subrutina de interrupción serial es SERIALINT.

Serialint- Realiza la recepción, control y almacenamiento de las cadenas de datos (protocolo) que el microcontrolador recibe. Cuando existe interrupción de comunicación serial se verifica si la interrupción es de recepción (bandera RI está activa). El control de recepción de datos se realiza a través de banderas que se habilitan cuando la cadena de datos que se recibe tiene error en el encabezado o longitud, en estos casos el microcontrolador espera una nueva cabecera para empezar una nueva recepción. Cada vez que llega un dato se activa esta interrupción, es decir el programa no desactiva a ningún instante la bandera de interrupción serial.

La rutina sigue varias secuencias para la recepción de los datos, la primera secuencia consiste en leer el dato y almacenarlo en un registro de memoria interna, revisar las banderas activas para determinar las acciones a tomarse.

La siguiente secuencia es estándar para almacenar las cadenas que se reciben: primero se lee un registro de memoria interna que ubica el puntero (dptr) del buffer de recepción en la localidad de memoria externa donde se copia el dato recibido. Luego se calcula el checksum (a partir del segundo dato que se recibe), se incrementa el puntero (dptr) y se almacena en

III.2.2.5. Transmisión.

La transmisión es un proceso que responde a tres eventos:

- Respuesta a una instrucción que el equipo recibe, esta operación se realiza en el programa principal. La transmisión por lo tanto está vinculada directamente con la interpretación y decodificación de la instrucción.
- Respuesta a cambio en entradas de alarma.
- Respuesta a ingreso de una tarjeta.

La transmisión se concibe de igual forma que la recepción es decir los datos que se van a transmitir son almacenados en el buffer de transmisión y se define puntero de transmisión que apunta a la dirección de memoria en la que se empezará a guardar los datos de la cadena que se va a transmitir. Al existir la posibilidad de que existan recepciones pendientes para validar también existen transmisiones pendientes.

Las subrutinas que administran la transmisión son: TRANS_PEND y TRANSMIT_BUFRX.

Trans_Pend.- Cumple la misma función que Val_Pend, la diferencia radica en la definición del puntero que en este caso es de transmisión.

Transmit_bufrx.- Utiliza el puntero definido por Trans_Pend, y realiza la transmisión del buffer definido por el puntero.

III.2.3. ADMINISTRACIÓN DE LECTORAS DE PROXIMIDAD.

Como se señala en el capítulo II, el equipo administra dos lectoras de proximidad. Los datos que transmite cada lectora son secuencia de unos y ceros a través de dos

líneas de datos (DATA0 y DATA1, como especifica el formato de salida Wiegand). La lectura de datos se realiza por interrupciones externas del microcontrolador (INT0 e INT1, cada una asociada con una lectora), es decir se lee los datos cuando hay un cambio de nivel lógico (de uno a cero) en las líneas de datos (la lectora está enviando datos).

III.2.3.1. Interrupciones.

Cada interrupción externa del microcontrolador está asociada a una lectora (en total dos).

La subrutina de atención a la interrupción externa INT0 es EXTINT0.

Extint0: Esta subrutina lee en cada interrupción generada un bit enviado por la lectora de proximidad, en total se leen 26 bits por cada tarjeta leída. Al ingresar a la subrutina se comienza verificando si existe una bandera que es habilitada cuando se ingresa por primera vez a la interrupción es decir cuando la lectora envía el primer bit de una lectura de tarjeta, si es la primera vez se define un contador con el número total de bits esperado (26 en total). Se habilita el pórtilo multiplexado 1 a través de las líneas de control T0 y T1 se leen los datos por los pines del pórtilo2 y los datos son almacenados en los registros de memoria interna. Cuando la lectora termina la transmisión de bits se habilita una bandera que indica que una tarjeta ha sido leída y se encera el contador de bits.

La subrutina de atención a la interrupción externa 1 es igual con la diferencia de que la lectura se realiza de los pines correspondientes a la lectora conectada a esta interrupción (ver Pórtilo Multiplexado en el Capítulo II).

El diagrama de bloques de la subrutina de interrupción externa 0 o 1 es el siguiente:

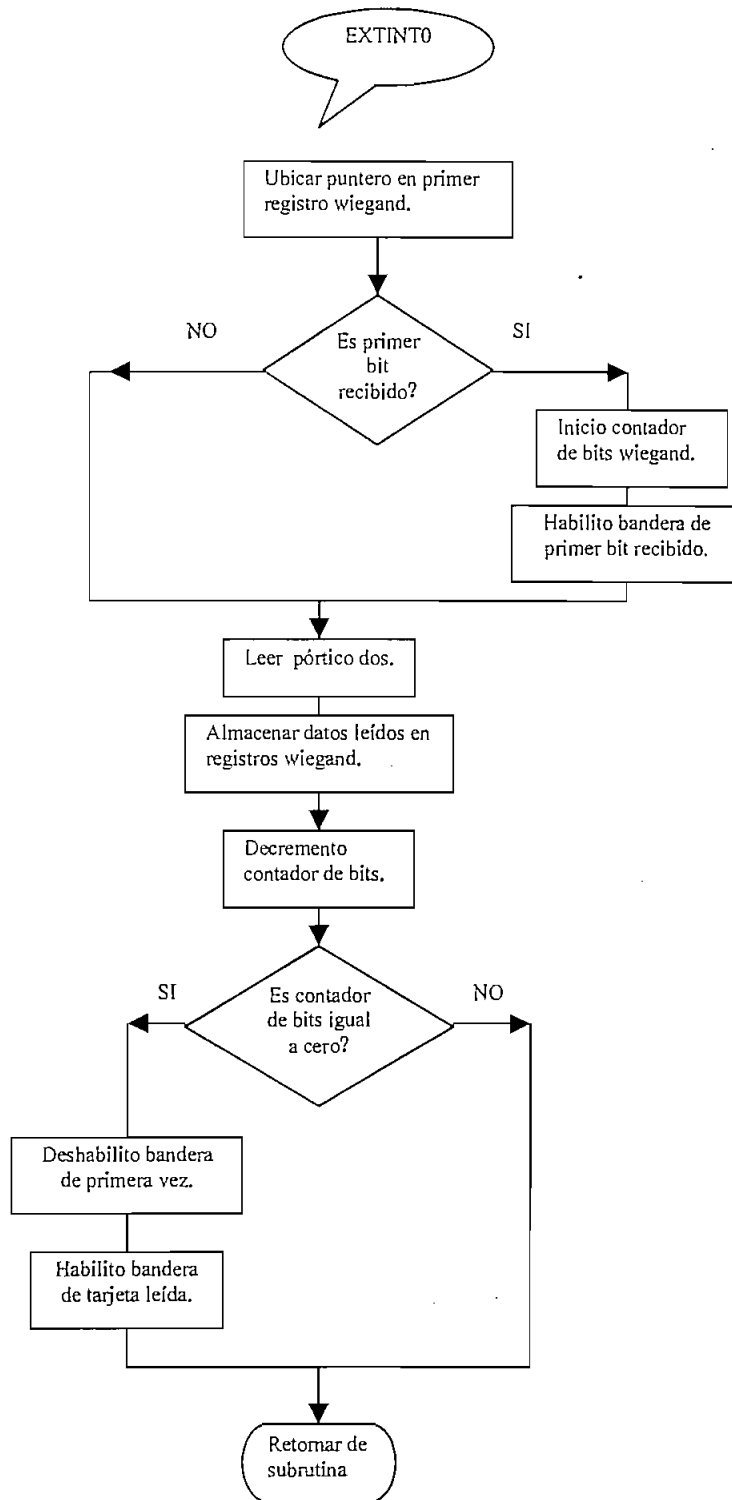


FIGURA III-6. DIAGRAMA DE BLOQUES DE EXTINT0 O EXTINT1.

III.2.3.2. Reconocimiento y validación.

En la ejecución del programa principal para ingresar al reconocimiento y validación, se

revisa el estado de dos banderas que son habilitadas cuando una tarjeta ha sido leída. Si una de las banderas está habilitada se ejecuta la siguiente subrutina: DECODTARJ.

Decodtarj.- Realiza dos funciones: verificación de usuario (tarjeta) definido, apertura de la cerradura de la puerta correspondiente.

La verificación de usuario definido comprende: primero comparación de número de tarjeta ingresada con los que se encuentran almacenados en la tabla de usuarios autorizados, definida previamente a través del comando USR; segundo si es usuario autorizado se procede a verificar el horario permitido de acceso (la programación de horarios se realiza a través del computador) de la siguiente forma: definiendo zonas de tiempo, se ubica el programa de la zona de tiempo definida para el usuario y se compara el horario definido con el actual (con el reloj-calendario del microcontrolador) si en la primera zona de tiempo el acceso no está permitido se sigue buscando en las siguientes zonas de tiempo definidas para el usuario.

La verificación de horarios se realiza calculando el número total de minutos actuales y se compara con los minutos totales definidos para la zona de tiempo, tanto para Horamáxima como para Horamínimo. Si la hora de ingreso está dentro del intervalo de Hora mínima a Hora máxima, el ingreso es aceptado y se visualiza en el Display el mensaje de aceptada.

Tercero, para cada mensaje de tarjeta no reconocida, acceso negado o acceso aceptado (ver *Definiciones*, Capítulo I) el microcontrolador escribe en el buffer de transmisión una cadena de datos a través de la subrutina TXTJR.

Txtjr.- Esta subrutina guarda en el buffer de transmisión mensaje de la tarjeta, la puerta y si fue aceptada, negada o no encontrada

Cuarta, cuando el ingreso de la tarjeta es aceptado se realiza la apertura de la cerradura. Se

inicializa registro de control de cerradura, el control de esta acción se realiza en el programa principal a través de la subrutina CERRADURA.

Cerradura.- Subrutina que controla el tiempo de monoestable (tiempo especificado para mantener un estado lógico) de apertura de las cerraduras.

El control se realiza utilizando banderas en el registro de control de cerradura (inicializado en la subrutina DECODTRJ). Los ocho bits de este registro son banderas que se activan o desactivan bajo determinadas condiciones. Las banderas definidas en este registro son: bit de encendido de cerradura1, bit de inicio de la cuenta de tiempo de cerradura1, bit de activado de cerradura2, bit de inicio de tiempo de apertura de cerradura2.

Se tiene por cada puerta un sensor que indica el estado de la puerta (abierta o cerrada completamente). Si la puerta está abierta se desactiva cerradura1, caso contrario se sigue cuenta de monoestable hasta que el tiempo transcurrido sea igual al definido para apertura de cerradura. El procedimiento descrito es similar para cerradura2.

III.2.4. ADMINISTRACIÓN DEL ECC, LCD Y MONITOREO DE LAS ENTRADAS ALARMA.

III.2.4.1. Administración del Reloj-Calendario.

La administración del reloj calendario se realiza a través de la subrutina ACTUALIZAecc, la misma que es llamada en el programa principal.

Actualizaecc.- Rutina para leer el ECC del microcontrolador DS5000T y cargar toda la información en memoria externa designada para almacenar hora y fecha actuales.

El ECC contiene información de tiempo, que esta almacenada en ocho registros de ocho

bits cada uno, organizada de la siguiente manera:

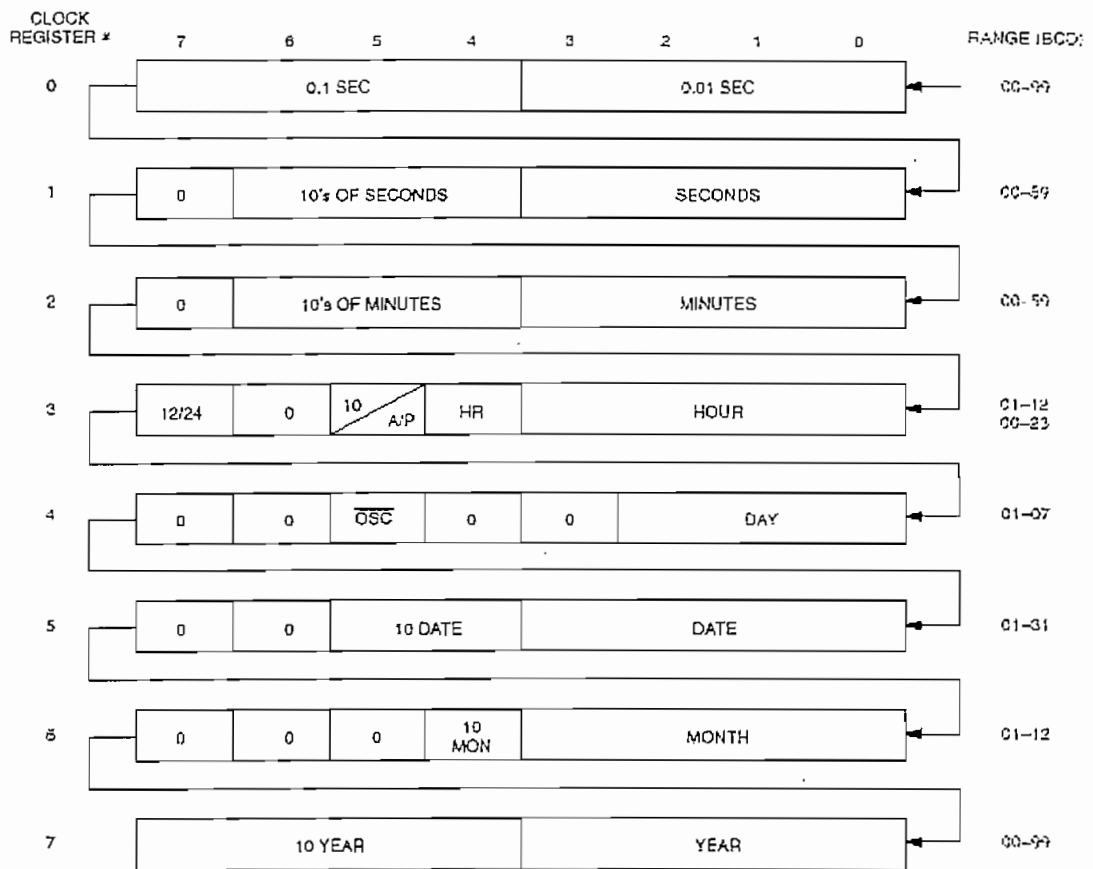


FIGURA III-7. DISTRIBUCIÓN DE LOS REGISTROS DEL ECC.

Los datos de los registros están normalmente en formato BCD excepto en el registro correspondiente a horas cuyo formato depende del modo de hora seleccionado AM/PM (12horas) o formato de 24 horas, el formato se selecciona a través de un bit del mismo registro.

La lectura del ECC se realiza a través de cuatro subrutinas: CLOSE, OPEN, WBYTE y RWRITE. Estas subrutinas vienen como parte de la documentación del microcontrolador DS5000T

Close.- Cierra los registros de tiempo a través 9 lecturas sucesivas de los registros de tiempo.

Open.- Ejecuta una secuencia de lecturas y escrituras que son requeridas para abrir la comunicación con el reloj. Al retornar de la subrutina se puede tener acceso a los datos de los registros de tiempo.

Rbyte.- Esta subrutina conmuta al banco de memoria del ECC activando el bit ECE2 del MCON. Entonces realiza la lectura de 8 bits, uno a uno y los acomoda dejando un byte completo en el Acumulador. Luego vuelve a conmutar al banco direccionado por el CE1, sin modificar ningún otro registro.

Wbyte.- Esta subrutina conmuta al banco del ECC activando el CE2 y entonces, escribe el byte contenido en el acumulador, bit a bit, en el ECC. Luego vuelve a conmutar al CE1 y regresa sin alterar ningún registro.

A través de la lectura del ECC (dentro de la rutina Actualizaecc) se definen dos banderas, que son utilizadas para manejo de cerraduras y para actualizar mensajes del display. Estas banderas se habilitan cuando ha transcurrido una décima de segundo o cuando ha transcurrido un segundo, esto se realiza comprobando el estado anterior de la localidad que almacena las décimas de segundo y unidades de segundos con los datos actuales leídos de los registros de tiempo del reloj. Las banderas son desactivadas en el programa principal.

La actualización de mensajes del display se refiere al tiempo de actualización de datos para visualizar los diferentes mensajes.

III.2.4.2. Manejo del display.

Para el manejo el display se utiliza subrutinas desarrolladas para la tarjeta MCP51 del Laboratorio de Control con Microcontroladores. Las subrutinas utilizadas son: RESETDISP, CLEARLCD, MOVCURSOR, PROMLCD, PRINTDIG,

BCDCONVERT.

Resetdisp.- Esta subrutina inicializa el display, es necesario correrla por lo menos una vez al inicio del programa para determinar el modo de trabajo del display, de acuerdo a las opciones que este presenta.

Clearlcd.- Borra el contenido de la RAM del display y ubica el cursor en la localidad 00H.

Movcursor.- Ubica el cursor en la localidad deseada (posición 0 a 31) para display haciendo transparente para el usuario el cambio de dirección en la mitad del display. Para utilizarla basta poner en la localidad de RAM interna CURSOR la posición deseada y hacer un llamado a la subrutina.

Promlcd.- Esta subrutina hace una escritura en pantalla de un grupo de 16 caracteres leídos en la ROM o parte del programa, creado como tabla, es ideal para escribir mensajes. Para utilizarla se asigna el número de fila que queremos mostrar en pantalla a la localidad de RAM interna ROMLCD, luego se hace el llamado a la subrutina.

Printdig.- Esta subrutina permite escribir directamente un número del 0 al 9 en el display. Se almacena el número en la localidad DIGITO y luego se llama a la subrutina.

Bcdconvert.- Transforma un número de tres bytes binarios a su equivalente BCD. Para utilizarla, se carga el número a transformar en las localidades AUX2, AUX1, AUX0 y el resultado lo almacena en BCD3, BCD2, BCD1, BCD0, considerando 0 como el byte menos significativo.

El display es utilizado como dispositivo de visualización y ayuda, pues indica el estado del sistema de control de accesos es decir la condición en la que se halla el equipo bajo la ejecución de alguna acción. Algunas funciones que el microcontrolador realiza están

asociadas un mensaje de respuesta a la acción que se despliega en el display, cuando el microcontrolador no realiza acción específica se visualiza la hora.

Para visualizar los diferentes mensajes en del display, en el programa principal se ejecuta la subrutina ACTUALIZALCD.

Actualizalcd- Rutina de actualización de mensajes de la segunda fila del display LCD en función del puntero PUNTRD que indica la página de memoria de la RAM del display que se requiere visualizar.

III.2.4.3. Monitoreo de las entradas de alarma.

En el programa principal se monitorea el estado del pórtilo cero (al que están conectadas las ocho entradas de alarma, ver **Interfaz de Entradas de Alarma** en el Capítulo II) y este valor se compara con el almacenado en un registro de memoria interna, si son diferentes se ejecuta la subrutina ACTUALIZAINPTS que almacena el nuevo valor del pórtilo en ese registro y se transmite el nuevo valor al computador.

III.3. DESARROLLO DE SOFTWARE EN EL COMPUTADOR PERSONAL.

III.3.1. DIAGRAMA DE BLOQUES DEL PROGRAMA DEL PC.

El programa realizado en el computador se divide en dos grandes grupos: desarrollo de la interfaz de LABVIEW y el desarrollo de la concepción de la base de datos en Microsoft Access.

La interfaz de LABVIEW comprende los siguientes aspectos:

- Desarrollo de una interfaz amigable hacia el operador del equipo. Comprende las diferentes pantallas para ingreso y visualización de datos de forma accesible y comprensible para el operador.
- Programación del protocolo de comunicación entre la red de equipos y el computador personal.
- Desarrollo de rutinas de comunicación entre LABVIEW y Access para el ingreso y consulta de datos desde y hacia la base de datos. Todo este conjunto de rutinas basadas en las facilidades que ofrece la comunicación de automatización OLE entre aplicaciones y la ayuda del objeto DAO de Microsoft Access.

La base de datos en Microsoft Access comprende la organización coherente de los datos a través de tablas interconectadas entre sí conforme las reglas de las bases de datos tipo relacional como lo es Access. En general Microsoft Access se presenta como un programa de entrada y salida de datos, pero no realiza ninguna actividad directa entre el operador y la base de datos. LABVIEW es quien trabaja los datos y mantiene relación directa máquina-operador.

Lo explicado se resume en el siguiente diagrama de bloques

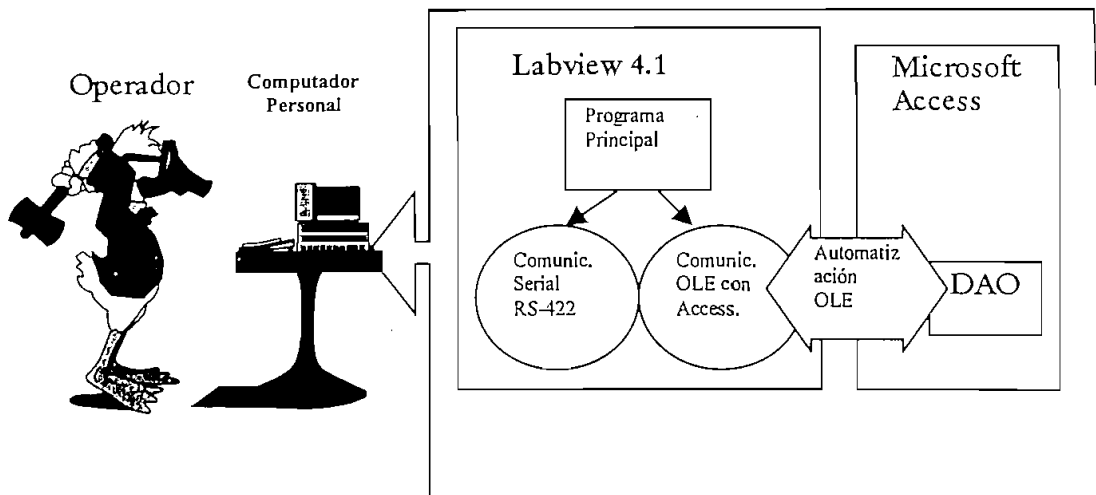


FIGURA III-8. DIAGRAMA DE BLOQUES DEL SOFTWARE DEL PC.

III.3.2. INTERFAZ DE LABVIEW.

Como se puede observar en la figura anterior, Labview desarrolla tres actividades íntimamente ligadas: el despliegue de la interfaz para el operador, la comunicación serial con los equipos y la comunicación con la base de datos. Todas estas actividades son independientes, pero tienen una administración “centralizada” en un programa principal encargado de “monitorear” las actividades realizadas por el operador o los equipos.

A continuación se describen los aspectos más importantes de la programación de las tres actividades mencionadas.

III.3.2.1. Programa Principal.

El programa principal desarrolla dentro de la programación las siguientes actividades: control de la visibilidad de la barra de herramientas, arranque y cierre de comunicación serial, control de mensajes, control de clave de operador al salir del programa o entrar a la configuración de los equipos, control de números de referencia de automatización con la base de datos. Además permite el acceso a las pantallas de programación y generación de

reportes.

El programa principal maneja dos variables globales: Salir y Database Refnum. La primera variable permite detener la ejecución de los programas de control de comunicación serial y OLE. La segunda es una variable de sincronización del programa principal con la base de datos.

Control de Visibilidad de la barra de herramientas.

La barra de herramientas es el interfaz de acceso a las diferentes pantallas de configuración del sistema. Para el diseño de la barra se ubicaron tres conjuntos de botones superpuestos en la pantalla principal, cada bloque corresponde a uno de los niveles de la barra de herramientas (Ver Anexo B, Figura B-2).

El control de la barra de herramientas se realiza en el programa principal a través de un lazo que cumple dos tareas:

- Control de selección de botones presionados.
- Control de visibilidad de los botones.

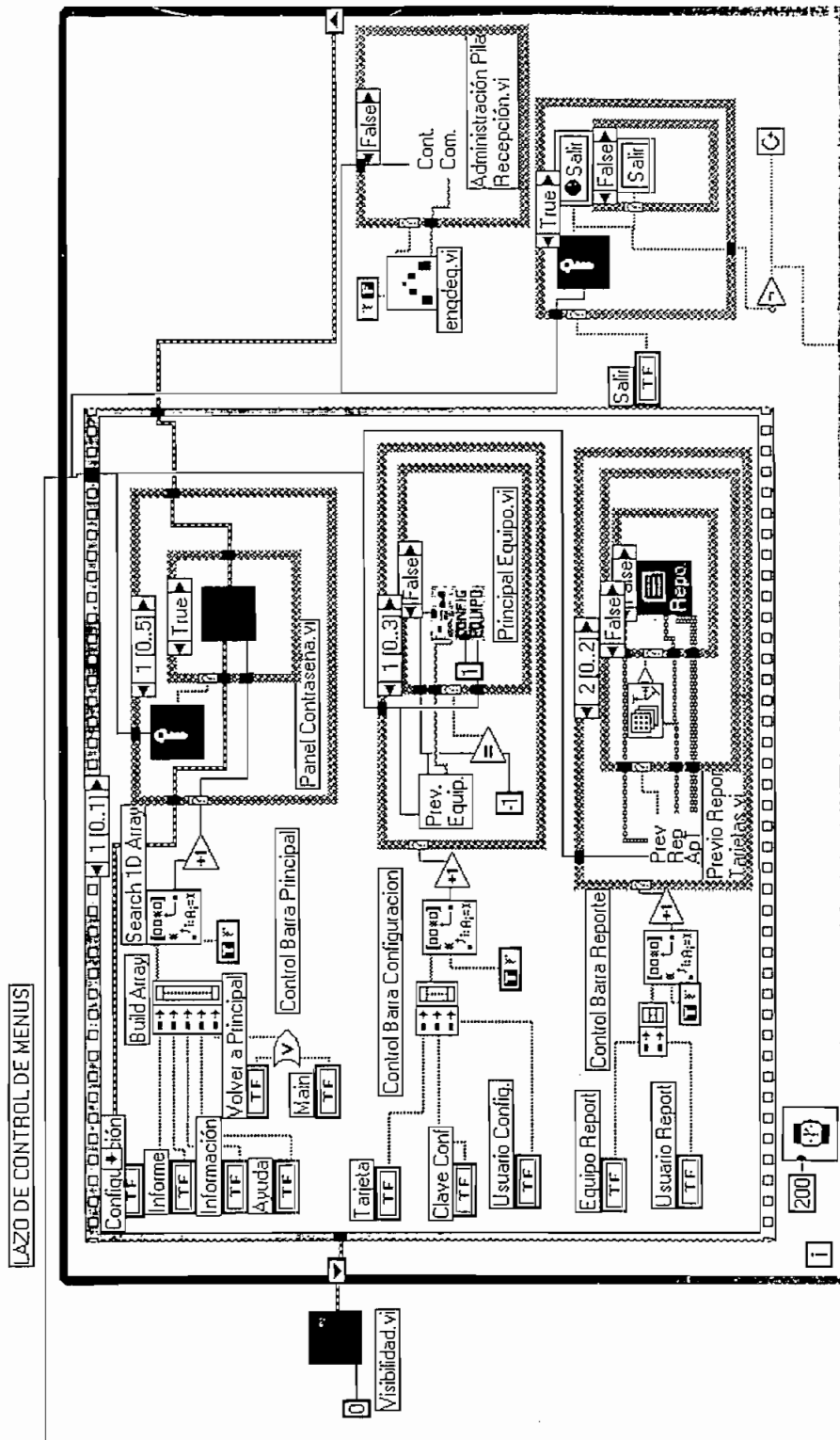


FIGURA III-9. LAZO DE CHEQUEO DE BOTONES.

El programa principal chequea continuamente si alguno de los botones es presionado (Figura III-9). Si el botón presionado permite el acceso a otro nivel de la barra de

herramientas llama a la subrutina Visibilidad.vi para escribir los atributos de visibilidad en los controles de la barra. El programa inicia con los valores de la barra de mayor nivel (barra principal).

La base del control de visibilidad es el SubVI Visibilidad.vi que consta en esencia de tres arreglos (clusters) con los valores booleanos de visibilidad (True = Visible, False = Invisible) de los controles de la barra de herramientas, como se muestra en la Figura III-10.

