ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN

TESIS DE GRADO

"DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL Y MONITOREO DE ACCESO Y SEGURIDAD SOBRE UNA RED ETHERNET UTILIZANDO TARJETAS SMART CARD"

Previa a la obtención del título de:

INGENIERO EN ELECTRICIDAD ESPECIALIZACIÓN ELECTRÓNICA Y AUTOMATIZACIÓN INDUSTRIAL
INGENIERO EN ELECTRICIDAD ESPECIALIZACIÓN ELECTRÓNICA Y AUTOMATIZACIÓN INDUSTRIAL
INGENIERO EN COMPUTACIÓN ESPECIALIZACIÓN SISTEMAS TECNOLÒGICOS

PRESENTADA POR:

CARLOS PATRICIO MOLINA CRESPO
FRANKLIN ANIBAL RODRIGUEZ ACOSTA
VICTOR XAVIER VERA ANCHUNDIA

GUAYAQUIL – ECUADOR

2007

AGRADECIMIENTO

A todos quienes contribuyeron de manera directa o indirecta en el desarrollo de este trabajo, en especial al Ing. Carlos Valdivieso A. y al Grupo de Investigación GUIMIC por su valiosa guía en todas las etapas del desarrollo del proyecto.

DEDICATORIA

Nuestra dedicatoria y homenaje A nuestros padres: Norma Crespo, Carlos Molina A., Lourdes Acosta, Félix Rodríguez, Rosa Anchundia Y Gonzalo Vera por su inagotable paciencia, comprensión y apoyo en la realización de nuestra tesis, así como en todas las etapas de nuestras vidas.

TRIBUNAL DE GRADUACIÓN

PRESIDENTE

Ing. Holger Cevallos Ulloa

DIRECTOR DE TESIS

Ing. Carlos Valdivieso A.

MIEMBROS PRINCIPALES

Ing. Marcelo Loor

Ing. Hugo Villavicencio

DECLARACIÓN EXPRESA

"La responsabilidad del contenido de esta Tesis de Grado, nos corresponde exclusivamente; y el patrimonio intelectual de la misma, a la Escuela Superior Politécnica del Litoral"

(Reglamento de exámenes y títulos profesionales de la ESPOL)

_____
Carlos Patricio Molina Crespo

_____
Franklin Aníbal Rodriguez Acosta

_____
Victor Xavier Vera Anchundia

RESUMEN

Desarrollo de un sistema de acceso remoto basado en un lector de tarjetas Smart Card que se comunica través de una red Ethernet, almacenando información de diferentes ambientes o habitaciones, en un servidor remoto.

En las tarjetas se almacenará la información del usuario con el número de habitación y su respectivo código de autorización de acceso. El huésped solamente puede acceder a la habitación asignada y podrá tener duplicados de tarjetas de acceso para otros usuarios de la misma habitación. Se puede contar con tarjetas que permitan el acceso a más de una habitación y con diferentes niveles de autorización; como es el caso de supervisores, personas de limpieza y personal de mantenimiento.

Los lectores de tarjetas Smart Card obtienen la información de los usuarios que desean entrar en una habitación y la almacena en un servidor central. El cual valida los datos con la lógica de la base de datos del sistema de acceso y envía la confirmación de ingreso al lector de tarjetas instalado en cada habitación.

ÍNDICE GENERAL

ABREVIATURAS

TCI	Tarjeta con circuito integrado
CPU	Unidad de procesamiento central
ROM	Memoria sólo de lectura
RAM	Memoria de acceso aleatorio
ISO 7816	Estándar para las tarjetas inteligentes
ISO/IEC 14443	Estándar de comunicación para las tarjetas inteligentes sin contacto.
ISO 15693	Estándar alternativo para las tarjetas inteligentes sin contacto.
RFID	Identificación por radio frecuencia.
SIM	Módulo de identificación del suscriptor
GSM	Grupo especial Móvil
API	Interfase de programación y aplicación
JVM	Máquina Virtual de JAVA
ISO	Organización de Estándares Internacionales
PPS	Selección de parámetros y protocolo
SAM	Módulo de aplicación de seguridad
ATR	Respuesta a un reset
STX	Inicio de texto
ETX	Fin de texto

APDU	Unidad de aplicación de protocolo de datos
MCU	Unidad microcontroladora
ICSP	Programación serial dentro del circuito.

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

INTRODUCCIÓN

En el desarrollo del ***Diseño e implementación de un sistema de control y monitoreo de acceso y seguridad sobre una red Ethernet utilizando tarjetas Smart Card*** cumplimos con los siguientes objetivos:

OBJETIVOS GENERALES

- Desarrollar un prototipo de hardware y software que permita la comunicación a una red Ethernet, mediante el uso de un microcontrolador 18F1320 y una tarjeta SBC45EC que sirve de enlace entre el PIC y la red Ethernet de códigos de seguridad obtenidos de la lectura de una tarjeta inteligente para el accionamiento de una cerradura eléctrica.
- Administrar los datos adquiridos desde un servidor central.

OBJETIVOS ESPECÍFICOS

- Comunicar lectores de tarjetas Smart Card a una red Ethernet.
- Permitir la existencia de usuarios de varios niveles.
- Monitorear el acceso a cualquier localidad de un edificio.
- Manejar una base de datos indicando: Hora de entrada, datos del usuario, y un código asignado al lugar de acceso.

- Generar reportes sobre los datos registrados.

Para que el lector tenga una idea rápida acerca de lo que encontrará en cada capítulo, a continuación brindamos una descripción general de cada uno de ellos:

- ***Capítulo 1:*** se identifica y se define el problema a tratar. También se presentan los antecedentes, causales y efectos del problema. Al final de este capítulo detallamos el alcance que tendrá la solución planteada.

- ***Capítulo 2:*** se definen los componentes substanciales del hardware utilizado y sus principales características. Y se brinda una descripción general del sistema de administración y control desarrollado.

- ***Capítulo 3:*** se define lo que es una tarjeta inteligente, ofreciéndose una introducción a los diferentes tipos de tarjetas existentes y a sus áreas de aplicación. También se brinda una descripción general de las normas internacionales que rigen el comportamiento de las tarjetas inteligentes.

- ***Capítulo 4:*** se explica el lector de tarjetas inteligentes escogido para la implementación del proyecto, así como: sus características, tipos de

tarjetas soportadas, parámetros de comunicación, formato de comandos y respuestas y su protocolo de transmisión y recepción.

- ***Capítulo 5:*** se refiere a la tarjeta inteligente utilizada para la implementación del proyecto. También se define la configuración interna de la misma, sus archivos y registros. En este capítulo se da a conocer los comandos para la tarjeta y se definen los mensajes de estado de la tarjeta.

- ***Capítulo 6*** se brinda una descripción de la tarjeta de comunicación utilizada para la realización del presente proyecto. Además en este capítulo se describe de manera general la pila TCP/IP del fabricante Microchip que permite manejar el controlador de red de la tarjeta usada.

- ***Capítulo 7:*** se detallan las especificaciones del software desarrollado para el control y monitoreo de accesos. En este capítulo se dan a conocer los distintos módulos de los que consta el sistema creado; así como su diseño y las pruebas realizadas para garantizar su correcta funcionalidad.

CAPÍTULO 1

1. ANÁLISIS DEL PROBLEMA

OBJETIVOS:

- Precisar el problema que se pretende solucionar en el presente proyecto de tesis.
- Puntualizar los antecedentes, causales y efectos detectados en el problema seleccionado.
- Definir hasta donde se pretende llegar en desarrollo del presente proyecto de tesis.

1.1. DEFINICION DEL PROBLEMA

Durante los últimos años se han introducido al mercado sistemas para controlar acceso ya sea para una puerta o para un sistema de redes de varios edificios basados en el esquema de control de acceso discrecional (DAC) el mismo que genera un grado dificultad para la actualización y colección de datos de estos productos.

La poca flexibilidad que presenta el esquema de control de acceso de los productos ofertados en la actualidad reduce el mercado potencial

para ofrecer este servicio por lo difícil de su adaptabilidad para varios de tipos de esquemas de negocios.

Existen varios métodos y tecnologías para controlar el acceso de usuarios los cuales no brindan el grado de confiabilidad y flexibilidad necesarias para garantizar la seguridad a sus clientes. En la Tabla 1.1. Se muestra una comparación de las distintas tecnologías de control de acceso disponibles en el mercado, así como sus características principales.

PARÁMETROS DEL SISTEMA	CÓDIGO DE BARRAS	OCR	RECONOCIMIENTO DE VOZ	BIOMETRÍA	BANDA MAGNÉTICA	SMART CARD
Cantidad típica de datos (bytes)	1 – 100	1 – 100			1 – 100	**16 – 64 K**
Densidad de datos	Bajo	Bajo	Alto	Alto	Bajo	**Muy alto**
Influencia a la suciedad/humedad	Muy alta	Muy alta			Alta	Posible
Costo de compra	Muy bajo	Medio	Muy Alto	Muy alto	Alto	**Bajo**
Copiado desautorizado/Modificación	Leve	Leve	Posible (Audio Tape)	Imposible	Leve	**Imposible**
Velocidad de Lectura	Normal	Normal	Bajo	Muy bajo	Bajo	Normal

Tabla 1.1. Comparación de diferentes sistemas de identificación.

1.2. ANTECEDENTES, CAUSALES Y EFECTOS DEL PROBLEMA

Antecedentes

Se tomó como punto de partida el requerimiento presentado por un empresario hotelero indicando la necesidad de elaborar el prototipo para un sistema de control de acceso de un hotel de la ciudad de Cuenca. Se plantearon las siguientes especificaciones:

- Control de acceso a 100 habitaciones distribuidas en 8 pisos.
- Permitir el acceso por niveles o jerarquía del usuario.
- Los accesos deben ser monitoreados y almacenados en un servidor central.

Causales

La empresa hotelera, al verse en una situación de mejoras del servicio y de implementación de nuevas tecnologías decidió instalar un sistema que le ahorre tiempo en el registro de los huéspedes y le ayude a optimizar la presentación de reportes de los usuarios y de los accesos de los empleados en la empresa.

Efectos

Al no contar con un sistema automatizado se pierde tiempo y dinero en el registro de clientes, teniendo poco o casi nulo control de seguridad de las localidades a las cuales accede el cliente.

Se complica la definición de áreas para uso exclusivo de clientes y/o para empleados y se continúa con el problema habitual de pérdidas de llaves de las localidades.

1.3. ALCANCE DEL PROYECTO

El prototipo cumple con los requerimientos solicitados por la empresa hotelera, el cual es también de fácil adaptación para otros tipos de negocios.

El sistema cuenta con el control de acceso y el repositorio de datos centralizado el cual puede ser un servidor o PC, el mismo que en base a la lógica del negocio, parámetros configurables por un operador de acuerdo al los niveles de la autorización da el acceso o lo restringe a la localidad solicitada.

Se generan reportes de bitácoras de acceso a las localidades a las cuales los usuarios han accedido y se mantiene el control de disponibilidad de las habitaciones según la demanda del hotel o negocio en el cual se encuentre implementado el sistema.

El sistema permite la creación de operadores y/o administradores para permitir accesos y realizar el respectivo mantenimiento del sistema administrador de accesos (SAC).

CAPÍTULO 2

2. ANÁLISIS Y DISEÑO GENERAL

OBJETIVOS:

- Presentar los componentes principales del hardware utilizado.
- Describir de manera general el sistema de administración y control desarrollado.

2.1. DIAGRAMA DE BLOQUES GENERAL

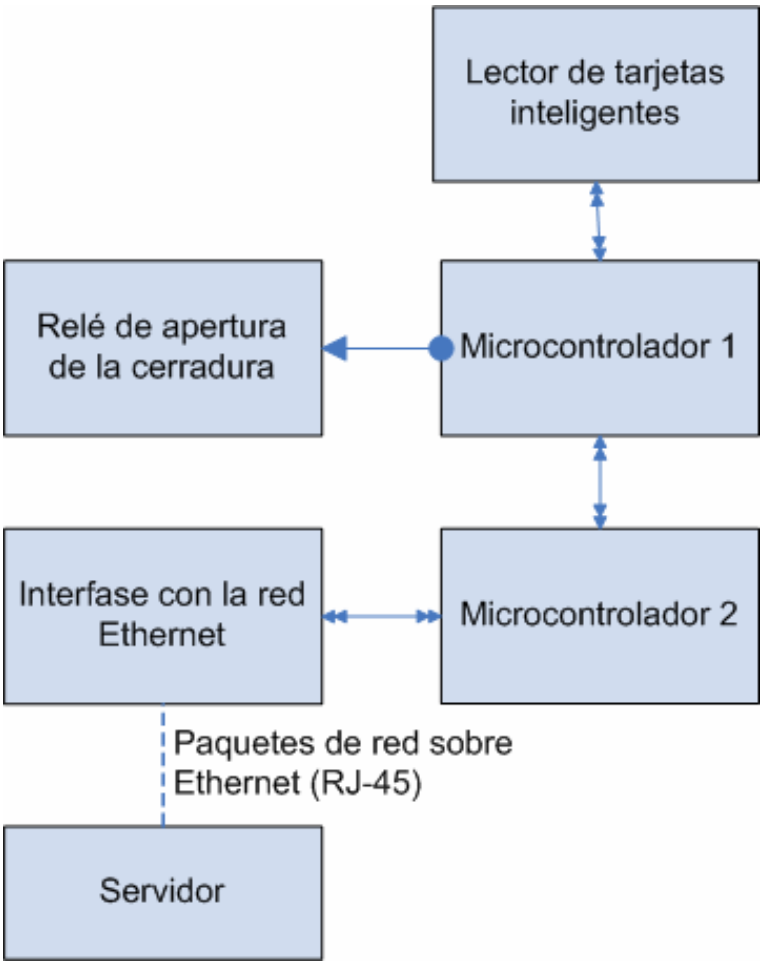


Figura 2.1. Diagrama de Bloques General del Proyecto.

2.2. DESCRIPCIÓN DEL DIAGRAMA DE BLOQUES

Lector de tarjetas inteligentes

El lector de tarjetas inteligentes ACR30 [1], recibe comandos desde el Microcontrolador 1, lee la información contenida en la tarjeta ACOS2 [2] (Tarjeta inteligente usada en el proyecto) y la envía al Microcontrolador 1.

Microcontrolador 1

Es un PIC18F1320 [3] el cual entre sus funciones tiene, comunicarse con el lector de tarjetas inteligentes ACR30 utilizando el protocolo RS-232 y enviar los datos obtenidos del lector al microcontrolador 2. Finalmente emite la instrucción para la activación del relé de apertura de puerta.

Microcontrolador 2

Es un PIC18F452 [4] que procesa la información recibida desde el Microcontrolador 1, encapsula los datos en paquetes UDP y los envía al servidor central para que sean procesados [5]. También recibe del servidor central la confirmación de acceso (paquete UDP) y la envía al microcontrolador 1.

Interfaz con la red Ethernet

Permite el envío y recepción de los paquetes UDP los cuales son guardados en un buffer para que el Microcontrolador 2 pueda descargarlos y procesarlos. Esta interfaz es necesaria para cubrir la distancia que existe entre el servidor central y los puntos de acceso.

Servidor

Representa a una computadora personal estándar que se encuentra conectada a una red Ethernet. La computadora tiene un sistema de adquisición de datos por medio del protocolo UDP. Este sistema procesa y gestiona con su base de datos el requerimiento de ingreso y posteriormente envía su respuesta al punto de acceso.

Relé de apertura de puerta

Este componente es necesario debido a las diferencias de voltajes y corrientes entre el microcontrolador y la cerradura eléctrica. El circuito toma un voltaje de lógica TTL desde uno de los pines del microcontrolador [6]. La corriente de salida del microcontrolador se amplifica por medio de un transistor que se usa para la activación del relé.

2.3. Comunicación con el lector de tarjetas ACR30

2.3.1. Hardware

El diagrama de bloques mostrado en la Figura 2.2. Representa el funcionamiento del circuito de interfase con el lector ACR30 y la tarjeta SBC45EC [7]. Como se puede observar el microcontrolador 1 sirve de enlace entre el ACR30 y la tarjeta de red.

Figura 2.2 Diagrama de bloques del circuito de interfase entre el lector ACR30 y la tarjeta SBC45EC.

La Figura 2.3. Representa el diagrama esquemático del circuito que se implementó para poder comunicar el lector de tarjetas inteligentes con la tarjeta madre SBC45EC que contiene el PIC18F452 y el integrado de interfase de red RTL8019. Los pines del microcontrolador B5 y B6 se utilizan como pines de comunicación USART con la tarjeta SBC45EC a 57600 baudios. Los pines B1 y B4 sirven como pines de transmisión y recepción respectivamente con el lector de tarjetas inteligentes; se comunican a 9600 baudios. Las señales de **busy* (Pin A0 del

microcontrolador) y *reset* (Pin A1 del microcontrolador) sirven como señales de control para el lector ACR30.

Las señales del lector de tarjetas inteligentes se explican en el Capítulo IV.

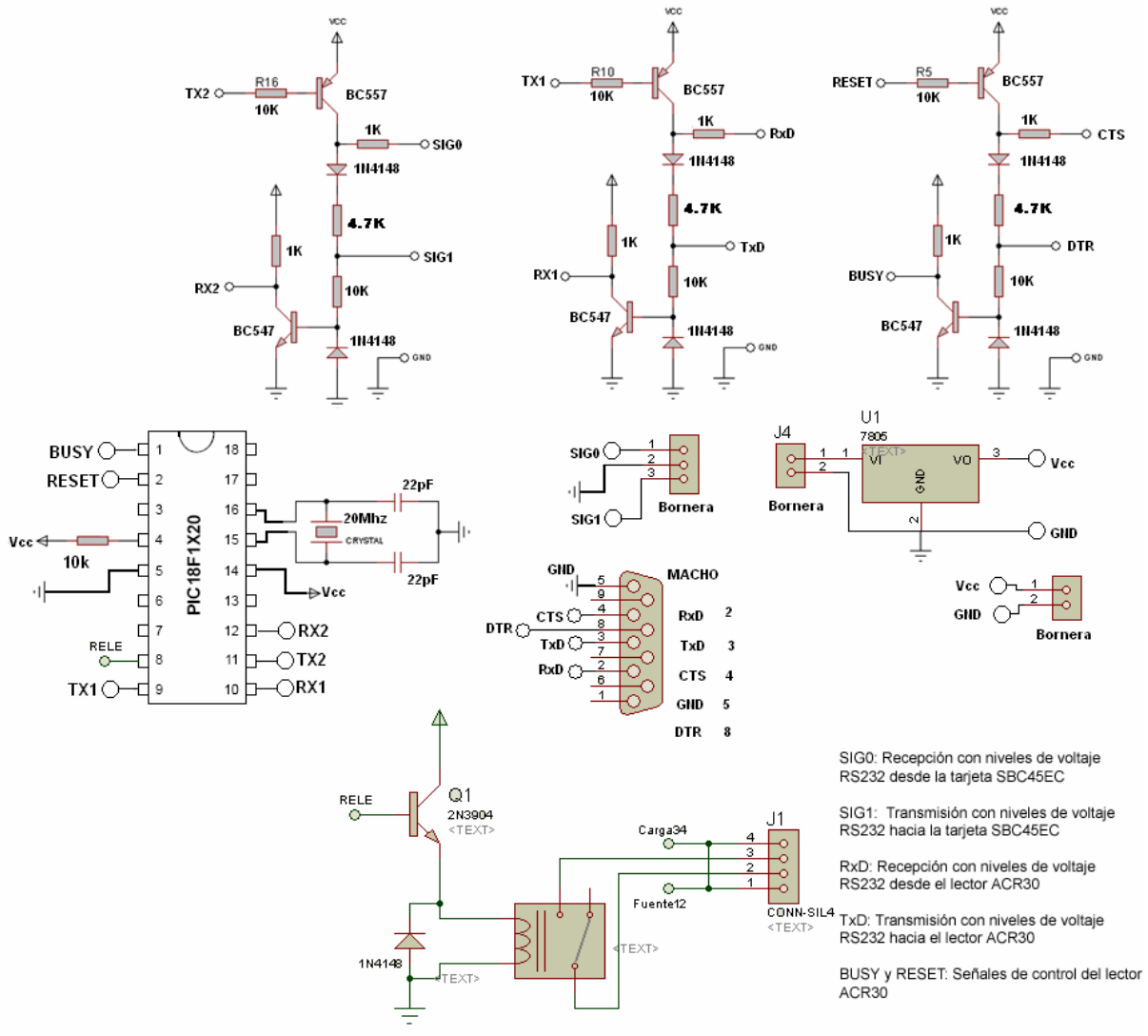


Figura 2.3. Diagrama esquemático del circuito de interfase entre el lector ACR30 y la tarjeta SBC45EC.

A continuación se muestran el circuito impreso del circuito de interfase entre el lector ACR30 y la tarjeta SBC45EC (Figura 2.4.), la distribución de los componentes en la placa (Figura 2.5.) y el Gráfico de la ubicación de los componentes en el circuito de interfase entre el lector ACR30 y la tarjeta SBC45EC y su conexión (Figura 2.6.).

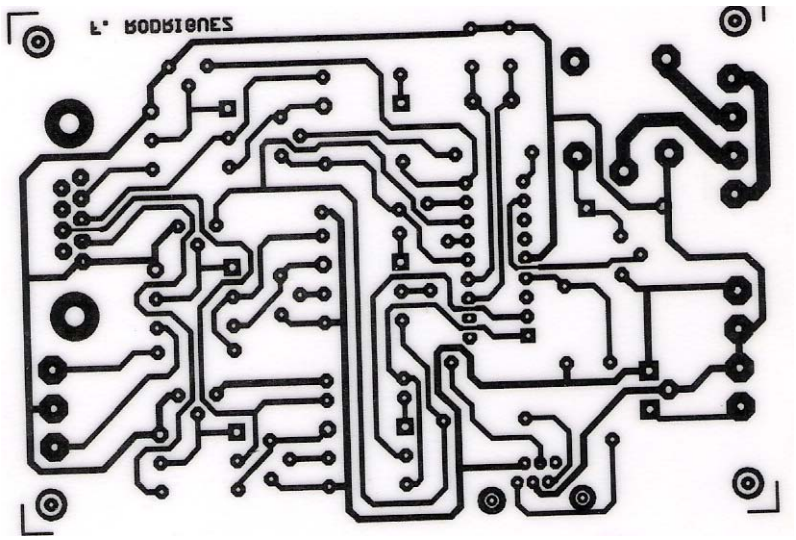


Figura 2.4. Circuito impreso del circuito de interfase entre el lector ACR30 y la tarjeta SBC45EC

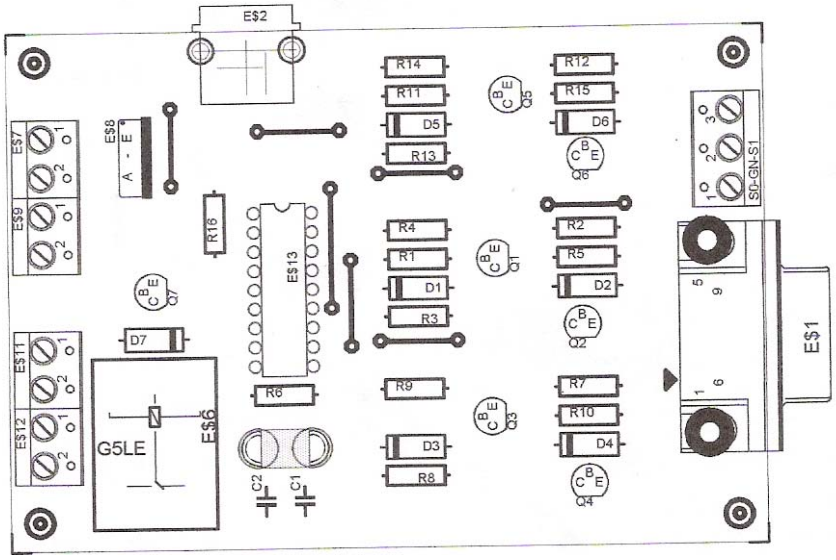


Figura 2.5. Componentes en la placa del circuito de interfase entre el lector ACR30 y la tarjeta SBC45EC

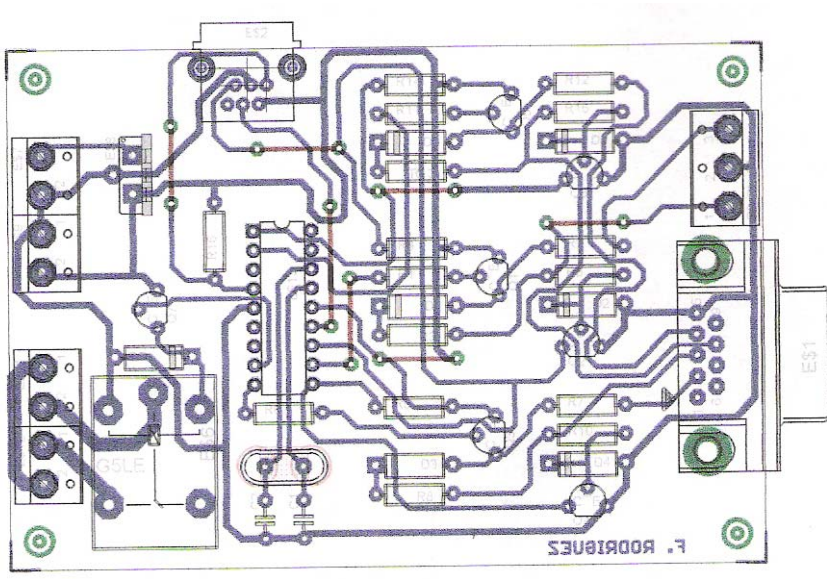
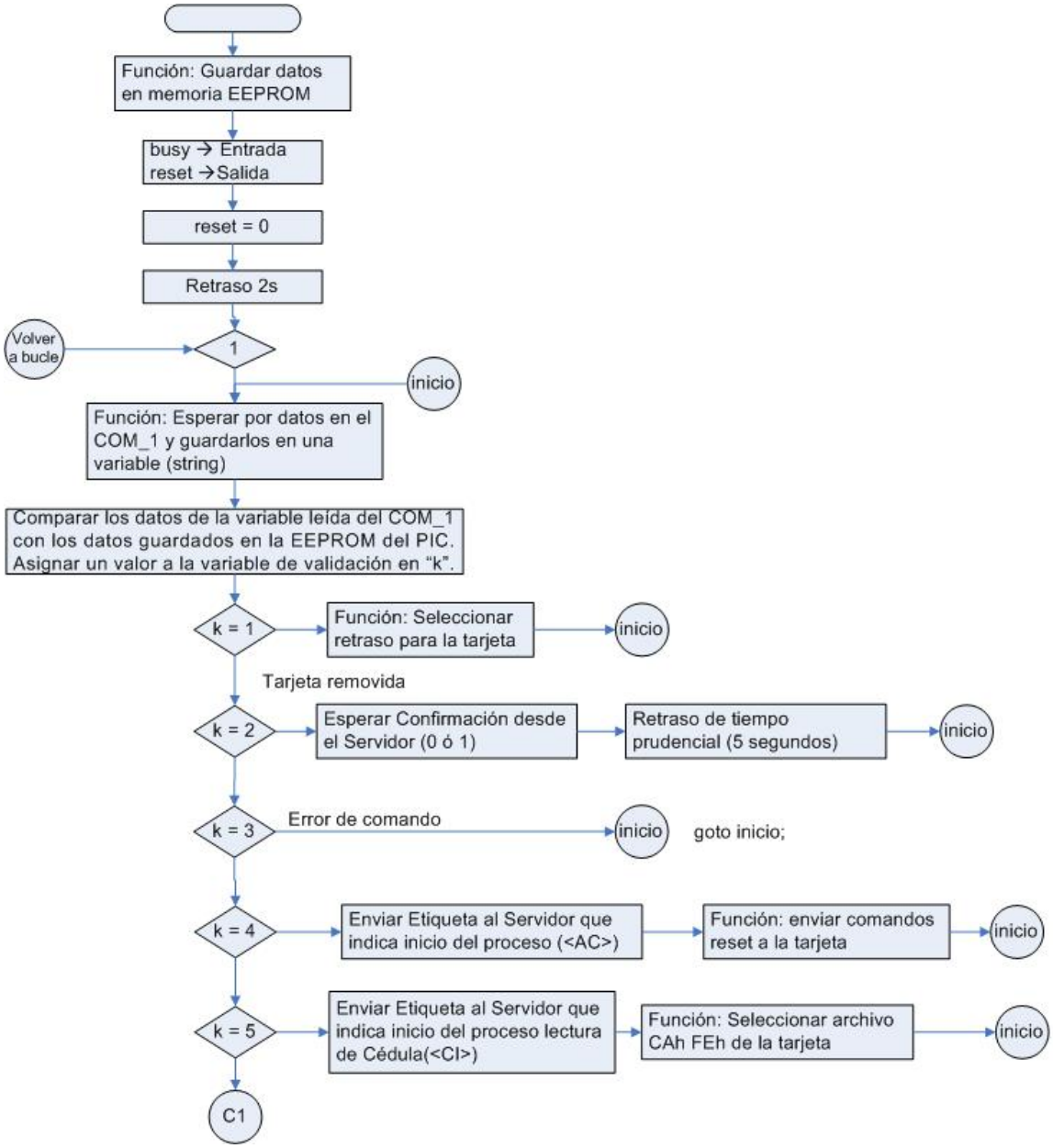


Figura 2.6. Gráfico de la ubicación de los componentes en el circuito de interfase entre el lector ACR30 y la tarjeta SBC45EC y su conexión.