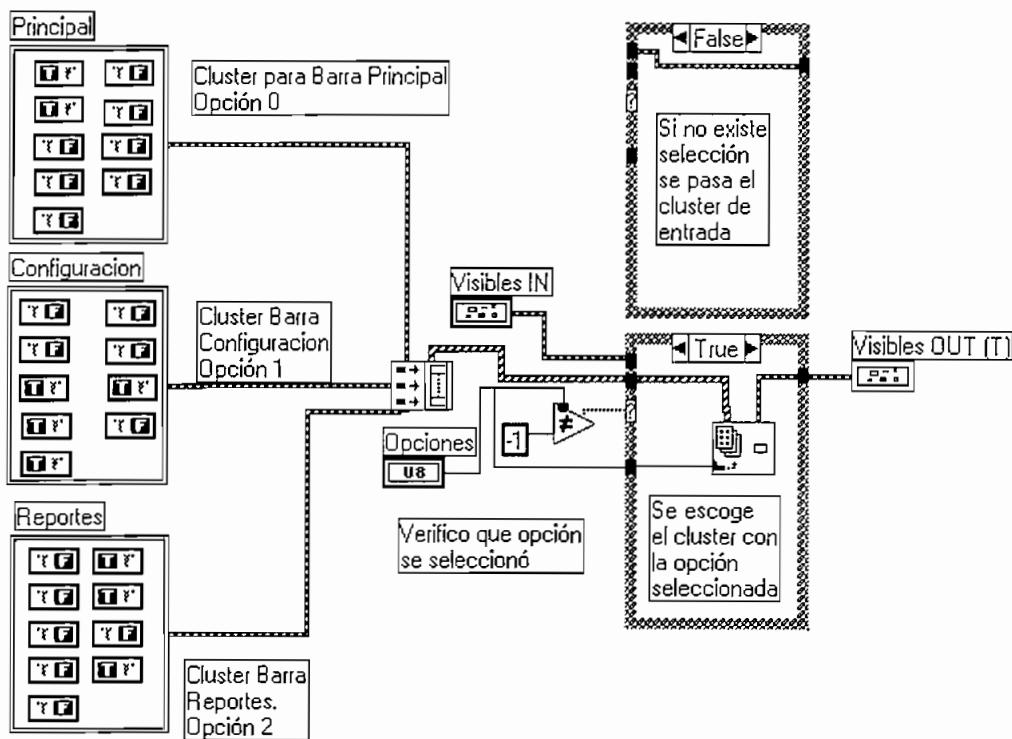


FIGURA III-10. SUBVI VISIBILIDAD.

Arranque de la comunicación serial.

El programa principal tiene una secuencia de inicialización que abre el puerto de comunicación a través de la rutina "Recepción Serial PROT422" (más detalles III.3.3 Desarrollo de programa de Comunicación Serial.).

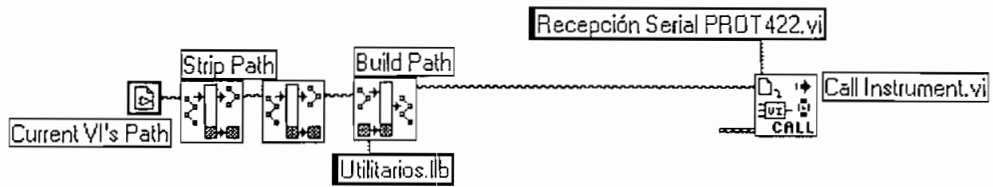


FIGURA III-11. APERTURA COMUNICACIÓN SERIAL.

La rutina Recepción Serial PROT 422 detiene su ejecución cuando el valor de la variable global salir es verdadera.

Control de Mensajes.

Cada modificación del operador a la configuración del sistema y la actividad de los equipos (lectura de tarjeta o cambio de entrada de alarma) se asocia con un mensaje que aparece en la pantalla. Para manejar los mensajes se utilizó como medio visual una tabla de tres columnas en la cual la primera representa la fecha de despliegue del mensaje, la segunda la hora y la tercera el mensaje propiamente dicho. Además cada operación se asocia con un color.

Debido a que cada operación de modificación se realiza en una pantalla diferente a la principal, se utilizó un “artificio” para poder recibir los mensajes en la pantalla principal. Este artificio consiste en hacer uso de los Vis dentro “Queue.llb”: Enqueue.vi, Dequeue.vi y Enqdeq.vi que vienen como ejemplo de Labview, en el cual se crea una pila que va almacenando datos en una matriz y que pueden ser recuperados posteriormente. Para realizar esta operación se creó como datos un cluster que contiene el mensaje y el color a desplegarse, los programas de modificación guardan los mensajes en este formato con la rutina “Enqueue Pantalla.vi” y el programa principal se encarga de recibir los mensajes y manejar los atributos del color del mensaje.

El programa principal a través de un lazo revisa continuamente si existen mensajes

pendientes para desplegar en pantalla. Cuando se detecta un mensaje se realiza la decodificación y despliega los mensajes. La pantalla de mensajes mantiene 48 mensajes activos, cada vez que este valor es superado se pierden los 16 mensajes más antiguos.

A continuación se presenta la programación que controla la decodificación del mensaje y de asignación de color.

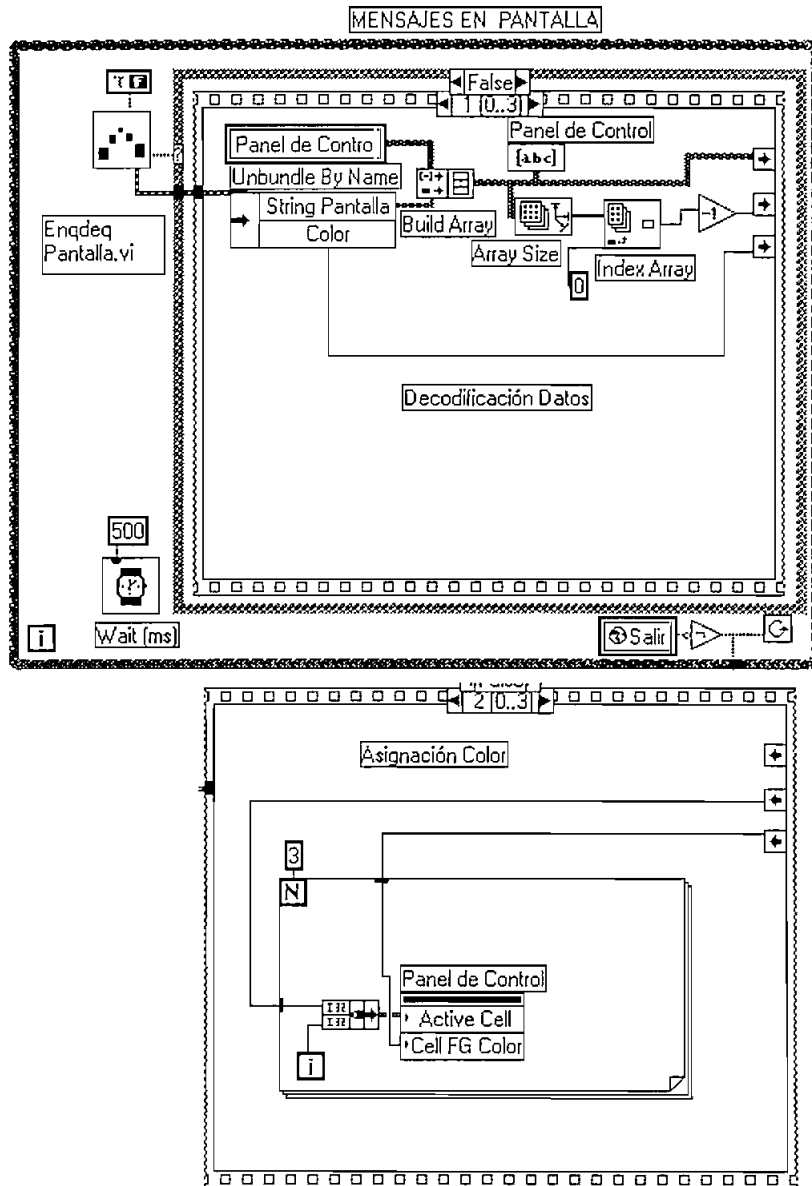


FIGURA III-12. PROGRAMACIÓN DE RECUPERACIÓN DE MENSAJES Y DESPLIEGUE EN PANTALLA.

Inicialización de la comunicación con la Base de Datos.

El programa principal al iniciar llama al vi “Manejo BD Control Accesos”. Este VI al iniciar abre la comunicación con el objeto DAO de Microsoft Access y da el número de referencia de la base de datos para el control de accesos (ver posteriormente en el presente capítulo Comunicación con el objeto DAO.DBEngine.). Este número de referencia es escrito en una variable global.

El programa principal detecta si el número de referencia a la base de datos es diferente de cero para acceder al lazo de control de la barra de herramientas como se muestra en la figura siguiente. Se debe realizar esta sincronización pues este lazo llama a otras pantallas que hacen uso de la base de datos y se generaría un comportamiento erróneo si no existe esta sincronización con la apertura de la comunicación con la base de datos.

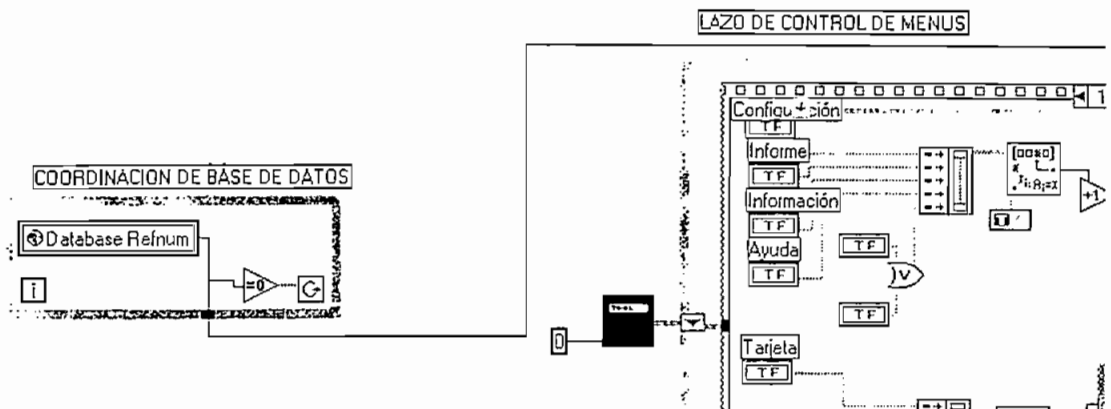


FIGURA III-13. SINCRONIZACIÓN DE LA BASE DE DATOS CON LA BARRA DE HERRAMIENTAS.

Control de Clave de Operador.

El control de seguridad del sistema se realiza llamando a la rutina “Contraseña”. Esta rutina se ejecuta cuando en el programa principal ocurre: el operador desea salir del sistema o el

operador desea ingresar a la barra de configuración.

La base de la programación de esta rutina consiste en leer los datos de la tabla "Contraseñas" de la base de datos. La lectura de una tabla se realiza abriendo los registros de la tabla y posteriormente abriendo los números de referencia de los campos que contiene los registros y por último se lee el dato en el formato almacenado (ver III.3.4 Desarrollo de rutinas de automatización Ole para comunicación entre Labview y Microsoft Access.) como se muestra en la siguiente figura:

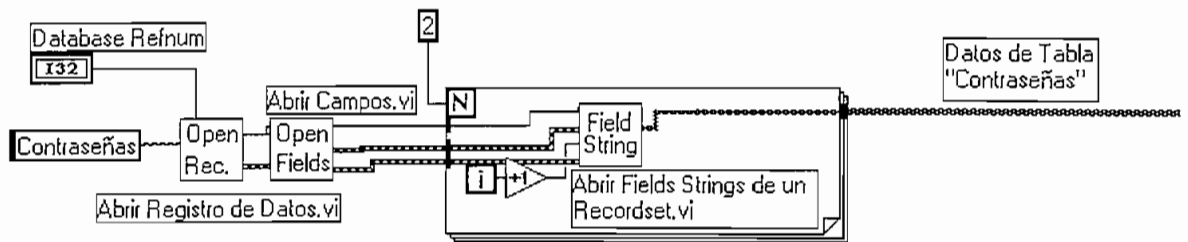


FIGURA III-14. LECTURA DATOS CONTRASEÑA

Los datos leídos contienen los nombres de operadores y contraseñas autorizadas (estos valores se encuentran "encriptados"), estos datos son comparados con los ingresados por el operador para verificar su autenticidad. Además se genera un mensaje (almacenado en la pila de los mensajes) cuando se ejecuta esta pantalla. Se muestra un ejemplo de la generación de mensajes en la siguiente figura:

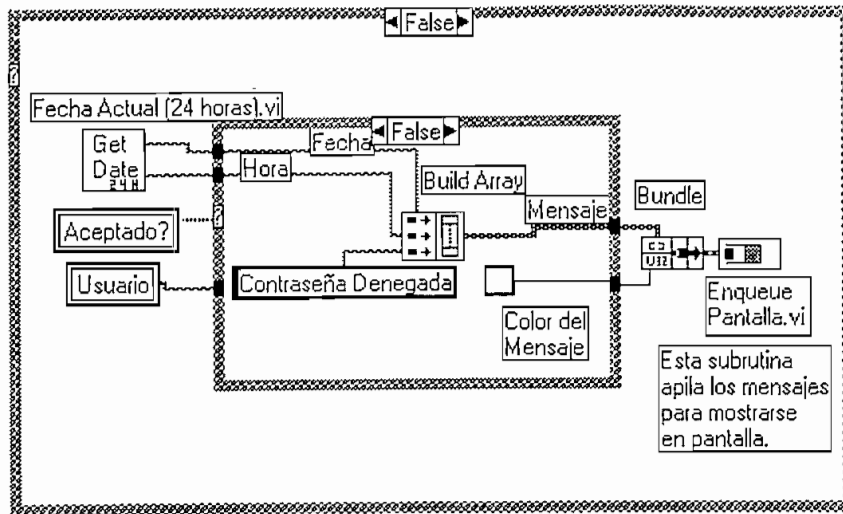


FIGURA III-15. EJEMPLO ALMACENAMIENTO DE MENSAJE PARA PANTALLA EN LA PILA GENERADO EN LA PANTALLA CONTRASEÑA.

En la base de datos se encuentra definido un operador inicial con una contraseña dada (nombre: *default*, contraseña: *def*).

III.3.2.2. Rutinas de Configuración.

La barra de herramientas es el mecanismo mediante el cual se puede acceder a las diferentes características de administración del programa que son: la configuración y la generación de reportes. El presente ítem describe las rutinas utilizadas por las diferentes pantallas de programación.

Rutina Configuración Operador.

La interfaz de esta rutina es la pantalla del vi “Configuración Clave” y se muestra en la Figura B-4 del Anexo B.

La programación de esta pantalla se basa en modificar la tabla “Contraseña” de la base de datos que contiene los datos de los operadores autorizados a través de las rutinas de automatización OLE entre LABVIEW y Access. El programa en primer lugar carga todos

los datos de los operadores autorizados, posteriormente inicializa los atributos de los controles, luego se ejecuta el lazo del programa que permite realizar las modificaciones en la tabla de la base de datos según las opciones que se presentan en la Figura B-4 y por último se ejecuta el cierre de la comunicación con la tabla respectiva.

Cuando se ejecuta el lazo de programación se manejan los registros de la tabla utilizando las **Rutinas de manejo de registros** descritas en el ítem III.3.4.2 del presente capítulo. Junto con este manejo de la base de datos se realiza la modificación de una matriz que contiene los datos de la tabla “Contraseña” utilizada para la visualización de los datos. Además cada cambio conlleva un mensaje a desplegarse en la pantalla principal.

Si se quiere duplicar el nombre de un operador, el programa sobrescribe los datos originales con los nuevos requeridos, la opción añadir en este caso no define un nuevo usuario en la base de datos.

Rutina Configuración Usuario.

De igual manera que en el caso anterior la interfaz de esta rutina es una pantalla asociada al vi “Principal Usuario” y que se muestra en la Figura B-5 del Anexo B.

La programación de esta pantalla se basa en modificar la tabla “Usuario” de la base de datos que contiene los datos de los usuarios autorizados: Número de Tarjeta, Nombre, Apellido Paterno, Apellido Materno y Número Telefónico. El programa realiza las mismas operaciones que en el caso de Rutina Configuración Operador.

El cambio en la programación con respecto a la de configuración de operador está en el uso de rutinas que ayudan a visualizar los datos y estas son: “Visibilidad de Botones Recordset” que genera los atributos de visibilidad de los botones siguiente y anterior registro tomando como base el registro actual que se está visualizando, “Leer

Registro de Usuarios” visualiza el registro al cual el operador se moviliza con uno de los botones de la pantalla. A continuación se muestra como se utiliza estas rutinas cuando se presiona el botón Siguiente Registro.

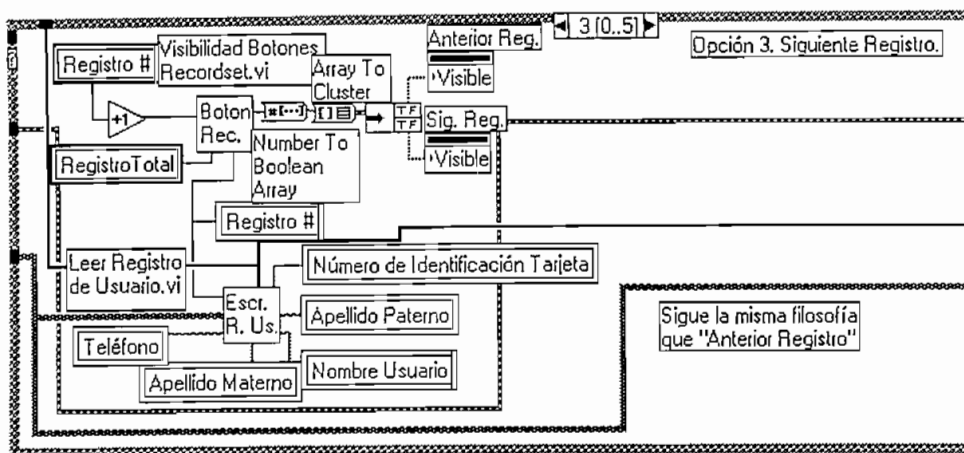


FIGURA III-16. PROGRAMACIÓN DE SELECCIÓN DE REGISTRO EN PANTALLA CONFIGURACIÓN USUARIO.

Rutinas Configuración Equipo.

Al ingresar a la configuración de equipos, el programa llama a dos rutinas: “Panel Previo” y “Panel Principal Configuración”.

Programación Panel Previo

La rutina de esta pantalla tiene dos funciones: manejar la tabla “Equipo” de la base de datos y verificar la comunicación con el equipo seleccionado para acceder a la pantalla principal de configuración equipo.

Cuando el operador presiona el botón Agregar, el programa llama a la rutina “Agregar Equipo”, esta rutina le pide al usuario el identificador del nuevo equipo a agregar en la red y su descripción. Luego utiliza una secuencia que primero verifica la comunicación con dicho equipo utilizando el comando IDE y si la comunicación existe inicializa al equipo

con el comando INI del Protocolo y por último agrega el equipo a la tabla "Equipo" de la base de datos. Además genera un mensaje para la pantalla principal. La figura III-17 muestra el proceso. (Ver ítem III.3.3 Desarrollo de programa de Comunicación Serial para el proceso de comunicación PC con el equipo).

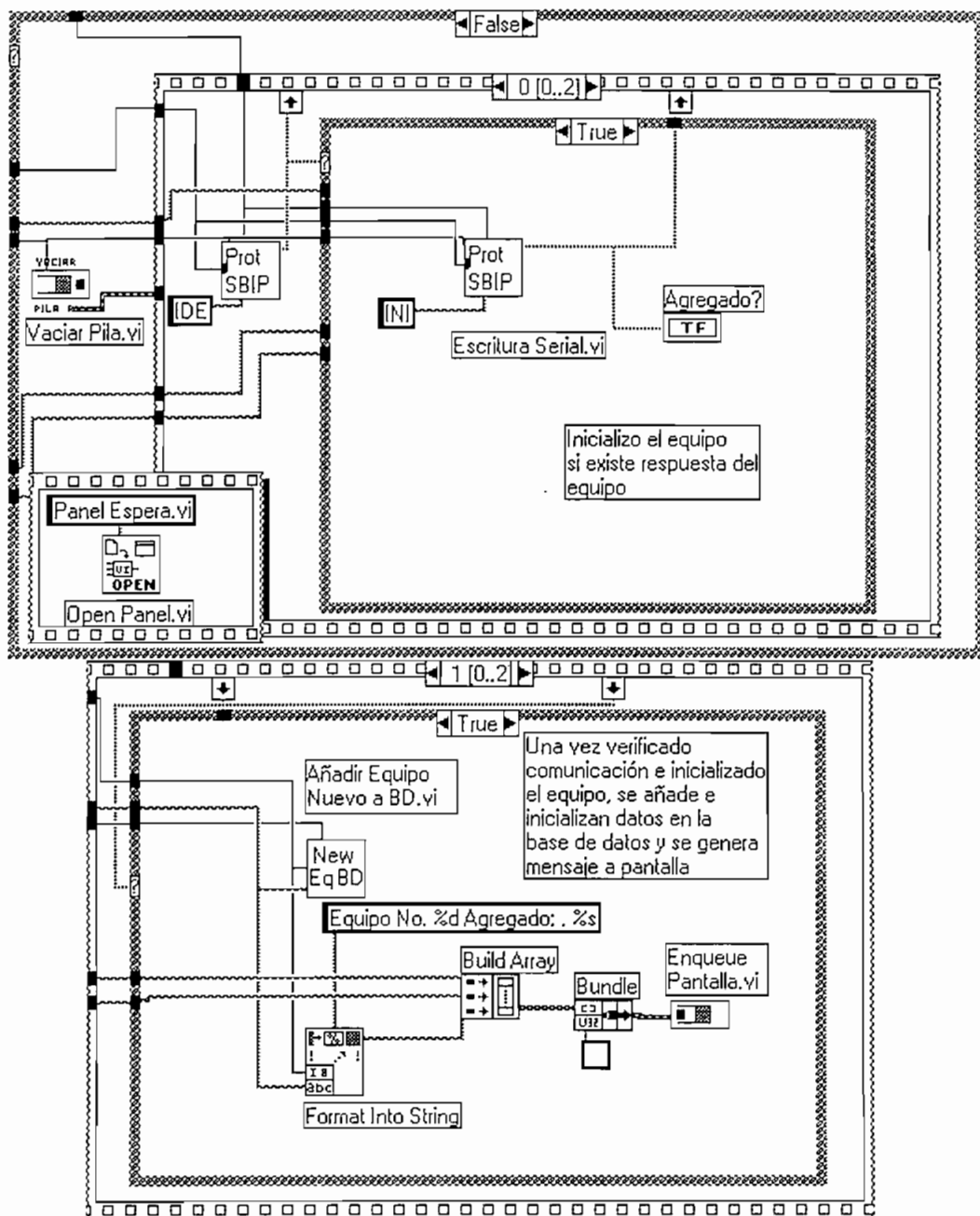


FIGURA III-17. PASOS PARA AGREGAR UN EQUIPO A LA BASE DE DATOS.

La segunda función del Panel Previo es verificar la comunicación con el equipo que

el operador selecciona para configurar, su programación es similar a la primera función de la rutina “Agregar Equipo”.

El panel previo tiene argumentos de salida que utiliza la pantalla “Principal Configuración Equipo” y son: un cluster de error que indica si la comunicación fue exitosa, una cadena de caracteres con el nombre del equipo y una variable numérica con el número del identificador del equipo seleccionado.

Programación Pantalla Principal Configuración Equipo.

La tarea de configuración no es más que una interfaz amigable para la comunicación del operador con el equipo. Cada opción de configuración involucra uno o dos comandos del protocolo de comunicación (ver ítem III.2.2.1 Protocolo) y el manejo de las tablas de la base de datos.

Las opciones de configuración son: lectura de entradas de alarma, consulta e igualación de reloj calendario, definición de zonas de tiempo, programación de horarios de zonas de tiempo, cambio de descripción de puertas, configuración de monoestable de cerradura, definición de usuarios autorizados por equipo. La primera opción forma parte de la pantalla principal de configuración, las demás son pantallas independientes.

La programación de la pantalla principal de configuración consta de un lazo que detecta cual opción se selecciona y accede a la pantalla de configuración correspondiente. La única opción que se controla en esta pantalla es la lectura de las entradas de alarma, en esta opción el equipo se comunica a través del comando INP y el estado de las entradas de alarma se muestra en un entorno visual de LEDS (cluster booleano) como se muestra en la figura III-18.

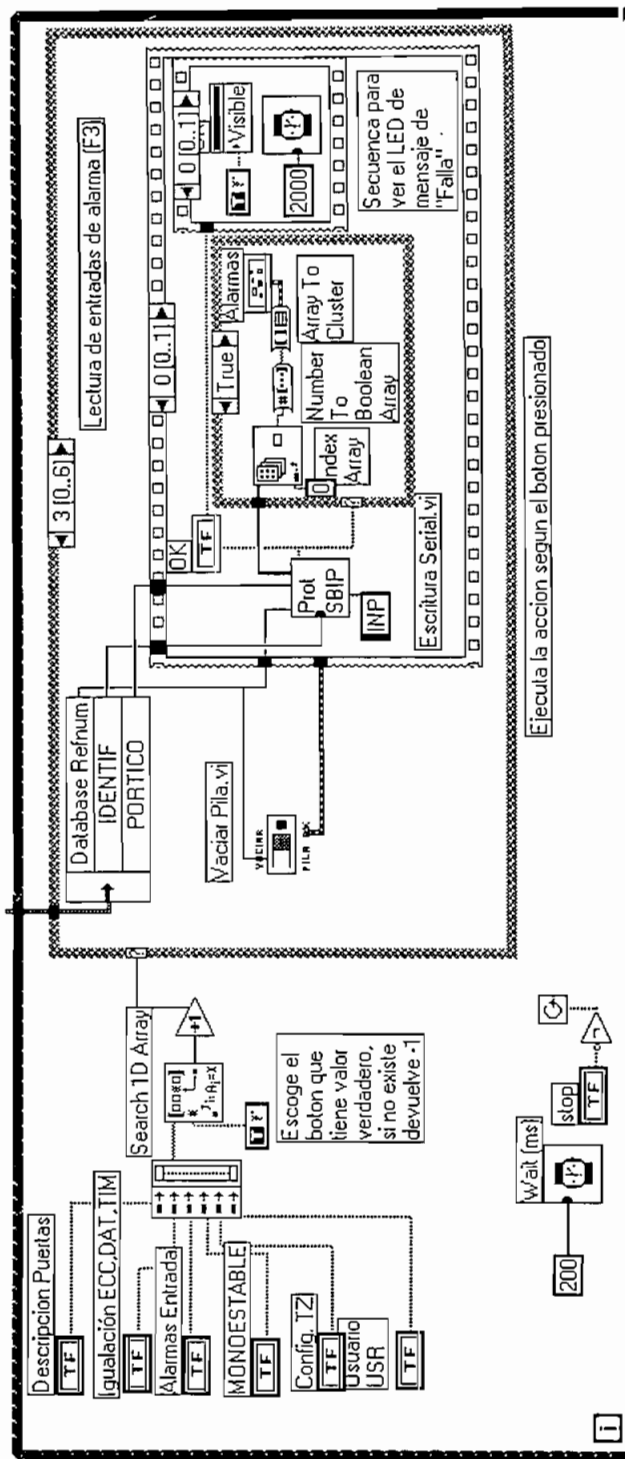


FIGURA III-18. PROGRAMACIÓN PRINCIPAL CONFIGURACIÓN EQUIPO.

Como filosofía dentro de la opción de entradas de alarma y de las pantallas de configuración que en adelante se detallarán, cuando existe una falla de comunicación aparecerá siempre un mensaje de “falla” (color rojo) en la pantalla durante

aproximadamente dos segundos, y si el comando se ejecuta correctamente aparecerá un mensaje “OK” (color verde) del mismo intervalo de tiempo.

Además con cada tarea de comunicación se genera un mensaje que aparece en la Pantalla Principal.

A continuación se detalla las pantallas para cada una de las opciones de configuración del equipo.

Programación Pantalla Cambio Descripción de Puertas.

En su programación lee los datos de la tabla “Puertas” de la base de datos y si se realiza alguna modificación se escribe los datos modificados en dicha tabla (ver III.3.4 Desarrollo de rutinas de automatización Ole para comunicación entre Labview y Microsoft Access.).