2.3.2. Programa del microcontrolador 2



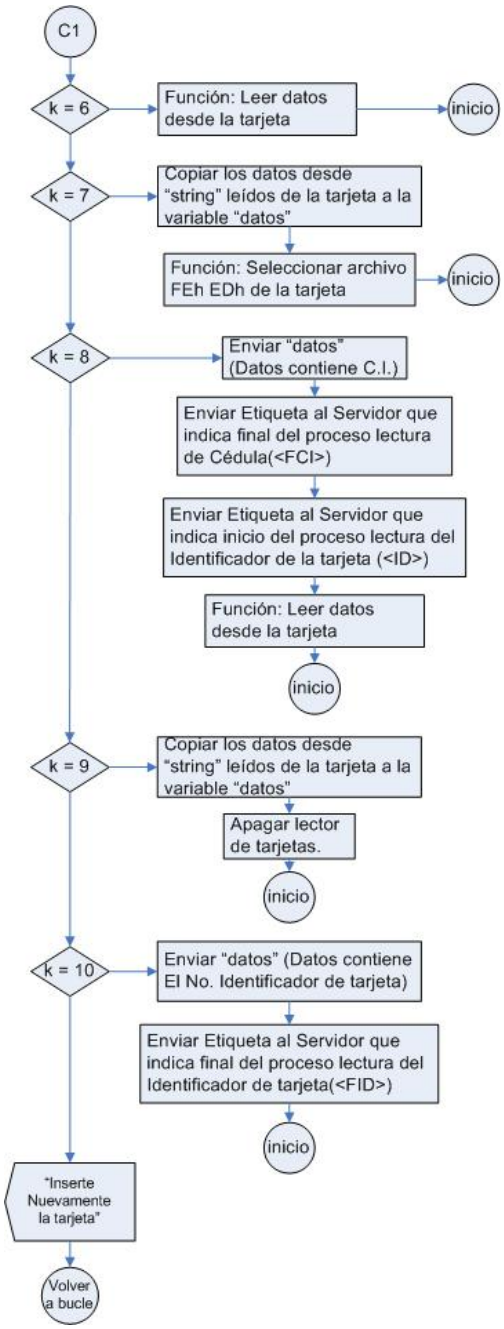
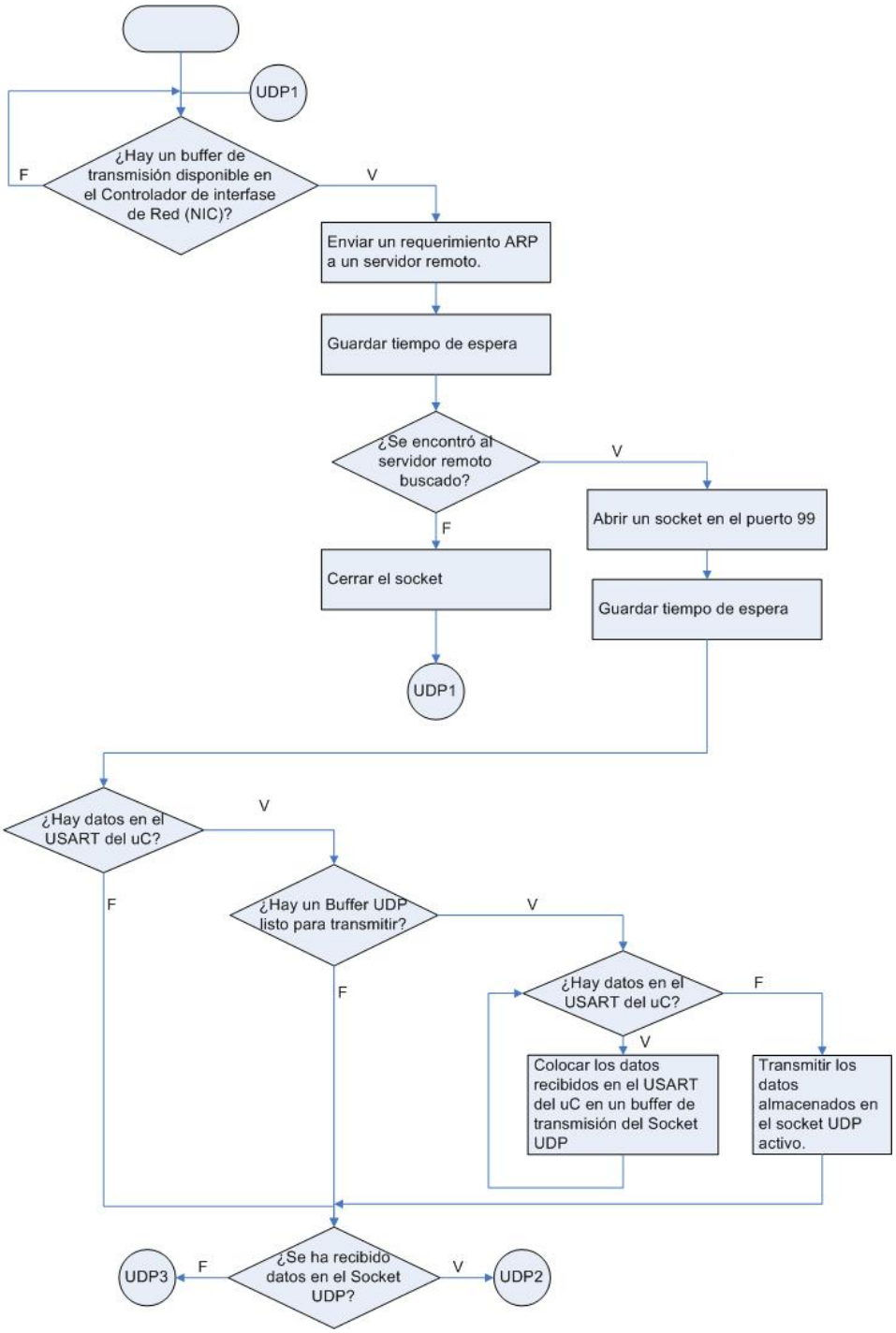


Figura 2.7. Diagrama de flujo de la función implementada en el *Microcontrolador 2*

Para una descripción detallada del diagrama de flujo de las funciones implementadas en el microcontrolador 2 refiérase al Apéndice C y para la descripción de las mismas refiérase al Apéndice D.

2.4. Comunicación con la red Ethernet

En esta sección se describe las funciones utilizadas para el programa del PIC18F452 [4] que se encuentra en la tarjeta SBC45EC [7].


UDP1
¿Hay un buffer de transmisión disponible en el Controlador de interfase de Red (NIC)?
F
V
Enviar un requerimiento ARP a un servidor remoto.
Guardar tiempo de espera
¿Se encontró al servidor remoto buscado?
V
F
Abrir un socket en el puerto 99
Cerrar el socket
Guardar tiempo de espera
UDP1
¿Hay datos en el USART del uC?
V
F
¿Hay un Buffer UDP listo para transmitir?
V
F
¿Hay datos en el USART del uC?
F
V
Colocar los datos recibidos en el USART del uC en un buffer de transmisión del Socket UDP
Transmitir los datos almacenados en el socket UDP activo.
UDP3
F
¿Se ha recibido datos en el Socket UDP?
V
UDP2

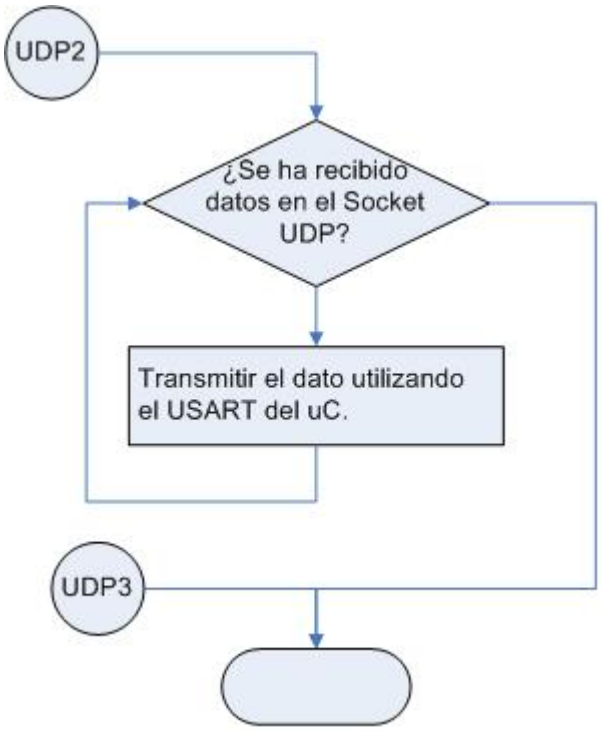


Figura 2.8. Diagrama de Flujo de la función implementada en el *Microcontrolador 1*.

Descripción del Protocolo de resolución de direcciones de la Pila de Microchip (ARP y ARPTask)

La capa ARP de la pila de Microchip es realmente implementada en dos módulos: ARP y ARPTask. El módulo ARP es implementado en el archivo "ARP.c" y crea las primitivas necesarias del ARP. El módulo ARPTask, implementado en el archivo "ARPTask.c", utiliza las primitivas y provee servicios completos para el protocolo ARP.

El módulo ARPTask esta implementado como una máquina de estados cooperativa, que responde a los requerimientos ARP desde el servidor remoto. Este módulo también mantiene un nivel de caché para almacenar las respuestas ARP y enviarlas a un nivel superior cuando es requerida. El módulo ARPTask no se implementa un mecanismo de reintento, por ende los módulos de niveles superiores o aplicaciones deben detectar las condiciones de tiempo de espera y responder acorde a esto [8].

El módulo ARPTask trabaja en dos modos: modo Servidor y modo Servidor/Cliente, una porción del código es habilitada y compilada para generar los requerimientos ARP desde el Servidor local. En el modo Servidor, los requerimientos del ARP no son compilados.

Cuando se define STACK_CLIENT_MODE se incluye la porción de código referente al cliente. En el modo Servidor/Cliente, el módulo ARPTask mantiene un caché de un nivel para almacenar las respuestas ARP del Servidor remoto .

Descripción del Protocolo de Datagrama de Usuario en la Pila de Microchip

La capa UDP de la pila TCP/IP de Microchip esta implementada en el archivo “UDP.c”. El archivo de encabezado “UDP.h” define los servicios que provee esta capa. En la arquitectura de la pila, UDP es una capa activa. Esta envía paquetes

UDP y notifica al respectivo *socket que un dato ha llegado o ha sido transmitido. El módulo UDP esta implementado como una tarea cooperativa, que desarrolla operaciones automáticas sin necesidad del reconocimiento de la aplicación principal.

La capa permite más de 254 *sockets UDP (número limitado solo por la cantidad de memoria y el compilador usado). Con más de un *socket, las aplicaciones de de niveles superiores pueden mantener comunicaciones UDP simultaneas. Más de una aplicación puede usar esta capa. En el lado del receptor, sólo hay un buffer de recepción. Si un *socket recibe un dato el dueño de este *socket debe enviar y desechar el buffer de recepción en el tiempo de una tarea para que así los demás *buffers reciban el dato.

Este diseño obliga a que una vez que la tarea detecte un paquete que le sea de interés, debe recibir el dato completamente en el tiempo de una tarea. Una tarea no puede enviar una parte de un paquete durante el tiempo de la tarea y esperar para enviar el resto del paquete después.

Las especificaciones UDP no obligan a que el cálculo del *checksum sea enviado en los paquetes UDP. Para reducir el tamaño del programa y los requerimientos de memoria, la pila TCP/IP de Microchip no implementa el

cálculo del checksum, por lo que se fija a cero este campo para indicar que este cálculo no se realizo [8].

2.5. Características generales del Sistema de Administración y Control de Acceso

El diseño del sistema administrador de accesos tiene como una de sus principales características la posibilidad de recibir por medio del protocolo UDP paquetes de datos que representan peticiones de ingresos a lugares físicos [9], para que estos sean validados contra una base de datos la cual contiene la lógica de qué personas tienen accesos a qué lugares y que luego sea enviada una respuesta de confirmación positiva o negativa [10].

Como otra característica del diseño que tiene el sistema se provee un módulo que permite la administración de accesos en el cuál se guarda la información de los usuarios/clientes en las tarjetas inteligentes y se define a qué lugares va a tener acceso con sus respectivos días y horas. Y de esta manera generan reportes de los accesos a todos los usuarios/clientes.

CAPÍTULO 3

3. TARJETAS INTELIGENTES

OBJETIVOS:

- Definir Qué es una tarjeta inteligente
- Dar una introducción a los tipos de tarjetas y sus distintas aplicaciones
- Dar a conocer al lector sobre las normas internacionales que rigen el comportamiento de la tarjeta, interfase con el usuario, dimensiones, etc.
- Dar un listado general de los distintos fabricantes de tarjetas y lectores.

3.1. INTRODUCCIÓN

Una tarjeta inteligente o tarjeta con circuito integrado (TCI), es una tarjeta con circuitos integrados incluidos. Aunque existe un diverso rango de aplicaciones, hay dos categorías principales de TCI. Las Tarjetas de memoria contienen sólo componentes de memoria no volátil y posiblemente alguna lógica de seguridad. Las Tarjetas microcontroladas contienen memoria y microcontroladores.

Debemos distinguir entre lo que es una tarjeta inteligente y lo que es una Tarjeta Chip. No se trata de lo mismo, ya que el chip no es lo que la hace "Inteligente", si no el microcontrolador, por esto existen diferentes tipos de tarjetas, de las cuales, unas son "Inteligentes", y otras son de "memoria".

La percepción estándar de una "tarjeta inteligente" es una tarjeta microcontrolada de las dimensiones de una tarjeta de crédito con varias propiedades especiales y es capaz de proveer servicios de seguridad.

3.2. ESTRUCTURA DE UNA TARJETA INTELIGENTE MICROCONTROLADA

Internamente, el chip de una tarjeta inteligente microcontrolada se compone de:

- **CPU** (*Unidad de procesamiento central*): el procesador de la tarjeta; suele ser de 8 bits, a 5 MHz. y 5 voltios. Pueden tener opcionalmente módulos hardware para operaciones criptográficas.
- **ROM** (*Memoria de sólo lectura*): memoria interna (normalmente entre 12 y 30 KB) en la que se incrusta el sistema operativo de la tarjeta, las rutinas del protocolo de comunicaciones y los algoritmos de seguridad

de alto nivel por software. Esta memoria, como su nombre indica, no se puede reescribir y se inicializa durante el proceso de fabricación.

- **EEPROM**: memoria de almacenamiento en el que está grabado el sistema de ficheros, los datos usados por las aplicaciones, claves de seguridad y las propias aplicaciones que se ejecutan en la tarjeta. El acceso a esta memoria está protegido en distintos niveles por el sistema operativo de la tarjeta.
- **RAM** (*Memoria de acceso aleatorio*): memoria volátil para el trabajo en el procesador.

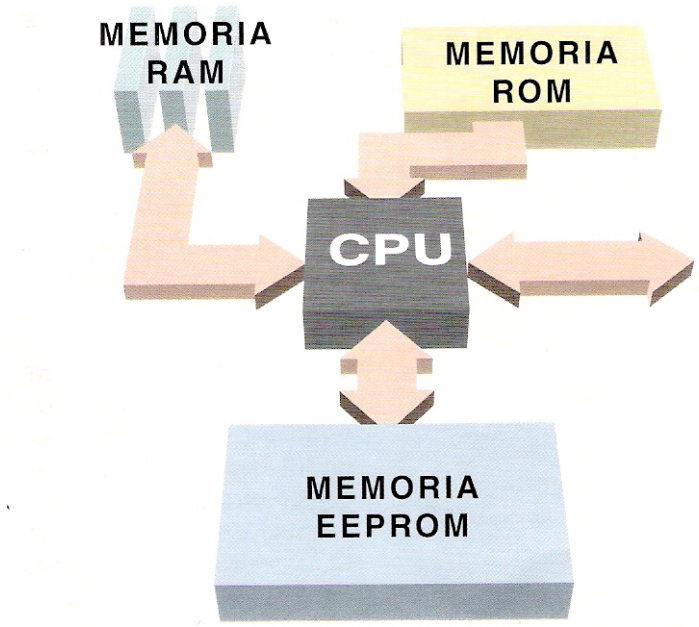


Figura 3.1. Estructura de una tarjeta inteligente microcontrolada

3.3. TIPOS DE TARJETA SEGÚN LA INTERFASE

3.3.1. TARJETA INTELIGENTE DE CONTACTO

La serie de estándares **ISO/IEC 7816** e **ISO/IEC 7810** definen:

- La forma física (parte 1)
- La posición de las formas de los conectores eléctricos (parte 2)
- Las características eléctricas (parte 3)
- Los protocolos de comunicación (parte 3)
- El formato de los comandos enviados a la tarjeta y las respuestas retornadas por la misma.
- La funcionalidad.

Las tarjetas no contienen baterías; la energía es suministrada por los lectores de tarjetas.

Los lectores de tarjetas inteligentes de contacto son utilizados como un medio de comunicación entre la tarjeta inteligente y un servidor, como por ejemplo un ordenador.

3.3.2. TARJETAS INTELIGENTES SIN CONTACTO

El segundo tipo es la tarjeta inteligente sin contacto en el cual el chip se comunica con el lector de tarjetas mediante inducción a una tasa de transferencia de 106 a 848 Kb/s).

El estándar de comunicación de tarjetas inteligentes sin contacto es el **ISO/IEC 14443** del 2001 define dos tipos de tarjetas sin contacto (A y B), permitidos para distancias de comunicación de hasta 10 cm. Ha habido propuestas para la ISO 14443 tipos C, D, E y F que todavía tienen que completar el proceso de estandarización. Un estándar alternativo de tarjetas inteligentes sin contacto es el **ISO 15693**, el cual permite la comunicación a distancias de hasta 50 cm.

Un ejemplo del amplio uso de tarjetas inteligentes sin contacto es la tarjeta Octopus en Hong Kong, la cual usa el estándar anterior al ISO/IEC 14443.

Una tecnología sin contacto relacionada es RFID (*Identificación por radio frecuencia*). En algunos casos puede ser utilizado para aplicaciones similares a las tarjetas inteligentes sin contacto, como el peaje electrónico. Las RFID generalmente no incluyen memoria de escritura o microcontroladores como las tarjetas inteligentes sin contacto.

3.3.3. TARJETAS HÍBRIDAS Y DUALES

Una tarjeta híbrida comienza con una tarjeta sin contacto a la cuál se le agrega un segundo chip de contacto. Ambos chips pueden ser:

- Chips microcontrolados.
- Chips de memoria.

El chip sin contacto es generalmente usado en aplicaciones que requieren transacciones rápidas. Por ejemplo el transporte, mientras que el chip de contacto es generalmente utilizado en aplicaciones que requieren de alta seguridad como las bancarias.

Una tarjeta de interfase dual es similar a la tarjeta híbrida en que la tarjeta presenta ambas interfaces con y sin contacto. La diferencia más importante es el hecho de que la tarjeta de interfase dual tiene solamente un solo circuito integrado.

3.4. TIPOS DE TARJETAS SEGÚN EL FORMATO

En el estándar ISO/IEC 7816 parte 1 se definen los siguientes tamaños para tarjetas inteligentes:

- **ID 000**: el de las tarjetas *SIM usadas para teléfonos móviles *GSM.
- **ID 00**: un tamaño intermedio poco utilizado comercialmente.
- **ID 1**: el más habitual, tamaño tarjeta de crédito.

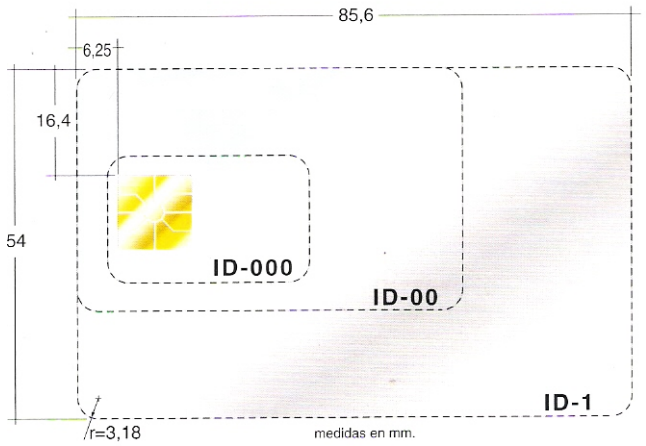


Figura 3.2. Tipos de tarjetas según el formato

3.5. TIPOS DE TARJETAS SEGÚN LA ESTRUCTURA DE SU SISTEMA OPERATIVO

3.5.1. BASADAS EN SISTEMAS DE FICHEROS, APLICACIONES Y COMANDOS

Estas tarjetas disponen del equivalente a un sistema de ficheros compatible con el estándar ISO/IEC 7816 parte 4 y un sistema

operativo en el que se incrustan una o más aplicaciones (durante el proceso de fabricación) que exponen una serie de comandos que se pueden invocar a través de *APIs de programación.

3.5.2. TARJETAS JAVA

Una tarjeta JAVA es una tarjeta capaz de ejecutar mini-aplicaciones Java. En este tipo de tarjetas el sistema operativo es una pequeña máquina virtual Java (JVM) y en ellas se pueden cargar dinámicamente aplicaciones desarrolladas específicamente para este entorno.

3.6. TIPOS DE TARJETAS SEGÚN SUS CAPACIDADES

Según las capacidades de su chip, las tarjetas más habituales son:

- **Memoria**: tarjetas que únicamente son un contenedor de ficheros pero que no albergan aplicaciones ejecutables. Por ejemplo, MIFARE, Éstas se usan generalmente en aplicaciones de identificación y control de acceso sin altos requisitos de seguridad.
- **Microcontroladas**: tarjetas con una estructura análoga a la de un ordenador (procesador, memoria volátil, memoria permanente). Éstas

albergan ficheros y aplicaciones y suelen usarse para identificación y pago con monederos electrónicos.

- **Criptográficas**: tarjetas microcontroladas avanzadas en las que hay módulos hardware para la ejecución de algoritmos usados en cifrados y firmas digitales. En estas tarjetas se puede almacenar de forma segura un certificado digital (y su clave privada) y firmar documentos o autenticarse con la tarjeta sin que el certificado salga de la tarjeta ya que es el procesador de la propia tarjeta el que realiza la firma. Un ejemplo de estas tarjetas son las emitidas por la Fábrica Nacional de Moneda y Timbre (FNMT) española para la firma digital.

3.7. SEGURIDAD

La seguridad es una de las propiedades más importantes de las tarjetas inteligentes y se aplica a múltiples niveles y con distintos mecanismos. Cada fichero lleva asociadas unas condiciones de acceso y deben ser satisfechas antes de ejecutar un comando sobre ese fichero.

En el momento de personalización de la tarjeta (durante su fabricación) se puede indicar qué mecanismos de seguridad se aplican a los ficheros. Normalmente se definirán:

- Ficheros de acceso libre
- Ficheros protegidos por claves: Pueden definirse varias claves con distintos propósitos. Normalmente se definen claves para proteger la escritura de algunos ficheros y claves específicas para los comandos de consumo y carga de las aplicaciones de monedero electrónico. De ese modo la aplicación que intente ejecutar comandos sobre ficheros protegidos tendrá que negociar previamente con la tarjeta la clave oportuna.
- Ficheros protegidos por PIN: El PIN es un número secreto que va almacenado en un fichero protegido y que es solicitado al usuario para acceder a este tipo de ficheros protegidos. Cuando el usuario lo introduce y el programa se lo pasa a la operación que va a abrir el fichero en cuestión el sistema valida que el PIN sea correcto para dar acceso al fichero.

Finalmente, indicar que la negociación de claves se realiza habitualmente apoyándose en un **Módulo SAM**, que no deja de ser otra tarjeta inteligente en formato *ID-000* alojada en un circuito interno propio dentro de la carcasa del lector principal y que contiene aplicaciones criptográficas que permiten negociar las claves oportunas con la tarjeta inteligente del usuario.

3.8. ESTÁNDAR PARA TARJETAS INTELIGENTES ISO/IEC 7816

La norma ISO 7816 es el estándar internacionalmente aceptado para las tarjetas inteligentes. La norma ISO 7816 es una familia de estándares que manejan principalmente aspectos de interoperabilidad relativos a: características de comunicación, de propiedades físicas, de aplicación de identificadores del chip implantado y de datos entre otros.

La Tabla 3.1. presenta un vistazo general que de los estándares ISO 7816. Si se desea profundizar en más en éste estándar y sus partes, puede obtenerlos directamente de la Organización de Estándares Internacionales (ISO) en Suiza.

Estándar	**Descripción**
ISO 7816 – 1	El estándar ISO 7816 – 1 especifica las características físicas de la tarjeta. Las características físicas de la tarjeta incluyen: ▪ Dimensiones ▪ Radiación electromagnética ▪ Esfuerzo mecánico ▪ Localización de la pista magnética

	▪ Localización del circuito integrado de la tarjeta ▪ Resistencia a la electrostática
ISO 7816 – 2	El estándar ISO 7816 – 2 define las dimensiones y localización de los contactos. Éste estándar trata acerca del número, función y localización de los contactos eléctricos. El circuito integrado de la tarjeta tiene 8 contactos eléctricos; están nombrados de C1 a C8, sin embargo, no todos los 8 contactos están conectados al chip del microcontrolador embebido y, por consiguiente, actualmente están sin uso. La Tabla 3.2. contiene la definición de los contactos de acuerdo al estándar ISO 7816 – 2.
ISO 7816 – 3	El estándar ISO 7816 – 3 describe las señales electrónicas y protocolos de transmisión de los circuitos integrados de las tarjetas. La mayor parte del estándar es importante para los fabricantes de lectores y desarrolladores que quieren establecer una comunicación con una tarjeta inteligente en un bajo nivel. A través del estándar ISO 7816 – 3 podemos apreciar las partes involucradas al escribir tu propio software de entrada / salida. Esto puede ser cualquier comunicación desde un microcontrolador ó un puerto RS-232 / paralelo /

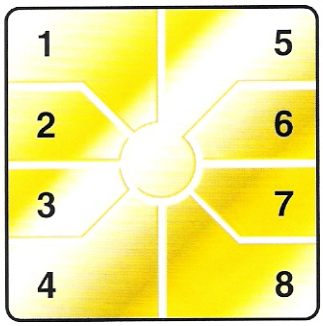
	USB de una computadora.
ISO 7816 – 4	El estándar ISO 7816 – 4 especifica: ▪ Medios de recuperación de elementos y objetos de datos de la tarjeta. ▪ Métodos de acceso a archivos y datos de la tarjeta. ▪ Arquitectura de seguridad que define los derechos de acceso a los archivos y datos en la tarjeta. ▪ Métodos de acceso a los algoritmos procesados por la tarjeta. No describe estos algoritmos. ▪ Comandos para la lectura, escritura y actualización de datos en la tarjeta.

Tabla 3.1. Estándares ISO 7816

Contacto	Designación	Uso
C1	**VCC**	Conector de energía a través del cual se provee energía al chip microprocesador en la tarjeta.
C2	**RST**	Línea a través de la cual se le puede enviar una señal al chip microprocesador de la tarjeta inteligente para iniciar la secuencia de instrucciones de reinicialización (reset).
C3	**CLK**	Línea de señal de reloj. Esta línea controla la velocidad de operación y provee un marco de trabajo común para la comunicación de datos entre el circuito integrado de la tarjeta y el lector.
C4	**RFU**	Reservado para uso futuro.
C5	**GND**	Línea que provee un punto eléctrico común entre el circuito integrado de la tarjeta y el lector.
C6	**VPP**	Conector de energía de programación usado para programar la memoria EEPROM del circuito integrado de la tarjeta.
C7	**I/O**	Línea de entrada / salida que provee un canal de

		comunicación bidireccional no simultánea entre el lector y la tarjeta inteligente.
C8	**RFU**	Reservado para uso futuro.

Tabla 3.2. Definición de contactos de acuerdo al estándar ISO 7816 – 2.



1	Vcc	1	Vcc
2	Reset	2	In
3	CLK	3	CLK
4	Nc	4	Reset
5	GND	5	GND
6	Nc	6	Vpp
7	I/O	7	Out
8	Nc	8	Nc

Figura 3.3. Definición de contactos de acuerdo al estándar ISO 7816 – 2.

3.9. APLICACIONES COMERCIALES

Las dos aplicaciones fundamentales de las tarjetas inteligentes son:

- **Identificación** del titular de la misma.
- **Pago** electrónico.

Las aplicaciones de las tarjetas inteligentes incluyen su uso como tarjeta de crédito, SIM para telefonía móvil, tarjetas de autorización para televisión por pago, identificación de alta seguridad, tarjetas de control de acceso y como tarjetas de pago del transporte público.

Las tarjetas inteligentes también son muy utilizadas como un monedero electrónico. Estas aplicaciones disponen normalmente de un fichero protegido que almacena un contador de saldo y comandos para decrementar e incrementar el saldo (esto último sólo con claves de seguridad especiales). Con esta aplicación, el chip de la tarjeta inteligente puede ser cargado con dinero, que puede ser utilizado en parquímetros, máquinas expendedoras u otros mercados. Protocolos criptográficos protegen el intercambio de dinero entre la tarjeta inteligente y la máquina receptora.

Cuando las tarjetas son criptográficas las posibilidades de identificación y autenticación se multiplican ya que se pueden almacenar de forma segura

certificados digitales o características biométricas en ficheros protegidos dentro de la tarjeta de modo que estos elementos privados nunca salgan de la tarjeta y las operaciones de autenticación se realicen a través del propio chip criptográfico de la tarjeta.

3.10. REFERENCIAS DE FABRICANTES DE TARJETAS Y LECTORES

- Gemplus
- Schlumberger
- Bull
- Oberthur
- Orga
- Solaic
- De la Rue (Philips)
- Fábrica Nacional de Moneda y Timbre (FNMT)
- Microelectrónica Española

CAPÍTULO 4

4. LECTOR DE TARJETAS INTELIGENTES ACR30

OBJETIVOS:

- Definir las características y tipos de tarjetas soportadas por el lector ACR30.
- Presentar los parámetros de comunicación del lector de tarjetas ACR30.
- Describir el formato de los comandos y respuestas del lector ACR30.
- Definir el protocolo de transmisión y recepción del lector.

4.1. INTRODUCCIÓN

El lector/grabador de tarjetas inteligentes ACR30 es una interfase de comunicación entre un controlador (por ejemplo, una PC o un microcontrolador) y una tarjeta inteligente. Diferentes tipos de tarjetas inteligentes tienen diferentes comandos y diferentes protocolos de comunicación. Esto evita en la mayoría de los casos la comunicación directa entre la tarjeta inteligente y el controlador. El lector/grabador ACR30

establece una interfase uniforme desde el controlador a la tarjeta inteligente para una amplia variedad de tarjetas.

El lector/grabador de tarjetas inteligentes ACR30 se conecta al controlador a través de una interfase serial asincrónica ó una interfase USB. El lector/grabador acepta comandos desde el controlador, realiza la función especificada y retorna el dato solicitado o información del estado [1].

4.2. CARACTERÍSTICAS

- Interfase de las tarjeta inteligentes compatible con la norma ISO7816-1/2/3
- Soporta tarjetas microcontroladas con protocolo T = 0 y/o T = 1
- Soporta la selección de parámetros y protocolo (PPS) con 9600 – 96000 bps en la lectura y escritura de las tarjetas inteligentes.
- Interfase RS-232 ó USB con el controlador.
- Soporta las tarjetas de memoria SLE4436 y SLE5536 (*firmware 2.10 en adelante)
- Módulos de aplicación de seguridad (SAM) dentro del lector, que soporta tarjetas microcontroladas con protocolo T = 0 y/o T = 1.