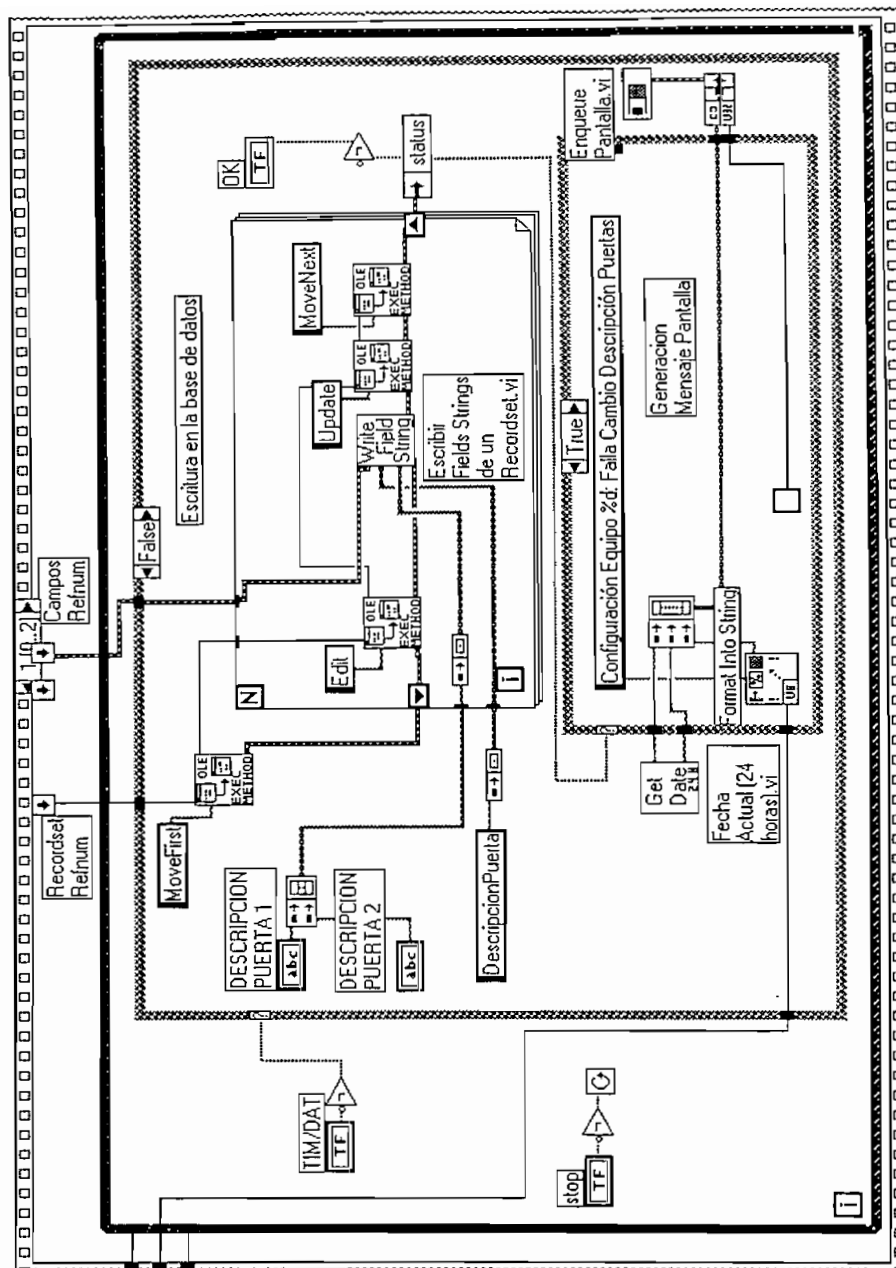


FIGURA III-19. LAZO PRINCIPAL DE PROGRAMACIÓN DESCRIPCIÓN PUERTAS.

Pantalla Consulta e Igualación del Reloj Calendario.

Presenta dos opciones: consultar (comando ECC) e igualar (comandos TIM y DAT) el reloj. (ver III.3.3 Desarrollo de programa de Comunicación Serial.)

Programación Pantalla del Monoestable de la Cerradura.

Esta pantalla define el intervalo de tiempo (en segundos) de apertura de la cerradura, utiliza para la programación el comando MON del Protocolo (ver III.3.3 Desarrollo de programa de Comunicación Serial.).

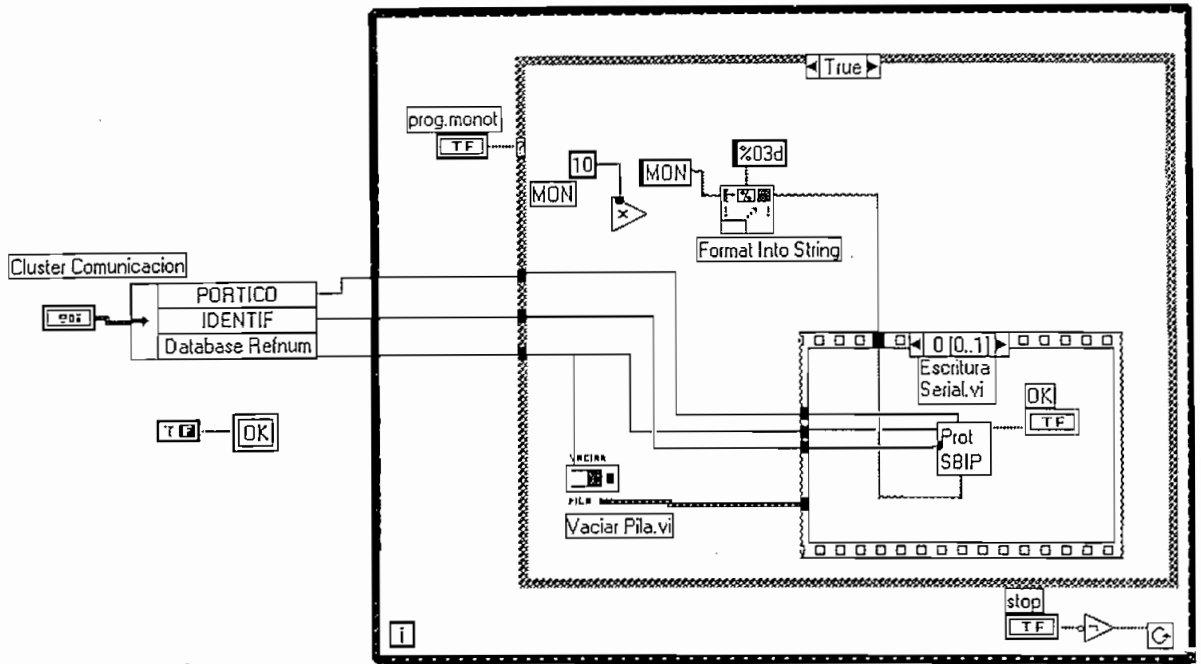


FIGURA III-20. PROGRAMACIÓN CONFIGURACIÓN MONOESTABLE CERRADURA.

Programación Pantalla Configuración de Zonas de Tiempo.

La programación de esta pantalla comprende:

Primero, se obtiene los datos de los horarios y descripción de las zonas de tiempo de las tablas "Horario de Timezones" y "Zonas de tiempo" de la base de datos y se visualiza en la pantalla los datos de los horarios correspondientes al día domingo.

En segundo lugar, se ingresa a un lazo que espera el ingreso de datos de los horarios del día respectivo, durante cada cambio de horario el sistema verifica que los datos ingresados sean coherentes. Si se realizó un cambio en el día y no se ejecutó el comando Guardar (tecla F1) LOS CAMBIOS REALIZADOS SE PERDERÁN. Si el operador cambia de día la

tabla se actualiza con los datos del día correspondiente.

Cuando se ejecuta el comando *Guardar* (F1) el programa se llama al Subvi “Escribir Horarios.vi”. Esta subrutina utiliza para programar los horarios los comandos HON (hora mínima) y HOF (hora máxima) (ver III.2.2.1 Protocolo), y si no existe un fallo de comunicación se graba el horario en la tabla “Horario Timezones” de la base de datos.

En última instancia, el programa verifica si existió algún cambio en la descripción de las Zonas de Tiempo y guarda estos cambios en la tabla “Zonas de Tiempo” de la base de datos y libera la comunicación con estas tablas (ver III.3.4 Desarrollo de rutinas de automatización Ole para comunicación entre Labview y Microsoft Access.).

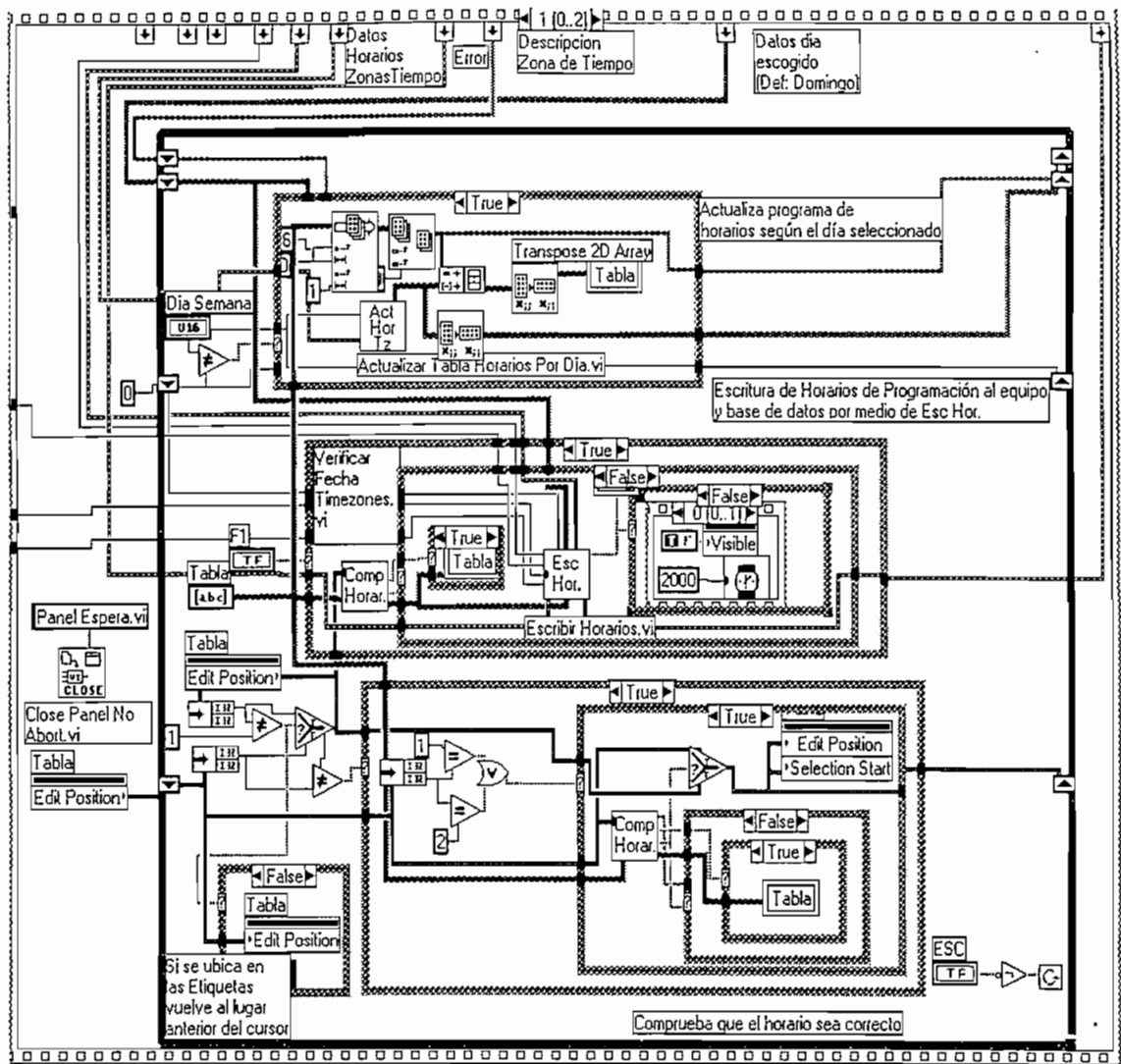


FIGURA III-21. PROGRAMACIÓN LAZO PRINCIPAL PANTALLA CONFIGURACIÓN TIMEZONES.

Programación Pantalla de Definición de Tarjetas Válidas para el Equipo.

Los botones para la configuración que presenta esta pantalla (ver Figura B-12) realizan dos tareas: desplazarse dentro de la tabla de las tarjetas y grabar los cambios efectuados.

La programación de esta pantalla se realiza en los siguientes pasos:

Primero se inicializa los controles de las columnas de la tabla y la lista de selección de las zonas de tiempo realizando consultas a la base de datos a través de las tablas "Registro de

Usuarios por Equipo”, “Zonas de Tiempo”, “Usuarios” y “Puertas” de la base de datos para poner las descripciones de las zonas de tiempo, nombres de los usuarios y puertas.

A continuación se ingresa al lazo del programa el cual realiza varias acciones. Monitorea el control de número de memoria (primera columna), si existe un cambio actualiza los datos de los demás controles de la tabla, ubica en la primera fila de la tabla el registro seleccionado y actualiza los datos en la lista de selección de zonas de tiempo de acuerdo al registro seleccionado. Cuando el operador cambia los datos del control del código de tarjeta o el nombre de usuario, el programa sincroniza el cambio de datos en ambos controles. El lazo además supervisa si los controles de avance o retroceso de página han sido presionados, estos controles se hacen invisibles cuando están en la última o primera fila. Por último cuando la opción *Modificar* (F1) es seleccionada actualiza los datos en el equipo a través de los comandos USR y UTZ (ver III.2.2.1 Protocolo) y si no existen fallas en la comunicación guarda los datos modificados en la tabla “Registro de Usuarios por Equipo” de la base de datos.

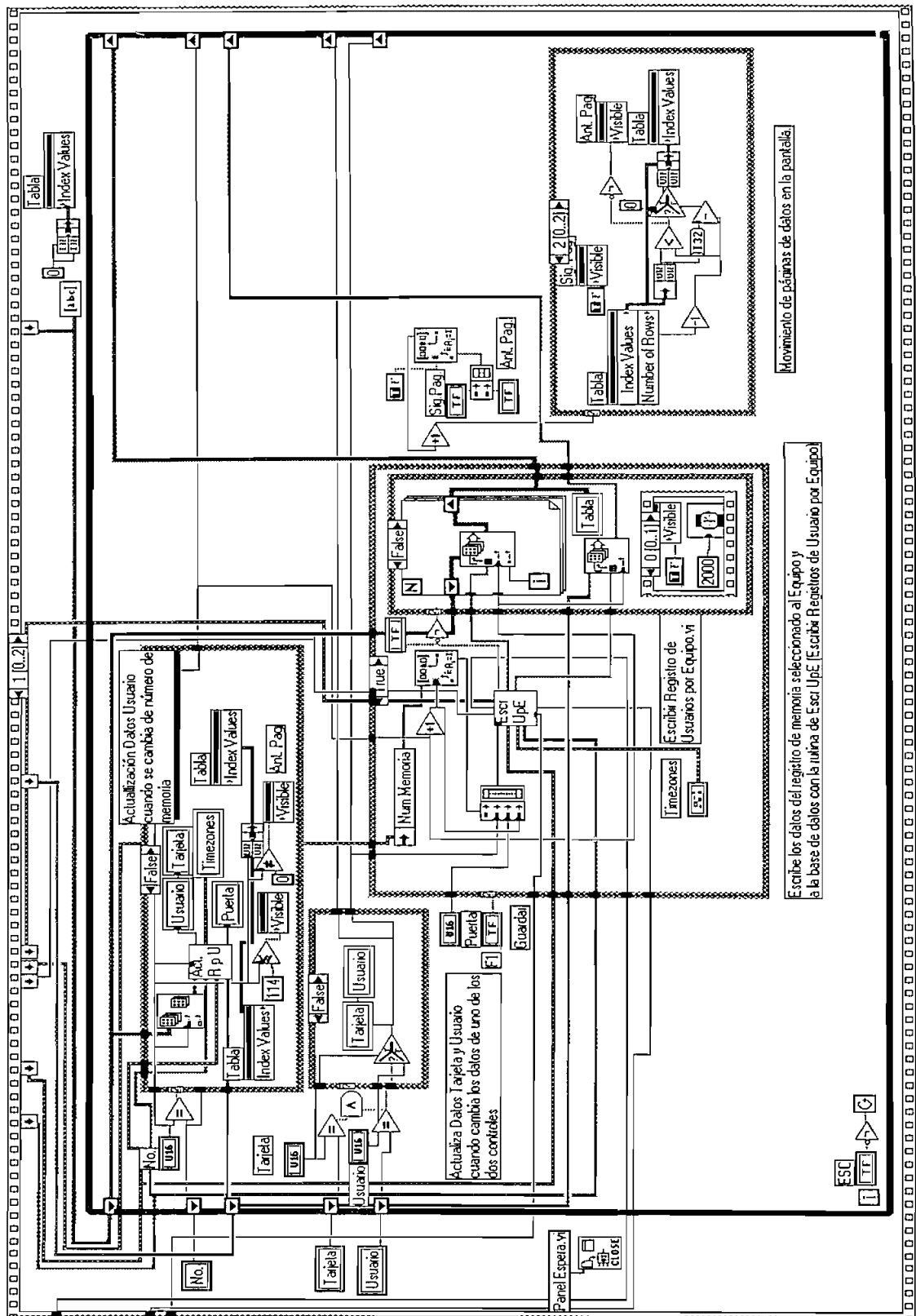


FIGURA III-22. LAZO PRINCIPAL DE PROGRAMACIÓN DE LA PANTALLA CONFIGURACIÓN
 TARJETAS VALIDAS PARA EL EQUIPO.

III.3.2.3. Rutinas Generación de Reportes.

La generación de reportes presenta dos opciones: Reporte por Usuario y Reporte por Equipo.

Al seleccionar cualquiera de las dos opciones se presenta interfaces similares y en consecuencia una programación similar. En las cuales se selecciona los parámetros para realizar la consulta en la tabla “Control de Accesos” de la base de datos. Los datos obtenidos son clasificados sea por equipo o por usuario según la opción seleccionada en la barra de herramientas.

Programación Pantalla Reporte por Usuario.

La programación comprende:

- Iniciar los controles. Dentro de la iniciación se leen los datos de las tablas “Usuarios” y “Equipo” de la base de datos. Además se inicializa la pantalla con las opciones de reporte de todas las tarjetas que ingresaron el día de “hoy”.
- Lazo de Control. Administra las opciones de la pantalla con ayuda del subVI “Manage Radio Button.vi”, una rutina que viene como parte de los ejemplos de Labview, que indica cuando el operador realiza un cambio entre opciones “mutuamente excluyentes” como son las opciones “Todas las Tarjetas” y “Seleccionar Tarjetas” y las opciones de selección de fecha “Hoy” y “Reporte entre” (ver Figura B-14). Cuando se cambia de una opción a otra se habilita los controles relacionados y sus atributos permitiendo de esta forma seleccionar los datos del para de informe que se puede realizar. Además las opciones “Por Fechas” y “Por Equipo” se controlan individualmente para habilitar los atributos de los controles relacionados con dichas opciones.

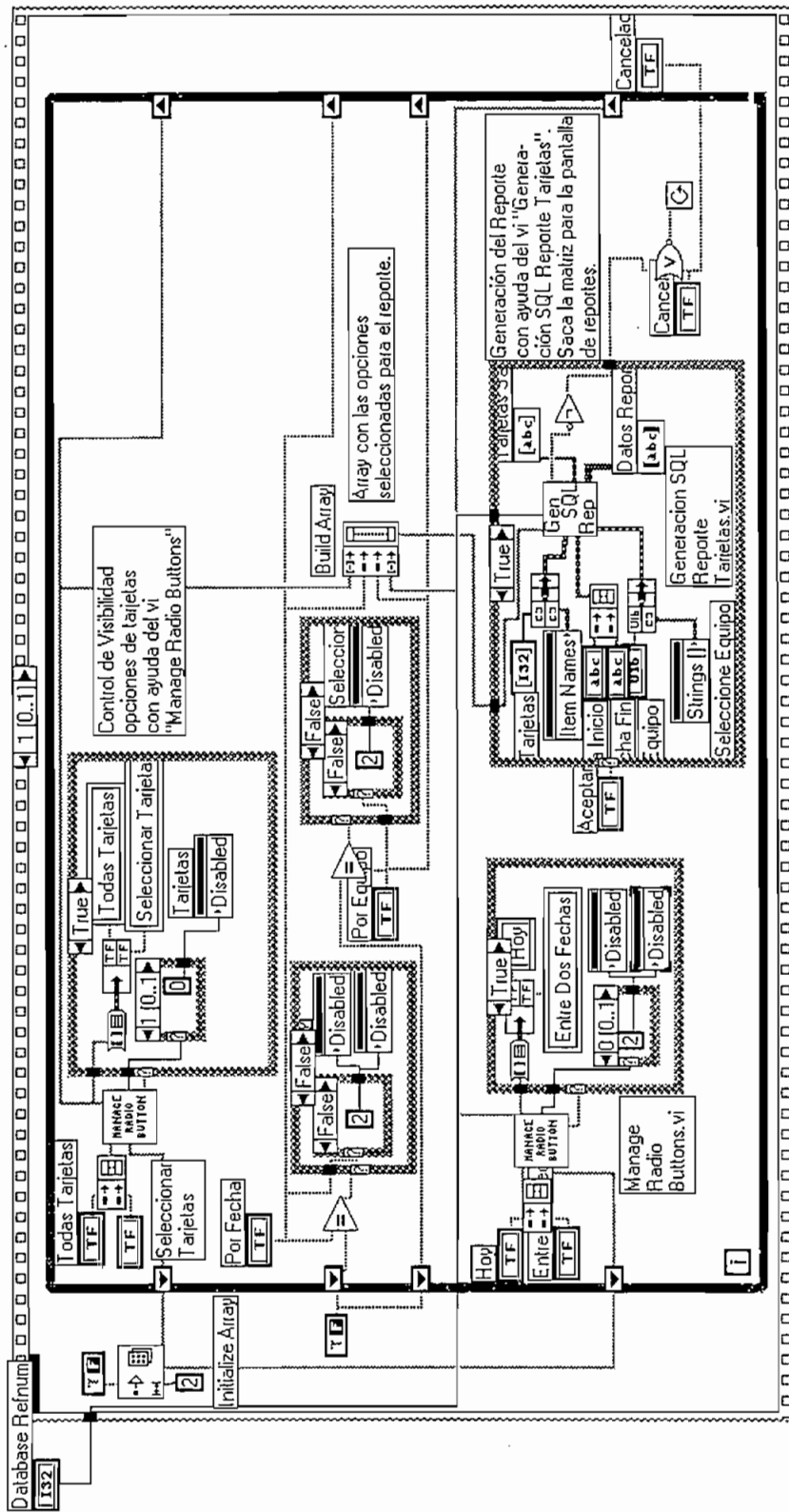


FIGURA III-23. LAZO PRINCIPAL DE PROGRAMACIÓN PARA PANEL PREVIO REPORTE POR TARJETAS.

- Generación de los datos a través de la consulta a la base de datos utilizando el subvi “Generación SQL Reporte Tarjetas.vi”, esta subrutina toma los datos de las opciones seleccionadas y genera una cadena SQL para leer la información desde la base de datos. Esta cadena como se muestra en la figura III-24 tiene la forma “SELECT [Control Accesos].* FROM [Control Accesos] WHERE %s %s %s ORDER BY IdTarjeta, HoraAccesos, Fecha Acceso, IdEquipo”, donde la forma %s son los parámetros que se seleccionó para realizar el reporte (ver ejemplo SQL en Introducción al Lenguaje SQL del Capítulo I). Los datos obtenidos en la consulta son almacenados en una matriz de 3 dimensiones que servirá como entrada para la pantalla de visualización del reporte.

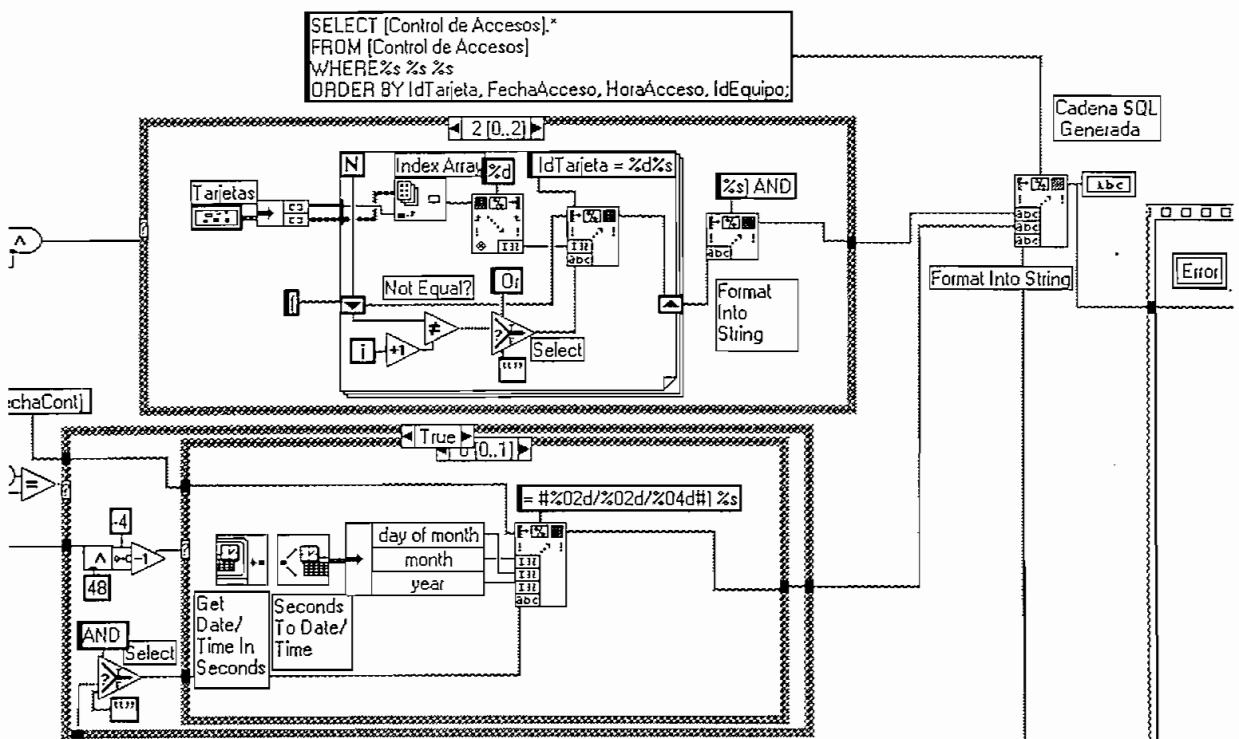


FIGURA III-24. GENERACIÓN DE LA CADENA SQL EN EL VI “GENERACIÓN REPORTE SQL TARJETA”.

Programación Pantalla Reporte por Equipo.

La filosofía de programación es la misma de la pantalla anterior es decir, esperar que el operador seleccione las opciones, para luego generar una consulta a la base de datos y almacenar los datos en una matriz para usarse en la pantalla de visualización de los reportes. (ver ítems **Programación Pantalla Reporte por Usuario.** y **Programación Pantalla Presentación Reportes.** del presente capítulo)

Programación Pantalla Presentación Reportes.

La programación de esta pantalla (Figura B-13) se basa en la visualización amigable de una matriz de tres dimensiones que genera la pantalla previa de reporte. Esta matriz se encuentra organizada de la siguiente forma:

- Se divide la matriz como submatrices de dos dimensiones.
- Cada matriz de dos dimensiones tiene los datos que aparecerán en la tabla de la pantalla.
- La tercera dimensión está relacionada con la tarjeta o equipo que aparece en el botón de selección que se encuentra en la parte izquierda de la pantalla.