4.3. TIPOS DE TARJETAS SOPORTADAS

El ACR30 puede funcionar con tarjetas microcontroladas con protocolo T = 0 y T = 1. El Apéndice A explica el valor de selección del tipo de tarjeta que debe ser especificado para varios tipos de tarjetas soportados por el lector ACR30.

4.3.1. TARJETAS INTELIGENTES DE MEMORIA (INTERFASE SINCRÓNICA)

- Tarjetas contadoras incluidas:
  - Gemplus GPM103,
  - Siemens SLE4406
  - Siemens SLE4436 y SLE5536 (firmware 2.10 en adelante)
- Tarjetas que siguen el protocolo I2C incluidas:
  - Atmel AT24C01/02/04/08/16
  - Gemplus GFM2K, GFM4K
  - SGS – Thompson ST14C02C, 14C04C
- Siemens inteligentes de 256 bytes de memoria EEPROM con función de protección de escritura:
  - SLE4432
  - SLE4442

- Siemens inteligentes de 1Kb de memoria EEPROM con función de protección de escritura:
  - SLE4418
  - SLE4428

4.3.2. TARJETAS INTELIGENTES MICROCONTROLADAS (INTERFASE ASINCRÓNICA)

El ACR30 soporta tarjetas basadas en microcontroladores con generación de voltaje de programación interno (VPP) y los siguientes parámetros de programación transmitidos en la respuesta a un reset (Answer to reset):

PI1 = 0 ó 5

I = 25 ó 50

El ARC30 ejecuta el procedimiento de selección de parámetros y protocolo (PPS) como se especifica en la norma ISO7816-3:1997.

Cuando la respuesta a un reset (ATR) de la tarjeta indica un modo de operación específico (TA_2 presente, el bit 5 de TA_2 debe ser 0), y que un modo particular es soportado por el ACR30, el lector reseteará la tarjeta

para fijarla en el modo de negociación. Si la tarjeta no puede ser fijada en modo de negociación, el lector rechazará la tarjeta.

Cuando la respuesta a un reset de la tarjeta (ATR) indica modo de negociación (TA_2 no está presente) y los parámetros de comunicación diferentes a los parámetros predeterminados, el ACR30 ejecutará la selección de parámetros y protocolo (PPS) e intentará usar los parámetros de comunicación que la tarjeta sugirió en su respuesta a un reset (ATR). Si la tarjeta no acepta la selección de parámetros y protocolo, el lector utilizará los parámetros predeterminados (F = 372, D = 1).

Para la explicación y significado de lo arriba mencionado, por favor referirse a la norma ISO 7816 – 3.

4.4. INTERFASE

La interfase entre el lector ACR30 y la tarjeta inteligente insertada obedece a las especificaciones de la norma ISO7816-3 con ciertas restricciones o mejoras para incrementar el funcionamiento práctico del ACR30.

4.4.1. FUENTE DE ALIMENTACIÓN PARA LAS TARJETAS INTELIGENTES VCC (C1)

El consumo de corriente de la tarjeta insertada no debe ser mayor que 50 mA.

4.4.2. VOLTAJE DE PROGRAMACIÓN VPP (C6)

De acuerdo a la norma ISO7816-3, el contacto C6 de la tarjeta inteligente (VPP) provee el voltaje de programación a la tarjeta inteligente. Dado que las tarjetas inteligentes más comunes en el mercado son basadas en EEPROM y no requieren de un voltaje externo de programación, el contacto C6 (VPP) ha sido implementado como una señal de control normal en el ACR30.

4.4.3. SELECCIÓN DEL TIPO DE TARJETA

El control siempre tiene que seleccionar el tipo de tarjeta mediante los comandos adecuados enviados al ACR30 antes de activar la tarjeta insertada. Esto incluye a ambas, tarjetas de memoria y tarjetas microcontroladas.

Para las tarjetas microcontroladas el lector permite seleccionar el protocolo preferido, T = 0 ó T = 1. Sin embargo, esta selección sólo es

aceptada y llevada a cabo por el lector a través de la selección de parámetros y protocolo (PPS) cuando la tarjeta insertada en el lector soporta ambos tipos de protocolo. Siempre que una tarjeta microcontrolada soporte sólo un tipo de protocolo, T=0 ó T=1, el lector automáticamente usa ese tipo de protocolo, independientemente del tipo de protocolo seleccionado por la aplicación.

4.4.4. INTERFASE CON TARJETAS INTELIGENTES MICROCONTROLADAS[1]

Para las tarjetas inteligentes microcontroladas solamente los contactos C1 (VCC), C2 (RST), C3 (CLK), C5 (GND) y C7 (I/O) son usados. Una frecuencia de 3.6864 / 4 MHz se aplica a la señal de reloj (C3).

[1] Refiérase a la Figura 3.3 que muestra la definición de contactos de acuerdo con el Estándar ISO 7816 – 2

4.4.5. PROTECCIÓN CONTRA DESCONEXIÓN

El ACR30 provee un mecanismo para proteger a la tarjeta insertada cuando es removida repentinamente mientras esta siendo energizada. La fuente de alimentación a la tarjeta y las líneas de señal entre el ACR30 y la tarjeta se desactivan inmediatamente cuando la tarjeta esta siendo removida. Como regla general, sin embargo, para evitar cualquier daño eléctrico, una tarjeta sólo debe ser removida del lector cuando esta apagado.

4.5. FUENTE DE ALIMENTACIÓN

El lector/grabador ACR30 requiere un voltaje de 5V DC, fuente de alimentación regulada 100 mA.

LEDs de estado

El diodo LED de color verde en la parte frontal del lector indica el estado de activación de la interfase de la tarjeta inteligente.

4.6. INTERFASE SERIAL

El ACR30 es conectado al controlador a través de una interfase serial asincrónica obedeciendo el estándar RS-232.

Para la comunicación entre el ACR30 y una controladora, cinco líneas de la interfase RS-232 son usadas: RxD, TxD, CTS, DTR y GND.

PIN	CPU	Lector	Función
2	RxD	TxD	Dato transmitido desde el controlador al lector ACR30.
3	TxD	RxD	Dato transmitido desde el lector ACR30 al controlador.
4	DTR	*RESET	Señal de entrada *RESET. Permite ejecutar una reset por hardware del lector a través de la interfase RS-232.
5	GND	*GND	Nivel de voltaje de referencia para la fuente de alimentación y la interfase serial.
8	CTS	*Busy	Indica al controlador si el ACR30 esta listo para recibir el siguiente comando.

Tabla 4.1. Alambrado de la interfase RS-232 del lector ACR30.

4.6.1. PARÁMETROS DE COMUNICACIÓN

Los siguientes parámetros de comunicación son usados por el ACR30 y no pueden ser modificados por el servidor:

Protocolo de transmisión: Serial asincrónico

Paridad : Ninguna

Bits de datos : 8

Bits de parada : 1

Handshake : A través de CTS

El ACR30 provee dos medios para seleccionar la velocidad de transmisión usada por el lector en su operación normal: hardware y/o software.

4.6.1.1. VELOCIDAD POR HARDWARE

El valor predeterminado por hardware es de 9600 baudios.

4.6.1.2. SELECCIÓN DE LA VELOCIDAD POR SOFTWARE

El comando SET_PROTOCOL permite fijar la velocidad de transmisión y un tiempo de retraso insertado entre los bytes transmitidos por el lector al controlador.

Se debe tener en cuenta que la programación de la velocidad hecha con este comando es volátil y se perderá cuando el lector sea *reseteado o encendido la próxima ocasión.

4.7. PROTOCOLO DE COMUNICACIÓN

La comunicación se lleva a cabo a través de intercambios sucesivos Comando – Respuesta. El controlador transmite un comando al lector y recibe una respuesta desde este después que el comando ha sido ejecutado. Un nuevo comando puede ser transmitido al ACR30 sólo después que la respuesta del comando previo ha sido recibida.

Hay dos casos donde el lector transmite datos sin haber recibido un comando desde el controlador y son el *mensaje de *reset* del lector y el *mensaje de estado de la tarjeta*.

4.7.1. FORMATO DE LOS COMANDOS

4.7.1.1. FORMATO NORMAL (LONGITUD < 255 BYTES)

Un comando consiste en cuatro bytes de protocolo y un número variable de bytes de datos con la siguiente estructura:

byte	1	2	3	4… N+3 (0<N< 255)	N+4
	Encabezado	Instrucción	Longitud del datos = N	Datos	*Checksum

Encabezado	Es siempre 01_H para indicar el inicio de un comando.
Instrucción	El código de instrucción a ser ejecutado por el ACR30
Longitud de datos	Longitud del dato en bytes. (0<N<255)
Datos	Contenido de datos del comando. Para un comando de LECTURA (READ), por ejemplo, los datos especificarían la dirección inicial y el número de bytes que serán leídos. Para un comando de ESCRITURA (WRITE) los datos especificarían la dirección inicial y los datos que se escribirán en la tarjeta. Los datos representan valores que serán escritos en la tarjeta y/o parámetros de comandos tales como direcciones, contador, etc.
***Checksum**	El *Checksum se realiza mediante una operación XOR de todos los bytes de comando incluyendo encabezado, instrucción, longitud de datos y los bytes de datos.

El siguiente ejemplo muestra la estructura de un comando con código de instrucción 91_H y tres bytes de datos con los valores 11_H, 22_H y 33_H respectivamente:

byte	1	2	3	4	5	6	7
	01_H	91_H	03_H	11_H	22_H	33_H	93_H

4.7.1.2. FORMATO EXTENDIDO (LONGITUD > 255 BYTES)

Un comando consiste en seis bytes de protocolo y un número variable de bytes de datos con la siguiente estructura:

byte	1	2	3	4…	5	6… N+5 (N>0)	N+6

Encabezado	Instrucción	Longitud de datos = N		Datos	*Checksum
		FF_H	Longitud de datos N		

Encabezado Es siempre 01_H para indicar el inicio de un comando.

Instrucción El código de instrucción a ser ejecutado por el ACR30

Longitud de datos Longitud del dato en bytes. Es codificado en tres bytes. El primer byte es FF_H. El segundo y tercer byte representa la longitud del dato N.

Datos Contenido de datos del comando.

Para un comando de LECTURA (READ), por ejemplo, los datos especificarían la dirección inicial y el número

	de bytes que serán leídos. Para un comando de ESCRITURA (WRITE) los datos especificarían la dirección inicial y los datos que se escribirán en la tarjeta. Los datos representan valores que serán escritos en la tarjeta y/o parámetros de comandos tales como direcciones, un contador, etc.
***Checksum**	El *Checksum se realiza mediante una operación XOR de todos los bytes de comando incluyendo encabezado, instrucción, longitud de datos y los bytes de datos.

4.7.2. FORMATO DE LA RESPUESTA

La respuesta del ACR30 a cualquier comando depende si el comando fue recibido por el lector sin errores.

4.7.2.1. RESPUESTA NORMAL SIN ERROR DE TRANSMISIÓN (LONGITUD < 255 BYTES)

La respuesta del ACR30 a un comando recibido correctamente consiste de tres bytes de protocolo, dos bytes de estado y un número variable de bytes de datos y tiene la siguiente estructura:

byte	1	2	3	4	5…	N+4	N+5
					(0<N< 255)		

Encabezado	SW1	SW2	Longitud de datos = N	Datos	*Checksum

Encabezado Es siempre 01_H para indicar el inicio de una respuesta.

SW1 Indica el estado de ejecución del comando.

- 90_H = Comando ejecutado exitosamente.
- 60_H = Error en comando *datos*; el comando no pudo ser ejecutado.
- 67_H = Error detectado en la ejecución del comando.
- FF_H = *Mensaje de estado* iniciado por el lector.

SW2 Información adicional del estado de ejecución de los comandos.

Una tabla indicando los posibles valores de los bytes de estado SW1 y SW2 y su correspondiente significado se muestra en el Apéndice B.

Longitud de datos Longitud de los datos en bytes. (0<N<255)

Datos Contenido de datos del comando.

Para un comando READ_DATA, por ejemplo, los datos tendrían el contenido de las direcciones de memoria leídas desde la tarjeta. Los datos pueden representar valores leídos desde el lector y/o información del estado.

***Checksum** El *Checksum se realiza mediante una operación XOR de todos los bytes de comando incluyendo encabezado, bytes de estado, longitud de datos y los bytes de datos.

El siguiente ejemplo muestra la estructura de una respuesta a un comando que ha sido satisfactoriamente ejecutado y que retorna tres datos con los valores 11_H, 22_H, 33_H, respectivamente:

byte	1	2	3	4	5	6	7	8
	01_H	90_H	00_H	03_H	11_H	22_H	33_H	92_H

4.7.2.2. RESPUESTA EXTENDIDA SIN ERROR DE TRANSMISIÓN

La respuesta del lector a un comando recibido correctamente consiste de tres bytes de protocolo, dos bytes de estado y un número variable de bytes de datos. Tiene la siguiente estructura:

byte	1	2	3	4	5	6	7… N+6	N+7

Encabezado	SW 1	SW2	Longitud de datos = N		Datos	*Checksum
			FF_H	Longitud de datos N		

Encabezado Es siempre 01_H para indicar el inicio de una respuesta.

SW1 Indica el estado de ejecución del comando.

- 90_H = Comando ejecutado exitosamente.
- 60_H = Error en comando datos; el comando no pudo ser ejecutado.
- 67_H = Error detectado en la ejecución del comando.
- FF_H = Mensaje de estado iniciado por el lector.

SW2	Información adicional del estado de ejecución de los comandos. Una tabla indicando los posibles valores de los bytes de estado SW1 y SW2 y su correspondiente significado se muestra en el Apéndice B.
Longitud de datos	Longitud del dato en bytes. Es codificado en tres bytes. El primer byte es FF_H. El segundo y tercer byte representa la longitud del dato N.
Datos	Contenido de datos del comando. Para un comando READ_DATA, por ejemplo, los datos tendrían el contenido de las direcciones de memoria leídas desde la tarjeta. Los datos pueden representar valores leídos desde el lector y/o información del estado.
***Checksum**	El *Checksum se realiza mediante una operación XOR de todos los bytes de comando incluyendo encabezado, bytes de estado, longitud de datos y los bytes de datos.

4.7.2.3. ERROR DE TRANSMISIÓN

Si la parte receptora de un comando (ACR30) o una respuesta (Controlador), detecta un error de longitud de datos o de Checksum

de un comando, no se toma en cuenta el dato recibido y se envía un mensaje de “NO RECONOCIDO” (NOT AKNOWLEDGE) a la parte transmisora.

byte	1	2
	05_H	05_H

Si el ACR30 responde con un mensaje de “NO RECONOCIDO” a un comando del controlador; el controlador normalmente transmitiría el comando nuevamente. Si el controlador detecta un error de transmisión en una respuesta del ACR30, puede enviar un mensaje de “NO RECONOCIDO” al lector a lo que el lector transmitiría nuevamente la respuesta más reciente.

4.7.3. MENSAJE DE RESET

Un reset del lector ocurre automáticamente siempre que el lector es energizado. El lector transmite una vez un mensaje de reset, que tiene la misma estructura que una respuesta normal a un comando y contiene lo siguiente:

byte	1	2	3	4	5	6
	Encabezado	**SW1**	**SW2**	**Longitud de datos = N**	**Datos**	***Checksum**
	01_H	FF_H	00_H	01_H	BAUD = 12_H	

BAUD Indica la velocidad fijada de los baudios del hardware (Valor predeterminado de la velocidad de comunicación en baudios), que es 9600 baudios. El lector no espera una señal de reconocimiento desde el controlador para enviar un mensaje de reset. Después de enviar un mensaje de reset el lector espera por el primer comando proveniente del controlador.