Toda esta relación se muestra en la siguiente figura:

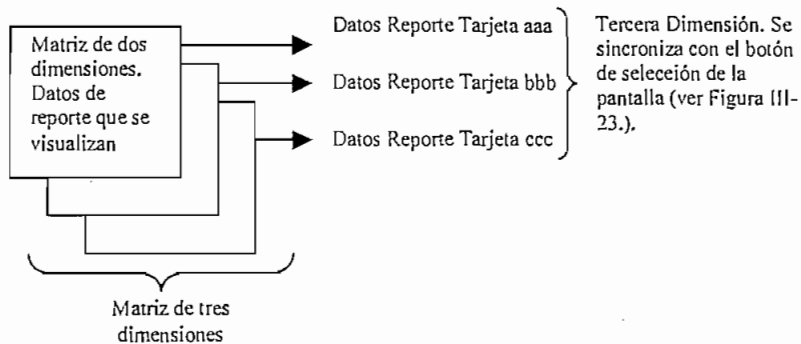


FIGURA III-25. ESTRUCTURA DE LOS DATOS DENTRO DE LA MATRIZ DE TRES DIMENSIONES QUE SIRVE PARA LA VISUALIZACIÓN DEL REPORTE.

La programación consiste en actualizar los datos de la tabla de la pantalla respecto a la tarjeta o botón seleccionado.

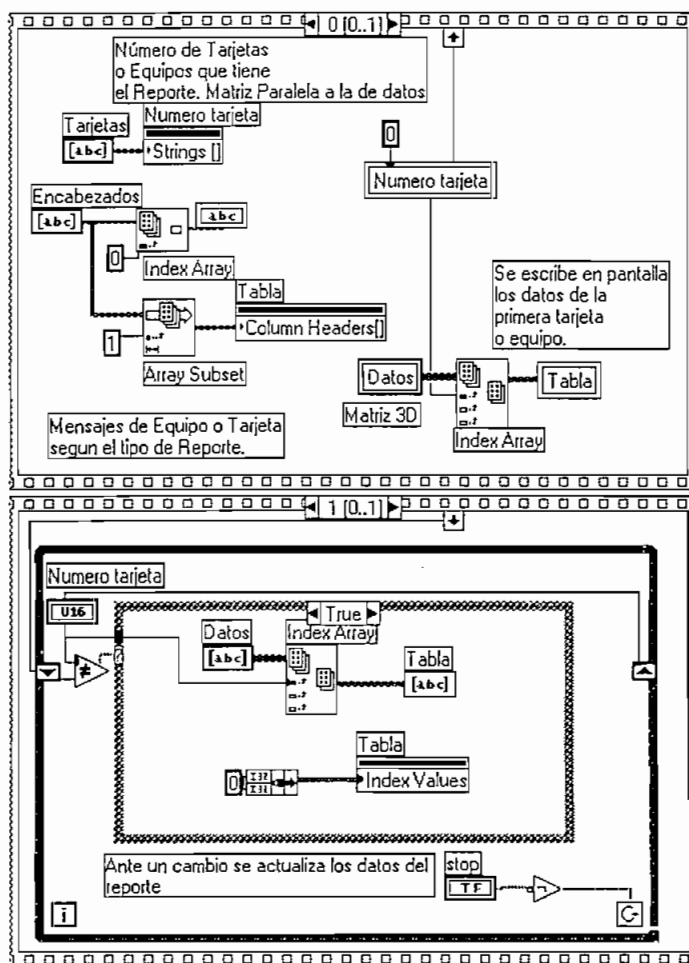


FIGURA III-26. PROGRAMACIÓN PANTALLA DE REPORTE.

III.3.3. DESARROLLO DE PROGRAMA DE COMUNICACIÓN SERIAL.

La comunicación serial se divide en dos partes: recepción y transmisión serial. La recepción se realiza a través de la subrutina “Recepción Serial Prot422”. La recepción serial es aleatoria y la decodificación se realiza en el lazo de programación del *Panel principal* a través de la subrutina “Administración de la pila de recepción.vi”, por esta razón los datos que se reciben son almacenados en una pila (de manera similar a lo que se realiza con la pila de mensajes pero los datos son cadenas de caracteres). A través de la subrutina de administración de la pila de recepción se decodifica los comandos que el equipo envía, los datos recibidos son de dos tipos: respuestas del equipo cuando ha ejecutado alguna instrucción y comandos enviados por el equipo para realizar una transferencia de datos. En cualquiera de los dos casos en la *Pantalla principal* se registra un mensaje que describe la acción realizada.

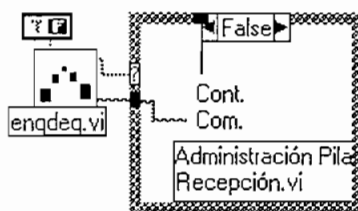


FIGURA III-27. RECEPCIÓN DE CADENAS DE DATOS RECIBIDAS EN EL PROGRAMA PRINCIPAL.

La transmisión serial no es aleatoria, se realiza solo cuando el operador escoge cualquier opción de las diferentes pantallas del programa que realiza alguna modificación en el equipo(microcontrolador). La transmisión se realiza a través de la subrutina “Escritura Serial”, antes de realizar una transmisión de datos se vacía la pila de recepción con ayuda de la rutina “Vaciar Pila RX”.

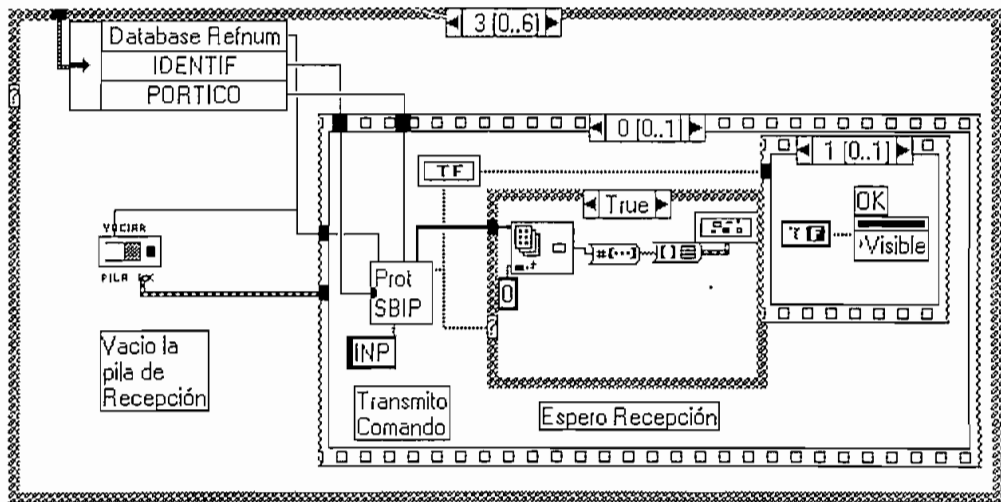


FIGURA III-28. EJEMPLO DE TRANSMISIÓN DEL COMANDO INP Y LA RECEPCIÓN DE DATOS.

III.3.4. DESARROLLO DE RUTINAS DE AUTOMATIZACIÓN OLE PARA COMUNICACIÓN ENTRE LABVIEW Y MICROSOFT ACCESS.

Como se explica en el capítulo I, Labview tiene Vis que le dan la capacidad de ser un cliente OLE, es decir, se puede controlar un programa (servidor OLE) a través de Labview.

El objetivo de las rutinas que se desarrollan a continuación es manejar datos contenidos en la base de datos sin abrir de manera visual Access. Microsoft Access dentro de su arquitectura como un servidor OLE presenta un objeto denominado Database Access Object que permite la manipulación de datos sin la necesidad de abrir la aplicación Microsoft Access y maneja el motor de la base de datos conocido como DBEngine (ver Arquitectura Microsoft Access, Capítulo I).

El proceso para manipular datos de Access a través del objeto DAO utilizando automatización OLE es:

Primero se establece comunicación con el objeto DAO.DBEngine. Seguidamente se abre la

base de datos en la cual se desea trabajar (método *OpenDatabase*). A continuación se abre el registro de datos de una tabla o consulta SQL (método *OpenRecordset*). Posteriormente se ejecuta generalmente una de las siguientes acciones: agregar registro (método *AddNew*), borrar registro (método *Delete*) o modificar registro (método *Edit*) y se graba los cambios (método *Update*).

A continuación se detallan las rutinas que permiten el manejo de la base de datos mediante Labview y los detalles relacionados con los pasos descritos anteriormente para la manipulación de los datos.

III.3.4.1. Rutinas de apertura de la comunicación con Access.

Comunicación con el objeto DAO.DBEngine.

Para establecer la comunicación con el objeto DAO, se debe crear un número de referencia de automatización (ver Labview como cliente de Automatización OLE. SubVis Incorporados en el Programa, Capítulo I). El nombre para abrir el objeto DAO de Microsoft Access 8.0 es *DAO.DBEngine.35*, para obtener este se debe ingresar al registro de Windows (Registry de Windows).

La programación comprende un SubVI para crear el número de referencia de DAO tal como se muestra a continuación (rutina “Abrir DAO.vi”).

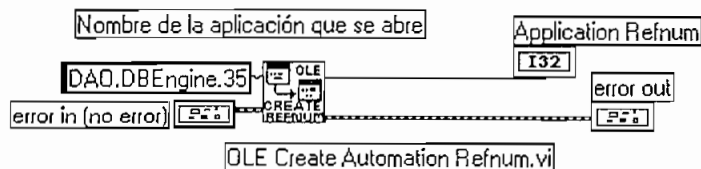


FIGURA III-29. PROGRAMA DE APERTURA DE COMUNICACIÓN CON DAO.

Apertura de la base de datos.

Antes de comenzar a realizar el manejo de los datos se debe especificar el nombre de la base de datos sobre la que se va a trabajar, para esto se dispone del método *OpenDatabase* que necesita como un argumento de entrada la ruta completa y el nombre de la base de datos a abrirse y devuelve como salida el número de referencia para trabajar con los elementos de la base de datos. La rutina “Abrir Base de Datos.vi” realiza este trabajo.

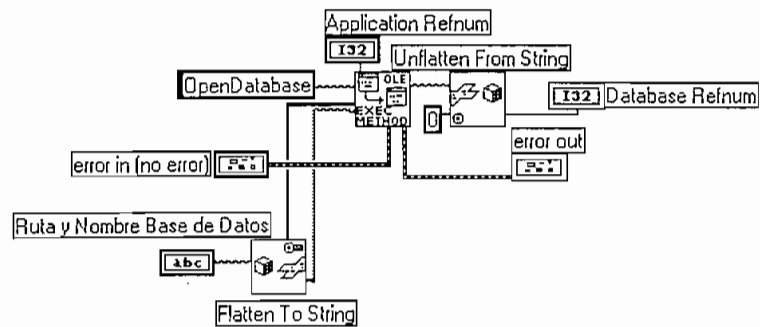


FIGURA III-30. PROGRAMACIÓN DE “ABRIR BASE DE DATOS.VI”

Apertura de un registro de datos (Recordset).

Para abrir un conjunto de registro de datos se utiliza el método *OpenRecordset* que toma como argumento de entrada el nombre de una tabla, una consulta existente o una consulta SQL, y presenta como salida el número de referencia de automatización de enlace a los registros de datos. La rutina “Abrir Registro de Datos.vi” genera este número de referencia.

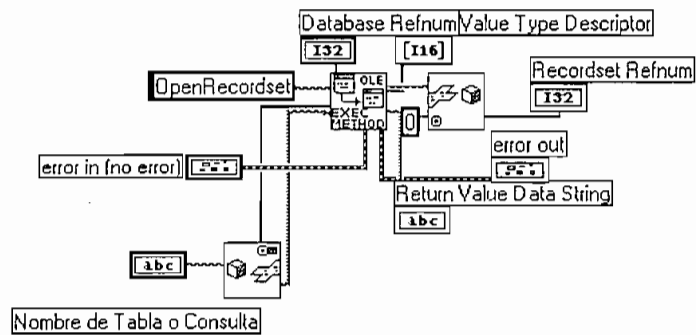


FIGURA III-31. PROGRAMACIÓN “ABRIR REGISTRO DE DATOS.VI”

Apertura de los campos de un registro de datos.

Un conjunto de registro de datos tiene varios campos. Cada campo necesita un número de referencia de automatización. La apertura de los campos de un conjunto de registros (de un Recordset) hace uso de los siguientes métodos: *Fields*, *Count* (métodos del objeto Recordset), *Item* (método del objeto Fields) (ver **Arquitectura de una aplicación de Microsoft Access**, capítulo I). Este último método es el que da el número de referencia para los campos, sin embargo cada campo tiene un nombre que ayuda a ubicar rápidamente al mismo, para saber el nombre de un campo se utiliza el método *Name*, el método *Type* permite conocer el tipo de datos de un campo.

Para la apertura de campos se utiliza el subvi “Abrir Campos.vi”, el mismo que entrega como salida un cluster con el nombre de los campos, los números de automatización y el tipo de datos.

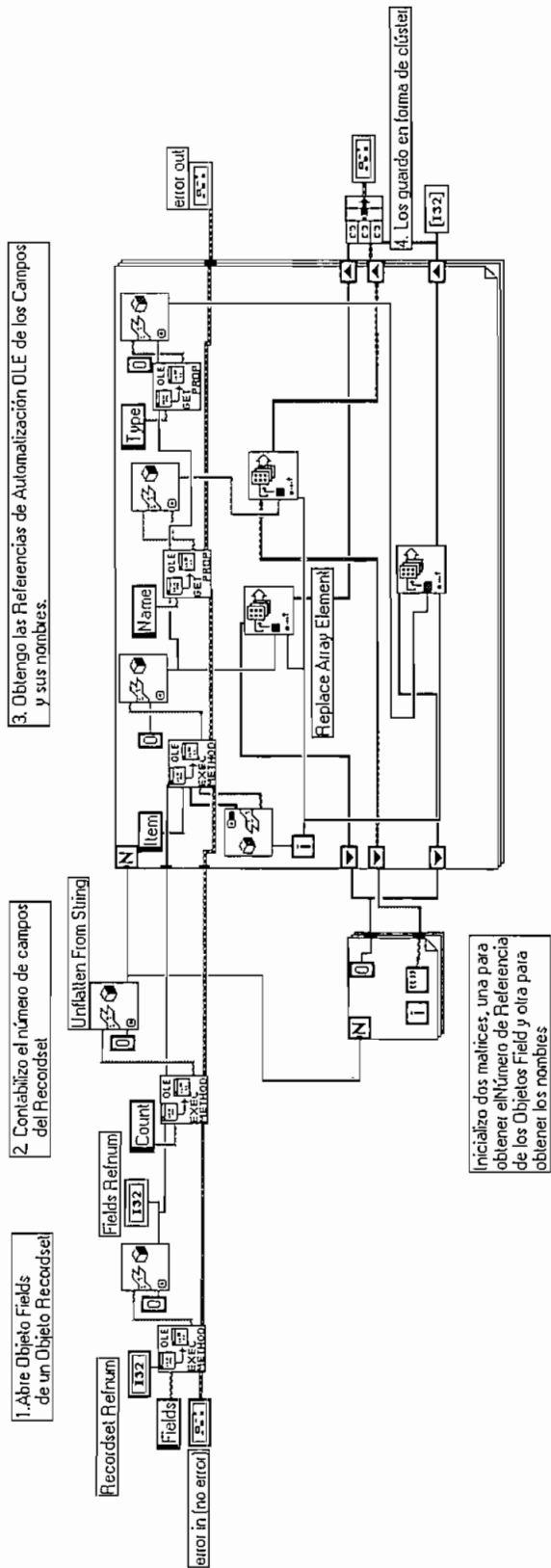


FIGURA III-32. PROGRAMACIÓN “ABRIR CAMPOS.VI”

III.3.4.2. Rutinas de manejo de registros.

Añadir datos a través de una consulta SQL de acción.

De manera general la modificación de un conjunto de registros de datos utiliza los métodos del objeto Recordset, sin embargo cuando se quiere añadir o eliminar datos a una tabla se puede utilizar una consulta SQL de acción.

Las consultas de acción requieren como parámetros el nombre de la tabla a la que se desea añadir o eliminar datos, el nombre de los campos y los valores con los que van a añadir un nuevo dato al registro de datos de la tabla (ver **Introducción al lenguaje SQL**, Capítulo I) y utilizar el método *Execute* del objeto Database.

Para añadir datos de esta forma se utiliza el subvi “Añadir Datos a una Tabla.vi”.

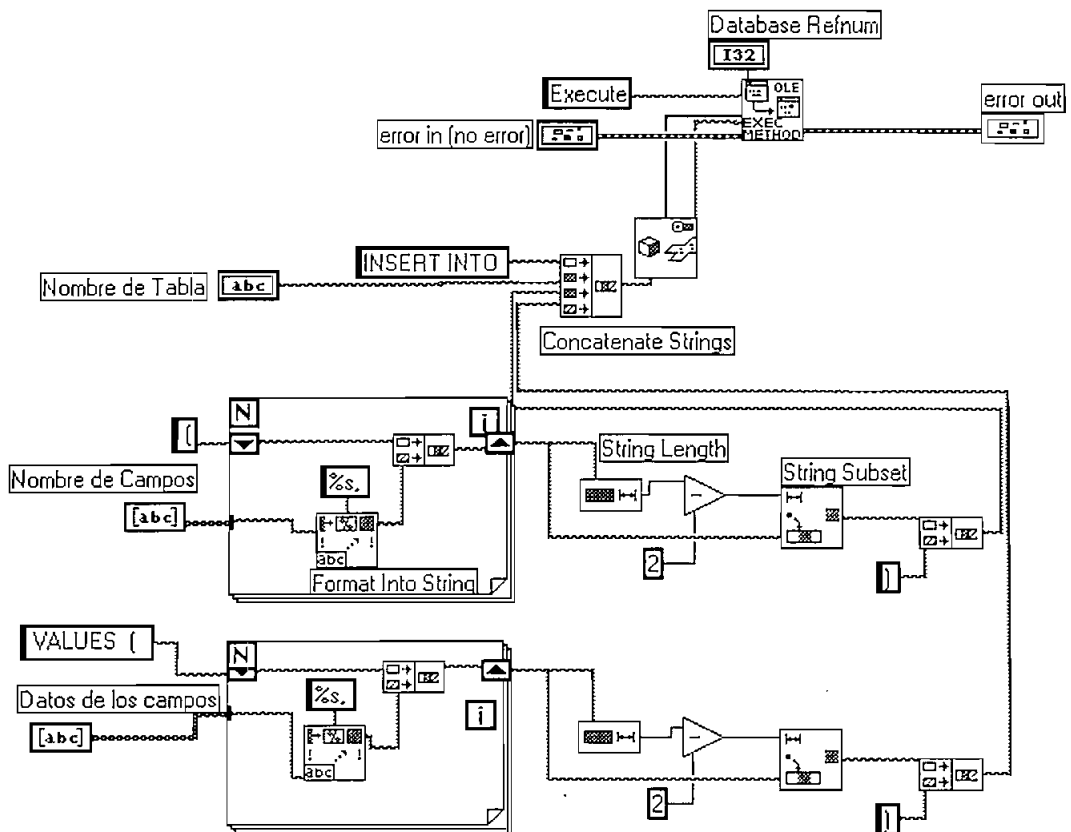


FIGURA III-33. PROGRAMACIÓN “AÑADIR DATOS A UNA TABLA”.

Escritura de datos a un objeto Recordset.

El proceso de escritura en una tabla o consulta tiene el siguiente proceso: si se va a añadir datos se utiliza el método *AddNew* del objeto Recordset, mientras si se edita el registro se debe mover al registro a modificarse con cualquiera de los métodos *Move* (ver Métodos y Propiedades de Microsoft Access, Capítulo I) del objeto Recordset y utilizar el método *Edit*. Posteriormente se debe escribir el nuevo valor de cada uno de los campos mediante el uso del método *Value* del objeto Field (cada campo tiene un número de referencia). Por último se utiliza el método *Update* del objeto Recordset para que el cambio se efectúe, si no se utiliza este último método las modificaciones realizadas se pierden.

Si al ingresar un dato a un campo no coincide con el tipo de datos que espera Access se genera un error que puede ser interceptable por programación y no se agrega el nuevo dato. Para escritura de los datos de las tablas se utilizan dos subvis: "Escribir Fields Numérico de un Recordset.vi" y "Escribir Fields Strings de un Recordset.vi".

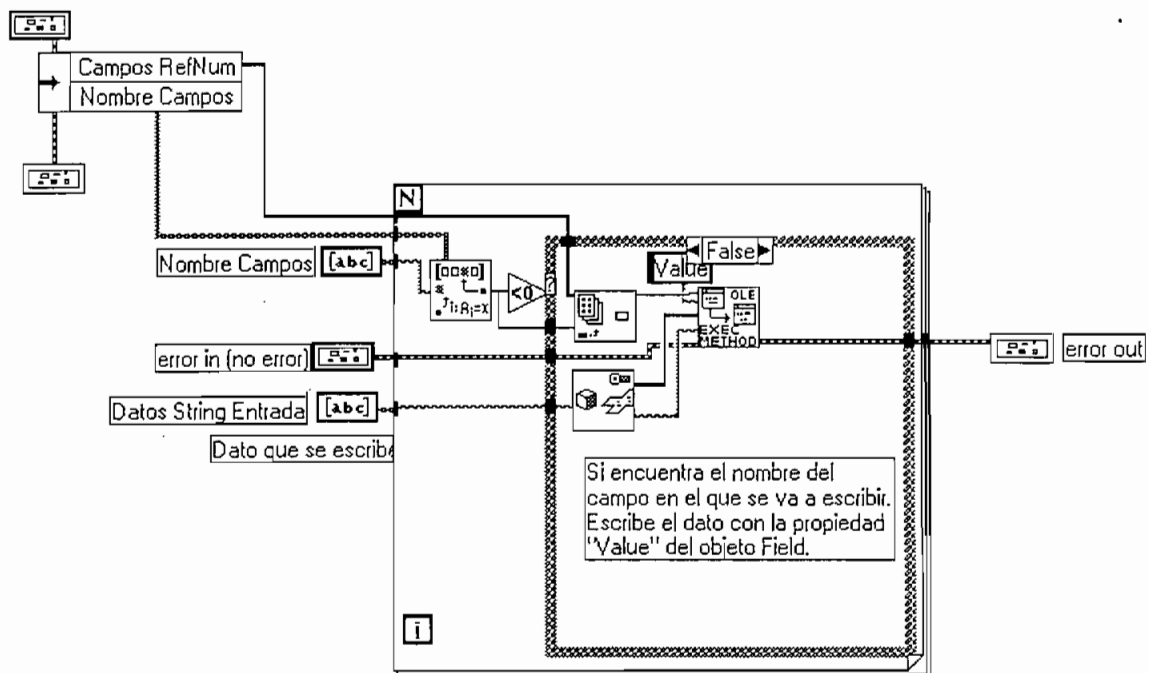


FIGURA III-34. PROGRAMACIÓN "ESCRIBIR FIELDS STRING DE UN RECORDSET".

Lectura de datos de un Recordset.

La lectura de un registro de un Recordset implica leer los valores de los campos a través del método *Value* los datos devueltos por los subvis de automatización OLE (“Get Property.vi” y “Execute Method.vi”) están en formato “plano” y se debe reconstruir a través de la función “Unflatten from String”, esta función requiere que se ponga un dato con el tipo de valor que se desea recibir, por ejemplo si se espera recibir un dato entero sin signo de 16 bits, se debe poner un valor constante con estas propiedades.

En el desarrollo del software se necesita para inicializar ciertos subvis leer todos los registros de un objeto Recordset, por esta razón se desarrollaron rutinas que permitan la lectura de todos estos registros, estas son: “Abrir Fields NumI8 de un Recordset.vi” “Abrir Fields NumI16 de un Recordset.vi”, “Abrir Fields NumI32 de un Recordset.vi” y “Abrir Fields Strings de un Recordset.vi”. Estas subrutinas utilizan el método *MoveFirst* para moverse al primer registro del Recordset, a continuación leen los datos de un campo específico con el método *Value*, luego se mueve al siguiente registro con el método *MoveNext* del objeto Recordset y se lee la propiedad *EOF* del objeto Recordset, si es falsa se continúa leyendo el siguiente registro, caso contrario se termina la lectura de los datos.

para eliminar el registro. En el desarrollo del software no se desarrolló una subrutina específica para borrar registros, pero se utiliza directamente en algunas pantallas del sistema como es la pantalla de Usuario.

III.3.4.3. Filosofía de manejo de la base de datos.

El software desarrollado maneja la base de datos de la siguiente manera:

- El programa principal llama a la subrutina “Manejo BD Control de Accesos.vi”, esta rutina abre la comunicación con el objeto DAO.DBEngine y la base de datos, luego escribe el número de referencia de automatización de la base de datos en una variable global, de manera que este expuesta a todas las rutinas del sistema, y continua ejecutándose esta rutina.
- El programa principal espera a que se inicie la comunicación para dejar al operador utilizar el programa.
- Luego las pantallas que utilizan la base de datos hacen uso de las rutinas de manejo de la base de datos.
- Cuando el operador cierra el programa principal, la rutina “Manejo BD Control Accesos.vi” cierra la comunicación con Access con ayuda del subVI FreeNum.vi.

III.3.5. DESARROLLO DE LA BASE DE DATOS.

El papel de la base de datos dentro del sistema de control de accesos es ser un medio de consulta y almacenamiento de datos. Por esta razón el desarrollo de la base de datos consiste en el diseño de tabla cuyos datos se relacionen entre sí conforme las reglas de las bases de datos relacionales y no implica el desarrollo de consultas, formularios, reportes y

programas para el desarrollo de un sistema de gestión de datos más complejo.

En el diseño se definieron las siguientes tablas: Contraseñas, Control de Accesos, Equipo, Horario Timezones, Puertas, Registro de Usuarios por Equipo, Usuarios y Zonas de Tiempo. A continuación se presenta las relaciones de las diferentes tablas de la base de datos.

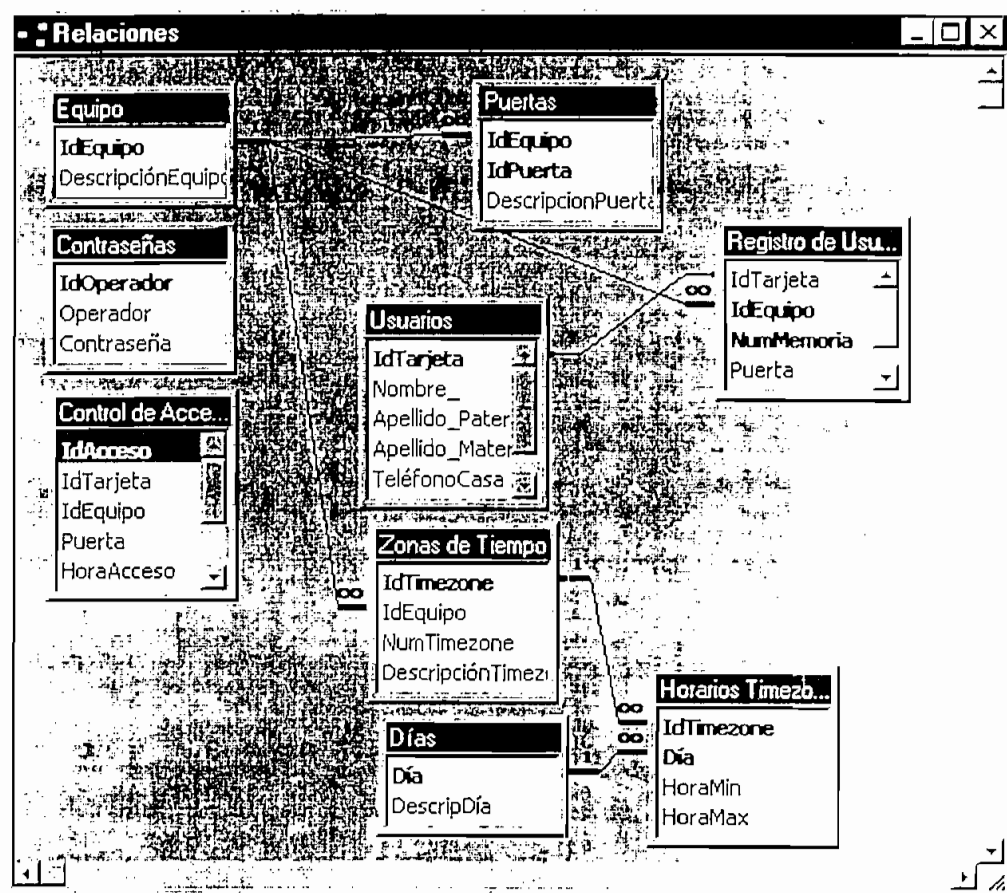


FIGURA III-36. RELACIONES DE LAS TABLAS DENTRO DE LA BASE DE DATOS.

Cada tabla se relaciona con una pantalla de Labview como se lo ha explicado durante el transcurso del presente capítulo. A continuación se detalla los campos que contiene cada tabla, el tipo de datos y la función de los campos más importantes de la tabla. (Si desea información detallada de las tablas por favor remítase a la documentación del software).

III.3.5.1. Tabla Contraseñas.

Almacena los nombres de los operadores y las contraseñas correspondientes a dicho operador. Contiene tres campos: IdOperador, Operador, Contraseña. El primer campo es autonumérico (se incrementa automáticamente con el ingreso de un nuevo registro) y los siguientes son campos de textos. Al desplegar los datos en Access, la contraseña aparece en el formato de contraseña, es decir, el administrador de la base de datos no puede saber cual es la contraseña del operador de manera directa.

III.3.5.2. Tabla Equipo.

Almacena sus registros en dos campos: IdEquipo y DescripciónEquipo, el primero como un campo numéricos y el segundo como un texto. El primer campo es el número de identificación del equipo dentro de la red y el segundo es una descripción del equipo que da el operador.

III.3.5.3. Tabla Usuarios.

Genera un conjunto de registros con los números de tarjetas disponibles para el sistema y los datos del usuario perteneciente a esta tarjeta. Tiene 5 campos: IdTarjeta, Nombre_, Apellido_Paterno, Apellido_Materno y TeléfonoCasa, donde el primer campo es tipo numérico y el resto de campos son tipo texto. Tiene como una restricción que el formato del número telefónico debe contener tres números seguidos de un guión y otros tres números (por ejemplo 535-984) y los tres primeros campos citados deben tener datos para ser un registro válido.

III.3.5.4. Tabla Puertas.

Contiene para cada puerta de los Equipo dentro del sistema la descripción que les da el

operador. Contiene tres campos.

IdEquipo: campo numérico que representa el número de Equipo dentro de la red.

IdPuerta: campo numérico que acepta solo dos valores 1 o 2, e indica el número de puerta del equipo (ver programa del microcontrolador para identificar a que relé se conecta puerta 1 y puerta 2).

DescripcionPuerta: campo de texto que especifica la descripción dada por el operador a dicha puerta.

III.3.5.5. Tabla Registro de Usuarios por Equipo.

Registra información del número de tarjetas definidas para cada equipo de la red del sistema de control de accesos. Tiene cinco campos.

IdTarjeta: campo numérico, almacena código de tarjeta definida para ese registro.

IdEquipo campo numérico.

NumMemoria: campo numérico, almacena número asignado al usuario en la memoria del microcontrolador.

Puerta: campo numérico, indica las puertas a las que tiene acceso el usuario. Acepta solo cuatro valores: 0= ninguna puerta, 1=puerta1, 2=puerta2, 3= ambas puertas.

TimezoneAcceso: campo texto, contiene un texto equivalente a un número binario de 16 bits. Cada bit representa la zona de tiempo del equipo, 1= zona de tiempo permitida, 0= zona de tiempo no permitida.

III.3.5.6. Tabla Control de Accesos.

Registra cada ingreso de una tarjeta al sistema de control de accesos. Está compuesta de ocho campos.

IdAcceso: campo autonumérico.

Id Tarjeta: campo numérico.

IdEquipo: campo numérico.

Puerta: campo numérico.

HoraAcceso: campo texto, almacena la hora de ingreso de la tarjeta al sistema.

FechaAcceso: campo texto, almacena la fecha de ingreso de la tarjeta al sistema.

Mensaje: campo texto, almacena mensaje de tipo de acceso registrado a la tarjeta ingresada.

Fecha_Cont: campo tipo fecha/hora, este campo crea una copia del campo FechaAcceso. Se utiliza para realizar consultas por fechas en los reportes de Labview.

III.3.5.7. Tabla Zonas de Tiempo.

Almacena un número único para cada timezone de cada equipo, además de una descripción para cada una de las timezones. Tiene cuatro campos.

IdTimezone: campo autonumérico, número de identificación de la timezones en la base de datos.

IdEquipo: campo numérico.

NumTimezone: campo numérico, contiene el número de timezones asignada al equipo.

DescripciónTimezone: campo texto, descripción de la timezone.

III.3.5.8. *Tabla Horario Timezones.*

Almacena los horarios de acceso de las zonas de tiempo. Tiene cuatro campos.

IdTimezone: campo numérico.

Día: campo numérico, indica el día de programación del horario. 0=domingo, 7=sábado.

HoraMin: campo texto, almacena hora mínima de acceso.

HoraMax: campo texto, almacena hora máxima de acceso.

CAPÍTULO IV:

PRUEBAS Y RESULTADOS

IV.1. PRUEBAS EN STAND ALONE.

El equipo (microcontrolador) que administra el sistema tiene la capacidad de realizar el control de accesos sin necesidad de conexión al computador central. Las pruebas que se realizan con el equipo funcionando en Stand Alone son:

- Detección de tarjeta.
- Funcionamiento de salidas de relé.
- Apertura manual.

Todas las pruebas mencionadas se realizan después de configurar el equipo, es decir definir tarjetas autorizadas, horarios de zonas de tiempo y tiempo de apertura de cerradura (salidas de relé).

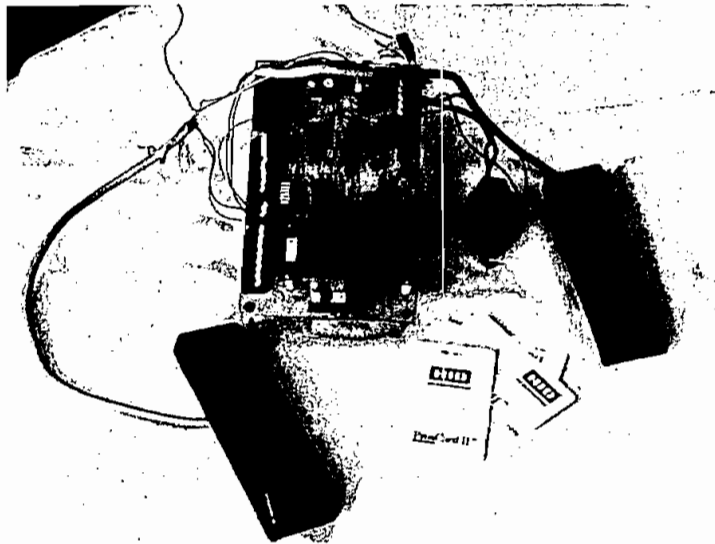


FIGURA IV-1. FOTO DE EQUIPO.

IV.1.1. DETECCIÓN DE TARJETA.

Esta prueba consiste en verificar si el equipo reconoce tarjetas de usuario. Los resultados se observan a través de mensajes que aparecen en el display, que indican el tipo de acceso registrado. Los mensajes que aparecen en el display son: Aceptada, No reconocida y Reconocida.

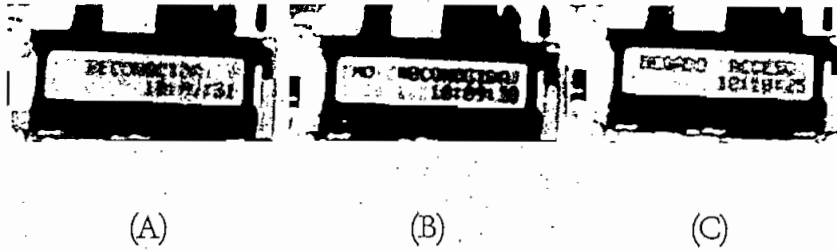


FIGURA IV-2. MENSAJES DESPLEGADOS POR EL EQUIPO.

(A) RECONOCIDA, (B) NO RECONOCIDA, (C) NEGADO ACCESO

IV.1.2. FUNCIONAMIENTO DE SALIDAS DE RELÉ.

Cuando una tarjeta es reconocida el equipo activa la salida de relé respectiva, esta prueba se realiza conectando la salida de relé a un circuito de encendido de una lámpara.

En la primera prueba se observa que la lámpara se activa automáticamente y permanece en este estado el tiempo programado para la apertura (tiempo del monoestable) manteniendo las entradas de los contactos de fin de carrera (que detectan la apertura total de la puerta) abiertas (uno lógico).

La segunda prueba de funcionamiento de la salida de los relés simula que la puerta se abre antes del tiempo programado para la apertura de los relés. Es decir durante la activación de la salida de relé se pone la entrada de los contactos de fin de carrera a cero lógico y se observa que la salida de relé desactiva la lámpara.

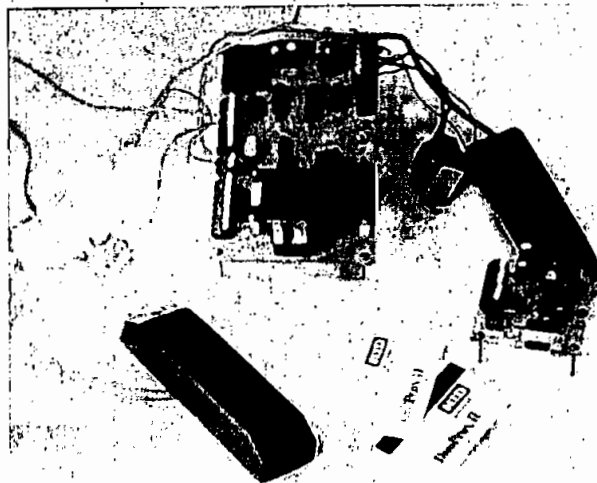


FIGURA IV-3. PRUEBAS DE RELÉ (FOTOGRAFÍA DEL CIRCUITO)

IV.1.3. APERTURA MANUAL

Esta prueba consiste en abrir la puerta manualmente es decir activar la salida de relé sin necesidad de detección de una tarjeta. Esto se realiza utilizando el mismo circuito de encendido de lámpara de la prueba anterior y simulando la activación manual a través de un pulsante que envía una señal al equipo (entrada de pulsante) para la apertura de relé. Cuando la señal de la entrada de pulsante se encuentra en uno lógico, la salida de relé no sufre ningún cambio, pero cuando la señal de entrada cambia a cero lógico la salida de relé activa la lámpara (abre la puerta).

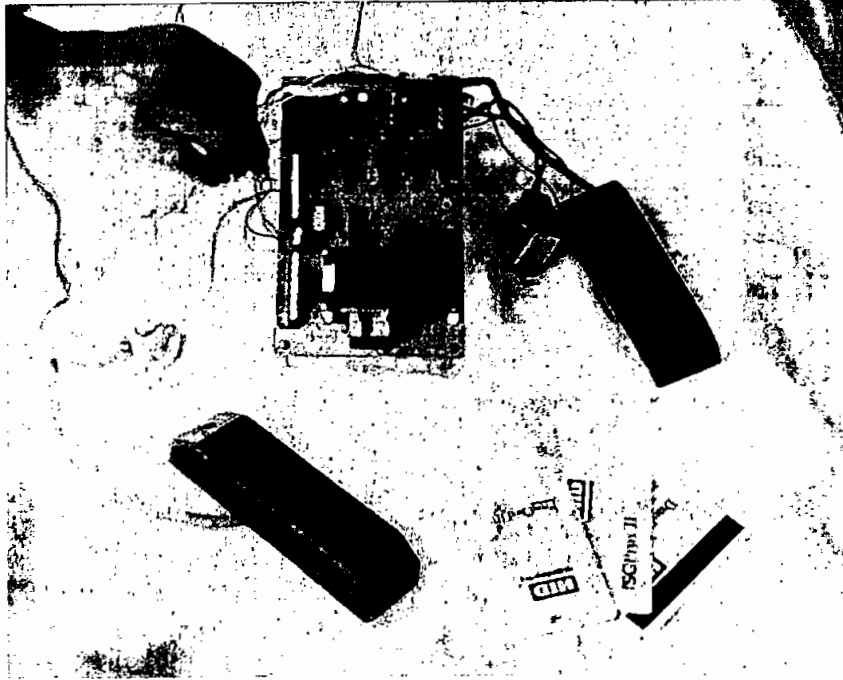


FIGURA IV-4. PRUEBA APERTURA MANUAL CERRADURA.

IV.2. PRUEBAS DEL SISTEMA DE CONTROL DE ACCESO

El sistema de control de accesos se conforma de los siguientes elementos: el equipo que realiza el control de accesos y el computador personal que monitorea el sistema. Las pruebas que se realizan son:

- Comunicación entre el equipo y el computador personal.
- Configuración del equipo.
- Base de Datos.
- Reportes

Estas pruebas se realizan cuando el equipo y el computador se encuentran interconectados utilizando la interfaz de comunicación que se describió en el capítulo II. En el computador

debe ejecutarse el programa desarrollado para esta aplicación (descrito en el capítulo III).

IV.2.1. COMUNICACIÓN ENTRE EL EQUIPO Y EL COMPUTADOR PERSONAL.

Para verificar la comunicación entre el equipo y el computador se ingresa al menú de configuración y en la pantalla de selección de equipo se escoge el equipo a través del número de identificador o nombre. Si se establece comunicación con el equipo se pasa a la pantalla de configuración de equipo, caso contrario aparece un mensaje que indica que no se estableció comunicación. Otro medio de verificación es observando el mensaje que aparece en la pantalla principal que indica si la comunicación fue exitosa o no.

A continuación se muestra el mensaje de error de comunicación con el equipo, para esta prueba se intentó configurar al equipo con identificador diferente al configurado (a través de los dipswitches).



FIGURA IV-5. PANTALLA DE ERROR DE COMUNICACIÓN.

A continuación se muestra el mensaje que aparece en la pantalla principal cuando ocurre una comunicación fallida.

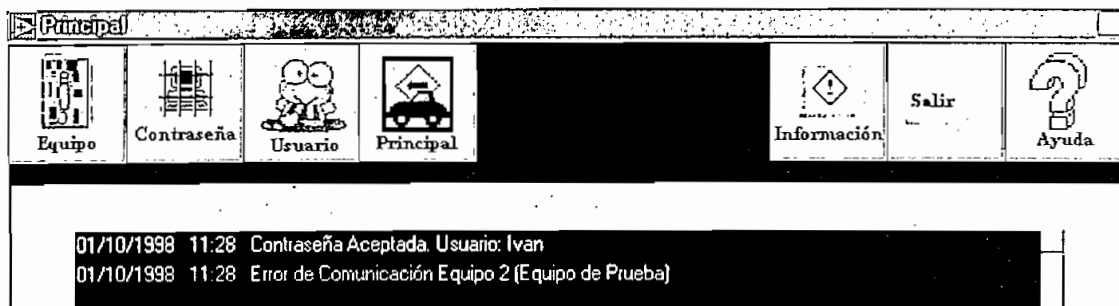


FIGURA IV-6. MENSAJE DE FALLA COMUNICACIÓN EN LA PANTALLA PRINCIPAL.

IV.2.2. CONFIGURACIÓN DE EQUIPO.

La configuración de equipo se realiza solo cuando la comunicación es exitosa. Las pruebas de configuración que se realiza son: consulta e igualación de reloj calendario, lectura de entradas de alarma, configuración de monoestable para apertura de puertas, configuración de zonas de tiempo y configuración de usuarios autorizados.

Las figuras que se encuentran en el Anexo B muestran pantallas obtenidas en las pruebas de configuración de un equipo:

A continuación se muestra una pantalla en la que se lee el estado de las entradas de alarma del Equipo con identificador 2 utilizado como equipo de Prueba.



FIGURA IV-7. PANTALLA PRINCIPAL DE CONFIGURACIÓN.

Esta Pantalla indica que tres estados de entradas de alarma están activas (para la prueba es un estado alto) y las demás están normales (estado bajo). Esta pantalla además es una prueba del funcionamiento de las entradas de alarma, parámetro que no puede detectarse realizando pruebas en stand alone.

IV.2.3. BASE DE DATOS

Las actividades del sistema de control de accesos se almacenan en la base de datos en la tabla Control de Acceso que se presenta a continuación.

Control de Accesos : Tabla							
Identificación	Id Tarjeta	Id Equipo	Puerta	Hora de Acces	Fecha de Acc	Mensaje	Fec
431	78	Equipo 31		1 09:44	24/06/1998	Denegado	2.
432	78	Equipo 31		2 09:45	24/06/1998	Aceptado	2.
433	78	Equipo 31		2 09:45	24/06/1998	Aceptado	2.
434	485	Equipo 31		2 09:45	24/06/1998	Aceptado	2.
435	485	Equipo 31		1 09:45	24/06/1998	Aceptado	2.
436	161	Equipo 31		1 09:45	24/06/1998	Aceptado	2.
437	161	Equipo 31		2 09:45	24/06/1998	Denegado	2.
438	161	Equipo 31		2 09:55	24/06/1998	Denegado	2.
439	161	Equipo 31		1 09:55	24/06/1998	Aceptado	2.
440	485	Equipo 31		2 09:56	24/06/1998	Aceptado	2.
441	485	Equipo 31		1 09:56	24/06/1998	Aceptado	2.
442	485	Equipo 31		1 09:56	24/06/1998	Aceptado	2.
443	485	Equipo 31		2 09:56	24/06/1998	Aceptado	2.
444	161	Equipo 31		1 10:17	24/06/1998	Aceptado	2.
445	161	Equipo 31		2 10:17	24/06/1998	Denegado	2.
446	161	Equipo 31		1 10:17	24/06/1998	Aceptado	2.
447	161	Equipo 31		1 10:17	24/06/1998	Aceptado	2.
448	485	Equipo 31		1 10:17	24/06/1998	Aceptado	2.
449	485	Equipo 31		2 10:17	24/06/1998	Aceptado	2.
450	161	Equipo 31		2 10:18	24/06/1998	Denegado	2.
451	78	Equipo 31		1 10:18	24/06/1998	Denegado	2.
452	78	Equipo 31		2 10:18	24/06/1998	Aceptado	2.

FIGURA IV-8. PANTALLA TABLA “CONTROL ACCESOS”.

A continuación se muestran los datos de las tablas “Contraseñas”, “Usuarios” y “Horarios Zonas de Tiempo” de la base de datos que se almacenaron durante la configuración del equipo de la tesis.

IV.2.4. REPORTES

La generación de reportes se realiza de acuerdo a los parámetros que se selecciona en la pantalla previa a la presentación del reporte (ver Pantalla Previo Reporte, Anexo B). Para la prueba se selecciona un reporte de todos los equipos y todas las tarjetas que ingresaron el día de hoy, en la figura se presentan los resultados.

REPORTE

Tarjeta	Puerta	Hora Acceso	Fecha Acceso	Mensaje
161	1	15:16	01/08/1998	Aceptado
161	1	15:16	01/08/1998	Aceptado
161	1	15:51	01/08/1998	Aceptado
161	1	15:51	01/08/1998	Aceptado
161	1	15:51	01/08/1998	Aceptado
161	1	15:51	01/08/1998	Aceptado
161	1	15:51	01/08/1998	Aceptado
161	1	15:51	01/08/1998	Aceptado
161	1	15:51	01/08/1998	Aceptado
161	1	15:51	01/08/1998	Aceptado
161	1	15:51	01/08/1998	Aceptado
161	1	15:54	01/08/1998	Aceptado
161	1	15:54	01/08/1998	Aceptado
161	1	15:57	01/08/1998	Aceptado
161	1	15:57	01/08/1998	Aceptado
20496	1	10:55	01/08/1998	No Encontrado
40992	1	10:56	01/08/1998	No Encontrado
40992	1	10:56	01/08/1998	No Encontrado

No. Equipo

1

Aceptar

FIGURA IV-9. PANTALLA PRESENTACIÓN REPORTES.

A continuación se presenta los datos filtrados de la tabla de la base de datos para los mismos parámetros de la pantalla anterior.

CAPÍTULO V:

CONCLUSIONES Y RECOMENDACIONES

V.1. CONCLUSIONES.

- La tecnología de lectoras de proximidad se encuentra al momento estandarizada, esta característica permite que el equipo desarrollado en esta tesis pueda utilizar cualquier lectora de proximidad con una salida tipo Wiegand, con el único detalle que el equipo solo lee 26 bits de datos de cada tarjeta de usuario. Si se quisiera leer más bits por tarjeta se requiere cambiar los contadores de bits de las rutinas de interrupción externa 0 y 1.
- El paquete Labview 4.1 es muy versátil para realizar comunicación hacia equipos externos, factor que ayudó a que la configuración del puerto serial sea muy sencilla durante el desarrollo del programa. Sin embargo, este paquete presenta muchas dificultades para comunicarse adecuadamente con otro software a través de Automatización OLE. Durante el desarrollo de las rutinas de enlace entre Labview y Microsoft Access se presentaron varias dificultades al leer o escribir datos a través de algunos métodos y propiedades, especialmente con la incongruencia de datos, por esta razón se necesitó desarrollar rutinas extensas para la comunicación. En conclusión si el sistema desarrollado solo tiene una proyección de trabajo exclusivo de control de acceso se recomendaría el desarrollo del software en un programa. Otra solución más costosa pero sencilla y potente sería sustituir las rutinas del software de Labview para comunicación con Access con la herramienta de manejo de base de datos a través de lenguaje SQL que se vende como un producto adicional de Labview.

- La topología de comunicación seleccionada para el trabajo (en anillo) presenta muchas facilidades respecto a la eliminación del desarrollo de rutinas en el microcontrolador que administren colisiones que pueden ocurrir en la red cuando se utiliza otra topología como son de bus o estrella utilizando una operación en modo full duplex (requerido en el protocolo RS-422), pero no se evita el esfuerzo de desarrollar otras técnicas para administrar la recepción y transmisión de manera independiente. En nuestro caso se necesitó el desarrollo de rutinas de administración propias para cada uno de estos trabajos, como son: definición de los buffer de recepción, transmisión y validación y manejo de punteros para los buffers, factores que demoraron la depuración de estas rutinas para su correcto funcionamiento. A pesar de las dificultades presentadas la topología en anillo es útil para manejar comunicación modo full-duplex.
- La decisión de desarrollar el trabajo a través de las zonas de tiempo presentó varias inquietudes sobre la forma en que se desarrollaría este concepto, pues en el sentido estricto las zonas de tiempo definen varios horarios durante el día, y el nivel de acceso es quien controla el área física al que se aplicarán estos horarios. Sin embargo se decidió el trabajar solo bajo zonas de tiempo y eliminar el concepto de nivel de acceso para facilitar el desarrollo de horarios en el microcontrolador y que el área física sea controlada por cada equipo y configurada en el computador.
- La selección del microcontrolador DS5000T ayudó al desarrollo del software y hardware. Así, para escribir en el microcontrolador el programa ensamblado se utilizó el puerto serial del computador, lo cual ahorra la necesidad de equipo especial para la programación del microcontrolador, esto gracias a su sistema de carga de programa a través del puerto serial utilizando como ayuda el programa Hyper Terminal (que viene como parte de Windows) y en hardware un chip MAX232. Por otro lado, la

incorporación del reloj calendario y memoria no volátil dentro del microcontrolador evitó el desarrollo de hardware que maneje estos elementos externamente.

- El reloj calendario del microcontrolador utiliza dos dígitos para el manejo del año, pero en el desarrollo del sistema el microcontrolador utiliza para su trabajo solo la parte del horario, mas no fechas. Pero en el desarrollo del software del computador las fechas de almacenamiento de los accesos se realiza con fecha de año a cuatro dígitos factor que elimina el problema que se tiene respecto al advenimiento del año 2000.
- Esta tesis es un sistema general de control de accesos, pero realizando pequeños cambios en el software y manteniendo el hardware se puede cambiar la aplicación del equipo para realizar tareas de seguridad, como por ejemplo activación de una sirena ante cambios de las entradas de alarma, sistema de control de guardia.

V.2. RECOMENDACIONES.

- El sistema está diseñado para manejar hasta 32 equipos, sin embargo si una aplicación requiere más número de equipos a controlar se debería utilizar, hardware adicional para ampliar la red y además prever cambios en la forma de identificación de los equipos dentro del formato del protocolo de comunicación.
- Debido a los problemas que se tuvo al realizar la comunicación entre Labview y Access se recomendaría el desarrollo de esta aplicación en un programa como Visual Basic (o Visual Basic para Aplicaciones que viene con Microsoft Access).
- Cuando el equipo funciona en Stand Alone no se visualiza los cambios en las entradas de alarma por lo que se recomienda ubicar algún elemento indicador como un led o generar en el display algún mensaje de alarma.

- La tabla para almacenar la actividad del sistema de control de accesos desarrollada en la base de datos influye en el funcionamiento del sistema pues va almacenando información a través de registros y estos van creciendo en número hasta llegar a un punto en que se vuelve inmanejable, por ello se recomienda realizar una partición de tabla cada cierto tiempo y el desarrollo de rutinas de enlace para la generación de reportes.
- La comunicación serial desde el computador hacia el equipo se realiza a través del p^ortico dos, se recomendaría una solución que a través software del computador se pueda seleccionar el p^ortico a través del cual se realizará la comunicación.

WIKI
NEWS

ANEXO A:

HOJA DE DATOS DALLAS DS5000T

DALLAS

SEMICONDUCTOR

DS5000(T)

Soft Microcontroller Module

FEATURES

- 8-bit 8051 compatible Microcontroller adapts to task-at-hand:
 - 8 or 32K bytes of nonvolatile RAM for program and/or data memory storage
 - Initial downloading of software in end system via on-chip serial port
 - Capable of modifying its own program and/or data memory in end use
- Crashproof operation:
 - Maintains all nonvolatile resources for 10 years in the absence of V_{CC}
 - Power-fail reset
 - Early warning power-fail interrupt
 - Watchdog timer
- Software Security Feature:
 - Executes encrypted software to prevent unauthorized disclosure
- On-chip, full-duplex serial I/O ports
- Two on-chip timer/event counters
- 32 parallel I/O lines
- Compatible with industry standard 8051 instruction set and pinout
- Optional Permanently Powered Real-Time Clock (DS5000T)

PIN ASSIGNMENT

P1.0	1	40	V_{CC}
P1.1	2	39	P0.0 AD0
P1.2	3	38	P0.1 AD1
P1.3	4	37	P0.2 AD2
P1.4	5	36	P0.3 AD3
P1.5	6	35	P0.4 AD4
P1.6	7	34	P0.5 AD5
P1.7	8	33	P0.6 AD6
RST	9	32	P0.7 AD7
RXD P3.0	10	31	\overline{EA}
TXD P3.1	11	30	ALE
$\overline{INT0}$ P3.2	12	29	PSEN
$\overline{INT1}$ P3.3	13	28	P2.7 A15
T0 P3.4	14	27	P2.6 A14
T1 P3.5	15	26	P2.5 A13
\overline{WR} P3.6	16	25	P2.4 A12
\overline{RD} P3.7	17	24	P2.3 A11
XTAL2	18	23	P2.2 A10
XTAL1	19	22	P2.1 A9
GND	20	21	P2.0 A8

40-PIN ENCAPSULATED PACKAGE

DESCRIPTION

The DS5000(T) Soft Microcontroller Module is a fully 8051 compatible 8-bit CMOS microcontroller that offers "softness" in all aspects of its application. This is accomplished through the comprehensive use of nonvolatile technology to preserve all information in the absence of system V_{CC} . The internal program/data memory space is implemented using either 8K or

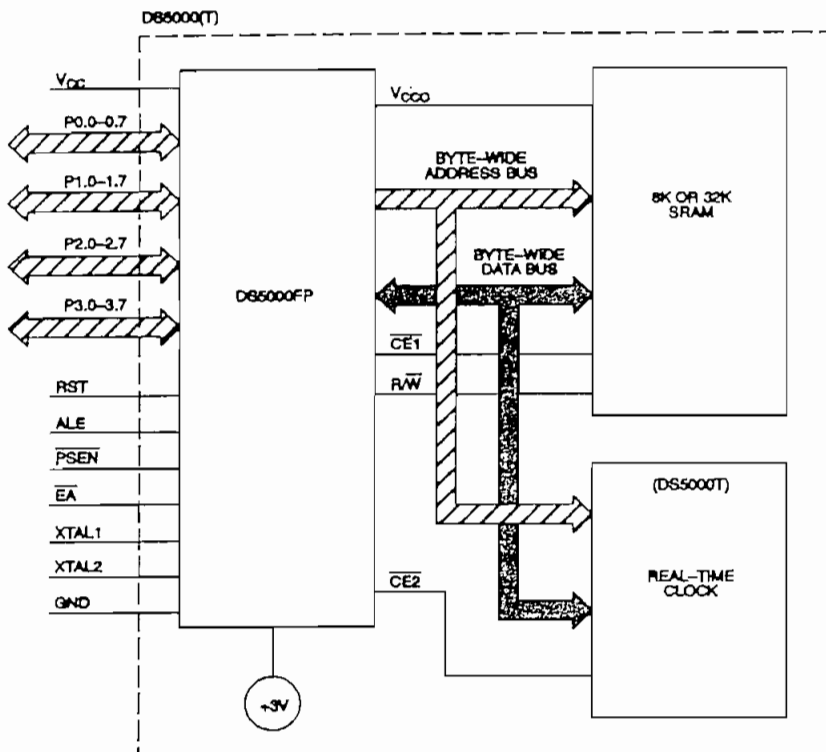
32K bytes of nonvolatile CMOS SRAM. Furthermore, internal data registers and key configuration registers are also nonvolatile. An optional real-time clock gives permanently powered timekeeping. The clock keeps time to a hundredth of a second using an on-board crystal.