4.7.4. MENSAJE DE ESTADO DE LA TARJETA

Cuando una tarjeta es insertada en el lector ó es removida desde el mismo, no en la ejecución de un comando, el lector transmite un mensaje

de estado de la tarjeta, para notificar al servidor sobre algún cambio de estado en la inserción de la tarjeta.

En un sistema donde estos mensajes no son deseados, pueden ser desactivados usando el comando SET_NOTIFICATION. Los mensajes de estado de la tarjeta tienen la siguiente estructura y contienen:

Mensaje de estado de tarjeta para: tarjeta insertada

byte 1 2 3 4 5 6

Encabezado	**SW1**	**SW2**	**Longitud de datos = N**	**Datos**	**Checksum**
01_H	FF_H	00_H	01_H	00_H	FF_H

Mensaje de estado de tarjeta para: tarjeta removida

byte 1 2 3 4 5 6

Encabezado	**SW1**	**SW2**	**Longitud de datos = N**	**Datos**	**Checksum**
01_H	FF_H	00_H	02_H	00_H	FC_H

Un mensaje de estado de tarjeta es transmitido solamente una vez por cada evento inserción o remoción de una tarjeta. El lector no espera una señal de reconocimiento del controlador. Después de enviar un mensaje de estado, el lector espera el siguiente comando del controlador.

4.7.5 PROTOCOLO DE TRANSMISIÓN

El inicio de un comando (al lector) o una respuesta (desde el lector, incluyendo los mensajes de reset y de estado de la tarjeta) se indica por la parte respectiva a través de la transmisión de un solo byte de *inicio de texto* (STX) con el valor de 02_H.

El fin de un comando o respuesta se indica a través de un solo byte de *fin de texto* (ETX) con el valor de 03_H.

Dentro de la transmisión del comando y la respuesta los caracteres ASCII están representados con dígitos hexadecimales (HEX). Cada byte de un comando o una respuesta se divide en sus nibbles más significativo y menos significativo. Para cada nibble se transmite el código ASCII que representa su respectivo valor hexadecimal. Por ejemplo, para transmitir el byte de dato $3A_H$, se envían dos bytes y son 33_H (Código ASCII para '3') seguido por 41_H (Código ASCII para 'A'):

Valor del byte de datos $3A_H$

Valores transmitidos 33_H = ‘3’ 41_H = ‘A’

El siguiente ejemplo muestra la transmisión de un comando con código de instrucción $A2_H$ y un byte de dato con el valor de $3D_H$. El comando tiene la siguiente estructura:

byte	1	2	3	4	5
	Encabezado	**Instrucción**	**Longitud de datos**	**datos**	**Checksum**
	01_H	$A2_H$	01_H	$3D_H$	$9F_H$

El comando se transmite en la interfase serial en 12 bytes como sigue:

byte	1	2	3	4	5	6	7	8	9	10	11	12
	STX	**‘0’**	**‘1’**	**‘A’**	**‘2’**	**‘0’**	**‘1’**	**‘3’**	**‘D’**	**‘9’**	**‘F’**	**ETX**
	02_H	30_H	31_H	41_H	32_H	30_H	31_H	33_H	44_H	39_H	46_H	03_H

Para representar los valores hexadecimales de los nibbles con sus respectivos caracteres ASCII en comandos, el ACR30 acepta los

caracteres en mayúsculas ‘A’… F’ (41_H… 46_H) y en minúsculas ‘a’… ‘f’ (61_H… 66_H):

byte	1	2	3	4	5	6	7	8	9	10	11	12
	STX	**‘0’**	**‘1’**	**‘A’**	**‘2’**	**‘0’**	**‘1’**	**‘3’**	**‘D’**	**‘9’**	**‘F’**	**ETX**
	02_H	30_H	31_H	41_H	32_H	30_H	31_H	33_H	44_H	39_H	46_H	03_H

Es equivalente a:

byte	1	2	3	4	5	6	7	8	9	10	11	12
	STX	**‘0’**	**‘1’**	**‘a’**	**‘2’**	**‘0’**	**‘1’**	**‘3’**	**‘d’**	**‘9’**	**‘f’**	**ETX**
	02_H	30_H	31_H	61_H	32_H	30_H	31_H	33_H	64_H	39_H	66_H	03_H

En sus mensajes de respuesta, el ACR30 usa caracteres en mayúscula ‘A’… ‘F’.

4.8. COMANDOS PARA TARJETAS MICROCONTROLADAS

Los comandos ejecutados por el ACR30 generalmente pueden ser divididos en dos categorías que son comandos de control y comandos de tarjeta.

Los comandos de control se preocupan de la operación interna del ACR30. Ellos no aceptan directamente a la tarjeta insertada en el lector y además son independientes del tipo de tarjeta seleccionada.

Los comandos de tarjeta son dirigidos hacia la tarjeta insertada en el ACR30 [1]. La estructura de estos comandos, datos transmitidos en los comandos y sus respuestas dependen del tipo de tarjeta seleccionada.

4.8.1. COMANDOS DE CONTROL

GET_ACR_STAT

Este comando retorna información relevante acerca del modelo ACR30, y el corriente estado de operación, tal como, el número de revisión del *firmware, la longitud máxima de datos de los comandos y la respuesta, los tipos de tarjetas soportadas, y si la tarjeta es insertada y energizada.

Formato del comando

Código de instrucción	Longitud de datos
01_H	00_H

Formato de dato de respuesta

INTERNO										MAX_C	MAX_R	C_TYPE		C_SEL	C_STAT

INTERNO 10 bytes de datos para uso interno solamente.

MAX_C El número máximo de comandos de datos.

MAX_R El número máximo de datos que pueden ser solicitados en una respuesta.

C_TYPE Tipos de tarjetas soportadas por el ACR30. Este campo es un mapa de bits; cada bit representa un tipo particular de tarjeta. Un bit fijado en '1' significa que el correspondiente tipo de tarjeta es soportado por el lector y puede ser seleccionado con el comando SELECT_CARD_TYPE.

Véase el Apéndice A para los valores correspondientes entre estos bits y sus respectivas tarjetas.

C_SEL El tipo de tarjeta seleccionada actualmente. Un valor de 00_H significa que no se ha seleccionado algún tipo de tarjeta.

C_STAT Indica si la tarjeta es insertada físicamente en el lector, también indica si esta encendida:

00_H : Tarjeta no insertada.

01_H : Tarjeta insertada y no encendida.

03_H : Tarjeta encendida.

SET_PROTOCOL

Este comando se usa para controlar la velocidad de comunicación entre el lector ACR30 y el dispositivo controlador. La velocidad de comunicación se controla por medio de dos factores: el factor de retraso y la tasa de baudios.

Formato del comando

Para cambiar solamente el factor de retraso, ó:

Código de instrucción	**Longitud de datos**	**Datos**
		Retraso N
03_H	01_H	

Para cambiar la factor de retraso y la velocidad de comunicación en baudios.

Código de instrucción	**Longitud de datos**	**Datos**	
		Retraso N	**Baudios**
03_H	02_H		

Retraso N Determina el tiempo de retraso insertado por el ACR30 entre dos bytes consecutivos para adaptarse a la velocidad más baja del sistema controlador. El tiempo esta dado por 0.1 * N ms. Con N desde 0… 255 (00 - FF_H). El valor predeterminado es N = 0.

Baudios Selecciona la velocidad en baudios de la interfase serial entre el lector y el sistema servidor. El valor predeterminado es de 9600 baudios.

Baudios	Velocidad serial en baudios
12_H	9600
11_H	19200
10_H	38400
03_H	14400
02_H	28800
01_H	57600
00_H	115200

Formato de datos de respuesta

No hay respuesta de datos.

SELECT_CARD_TYPE

Este comando fija el tipo de tarjeta requerida. El firmware en el ACR30 ajusta el protocolo de comunicación entre el lector y la tarjeta insertada de acuerdo al tipo de tarjeta seleccionada.

Formato del comando

Código de instrucción	**Longitud de datos**	**Datos**
		Tipo
02_H	01_H	

Tipo Véase el Apéndice A para obtener el valor que debe ser especificado en este comando para una tarjeta particular a ser usada.

Formato de datos de respuesta

No hay respuesta de datos.

RESET

Esta sección describe el comando RESET sólo para el caso en que ningún tipo de tarjeta ha sido seleccionado ó cuando el tipo de tarjeta 00_H es seleccionado. Para todos los demás casos hay que referirse a la sección específica descrita para cada tipo de tarjeta.

Formato del comando

Código de instrucción	Longitud de datos
80_H	00_H

Formato de datos de respuesta

ATR			

ATR Respuesta a un reset retornada por la tarjeta.

El retorno de códigos de estado para este comando es 90 00_H cuando la tarjeta insertada es una tarjeta con protocolo T = 0, 90 01_H cuando la tarjeta insertada es una tarjeta con protocolo T = 1, y 90 10_H cuando la

tarjeta insertada es una tarjeta de memoria bajo otras circunstancias el código de estado es 60 20_H.

SET_NOTIFICATION

Este comando habilita o deshabilita los mensajes de estado de la tarjeta transmitidos por el lector para notificar al controlador si una tarjeta ha sido insertada o removida.

Formato del comando

Código de instrucción	**Longitud de datos**	**Datos**
		Notificación
06_H	01_H	

Notificación Especifica si el mensaje de estado de la tarjeta será transmitido para notificar al controlador si una tarjeta ha sido insertada o removida.

01_H : Transmite el *mensaje de estado* de la tarjeta.

02_H : No transmite el *mensaje de estado* de la tarjeta.

Formato de datos de respuesta

No hay respuesta de datos.

4.8.2. COMANDOS DE LA TARJETA

Los comandos disponibles y los parámetros especificados en los comandos de respuesta así como los datos transmitidos en la respuesta por el ACR30 dependen del tipo de tarjeta seleccionada.

Las tarjetas basadas en microcontroladores o tarjetas microcontroladas tienen los siguientes comandos:

***RESET**

Este comando enciende la tarjeta insertada en el lector y realiza un reset en la tarjeta. La fuente de alimentación de la tarjeta no se apaga.

Formato del comando

Código de instrucción	Longitud de datos
80_H	00_H

Formato de datos de respuesta

ATR					

ATR Respuesta a un reset transmitida por la tarjeta del acuerdo al estándar ISO 7816 – 3.

POWER_OFF

Este comando apaga la tarjeta insertada en el lector de tarjetas inteligentes.

Formato del comando

Código de instrucción	Longitud de datos
81_H	00_H

Formato de datos de respuesta

No hay respuesta de datos.

EXCHANGE_APDU

Sirve para intercambiar un par comando/respuesta en la Unidad de Aplicación de Protocolo de datos (APDU) entre la tarjeta microcontrolada insertada en el ACR30 y el controlador.

Formato del comando

Código de Instrucción	Longitud de datos	Datos									
	LEN	CLA	INS	P1	P2	L_C	BYTE 1	… 2	…	BYTE N	L_e
$A0_H$											

LEN Longitud del dato de comando del APDU, N, + 6 (0 < N < MAX_R).

CLA Clase de instrucción APDU.

INS Instrucción APDU.

P1 Byte 1 de Parámetro APDU.

P2 Byte 2 de Parámetro APDU.

L_C Comando APDU de longitud de datos.

BYTE X Comando de datos APDU.

L_e Longitud esperada en respuesta del APDU.

Formato de datos de respuesta

BYTE 1	…	…	BYTE N	SW1	SW2

BYTE X Respuesta de datos desde la tarjeta (Si hay alguna).

SW1, SW2 Código de estado retornado por la tarjeta.

CAPÍTULO 5

5. TARJETA INTELIGENTE ACOS2

OBJETIVOS:

- Definir la configuración interna de la tarjeta, sus archivos y registros.
- Aprender Como personalizar la tarjeta y configurarla correctamente.
- Dar a conocer los comandos de la tarjeta.
- Definir los mensajes de estado de la tarjeta.

5.1. CICLO DE VIDA DEL CHIP

Durante el ciclo entero de vida de la tarjeta con chip, tres etapas y un modo de operación pueden ser distinguidos:

- Etapa de Manufactura
- Etapa de Personalización
- Etapa de Usuario
    - Etapa de Usuario – Modo Emisor (Modo de operación)

5.1.1. ETAPA DE MANUFACTURA

La etapa de Manufactura entra en operación desde el momento de fabricación del chip hasta que el fusible asociado (bit determinado en la memoria EEPROM), el cual es llamado *fusible de manufactura*, sea programado.

Una vez que el fusible de manufactura ha sido programado este no se puede regresar a su estado original.

En la etapa de Manufactura, cualquier acceso a escritura en los archivos internos, así también la lectura de los archivos de seguridad, solo es posible después de la presentación correcta del código inicial (IC).

El código IC es programado en el microcontrolador ACOS2 durante el proceso de fabricación del chip [2].

El código IC es presentado a la tarjeta de una manera simple, sin encriptación.

Los siguientes datos son escritos en la memoria EEPROM en la Etapa de Manufactura.

- *El archivo de manufactura,* contiene 3 registros de 8 bytes cada uno asociado a la etapa de fabricación. Este archivo sólo puede ser escrito en esta etapa. Después de programar el fusible de fabricación el archivo de manufactura es solo de lectura.
- *El código IC* para la etapa de personalización, debe ser presentado antes para que la tarjeta permita el acceso para escribir datos en los archivos en la etapa de personalización, que es aplicado inmediatamente después de la etapa de manufactura.
- *El fusible de fabricación*, cambia irreversiblemente el ciclo de vida de la tarjeta de la etapa de manufactura a la etapa de personalización. Este fusible es un bit en los 16 bytes del archivo de manufactura.

La tarjeta esta en cualquier momento en una de tres etapas. El siguiente diagrama muestra las posibles transiciones entre las etapas.

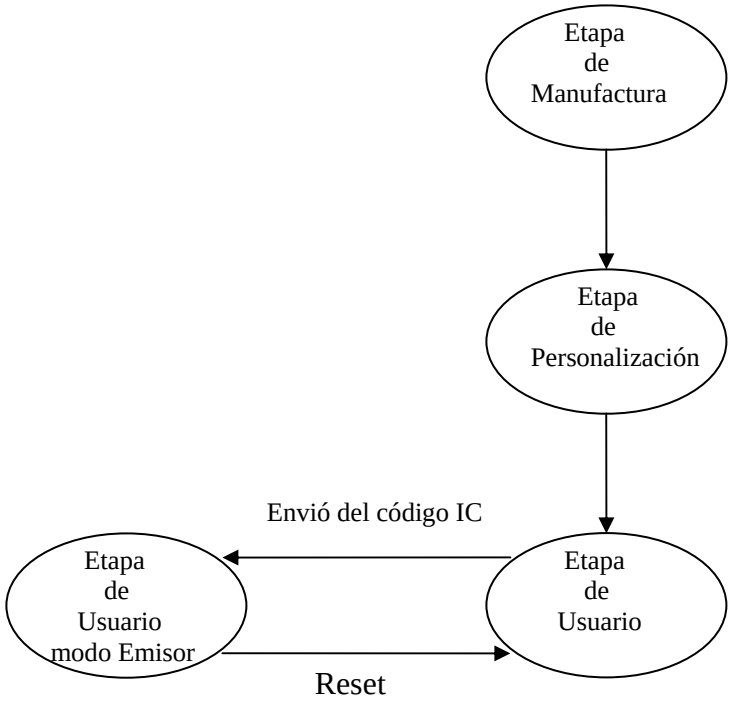


Figura 5.1. Ciclo de vida de la tarjeta inteligente ACOS2

La etapa actual del ciclo de vida del chip esta determinada por el sistema operativo de la tarjeta inmediatamente después de ejecutar un reset, la cual no cambia durante la operación de la tarjeta.