ORDERING INFORMATION

PART NUMBER	RAM SIZE	MAX CRYSTAL SPEED	TIMEKEEPING?
DS5000-8-16	8K bytes	16 MHz	No
DS5000-32-16	32K bytes	16 MHz	No
DS5000-8-18	8K bytes	16 MHz	Yes
DS5000T-32-16	32K bytes	16 MHz	Yes

Operating information is contained in the User's Guide section of the Secure Microcontroller Data Book. This data sheet provides ordering information, pinout, and electrical specification.

DS5000(T) BLOCK DIAGRAM Figure 1



PIN DESCRIPTION

PIN NUMBER	DESCRIPTION
1-8	P1.0 – P1.7. General purpose I/O Port 1
9	RST – Active high reset input. A logic 1 applied to this pin will activate a reset state. This pin is pulled down internally so this pin can be left unconnected if not used.
10	P3.0 RXD. General purpose I/O port pin 3.0. Also serves as the receive signal for the on board UART. This pin should not be connected directly to a PC COM port.
11	P3.1 TXD. General purpose I/O port pin 3.1. Also serves as the transmit signal for the on board UART. This pin should not be connected directly to a PC COM port.
12	P3.2 INT0. General purpose I/O port pin 3.2. Also serves as the active low External Interrupt 0.
13	P3.3 INT1. General purpose I/O port pin 3.3. Also serves as the active low External Interrupt 1.
14	P3.4 T0. General purpose I/O port pin 3.4. Also serves as the Timer 0 input.
15	P3.5 T1. General purpose I/O port pin 3.5. Also serves as the Timer 1 input.
16	P3.6 WR. General purpose I/O port pin. Also serves as the write strobe for Expanded bus operation.
17	P3.7 RD. General purpose I/O port pin. Also serves as the read strobe for Expanded bus operation.
18, 19	XTAL2, XTAL1. Used to connect an external crystal to the internal oscillator. XTAL1 is the input to an inverting amplifier and XTAL2 is the output.
20	GND. Logic ground.
21-28	P2.0-P2.7. General purpose I/O Port 2. Also serves as the MSB of the Expanded Address bus.
29	PSEN – Program Store Enable. This active low signal is used to enable an external program memory when using the Expanded bus. It is normally an output and should be unconnected if not used. PSEN also is used to invoke the Bootstrap Loader. At this time, PSEN will be pulled down externally. This should only be done once the DS5000(T) is already in a reset state. The device that pulls down should be open drain since it must not interfere with PSEN under normal operation.
30	ALE – Address Latch Enable. Used to de-multiplex the multiplexed Expanded Address/Data bus on Port 0. This pin is normally connected to the clock input on a '373 type transparent latch. When using a parallel programmer, this pin also assumes the PROG function for programming pulses.
31	EA – External Access. This pin forces the DS5000(T) to behave like an 8031. No internal memory (or clock) will be available when this pin is at a logic low. Since this pin is pulled down internally, it should be connected to +5V to use NV RAM. In a parallel programmer, this pin also serves as V _{PP} for super voltage pulses.

PIN NUMBER	DESCRIPTION
32-39	P0.7-P0.0. General purpose I/O Port 0. This port is open-drain and can not drive a logic 1. It requires external pull-ups. Port 0 is also the multiplexed Expanded Address/Data bus. When used in this mode, it does not require pull-ups.
40	V _{CC} - +5 volts.

INSTRUCTION SET

The DS5000(T) executes an instruction set which is object code compatible with the industry standard 8051 microcontroller. As a result, software development packages which have been written for the 8051 are compatible with the DS5000(T), including cross-assemblers, high-level language compilers, and debugging tools.

A complete description for the DS5000(T) instruction set is available in the User's Guide section of the Secure Microcontroller Data Book.

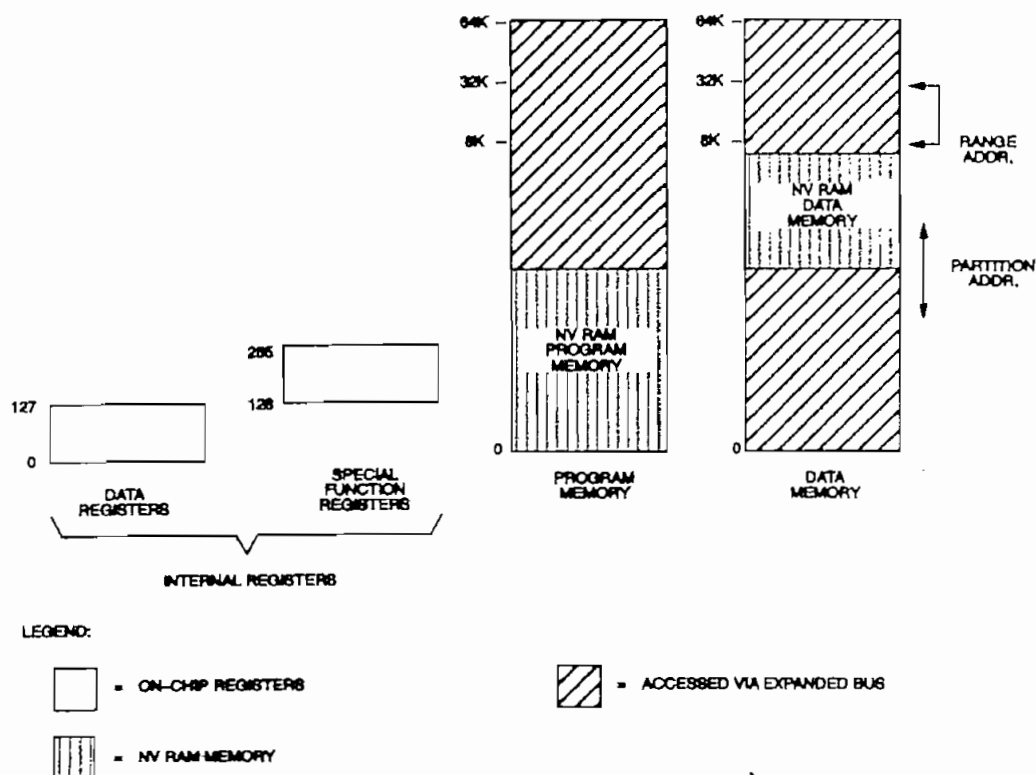
MEMORY ORGANIZATION

Figure 2 illustrates the address spaces which are accessed by the DS5000(T). As illustrated in the figure, separate address spaces exist for program and data

memory. Since the basic addressing capability of the machine is 16 bits, a maximum of 64K bytes of program memory and 64K bytes of data memory can be accessed by the DS5000(T) CPU. The 8K or 32K byte RAM area inside of the DS5000(T) can be used to contain both program and data memory.

The Real-time Clock (RTC) in the DS5000T is reached in the memory map by setting a SFR bit. The MCON.2 bit (ECE2) is used to select an alternate data memory map. While ECE2=1, all MOVXs will be routed to this alternate memory map. The real-time clock is a serial device that resides in this area. A full description of the RTC access and example software is given in the User's Guide section of the Secure Microcontroller Data Book. If the ECE2 bit is set on a DS5000 without a timekeeper, the MOVXs will simply go to a nonexistent memory. Software execution would not be affected otherwise.

DS5000(T) LOGICAL ADDRESS SPACES Figure 2



PROGRAM LOADING

The Program Load Modes allow initialization of the NV RAM Program/Data Memory. This initialization may be performed in one of two ways:

1. Serial Program Loading which is capable of performing Bootstrap Loading of the DS5000(T). This feature allows the loading of the application program to be delayed until the DS5000(T) is installed in the end system. Dallas Semiconductor strongly recommends the use of serial program loading because of its versatility and ease of use.
2. Parallel Program Load cycles which perform the initial loading from parallel address/data information presented on the I/O port pins. This mode is timing-set compatible with the 8751H microcontroller programming mode.

The DS5000(T) is placed in its Program Load configuration by simultaneously applying a logic 1 to the RST pin and forcing the PSEN line to a logic 0 level. Immediately following this action, the DS5000(T) will look for a parallel Program Load pulse, or a serial ASCII carriage return (0DH) character received at 9600, 2400, 1200, or 300 bps over the serial port.

The hardware configurations used to select these modes of operation are illustrated in Figure 3.

PROGRAM LOADING CONFIGURATIONS Figure 3

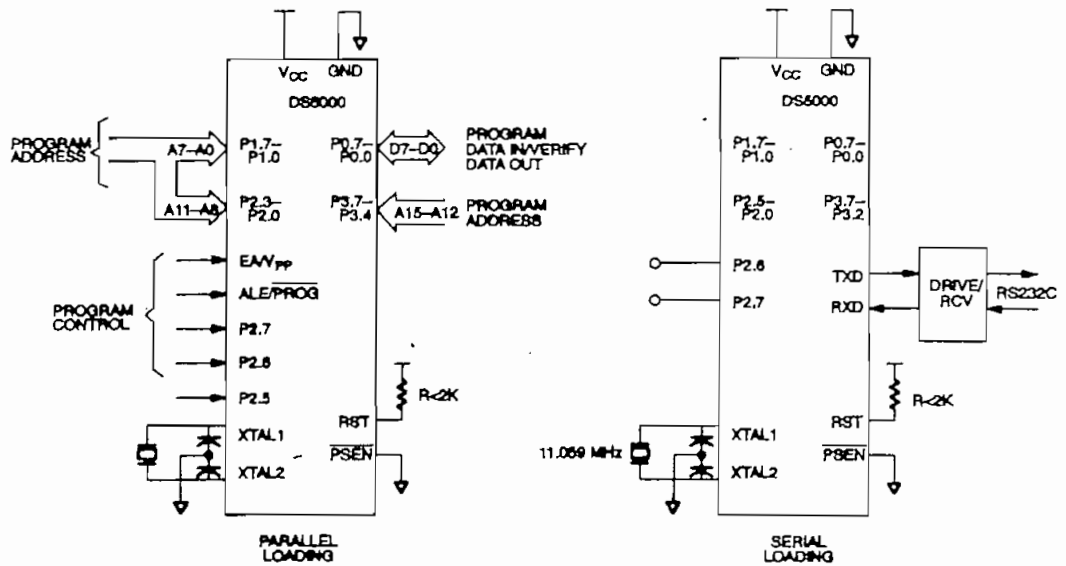


Table 1 summarizes the selection of the available Parallel Program Load cycles. The timing associated with these cycles is illustrated in the electrical specs.

SERIAL BOOTSTRAP LOADER

The Serial Program Load Mode is the easiest, fastest, most reliable, and most complete method of initially loading application software into the DS5000(T) nonvolatile RAM. Communication can be performed over a standard asynchronous serial communications port. A typical application would use a simple RS232C serial interface to program the DS5000(T) as a final production procedure. The hardware configuration which is required for the Serial Program Load mode is illustrated in Figure 3. Port pins 2.7 and 2.6 must be either open or pulled high to avoid placing the DS5000(T) in a parallel load cycle. Although an 11.0692 MHz crystal is shown in Figure 3, a variety of crystal frequencies and loader baud rates are supported, shown in Table 2. The serial loader is designed to operate across a three-wire interface from a standard UART. The receive, transmit, and ground wires are all that are necessary to establish communication with the DS5000(T).

The Serial Bootstrap Loader implements an easy-to-use command line interface which allows an application

program in an Intel hex representation to be loaded into and read back from the device. Intel hex is the typical format which existing 8051 cross-assemblers output. The serial loader responds to single character commands which are summarized below:

COMMAND	FUNCTION
C	Return CRC-16 checksum of embedded RAM
D	Dump Intel Hex File
F	Fill embedded RAM block with constant
K	Load 40-bit Encryption Key
L	Load Intel Hex File
R	Read MCON register
T	Trace (Echo) incoming Intel Hex data
U	Clear Security Lock
V	Verify Embedded RAM with incoming Intel Hex
W	Write MCON register
Z	Set Security Lock
P	Put a value to a port.
G	Get a value from a port.

PARALLEL PROGRAM LOAD CYCLES Table 1

MODE	RST	PSEN	PROG	EA	P2.7	P2.6	P2.5
Program	1	0	0	V _{PP}	1	0	X
Security Set	1	0	0	V _{PP}	1	1	X
Verify	1	X	X	1	0	0	X
Prog Expanded	1	0	0	V _{PP}	0	1	0
Verify Expanded	1	0	1	1	0	1	0
Prog MCON or Key registers	1	0	0	V _{PP}	0	1	1
Verify MCON registers	1	0	1	1	0	1	1

The Parallel Program Cycle is used to load a byte of data into a register or memory location within the DS5000(T). The Verify Cycle is used to read this byte back for comparison with the original loaded value to verify proper loading. The Security Set Cycle may be used to enable and the Software Security feature of the DS5000(T). One may also enter bytes for the MCON register or for the five encryption registers using the Program MCON cycle. When using this cycle, the absolute register address must be presented at Ports 1 and 2 as in the normal program cycle (Port 2 should be 00H). The MCON contents can likewise be verified using the Verify MCON cycle.

When the DS5000(T) first detects a Parallel Program Strobe pulse or a Security Set Strobe pulse while in the Program Load Mode following a Power On Reset, the internal hardware of the DS5000(T) is initialized so that an existing 4K byte program can be programmed into a DS5000(T) with little or no modification. This initialization automatically sets the Range Address for 8K bytes and maps the lowest 4K byte bank of Embedded RAM

as program memory. The next 4K bytes of Embedded RAM are mapped as Data Memory.

In order to program more than 4K bytes of program code, the Program/Verify Expanded cycles can be used. Up to 32K bytes of program code can be entered and verified. Note that the expanded 32K byte Program/Verify cycles take much longer than the normal 4K byte Program/Verify cycles.

A typical parallel loading session would follow this procedure. First, set the contents of the MCON register with the correct range and partition only if using expanded programming cycles. Next, the encryption registers can be loaded to enable encryption of the program/data memory (not required). Then, program the DS5000(T) using either normal or expanded program cycles and check the memory contents using Verify cycles. The last operation would be to turn on the security lock feature by either a Security Set cycle or by explicitly writing to the MCON register and setting MCON.0 to a 1.

SERIAL LOADER BAUD RATES FOR DIFFERENT CRYSTAL FREQUENCIES Table 2

CRYSTAL FREQ (MHz)	BAUD RATE					
	300	1200	2400	9600	19200	57600
14.7456		Y	Y	Y	Y	
11.0582	Y	Y	Y	Y	Y	Y
9.21600	Y	Y	Y	Y		
7.37280	Y	Y	Y	Y		
5.52960	Y	Y	Y	Y		
1.84320	Y	Y	Y	Y		

ADDITIONAL INFORMATION

A complete description for all operational aspects of the DS5000(T), is provided in the User's Guide section of the Secure Microcontroller Data Book.

DEVELOPMENT SUPPORT

Dallas Semiconductor offers a kit package for developing and testing user code. The DS5000TK Evaluation

Kit allows the user to download Intel hex formatted code directly to the DS5000(T) from a PC-XT/AT or compatible computer. The kit consists of a DS5000T-32, an interface pod, demo software, and an RS232 connector that attaches to the COM1 or COM2 serial port of a PC. See the Development Tools section of the Secure Microcontroller Data Book for further details.

ABSOLUTE MAXIMUM RATINGS*

Voltage on Any Pin Relative to Ground
 Operating Temperature
 Storage Temperature
 Soldering Temperature

-0.3V to +7.0V
 0°C to 70°C
 -40°C to +70°C
 260°C for 10 seconds

* This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operation sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods of time may affect reliability.

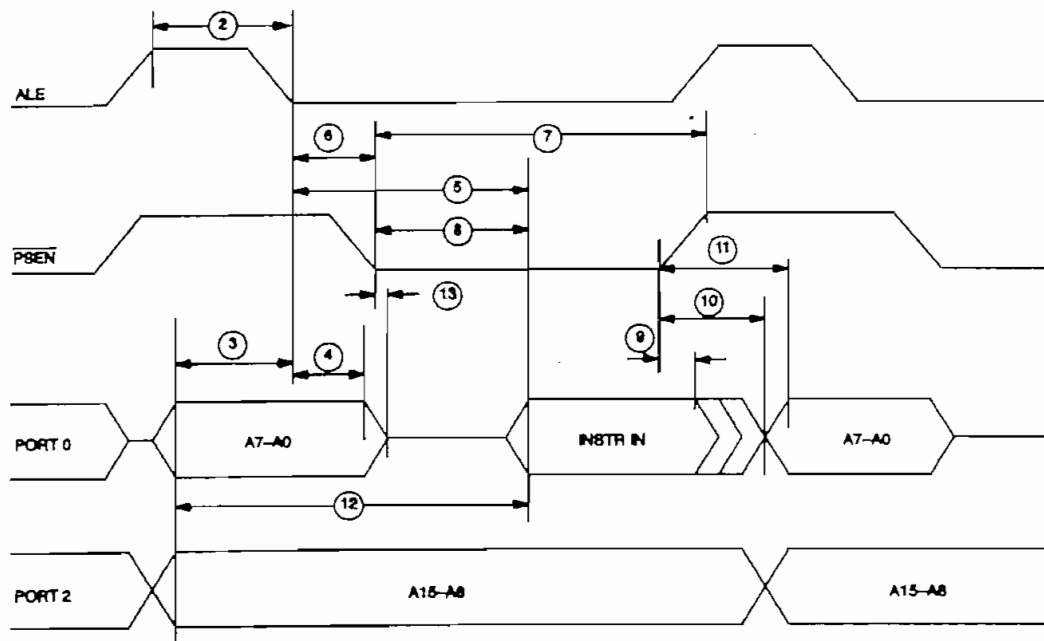
DC CHARACTERISTICS(t_A = 0°C to 70°C; V_{CC} = 5V ± 5%)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Input Low Voltage	V _{IL}	-0.3		0.8	V	1
Input High Voltage	V _{IH1}	2.0		V _{CC} + 0.3	V	1
Input High Voltage RST, XTAL1	V _{IH2}	3.5		V _{CC} + 0.3	V	1
Output Low Voltage @ I _{OL} = 1.6 mA (Ports 1, 2, 3)	V _{OL1}		0.15	0.45	V	
Output Low Voltage @ I _{OL} = 3.2 mA (Ports 0, ALE, PSEN)	V _{OL2}		0.15	0.45	V	1
Output High Voltage @ I _{OH} = 80 μA (Ports 1, 2, 3)	V _{OH1}	2.4	4.8		V	1
Output High Voltage @ I _{OH} = 400 μA (Ports 0, ALE, PSEN)	V _{OH2}	2.4	4.8		V	1
Input Low Current V _{IN} = 0.45V (Ports 1, 2, 3)	I _{IL}			-60	μA	
Transition Current; 1 to 0 V _{IN} = 2.0V (Ports 1, 2, 3)	I _{TL}			-500	μA	
Input Leakage Current 0.45 < V _{IN} < V _{CC} (Port 0)	I _L			±10	μA	
RST, EA Pulldown Resistor	R _{RE}	40		125	KΩ	
Stop Mode Current	I _{SM}			80	μA	4
Power Fail Warning Voltage	V _{PPW}	4.15	4.6	4.75	V	1
Minimum Operating Voltage	V _{CCmin}	4.05	4.5	4.65	V	1
Programming Supply Voltage (Parallel Program Mode)	V _{PP}	12.5		13	V	1
Program Supply Current	I _{PP}		15	20	mA	
Operating Current DS5000-8K @ 8 MHz DS5000-32K @ 12 MHz DS5000T-32-16 @ 16 MHz	I _{CC}		25.2 35.7 45.6	43 48 54	mA	2
Idle Mode Current @ 12 MHz	I _{CC}		4.5	6.2	mA	3

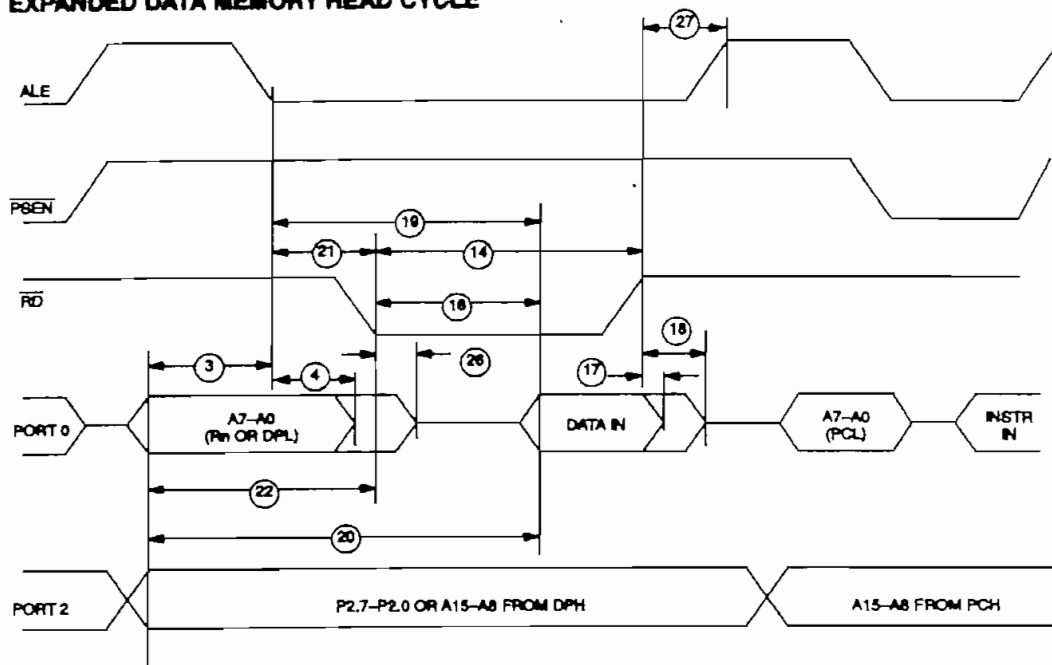
AC CHARACTERISTICS
EXPANDED BUS MODE TIMING SPECIFICATIONS
(t_A = 0°C to 70°C; V_{CC} = 5V ± 5%)

#	PARAMETER	SYMBOL	MIN	MAX	UNITS
1	Oscillator Frequency	1/CLK	1.0	16	MHz
2	ALE Pulse Width	t _{ALPW}	2t _{CLK} -40		ns
3	Address Valid to ALE Low	t _{AVALL}	t _{CLK} -40		ns
4	Address Hold After ALE Low	t _{AVAHV}	t _{CLK} -35		ns
5	ALE Low to Valid Instr. In ● 12 MHz ● 16 MHz	t _{ALLVI}		4t _{CLK} -150 4t _{CLK} -90	ns ns
6	ALE Low to PSEN Low	t _{ALLPSL}	t _{CLK} -25		ns
7	PSEN Pulse Width	t _{PPW}	3t _{CLK} -35		ns
8	PSEN Low to Valid Instr. In ● 12 MHz ● 16 MHz	t _{PSLVI}		3t _{CLK} -150 3t _{CLK} -90	ns ns
9	Input Instr. Hold after PSEN Going High	t _{PSIV}	0		ns
10	Input Instr. Float after PSEN Going High	t _{PSIX}		t _{CLK} -20	ns
11	Address Hold after PSEN Going High	t _{PSAHV}	t _{CLK} -8		ns
12	Address Valid to Valid Instr. In ● 12 MHz ● 16 MHz	t _{AVVI}		5t _{CLK} -150 5t _{CLK} -90	ns ns
13	PSEN Low to Address Float	t _{PSLAZ}	0		ns
14	RD Pulse Width	t _{RDPW}	5t _{CLK} -100		ns
15	WR Pulse Width	t _{WRPW}	5t _{CLK} -100		ns
16	RD Low to Valid Data In ● 12 MHz ● 16 MHz	t _{RDLVD}		5t _{CLK} -165 5t _{CLK} -105	ns ns
17	Data Hold after RD High	t _{RDHVD}	0		ns
18	Data Float after RD High	t _{RDHVZ}		2t _{CLK} -70	ns
19	ALE Low to Valid Data In ● 12 MHz ● 16 MHz	t _{ALLVD}		8t _{CLK} -150 8t _{CLK} -90	ns ns
20	Valid Addr. to Valid Data In ● 12 MHz ● 16 MHz	t _{AVVD}		9t _{CLK} -165 9t _{CLK} -105	ns ns
21	ALE Low to RD or WR Low	t _{ALLRDL}	3t _{CLK} -50	3t _{CLK} +50	ns
22	Address Valid to RD or WR Low	t _{AVRDL}	4t _{CLK} -130		ns
23	Data Valid to WR Going Low	t _{VDWRL}	t _{CLK} -80		ns
24	Data Valid to WR High ● 12 MHz ● 16 MHz	t _{VDWRH}	7t _{CLK} -150 7t _{CLK} -90		ns ns
25	Data Valid after WR High	t _{WRHVD}	t _{CLK} -50		ns
26	RD Low to Address Float	t _{RDLAZ}		0	ns
27	RD or WR High to ALE High	t _{RDHALH}	t _{CLK} -40	t _{CLK} +50	ns

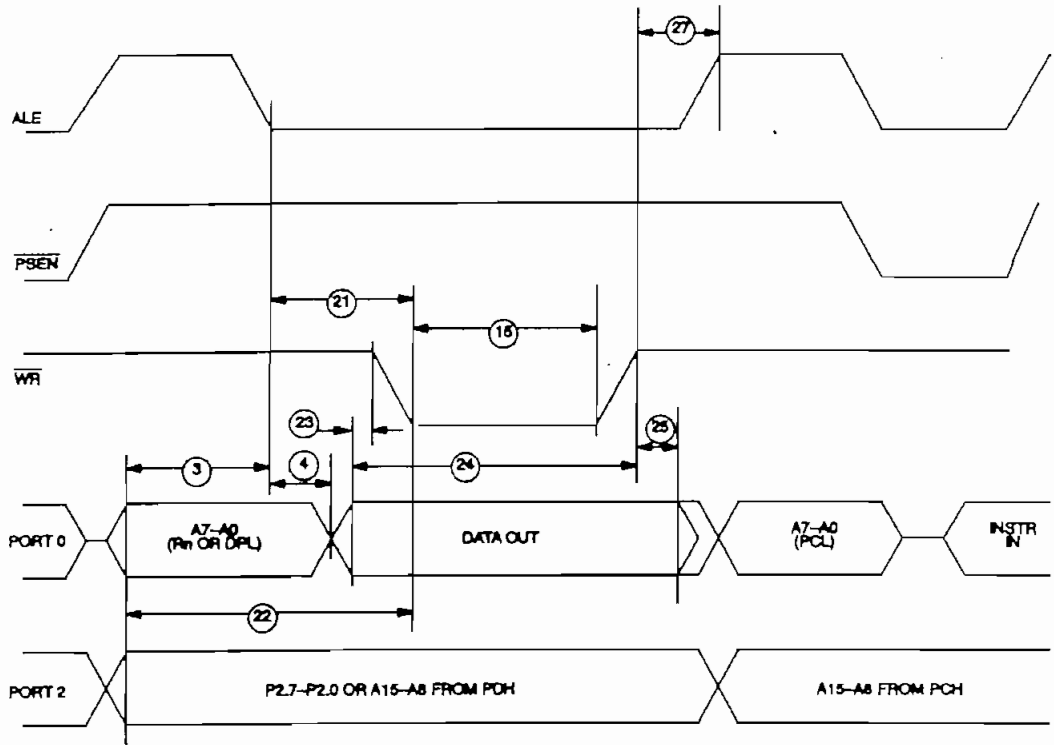
EXPANDED PROGRAM MEMORY READ CYCLE



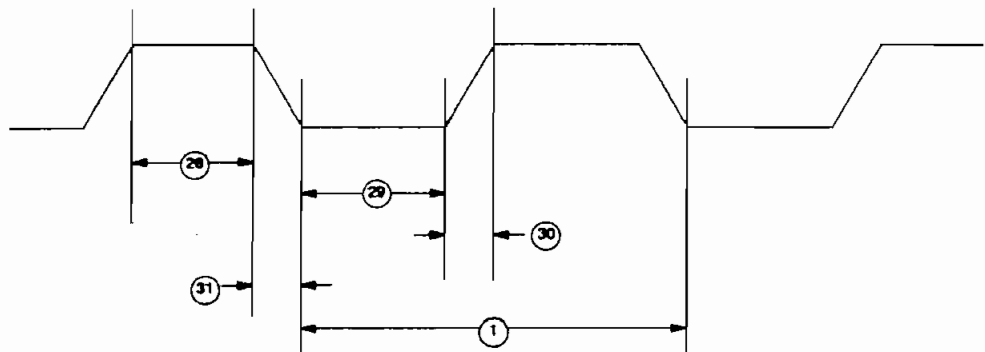
EXPANDED DATA MEMORY READ CYCLE



EXPANDED DATA MEMORY WRITE CYCLE



EXTERNAL CLOCK TIMING



AC CHARACTERISTICS (cont'd)
EXTERNAL CLOCK DRIVE

($t_A = 0^\circ\text{C to } 70^\circ\text{C}$; $V_{CC} = 5V \pm 5\%$)

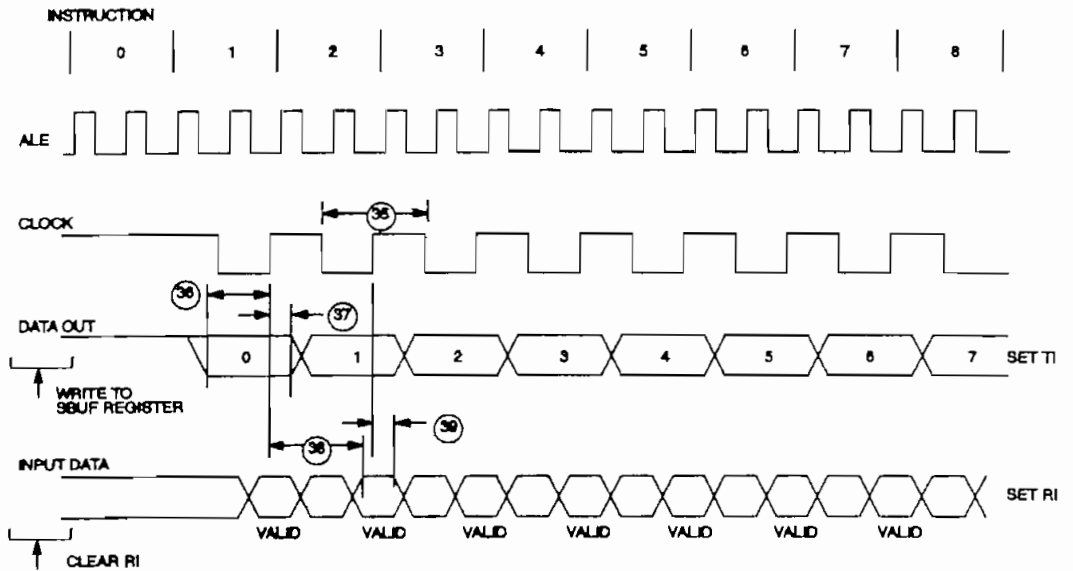
#	PARAMETER	SYMBOL	MIN	MAX	UNITS
28	External Clock High Time @ 12 MHz @ 16 MHz	t_{CLKHPW}	20 15		ns ns
29	External Clock Low Time @ 12 MHz @ 16 MHz	t_{CLKLPW}	20 15		ns ns
30	External Clock Rise Time @ 12 MHz @ 16 MHz	t_{CLKR}		20 15	ns ns
31	External Clock Fall Time @ 12 MHz @ 16 MHz	t_{CLKF}		20 15	ns ns

AC CHARACTERISTICS (cont'd)
SERIAL PORT TIMING - MODE 0

($t_A = 0^\circ\text{C to } 70^\circ\text{C}$; $V_{CC} = 5V \pm 5\%$)

#	PARAMETER	SYMBOL	MIN	MAX	UNITS
35	Serial Port Cycle Time	t_{SPCLK}	$12t_{CLK}$		μs
36	Output Data Setup to Rising Clock Edge	t_{DOCH}	$10t_{CLK} - 133$		ns
37	Output Data Hold after Rising Clock Edge	t_{CHDO}	$2t_{CLK} - 117$		ns
38	Clock Rising Edge to Input Data Valid	t_{CHDV}		$10t_{CLK} - 133$	ns
39	Input Data Hold after Rising Clock Edge	t_{CHDV}	0		ns

SERIAL PORT TIMING - MODE 0

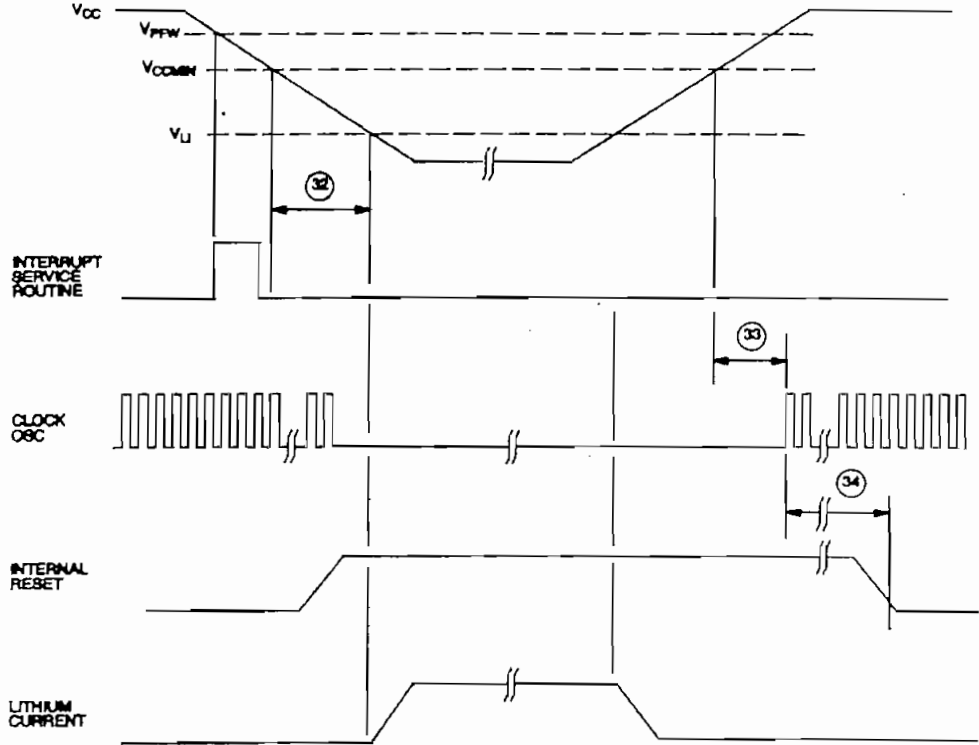


AC CHARACTERISTICS (cont'd)
POWER CYCLING TIMING

($t_A = 0^\circ\text{C}$ to 70°C ; $V_{CC} = 5V \pm 5\%$)

#	PARAMETER	SYMBOL	MIN	MAX	UNITS
32	Slew Rate from V_{CCmin} to 3.3V	t_F	40		μs
33	Crystal Start up Time	t_{CSU}		(note 5)	
34	Power On Reset Delay	t_{POR}		21504	t_{CLK}

POWER CYCLE TIMING



AC CHARACTERISTICS (cont'd)

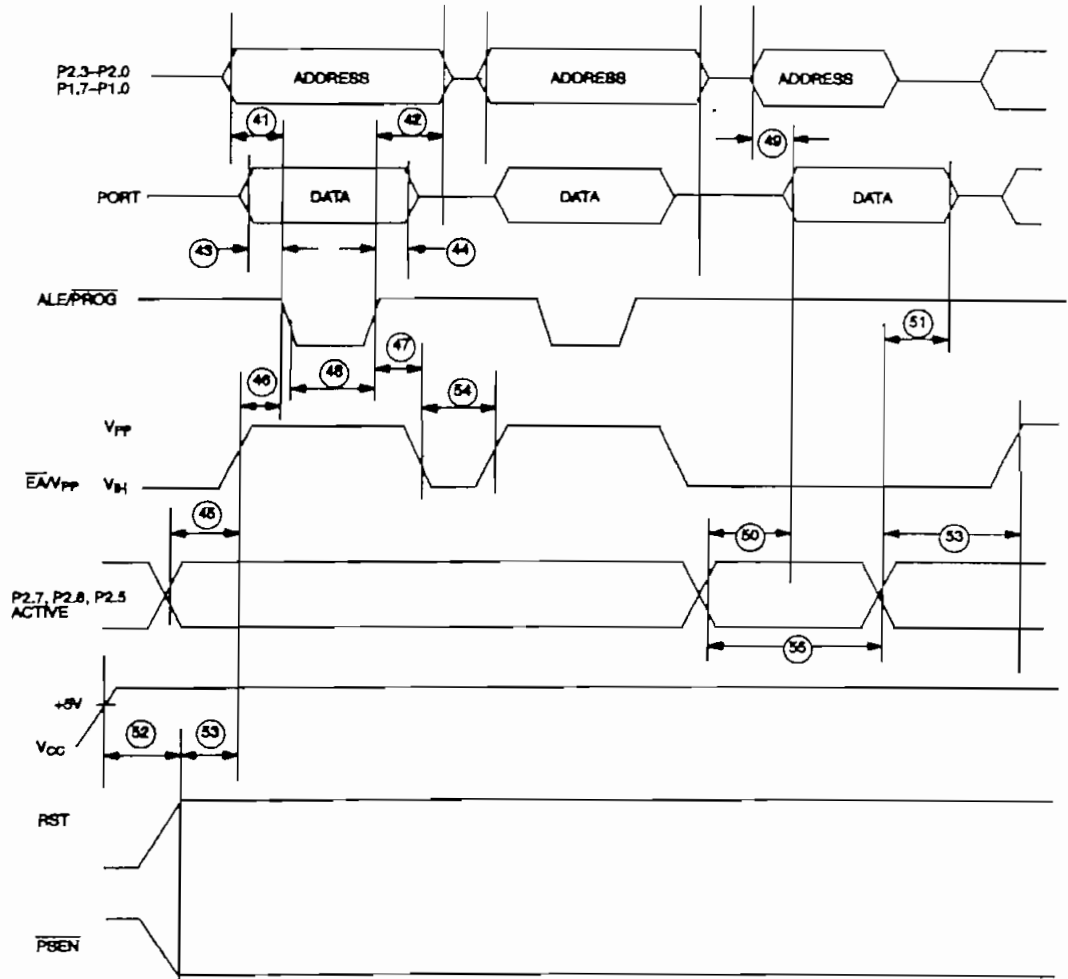
PARALLEL PROGRAM LOAD TIMING

 $(t_A = 0^\circ\text{C to } 70^\circ\text{C}; V_{CC} = 5V \pm 5\%)$

#	PARAMETER	SYMBOL	MIN	MAX	UNITS
40	Oscillator Frequency	$1/t_{CLK}$	1.0	12.0	MHz
41	Address Setup to PROG Low	t_{AVPRL}	0		
42	Address Hold after PROG High	t_{PRHUV}	0		
43	Data Setup to PROG Low	t_{DVPRL}	0		
44	Data Hold after PROG High	t_{PRHDV}	0		
45	P2.7, 2.8, 2.5 Setup to V_{pp}	t_{P27HVP}	0		
46	V_{pp} Setup to PROG Low	t_{VPHPRL}	0		
47	V_{pp} Hold after PROG Low	t_{PRHVPL}	0		
48	PROG Width Low	t_{PRW}	2400		t_{CLK}
49	Data Output from Address Valid	t_{AVDV}		48 1800*	t_{CLK}
50	Data Output from P2.7 Low	t_{DVP27L}		48 1800*	t_{CLK}
51	Data Float after P2.7 High	t_{P27HDZ}	0	48 1800*	t_{CLK}
52	Delay to Reset/PSEN Active after Power On	t_{PORPV}	21504		t_{CLK}
53	Reset/PSEN Active (or Verify Inactive) to V_{pp} High	t_{RAVPH}	1200		t_{CLK}
54	V_{pp} Inactive (Between Program Cycles)	t_{VPPPC}	1200		t_{CLK}
55	Verify Active Time	t_{VFT}	48 2400*		t_{CLK}

* Second set of numbers refers to expanded memory programming up to 32K bytes.

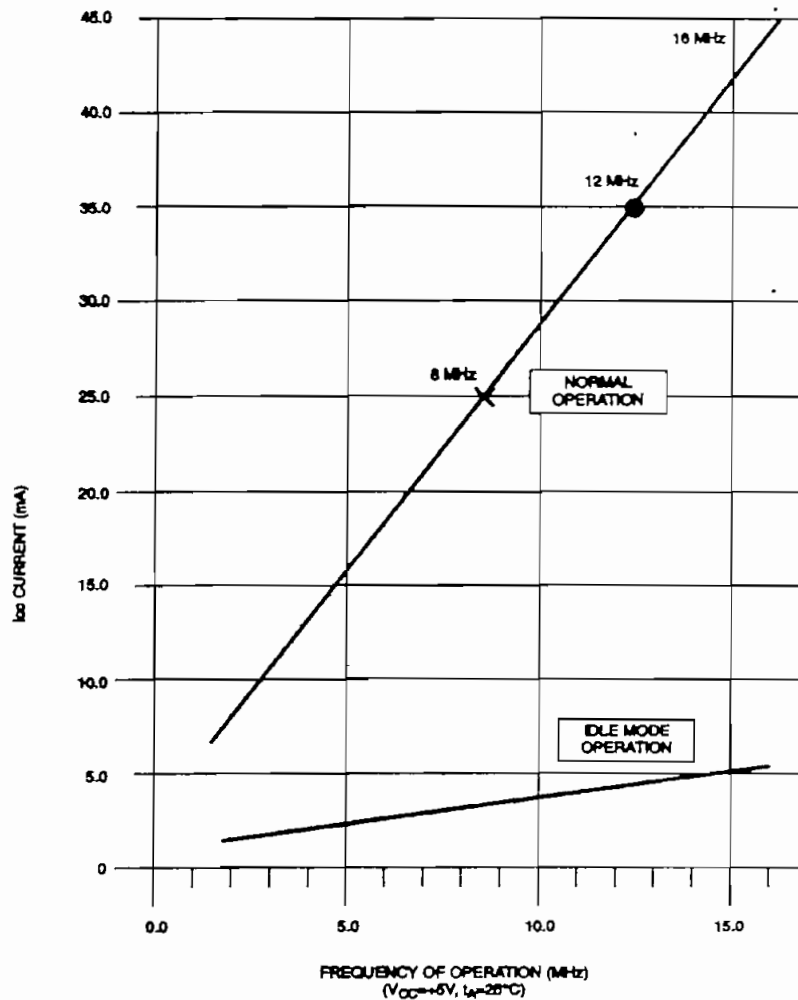
PARALLEL PROGRAM LOAD TIMING



CAPACITANCE

(test frequency = 1 MHz; $t_A = 25^\circ\text{C}$)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Output Capacitance	C_0			10	pF	
Input Capacitance	C_1			10	pF	

DS5000(T) TYPICAL I_{CC} VS. FREQUENCY

Normal operation is measured using:

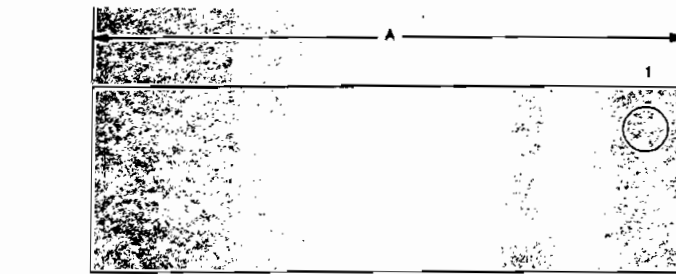
- 1) External crystals on XTAL1 and 2
- 2) All port pins disconnected
- 3) RST=0 volts and EA= V_{CC}
- 4) Part performing endless loop writing to internal memory.

Idle mode operation is measured using:

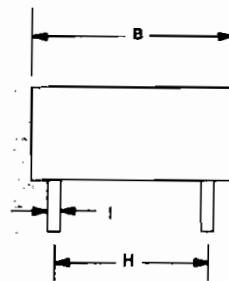
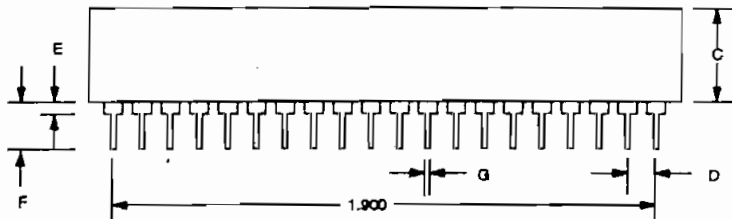
- 1) External clock source at XTAL1; XTAL2 floating
- 2) All port pins disconnected
- 3) RST=0 volts and EA= V_{CC}
- 4) Part set in IDLE mode by software.

NOTES:

1. All voltages are referenced to ground.
2. Maximum operating I_{CC} is measured with all output pins disconnected; XTAL1 driven with t_{CLKR} , $t_{CLKF}=10$ ns, $V_{IL} = 0.5V$; XTAL2 disconnected; $\overline{EA} = RST = PORT0 = V_{CC}$.
3. Idle mode I_{CC} is measured with all output pins disconnected; XTAL1 driven with t_{CLKR} , $t_{CLKF} = 10$ ns, $V_{IL} = 0.5V$; XTAL2 disconnected; $\overline{EA} = PORT0 = V_{CC}$, $RST = V_{SS}$.
4. Stop mode I_{CC} is measured with all output pins disconnected; $\overline{EA} = PORT0 = V_{CC}$; XTAL2 not connected; $RST = V_{SS}$.
5. Crystal start up time is the time required to get the mass of the crystal into vibrational motion from the time that power is first applied to the circuit until the first clock pulse is produced by the on-chip oscillator. The user should check with the crystal vendor for the worst case spec on this time.

PACKAGE DRAWING

DIM	INCHES	
	MIN	MAX
A IN.	2.080	2.100
B IN.	0.680	0.700
C IN.	0.290	0.325
D IN.	0.090	0.110
E IN.	0.030	0.060
F IN.	0.145	0.185
G IN.	0.016	0.020
H IN.	0.590	0.610
I IN.	0.009	0.015



DATA SHEET REVISION SUMMARY

The following represent the key differences between the dates 07/20/95 to 07/24/96 of the DS5000(T) data sheet. Please review this summary carefully.

1. Correct Figure 3 to show RST active high.
2. Add Data Sheet Revision Summary.

ANEXO B:
MANUAL DE USUARIO

B.1. PANTALLA PRINCIPAL.

En la Figura B-1 se muestra la pantalla principal del programa. Consta básicamente de dos áreas, la primera es una barra de herramientas que permite la configuración de equipos, generación de reportes y ayudas del software, la segunda es una pantalla que indica la actividad del sistema a través de mensajes como son: control de accesos en sí (tarjetas reconocidas, no reconocidas y accesos denegados), cambio de estados de alarma, cambios de configuración de equipos.

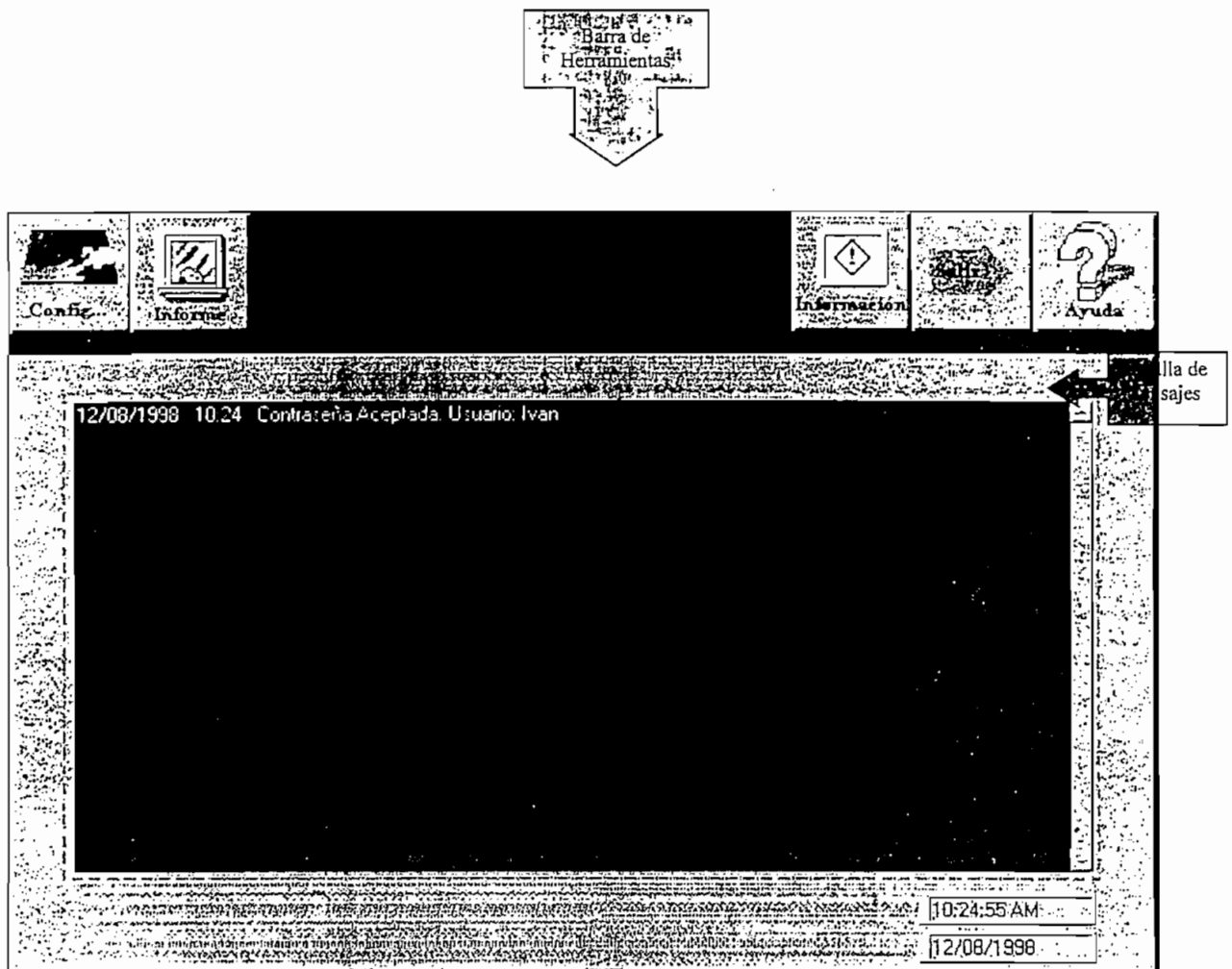


FIGURA B-1. PANTALLA PRINCIPAL EN LABVIEW

PANTALLA DE MENSAJES.

Esta sección de la pantalla principal es una interfaz de visualización del estado de la actividad del sistema.

Los mensajes se componen de fecha y hora de generación, el mensaje propiamente dicho y adicionalmente un código de color específico para cada mensaje, un ejemplo se muestra en la Figura III-10 (sin observarse el color del mensaje).

BARRA DE HERRAMIENTAS.

La barra de herramientas es el mecanismo mediante el cual se puede acceder a las diferentes características de administración del programa.

Tiene tres niveles denominados: barra principal, barra de configuración y barra de reportes.

Además de tres botones fijos: ayuda, información y salir, como lo indica la figura siguiente.



FIGURA B-2. NIVELES DE LA BARRA DE HERRAMIENTA

SEGURIDAD.

Cuando un operador quiere ingresar a configurar el sistema o desea salir, el software presenta una pantalla de seguridad que solicita el nombre del operador y su contraseña respectiva para poder realizar las tareas mencionadas anteriormente. El operador tiene tres oportunidades para ingresar su nombre y/o contraseña autorizada.

La definición de operadores autorizados se realiza en el nivel de barra configuración (ver ítem Configuración Operador.).

Esta pantalla siempre genera un mensaje dentro de la sección **Pantalla de Mensajes**.

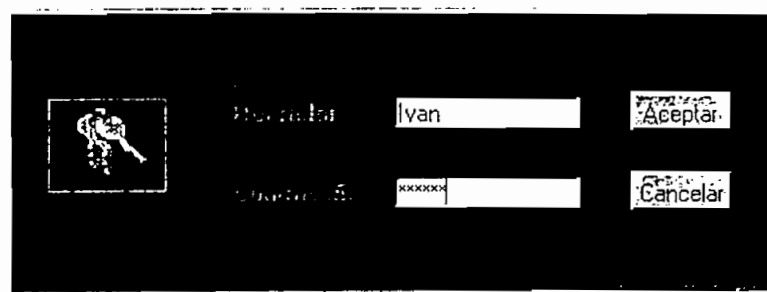


FIGURA B-3. PANTALLA INGRESO DE CONTRASEÑA.

B.2. PANTALLA CONFIGURACIÓN EQUIPO (NIVEL CONFIGURACIÓN).

Se puede observar en la Figura B-2 que este nivel presenta tres opciones: Configuración de Equipo, Configuración de Usuarios (Tarjetas) disponibles para los Equipos y Configuración de Operadores autorizados.

La lógica de cómo se debe configurar del sistema es la siguiente: primero configurar los operadores autorizados para navegar dentro del sistema, en segundo lugar definir las tarjetas de usuario que estarán disponibles para la configuración de los equipos y por último

se debe realizar la configuración de los equipos y sus respectivas características (ver ítem Configuración Equipo.).

A continuación se detalla cada una de las pantallas a las que se ingresa cuando se presiona la opción respectiva.

CONFIGURACIÓN OPERADOR.

La pantalla configuración de operador (Figura B-4) presenta tres opciones: cambiar, borrar y añadir operador. Además presenta un control en el cual se puede seleccionar los operadores autorizados. Los controles Operador y Contraseña permiten ingresar los datos desde el teclado.

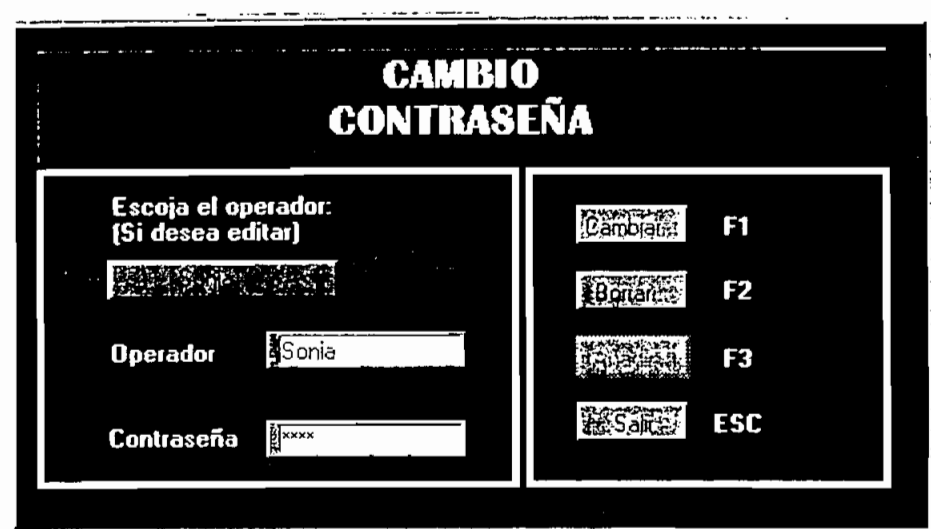


FIGURA B-4. PANTALLA CONFIGURACIÓN OPERADOR.