5.1.2. ETAPA DE PERSONALIZACIÓN

La Etapa de Personalización es efectiva desde el momento que finaliza la etapa de manufactura hasta que el bit asociado en la EEPROM, llamado *bit de personalización*, sea programado.

Una vez que el bit de personalización ha sido programado y la etapa de personalización ha sido terminada, el usuario puede volver a la etapa de personalización cuando se encuentre en el *modo de usuario*, ejecutando el comando CLEAR CARD. Este comando borra la memoria EEPROM, excepto el área de fabricación, y así de esta manera la tarjeta regresa al estado que tenia antes de que el bit de personalización fuera programado. Note que si el comando CLEAR CARD esta deshabilitado no hay forma de volver a la etapa de personalización.

En la etapa de Personalización, cualquier acceso de escritura en los archivos internos, así también la lectura de los archivos de seguridad, sólo es posible después de la presentación correcta del código IC.

El código IC es presentado a la tarjeta de una manera simple, sin encriptación.

Los siguientes datos son escritos en la memoria EEPROM en la Etapa de Personalización.

- *El Archivo de Personalización*, contiene 3 registros de 4 bytes, cada uno asociado a la *Etapa de Personalización*, incluyendo el Registro de Opciones. Esta área solo puede ser escrita en la *Etapa de Personalización* después de programar el *bit de Personalización*, el *Archivo de Personalización* es solo de lectura. Los 10 primeros bytes *del Archivo de Personalización* son transmitidos en los bytes del Historial.
- *El Archivo de Definición de Bloques* requeridos por los *Archivos de Datos del Usuario.*
- *El Bit de Personalización*, el cual cambia el ciclo de vida de la *Etapa de Personalización* a la *Etapa de Usuario.*

5.1.3. ETAPA DE USUARIO

La *Etapa de Usuario* designa el modo de operación 'normal' de la tarjeta. La *Etapa de Usuario* se ejecuta desde el momento en el que la Etapa de Personalización se termina hasta que el *Código Emisor* es enviado a la tarjeta. Una vez que este código es enviado, cambia la

operación de la tarjeta y ésta pasa a la etapa de *Modo Emisor* (modo privilegiado que permite el acceso a determinadas áreas de memoria, que de otra manera no seria accesible).

5.2. MANEJO DE LA MEMORIA EEPROM

Los 8k Bytes de memoria EEPROM provistas por el chip de la tarjeta están básicamente segregados en *Memoria de Datos Internos* y *Memoria de Datos del Usuario:*

- *La Memoria de Datos Internos* se usa para guardar la configuración de los datos y también es usada por el sistema operativo para manejar determinadas funciones.
- *La memoria de Datos del Usuario* guarda los datos manipulados en el uso normal de la tarjeta bajo el control de la aplicación.

5.2.1. ARCHIVOS DE DATOS

Acceder a la *Memoria de Datos Interna* y a la *Memoria* de *Datos del Usuario* es posible dentro del rango de los archivos de datos y los registros de datos. Los *Archivos de Datos* son referidos como *Archivos de Datos internos.* Los *Archivos de Datos* en la *Memoria de Datos del Usuario* son llamados *Archivos de datos del Usuario.*

Los *Archivos de Datos* son la entidad más pequeña que posee atributos de seguridad individuales que pueden ser asignados para controlar el acceso a la lectura y escritura de los datos guardados en la EEPROM.

Los *Archivos de Datos* están compuestos por registros. Un registro de datos es la unidad de datos más pequeña que individualmente puede ser direccionada en un *archivo de datos*. Cada *archivo de datos* contiene N registro de datos. El número de registro debe ser especificado cuando un registro (o un dato dentro de un registro) es leído o escrito en un archivo. Un *archivo de datos* contiene máximo 255 registros. La longitud del registro puede ser diferente para archivos diferentes pero es siempre fijo dentro de un archivo. *

La estructura de los *Archivos Internos de Datos* (dimensión archivo, identificador archivo, longitud registro, atributos seguridad) es definida por el sistema operativo y no puede ser cambiado. La estructura para la *Memoria de Datos del usuario* es determinada en la personalización de la tarjeta. Después de programar el parámetro N_OF_FILE en la Etapa de Personalización. La estructura del archivo es fija.

El acceso a todos los archivos sólo es posible a través de los comandos *READ RECORD y *WRITE RECORD.

Cada archivo es identificado por dos bytes del Archivo Identificador. El *Archivo de Identificación* es asignado a cada archivo cuando el archivo es definido durante la *Etapa de Personalización*. El sistema operativo no realiza ningún chequeo de unicidad de cada Archivo Identificador. Si el mismo identificador ha sido asignado a más de un archivo, puede ocurrir una mala operación en la tarjeta.

El valor de FF es el primer byte del archivo identificador es usado por los Archivos de Datos Internos y no puede ser usado por el Archivo de Datos del Usuario.

Después de acceder a un archivo, ya sea por lectura o escritura (*READ *RECORD o *WRITE RECORD), el archivo debe ser abierto a través del comando *SELECT FILE. Solo un archivo puede ser seleccionado a la vez. Los comandos *READ RECORD, *WRITE RECORD se ejecutan sobre el archivo recientemente seleccionado.

5.2.2. ARCHIVO DE CONTROL DE ACCESO

Dos archivos de seguridad son asignados a cada *Archivo de Datos*: el *Atributo de Seguridad de Lectura* y *Atributo de Seguridad de Escritura.* Los *atributos de Seguridad* definen las condiciones de seguridad que deben ser llenadas para permitir la respectiva operación:

- El Atributo de Seguridad de Lectura controla el acceso a la lectura de los datos en el archivo a través del comando READ RECORD. Si la condición de seguridad especificada en el *Atributo de Seguridad de Lectura* no es llenado la tarjeta rechazará el comando *READ RECORD para ese archivo.
- El Atributo de Seguridad de Escritura controla el acceso a la escritura de los datos en el archivo a través del comando *WRITE RECORD. Si la condición de seguridad especificada en el *Atributo de Seguridad de Escritura* no es llenado la tarjeta rechazará el comando WRITE RECORD para ese archivo.

El Atributo de Seguridad de Lectura y Escritura para cada archivo específica que código de aplicación, si hay alguno, debe ser enviado correctamente a la tarjeta para que esta permita la respectiva operación así como el Código de Usuario, el código PIN o el IC (código inicial) debe ser enviado para una aplicación especifica.

EL código de aplicación AC0 puede ser especificado en el Atributo de Seguridad, pero no puede ser enviado a la tarjeta.

Para los Archivos de Datos Internos, los atributos de seguridad son fijos en el sistema operativo de la tarjeta. Para los Archivos de Datos del

Usuario, los atributos de seguridad de un archivo son almacenados en el Bloque de Definición de Archivos.

La siguiente tabla lista algunos ejemplos de condiciones de seguridad que pueden ser especificados para los Archivos de Datos del Usuario:

Atributo de Seguridad	**Condición de Seguridad**
-	No tiene restricción, acceso libre
AC x	Acceso solo después del envió correcto del AC x
AC x, AC y, AC z	Acceso solo después del envió correcto de AC x, AC y, AC z
IC	Acceso solo después del envió correcto de IC
PIN	Acceso solo después del envió correcto del PIN
PIN, IC	Acceso solo después del envió correcto del PIN y del IC
AC x, IC	Acceso solo después del envió correcto del AC x del IC
AC x, PIN, IC	Acceso solo después del envió correcto de AC x, PIN y del IC
AC0	No hay Acceso
AC x	Requiere un Código de Aplicación
PIN	Requiere el código PIN
IC	Requiere el Código Emisor (código de usuario)

Tabla 5.1. Condiciones de seguridad que pueden ser especificados para los Archivos de Datos del Usuario.

Un *Atributo de Seguridad* esta definido en un byte como el siguiente:

Bit 7 bit 0

IC	PIN	AC5	AC4	AC3	AC2	AC1	AC0

Cada bit del byte representa un código. Si el bit es fijado a ‘1’, el correspondiente código debe ser enviado. Si el bit es fijado a ‘0’, el correspondiente código es una condición irrelevante para el acceso.

5.2.3. ARCHIVOS DE DATOS INTERNOS

Con la excepción de la Estructura de Datos de Cuentas, que están asociadas a un conjunto especial de comandos, las áreas de memoria de los *Datos Internos de Memoria* son procesadas como archivos de datos.

Los atributos de los Archivos Internos de Datos son definidos en el sistema operativo y no pueden ser cambiados. Sin embargo, los atributos de seguridad dependen de la etapa del ciclo de vida de la tarjeta.

Los siguientes *Archivos de Datos Internos* son los definidos:

	ARCHIVO DE ATRIBUTOS DE SEGURIDAD				Organización
Área de memoria	**ID del archivo interno**	**Etapa de Manufactura**	**Etapa de Personalización**	**Etapa de Usuario**	**de los registros**
Archivo MCU-ID	FF 00	R: LIBRE W: SIN ACCESO	R: LIBRE W: SIN ACCESO	R: LIBRE W: SIN ACCESO	2 X 8 bytes
Archivo de Manufactura	FF O1	R: LIBRE W: IC	R: LIBRE W: SIN ACCESO	R: LIBRE W: SIN ACCESO	2 X 8 bytes
Archivo de Personalización	FF 02	R: LIBRE W: IC	R: LIBRE W: IC	R: LIBRE W: SIN ACCESO	3 X 4 bytes
Archivo de Seguridad	FF 03	R: IC W: IC	R: IC W: IC	R: SIN ACCESO W: IC	12 X 8 bytes
Archivo de manejo de archivos de Usuario	FF 04	R: LIBRE W: IC	R: LIBRE W: IC	R: LIBRE W: IC	N_OF_FILES X 6 bytes
Archivo de Cuentas	FF 05	R: LIBRE W: IC	R: LIBRE W: IC	R: IC W: IC	8 X 4 bytes
Archivo de seguridad de cuentas	FF 06	R: LIBRE W: IC	R: LIBRE W: IC	R: SIN ACCESO W: IC	4 X 8 bytes

Área de archivos de datos del usuario	xx yy xx ≠ FF	DEACUERDO CON LAS DEFINICIONES DEL ARCHIVO

Tabla 5.2. Archivo de atributos de seguridad

5.2.3.1. ARCHIVO DE IDENTIFICACIÓN DEL MCU

El archivo de identificación del microcontrolador (MCU) contiene dos registros de 8 bytes cada uno. El contenido de estos archivos es determinado durante el proceso de fabricación del chip y no puede ser alterado.

5.2.3.2. ARCHIVO DEL FABRICANTE

El archivo de manufactura comprende dos registros de 8 bytes cada uno que son escritos en la etapa de manufactura del ciclo de vida de la tarjeta. Después de que finaliza la *Etapa de Manufactura*, este archivo es sólo de lectura.

NOTA: En las tarjetas ACOS2, el *Archivo de Manufactura* es normalmente escrito y el Fusible de Manufactura es programado

antes de ser entregadas, por lo tanto, el usuario de las tarjetas no tiene el control sobre los contenidos del Archivo de Manufactura.

5.2.3.3. ARCHIVO DE PERSONALIZACIÓN

El Archivo de Personalización comprende 12 bytes, organizados en 3 registros de 4 bytes cada uno.

Luego de la finalización de la *Etapa de Personalización* este archivo es sólo de lectura y solamente se tiene libre acceso para la *lectura.*

La terminación de la *Etapa de Personalización* se indica por la escritura de un ‘1’ en el bit más significativo del 4 byte del primer registro en el *Archivo de Personalización* (Bit de Personalización). El cambio de Etapa se hace efectivo inmediatamente después del próximo reseteo de la tarjeta.

Archivo de Personalización:

Bit de Personalización

		Byte 1			Byte 4
Registro	1	Registro de Opciones	Registro de Opciones de Seguridad	N_OF_FILE	P
	2				
	3				

Los tres primeros bytes del primer registro del Archivo de Personalización se usan para fijar determinados parámetros y habilitar o deshabilitar cualidades del sistema operativo de la tarjeta.

5.2.3.4. ARCHIVO DE SEGURIDAD

El Archivo de Seguridad guarda la siguiente información:

- El Código de Usuario IC.
- El código PIN
- Contadores de Error para limitar el número de presentaciones de códigos incorrectos y autentificación.

El *Archivo de Seguridad* sólo puede ser de lectura durante la *Etapa de Manufactura* y en la *Etapa de Personalización* del ciclo de vida de la tarjeta, luego de la correcta presentación del IC correcto.

Después de la terminación de la Etapa de Manufactura, no hay posibilidad de leer el Archivo de Seguridad.

El Archivo de Seguridad puede ser escrito en la *Etapa de Manufactura* y *en la Etapa de Personalización* después de presentar el código IC correcto.

El Archivo de Seguridad comprende 10 registros de 8 bytes de longitud y cada uno esta organizado como sigue:

		Byte 1 byte 8
Registro	1	Código IC
	2	PIN
	3	Número aleatorio para RNDc
	4	Código de aplicación AC1
	5	Código de aplicación AC2
	6	Código de aplicación AC3

	7	Código de aplicación AC4
	8	Código de aplicación AC5
	9	Contador para claves incorrectas
	10	Copia del Contador para claves incorrectas

5.2.4. ARCHIVOS DE DATOS DEL USUARIO

ARCHIVO DE DIRECCIONES DE LOS ARCHIVOS DEL USUARIO

El Archivo de Direcciones de los Archivos del Usuario esta formado por los registros definidos en N_OF_FILE de 6 bytes cada uno y guardados en el *Bloque de Archivos de Definición* que esta localizado en el *Archivo de Datos del Usuario* en cada registro.

El *Bloque de Archivos de Definición* es escrito durante la Etapa de Personalización del ciclo de vida de la tarjeta. Después de la terminación de la *Etapa de Personalización*, este archivo es de libre acceso y puede ser escrito después de enviar el código correcto IC.

Cuando un comando *SELECT FILE es enviado, el sistema operativo busca en todos los Bloques de Definición por el Identificador de Archivo que coincida con el valor especificado en el comando *SELECT FILE.

El Sistema Operativo de la tarjeta no provee un error referente a la inconsistencia del número de archivos escritos en el parámetro N_OF_FILE. Cualquier inconsistencia provocara un malfuncionamiento en la tarjeta.

Los *Archivos de Datos del Usuario* son destinados en la *Etapa de Personalización* del ciclo de vida de la tarjeta. Los Datos guardados el los *Archivos del Usuario* pueden ser leídos a través del comando *READ RECORD y actualizados a través del comando *WRITE RECORD cuando las condiciones de seguridad hayan sido correctamente enviadas a la tarjeta.

Los *Archivos de Datos del Usuario* son definidos escribiéndolos en el respectivo *Bloque de Definiciones* en los registros del usuario, durante la *Etapa de Personalización*. Y no es posible cambiar el número de registros del archivo una vez que cualquiera de los Archivos de Datos del Usuario haya sido usado.

El *Archivo de Datos del Usuario* puede contener hasta 255 registros de máximo 32 bytes de longitud cada uno.