Cada vez que se presiona una de las opciones se presenta una pantalla que solicita la verificación de la contraseña para poder realizar el cambio respectivo.

Si se quiere duplicar el nombre de un operador, el programa sobrescribe los datos originales con los nuevos requeridos, el comando añadir en este caso no define un nuevo usuario en la base de datos.

CONFIGURACIÓN USUARIO.

De manera similar a la configuración de operador, esta pantalla (Figura B-5) modifica la tabla "Usuarios" dentro de la base de datos, por lo que se presenta las mismas tres opciones de configuración: eliminar, agregar y editar los datos del usuario. La base de programación es la misma que en configuración de operador, aunque la interfaz difiere un poco.

Los datos requeridos para configurar usuario son: Número de Tarjeta, Nombre, Apellido Paterno, Apellido Materno y Número Telefónico.

La Interfaz presenta dos botones que permiten moverse dentro de los diferentes registros de la tabla y son: Siguiente Registro y Anterior Registro. Además si se presiona el casillero donde se encuentra el número de registro actual aparece una pantalla que permite movilizarse a un determinado registro de manera directa.

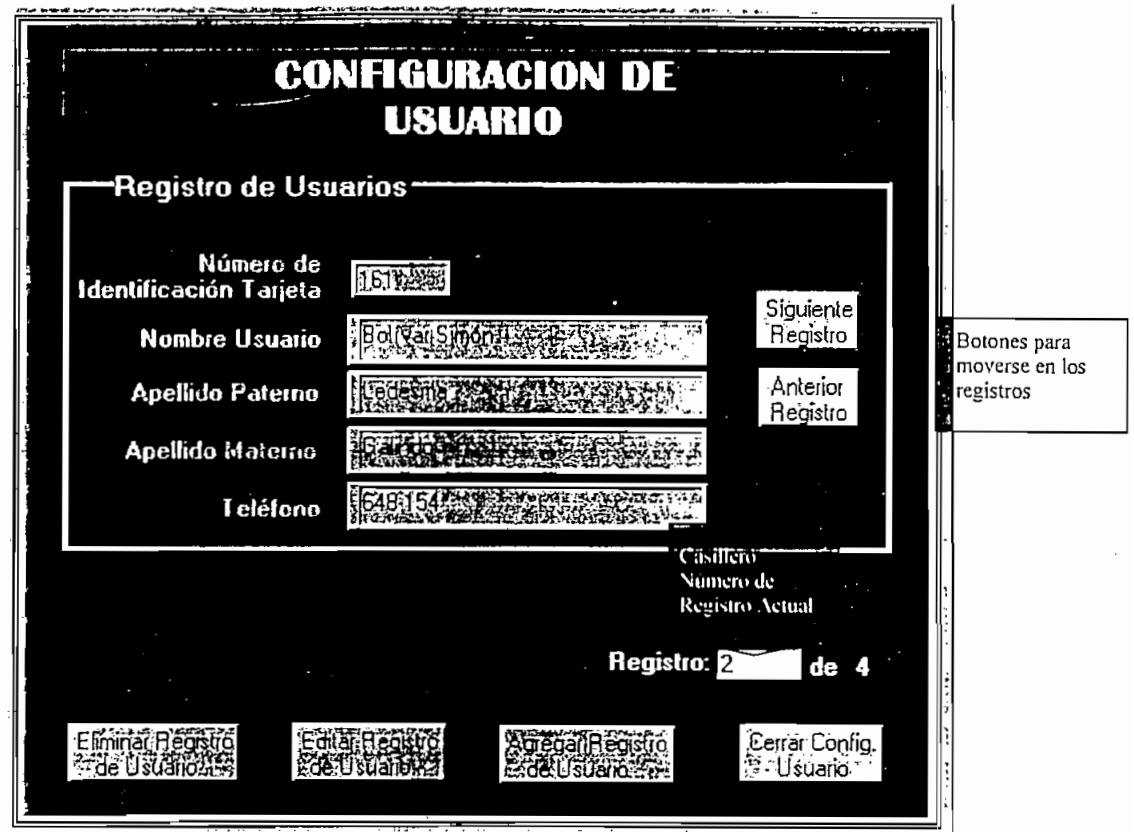


FIGURA B-5. PANTALLA CONFIGURACIÓN USUARIO.

CONFIGURACIÓN EQUIPO.

La configuración de equipo se realiza en dos partes: la primera consiste en seleccionar el equipo al que se va a configurar (Panel Previo) y en segundo lugar la configuración propiamente dicha (Panel Principal Configuración).

Panel Previo.

Esta pantalla presenta las siguientes opciones: seleccionar equipo y agregar equipo. La primera opción permite seleccionar el equipo al que se desea realizar las modificaciones y permite ingresar a la pantalla de configuración "Pantalla Principal Configuración Equipo" siempre y cuando se verifique la comunicación con el equipo. La segunda opción despliega una pantalla que solicita el número de equipo (dirección dentro de la red) y un nombre con

el que se identifica al mismo, esta pantalla verifica si la comunicación con el equipo existe para inicializar los datos del equipo, caso contrario el equipo no se agrega.

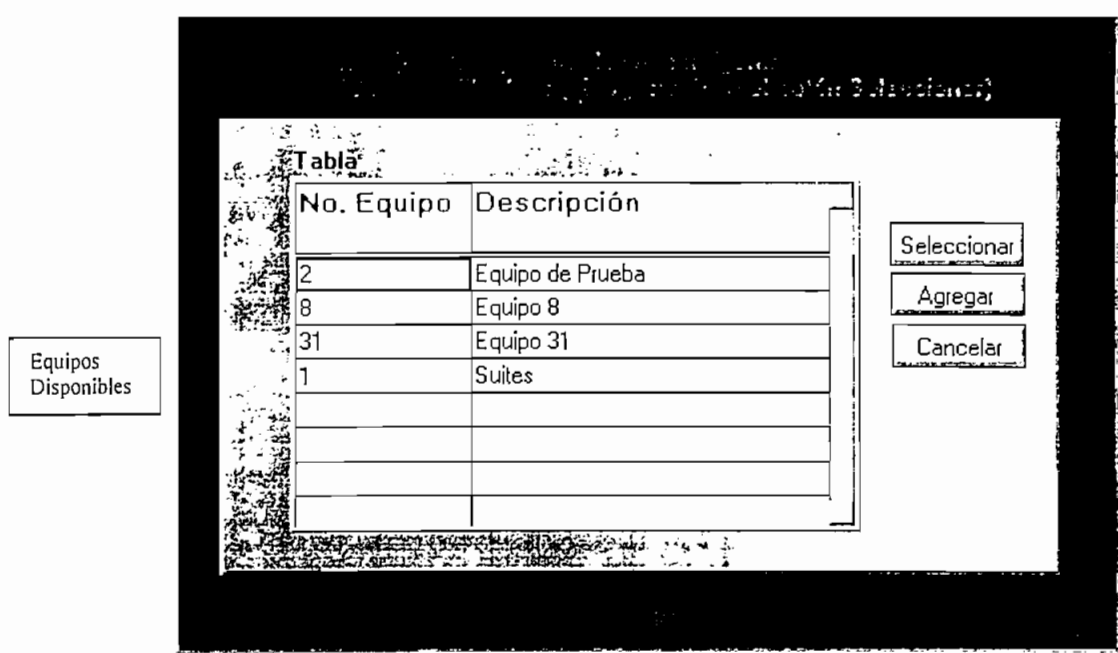


FIGURA B-6. PANTALLA PANEL PREVIO CONFIGURACIÓN EQUIPO.

Pantalla Principal Configuración Equipo.

A través de esta pantalla se puede realizar las siguientes tareas de configuración al equipo seleccionado: lectura de entradas de alarma, consulta e igualación de reloj calendario, definición de zonas de tiempo, programación de horarios de zonas de tiempo, cambio de descripción de puertas, configuración de monoestable de cerradura, definición de usuarios autorizados por equipo.



FIGURA B-7. PANTALLA PRINCIPAL DE CONFIGURACIÓN.

Como filosofía dentro de la opción de entradas de alarma y las pantallas de configuración que en adelante se detallarán, cuando existe una falla de comunicación aparecerá siempre un mensaje de “falla” (color rojo) en la pantalla durante aproximadamente dos segundos, y si el comando se ejecuta correctamente aparecerá un mensaje “OK” (color verde) del mismo intervalo de tiempo.

Además con cada tarea de comunicación se genera un mensaje que aparece en la Pantalla Principal sección Mensajes.

A continuación se detalla las pantallas para cada una de las opciones de configuración del equipo.

Pantalla Cambio Descripción de Puertas.

Esta pantalla genera un nombre descriptivo para cada puerta que controla el equipo.

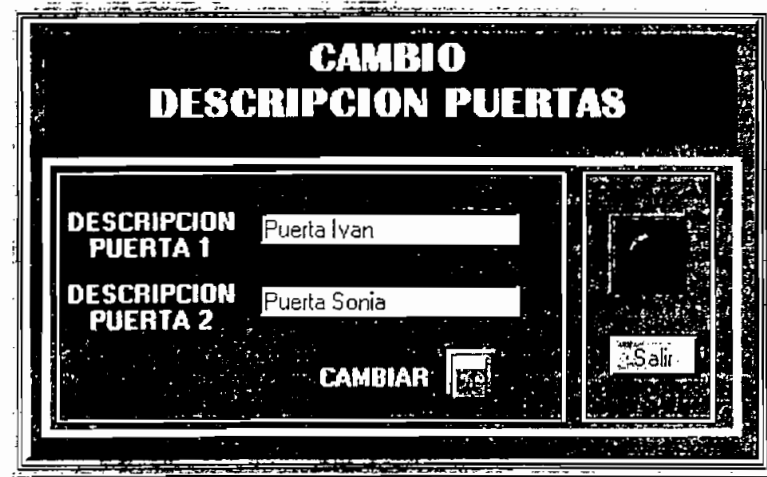


FIGURA B-8. PANTALLA CAMBIO DESCRIPCIÓN PUERTAS.

Pantalla Consulta e Igualación del Reloj Calendario.

Esta pantalla presenta dos opciones: consultar e igualar el reloj. Cuando se iguala el reloj calendario, se toma como base la fecha y hora del computador.

Esta pantalla de configuración es importante, pues de la sincronización entre el estado del reloj del equipo y del computador dependerá la correcta operación de control de accesos del sistema.

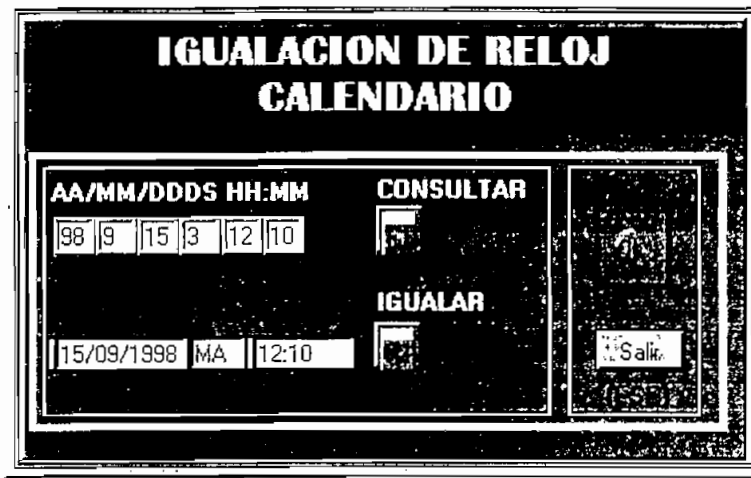


FIGURA B-9. PANTALLA CONSULTA E IGUALACIÓN DEL RELOJ-CALENDARIO.

Pantalla del Monoestable de la Cerradura.

Esta pantalla define el intervalo de tiempo (en segundos) de apertura de la cerradura. La interfaz gráfica es similar a la de la pantalla de Igualación del Reloj-Calendarario.



FIGURA B-10. PANTALLA CONFIGURACIÓN MONOESTABLE CERRADURA.

Pantalla Configuración de Zonas de Tiempo.

Esta pantalla esta compuesta de las siguientes partes: un control que permite escoger el día de la semana a configurar, una tabla que contiene la descripción de las zonas de tiempo y los horarios de programación de las mismas (según el día escogido).

El operador puede programar los horarios según las necesidades del sistema pero se tienen dos horarios especiales denominados: "Acceso 24 horas" y "No Acceso", el primer horario permite un acceso ilimitado durante el día especificado, mientras el segundo horario niega acceso durante todo el día. Por defecto todas las zonas de tiempo son inicializadas con un horario para Acceso las 24 horas durante todos los días.

En esta pantalla el operador también puede cambiar la Descripción de las zonas de Tiempo.

PROGRAMACION DE ZONAS DE TIEMPO		
Descripción	Minima	Maxima
Timezone 1	07:00	17:00
Timezone 2	07:00	18:00
Timezone 3	00:00	24:00
Timezone 4	00:00	24:00
Timezone 5	00:00	24:00
Timezone 6	00:00	24:00
Timezone 7	00:00	24:00
Timezone 8	00:00	24:00
Timezone 9	00:00	24:00
Timezone 10	00:00	24:00
Timezone 11	00:00	24:00
Timezone 12	00:00	24:00
Timezone 13	00:00	24:00
Timezone 14	00:00	24:00
Timezone 15	00:00	24:00
Timezone 16	00:00	24:00

Escoja el día de Programación

MARTES

Guardar Horario del Día (F1)

Salir (ESC)

NOTA:

	HORA MIN	HORA MAX	
Los cambios solo se ejecutaran si se ejecuta el comando "Guardar Horario del día"	24:00	00:00	Acceso 24 h
	00:00	24:00	No Acceso

FIGURA B-11. PANTALLA CONFIGURACIÓN ZONAS DE TIEMPO.

Si se realizó un cambio en la programación del día y no se ejecutó el comando Guardar (tecla F1) LOS CAMBIOS REALIZADOS SE PERDERÁN. Si el operador cambia de día la tabla se actualiza con los datos del día correspondiente.

Pantalla de Definición de Tarjetas Válidas para el Equipo.

Esta pantalla tiene la información para definir tarjetas que admitirá el equipo y asignar las zonas de tiempo a las que tendrá acceso (y que fueron definidos en la pantalla configuración de zonas de tiempo).

La pantalla consta de tres secciones: una tabla con los datos de las tarjetas, una lista de selección de las zonas de tiempo que pueden asignarse y una zona de botones para la configuración.

La tabla tiene cuatro columnas, cada columna tiene en la parte superior controles para poder configurar rápidamente el equipo. Las columnas son: número de usuario (en la memoria del equipo), código de tarjeta, nombre del usuario y puertas de acceso. La segunda y tercera columna están relacionadas, pues a cada código de tarjeta le corresponde un nombre de usuario, los controles de estas columnas permiten asignar los usuarios al equipo, las tarjetas que se pueden asignar son las que fueron definidas en la Pantalla de Configuración de Usuario. La última columna asigna la(s) puerta(s) que el usuario tiene acceso.

La lista de selección de las zonas de tiempo permite asignar a cual(es) zona(s) de tiempo tiene acceso el usuario. El equipo siempre tomará la unión de los horarios de las zonas de tiempo para el control de accesos.

Los botones para la configuración realizan dos tareas: desplazarse dentro de la tabla de las tarjetas y grabar los cambios efectuados.

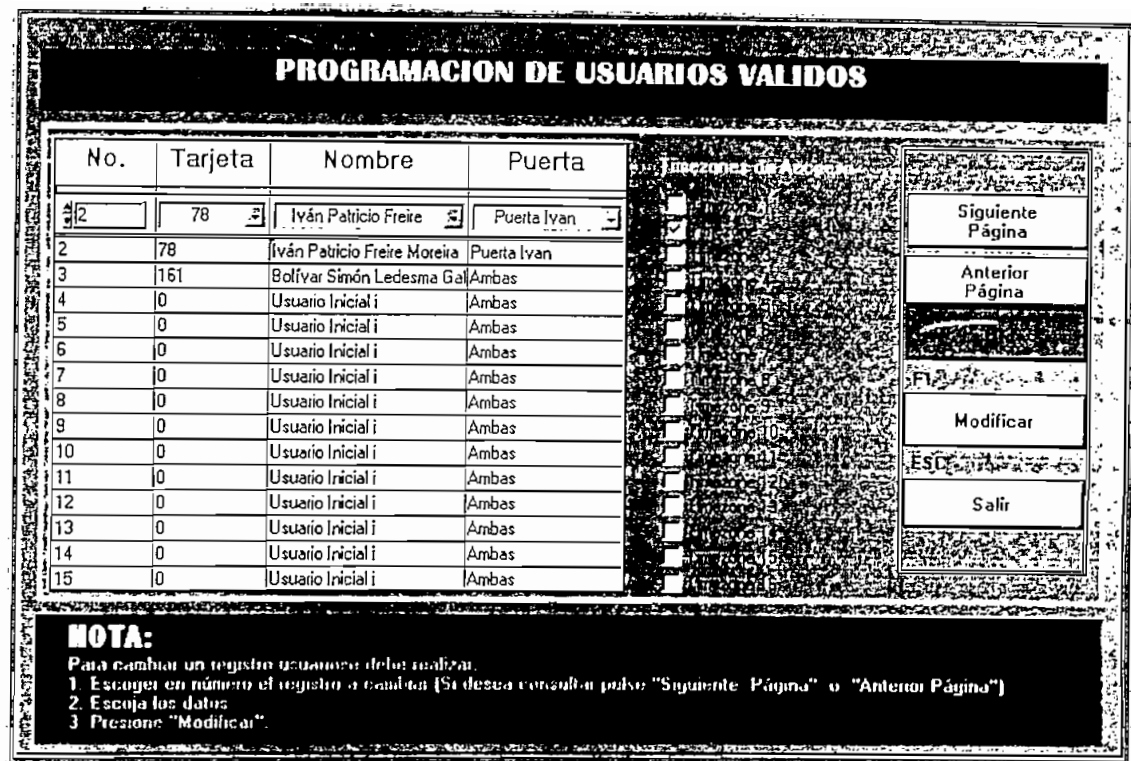


FIGURA B-12. PANTALLA DE DEFINICIÓN DE TARJETAS VALIDAS PARA EL EQUIPO.

B.3. PANTALLA REPORTES (NIVEL REPORTES).

La Barra de Herramientas en el nivel de reportes presenta dos opciones (ver Figura B-2): Reporte por Usuario y Reportes por Equipo.

Al seleccionar cualquiera de las dos opciones se presenta interfaces similares. En las cuales se selecciona los parámetros para realizar la consulta de los datos del control de accesos. Los datos obtenidos son clasificados sea por equipo o por usuario según la opción seleccionada en la barra de herramientas como se muestra en la Figura B-13.

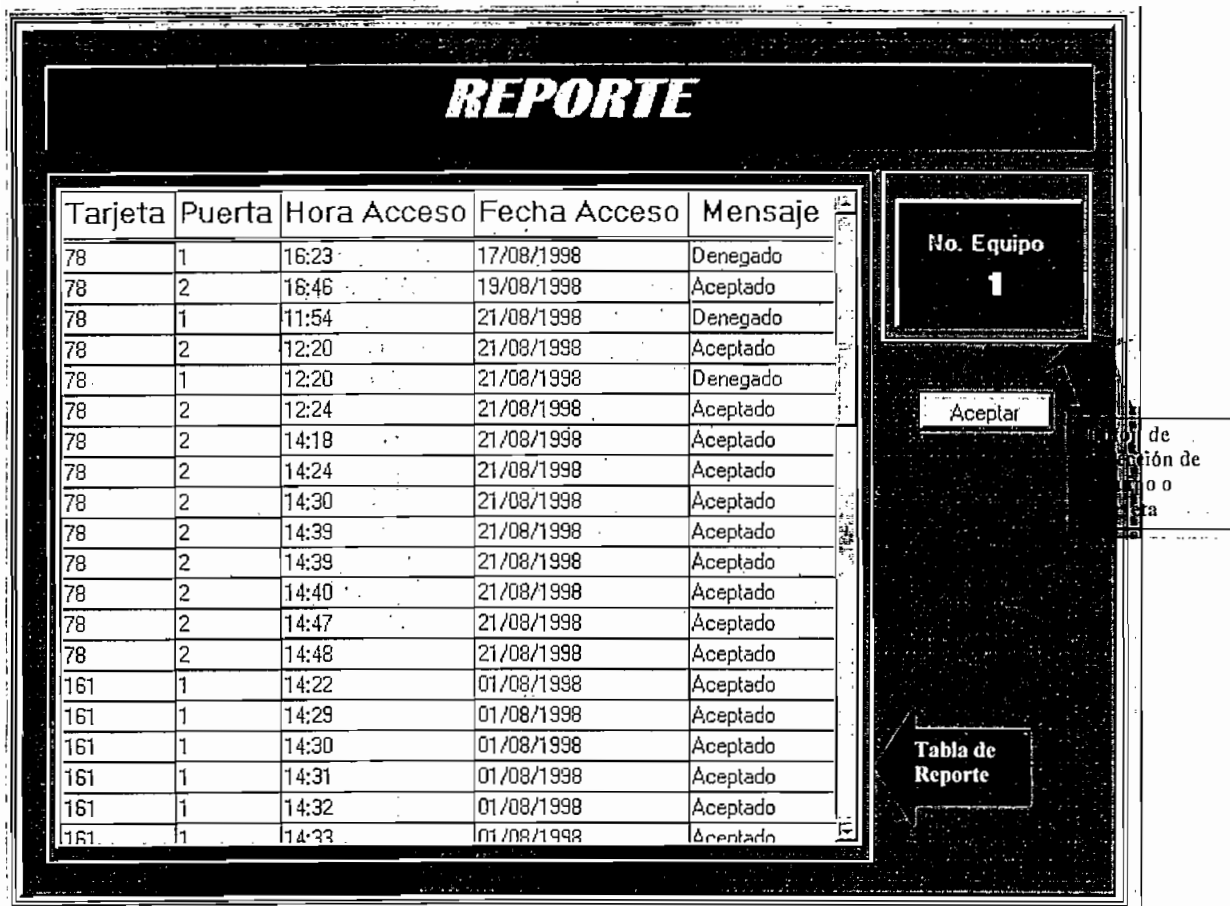


FIGURA B-13. PANTALLA DE PRESENTACIÓN DE REPORTES.

PANTALLA REPORTE POR USUARIO.

Esta pantalla permite visualizar información almacenada por el sistema de control de accesos. Se presentan las siguientes opciones:

- Selección de tarjetas se realiza a través de dos botones que activan dos opciones: todas las tarjetas o por tarjetas.
- Opciones de parámetros disponibles para el informe: por equipos y/o por fechas. Si selecciona por equipos se habilita un control que despliega la información almacenada en la tabla "Equipos" de la base de datos. Al seleccionar por fechas se habilitan dos controles para escribir el intervalo de tiempo. Se tiene una opción adicional que permite obtener información del día actual.

Reporte por Usuario

Tarjetas que aparecerán en el Reporte

Todas las Tarjetas

Seleccionar Tarjetas

78 (Juan Polanco Fresno Mo...
161 (Sofía Sanjón Ledesma...
455 (Cora María Lechón...
0 (Luisa Inés...)

Escoja por cual parámetro desea realizar el reporte

Por Equipo

Por Fechas

Seleccione Equipo

(Id. 2) Equipo de

Hoy

Reporte entre:
(Mes (2)/Día (2) /Año (4))

08/07/1998 y
09/07/1998

Aceptar **Cancelar**

* Si no se selecciona ningún ítem se genera un reporte con todos los equipos y los accesos generados hasta el día de hoy

FIGURA B-14. PANTALLA DE REPORTE POR USUARIO.

PANTALLA REPORTE POR EQUIPO.

Esta pantalla presenta una interfaz similar a la Pantalla de Reporte por Usuario (ver Figuras B-14 y B-15), presenta como opciones: la selección de equipos para la generación del informe, selección de la puerta (útil cuando se selecciona un solo equipo para el reporte), y las opciones de selección de tarjetas y fechas.

La selección de las opciones se la realiza de manera similar que en la Pantalla de Reporte por Usuarios.

Reporte de Accesos por Equipo

Tarjetas que aparecerán en el Reporte

Todos los Equipos

Seleccionar Equipos

Id. 01 Equipo de Fr...
 Id. 01 Equipo B...
 Id. 01 Equipo C...
 Id. 01 Equipo...

Escoja por cual parámetro desea realizar el reporte

Por Tarjetas

Por Fechas

Hoy

Reporte entre: [Mes/Día/Año (4 dígitos)]

y

Todas las Tarjetas

Seleccionar Tarjetas

78 Hean Fancio Piero Moran
 161 (Botrys Simon Ledezma
 485 (Sara Maria Loachuan
 0 (Usuario Invitado)

Escoja puerta con la que desea realizar el reporte

Puerta 1

Puerta 2

Ambas

* Si no se selecciona ningún ítem se genera un reporte con todos los equipos y los accesos generados hasta el día de hoy

Aceptar

Cancelar

FIGURA B-15. PANTALLA PREVIO REPORTE TARJETA.

PANTALLA PRESENTACIÓN REPORTES.

Esta pantalla visualiza los datos del reporte. Presenta una tabla de cinco columnas que son: Equipo o Tarjeta (según el tipo reporte escogido), puerta, hora acceso, fecha acceso y mensaje. Además presenta un control que permite seleccionar la tarjeta o el equipo del que se desea tener información (ver Figura B-13).

BIBLIOGRAFÍA

- [1] VIESCAS, John, "Guía Completa de Microsoft Access 2", McGrawHill, México, 1994.
- [2] NATIONAL INSTRUMENTS, "LabVIEW Graphical Programming for Instrumentation. User Manual", National Instruments, 1996.
- [3] NATIONAL INSTRUMENTS, "LabVIEW Graphical Programming for Instrumentation. Function and VI Reference Manual", National Instruments, 1997.
- [4] McKINNEY, Bruce, "Programación Avanzada para Visual Basic", McGraw Hill, Microsoft Press, ,1995.
- [5] CORNELL, Gary, "Manual de Visual Basic 4 para Windows 95", McGeaw Hill, Madrid, 1996.
- [6] DALLAS SEMICONDUCTOR, "DS5000 Soft Microcontroller, User's Guide", Texas, 1990.
- [7] NORTHERN COMPUTERS, "WIN-PAK, User's Manual. Version 1.15".
- [8] ALVAREZ, Carlos, "Control de Accesos", EPN, Quito, Tesis de Grado 1991.
- [9] <http://www.natinst.com>, archivo AN081.pdf.
- [10]<http://www.dalsemi.com>.
- [11]<http://codeaccess.com/wiegand.html>

REFERENCIAS BIBLIOGRÁFICAS.

- [1] <http://codeaccess.com/rk-which.html>, "Which technology.....".
- [2] <http://codeaccess.com/wiegand.html>, "Wiegand Background...."
- [3] ALVAREZ, Carlos, "Control de Accesos", EPN, Quito, Tesis de Grado 1991, pág. 20.
- [4] NORTH
- [5] <http://www.natinst.com>, archivo AN081.pdf, pág. 3.
- [6] VIESCAS, John, "Guía Completa de Microsoft Access 2", McGrawHill, México, 1994, pág. 7.
- [7] Base de datos de ejemplo "Neptuno.mdb" incluido con Microsoft Access, Tabla "Empleados".
- [8] Base de datos de ejemplo "Neptuno.mdb" incluido con Microsoft Access, Formulario "Empleados".
- [9] Base de datos de ejemplo "Neptuno.mdb" incluido con Microsoft Access, Informe "Catálogo".
- [10] VIESCAS, John, "Guía Completa de Microsoft Access 2", McGrawHill, México, 1994, pág. 39, Figura 3- 2.
- [11] VIESCAS, John, "Guía Completa de Microsoft Access 2", McGrawHill, México, 1994, pág. 781, Figura 21-7.

[12]Ayuda en línea de Microsoft Access 8.0, Ayuda>Contenido> Referencia DAO >Modelo de objeto de acceso a datos para espacios de Trabajo Microsoft Jet.

[13]<http://www.natinst.com>, archivo AN081.pdf, pág. 4, Figura 1.

[14]<http://www.natinst.com>, archivo AN081.pdf, pág. 5, Figura 2.

[15]<http://www.natinst.com>, archivo AN081.pdf, pág. 8, Figura 5.

[16]<http://www.natinst.com>, archivo AN081.pdf, pág. 9, Figura 6.