El tamaño del *Archivo de Datos del Usuario* es calculado como ***Número de Registros*longitud el registro (bytes***), la cantidad de espacio de

memoria ocupada por los archivos de datos es la suma de los tamaños individuales de cada archivo.

Hay que tener mucho cuidado de no exceder con datos la cantidad de memoria disponible, ya que se puede provocar un mal funcionamiento en la tarjeta.

5.2.4.2. DEFINICIÓN DEL BLOQUE DEL ARCHIVO DEL USUARIO

Cada *Archivo de Datos del Usuario* es descrito en asociación con el Bloque de Definición que contiene el archivo identificador, longitud del registro, longitud del archivo y los atributos de seguridad. Cada archivo de bloque de Definición comprende seis bytes:

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5 / 6
Longitud del registro	Número de registros	Atributo de seguridad de lectura	Atributo de seguridad de escritura	Archivo identificador

El archivo de definición de bloques de todos los archivos es guardado en el *archivo de direcciones del usuario*. Estos pueden ser leídos a través de los comandos READ RECORD después de seleccionar el archivo de direcciones del usuario con el comando SELECT FILE.

El número de registros en el archivo de direcciones del usuario es dado por el valor del parámetro N_OF_FILE en el registro de opciones.

5.2.4.3. DIRECCIONAMIENTO DE LOS ARCHIVOS DEL USUARIO

Para el direccionamiento de los archivos de datos del usuario en una nueva tarjeta, siga los pasos listados a continuación. Se asume que el código IC ha sido presentado a la tarjeta antes de que los archivos de datos internos puedan ser escritos.

1. Usar el comando SELECT FILE con el archivo de identificación ID=FF02 para seleccionar el archivo de personalización.
2. Escribir el número de archivos de datos de usuario requeridos (N_OF_FILE) en el registro de opciones, que es el tercer byte del primer registro del archivo de personalización para asignar el espacio requerido (número de registros) en el archivo de dirección del usuario.
3. Usar el comando SELECT FILE con el archivo de identificación ID=FF04 para seleccionar el Archivo de Direccionamiento de Archivos del Usuario.
4. Escribir los bloques de definición del archivo N_OF_FILE en el archivo de direccionamiento del Usuario con el comando WRITE RECORD. Escriba los seis bytes de cada Bloque del Archivo de Definición.

Ahora los Archivos de Datos del Usuario pueden ser seleccionados, leídos y escritos.

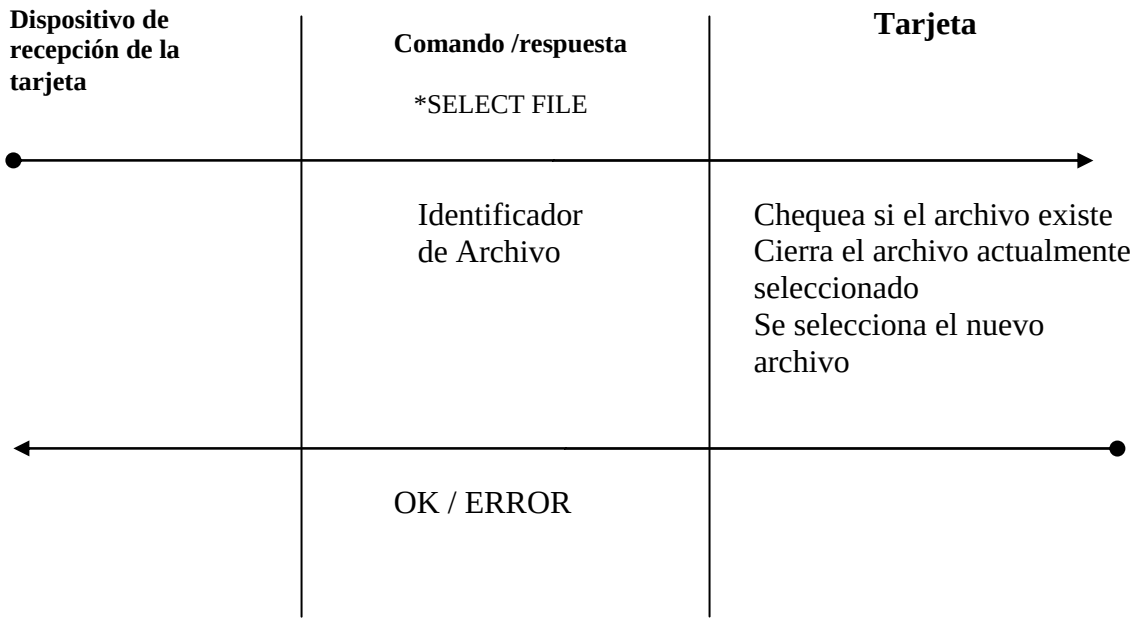
5.2.5. ACCESO DE LOS ARCHIVOS DE DATOS

El proceso de acceso a los archivos de Datos es idéntico para los Archivos de Datos Internos y los Archivos de Datos de Usuario.

5.2.5.1. SELECCIÓN DE UN ARCHIVO

El comando *SELECT_FILE puede ser ejecutado en cualquier momento. El archivo especificado – si existe- será seleccionado y el archivo previamente seleccionado – si hubiera uno- será cerrado. Si el archivo especificado no existe, la tarjeta retornara un código de error y no habrá ningún cambio en el archivo seleccionado. Después de un reset en la tarjeta, ningún archivo es seleccionado.

El comando *SELECT FILE es llevado a cabo de la siguiente manera:



5.2.5.2. LECTURA DE UN REGISTRO

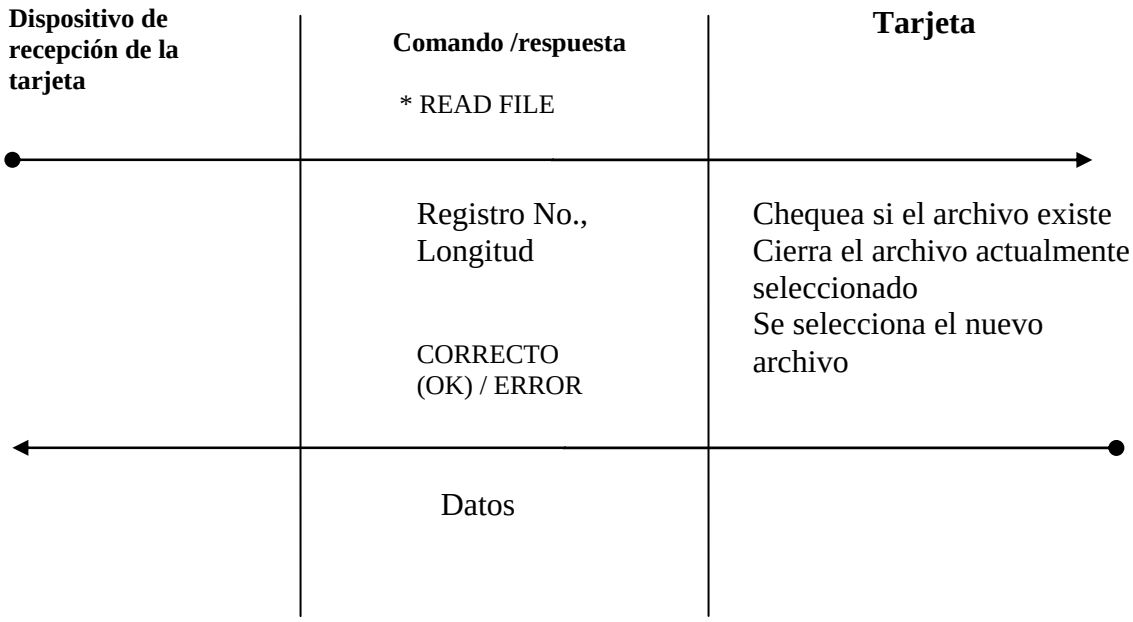
El comando *READ RECORD puede ser ejecutado una vez que el archivo haya sido seleccionado a través del comando *SELECT FILE.

Las condiciones de seguridad asociadas al archivo actualmente seleccionado son revisadas por la tarjeta antes de la ejecución del comando. Si las condiciones de seguridad no son cumplidas, el comando es rechazado por la tarjeta.

El número máximo de bytes que va ha ser leído es igual a la longitud del registro.

Si el número de bytes leídos (= N) es menor que la longitud del registro, los primeros N bytes del registro son retornados por la tarjeta.

El comando READ FILE es llevado a cabo de la siguiente manera:



Registro No.	Byte lógico que indica el número del registro
Longitud	número de datos a ser leídos del registro, máximo 32
Datos	Registro de Datos, *Longitud* bytes

5.2.5.3. ESCRITURA DE UN REGISTRO

El comando *WRITE RECORD puede ser ejecutado una vez que el archivo haya sido seleccionado a través del comando *SELECT FILE.

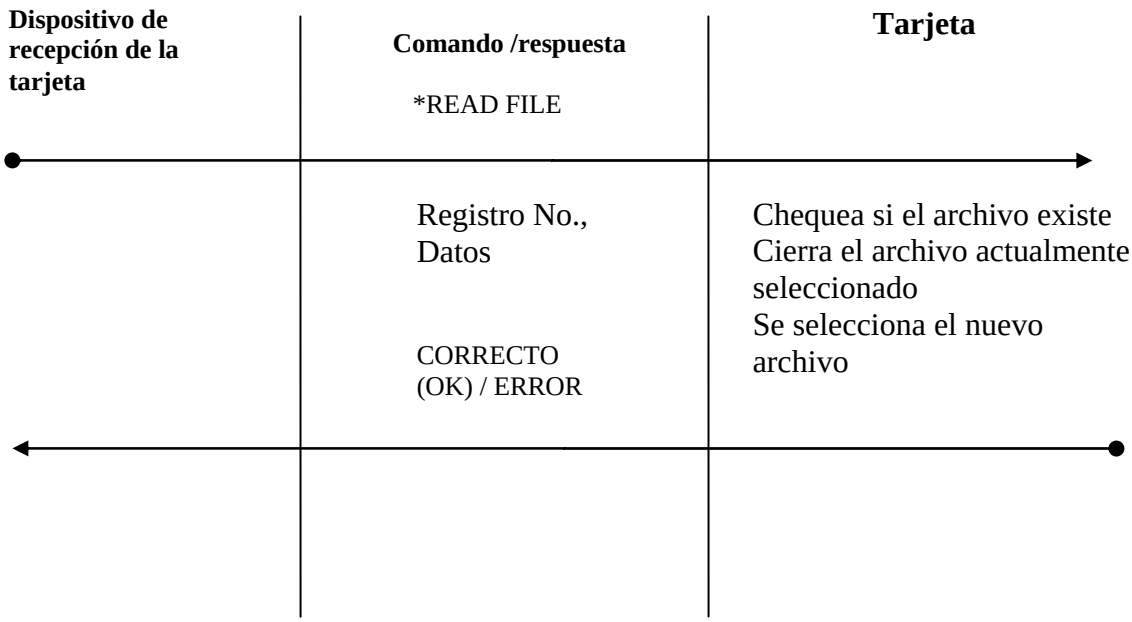
Las condiciones de seguridad asociadas al archivo actualmente seleccionado son revisadas por la tarjeta antes de la ejecución del comando. Si las condiciones de seguridad no son cumplidas, el comando es rechazado por la tarjeta.

Los datos pueden ser escritos en un solo registro en cada operación del comando *WRITE RECORD. El número de bytes a ser escritos es especificado en el comando de escritura.

El número máximo de bytes a ser escritos es igual a la longitud del registro.

Si el número de bytes a ser escritos (=N) es menor que la longitud del registro, los primeros N bytes del registro son sobrescritos con los nuevos datos. Los datos restantes en el registro no son modificados.

El comando *READ FILE es llevado a cabo de la siguiente manera:



Registro No. Byte lógico que indica el número del registro
Datos Datos que van a ser escritos en el registro.

5.3. NORMA ISO Y RESPUESTA A UN *RESET (ANSWER TO RESET)

Después de un reset la tarjeta transmite una respuesta a este comando en concordancia con el estándar ISO 7816-3. ACOS2 soporta el protocolo tipo T=0. La selección del tipo de protocolo no esta implementada.

La convención directa es usada para codificar los bits en la comunicación con la tarjeta, el nivel lógico de UNO corresponde al estado de alta impedancia en las líneas de entrada/salida.

Catorce bytes de datos son transmitidos en los caracteres del historial, como esta descrito a continuación.

Los siguientes datos son transmitidos en la respuesta:

TS	T0	TA1	TB1	TD1	14 caracteres del historial

Los 14 bytes de la cadena de caracteres transmitido en los caracteres del historial esta compuesto como se muestra a continuación:

T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14
41H	01H	10H/20H	REGISTRO DE OPCIONES			ARCHIVO DE PERSONALIZACIÓN BYTES 4-8					Etapa de Ciclo de vida	90H	00H

Etapa del Ciclo de Vida	Codifica el ciclo de vida de la tarjeta en un único byte 0: Etapa de Usuario 1: Etapa de Manufactura 2: Etapa de Personalización
Bytes de la Versión	El contenido de los bytes de la versión son: T1 = 41H = ACOS T2 = 01H = Version 1 T3 = 10H/20H/38H= Revision 1.0/Revision 2.0/Revision 3.8
Registro de Opciones	El contenido de los tres bytes del registro de opciones: T4= Registro de Opción T5 = registro de Opciones de Seguridad T6 = N_OF_FILE
Archivo de Personalización De memoria	Los cinco bytes siguientes del registro de Opciones del Archivo de personalización en la EEPROM

5.4. COMANDOS

La siguiente sección describe en detalle el formato de todos los comandos de la tarjeta ACOS2 y las posibles respuestas. Todos los valores son dados en HEX. Un sumario de los códigos retornados por la tarjeta está dado en la sección 5.6 acerca de los CÓDIGOS DE ESTADO.

Los siguientes comandos son los provistos para la tarjeta ACOS2 han sido utilizados en el desarrollo del proyecto:

***SELECT FILE**	Selección de un archivo de datos para lectura o escritura
***READ RECORD**	Lectura de un dato de un registro del archivo recientemente abierto
***WRITE RECORD**	Escritura de un dato de un registro del archivo recientemente abierto

5.4.1. SELECCIÓN DE ARCHIVO

Selecciona un dato de un archivo para subsecuentemente ejecutar los comandos *READ RECORD y *WRITE RECORD.

Comando:

CLA	INS	P1	P2	P3	DATO
80	A4	00	00	02	ID del archivo

ID de Archivo Identificador de Archivo de 2 bytes

Respuesta:

SW1 SW2
Estado

Códigos de Estado Específicos:

SW1	SW2	SIGNIFICADO
6A	**82**	El archivo no existe
91	**xx**	Selección de archivo (solo par Archivo de Datos de Usuario), xx es el número de registro del Archivo de Direcciones de Usuario (ID de archivo: FF04H) que contiene el Archivo de Bloques de Definiciones del archivo seleccionado.

5.4.2. LECTURA DE UN REGISTRO

Lee un número de bytes (máximo la longitud del registro) de un registro en el archivo actualmente seleccionado.

Comando:

CLA	INS	P1	P2	P3
80	B2	Reg. No.	00	Longitud

Registro No. Número del registro a leer

Longitud Número de bytes a ser leídos del Registro No.

Respuesta:

DATOS	SW1 SW2
Byte 1 ……byte N	Estado

Códigos de Estado Específicos:

SW1	SW2	SIGNIFICADO
69	**82**	Condiciones de seguridad no satisfechas
6A	**83**	Archivo no encontrado – archivo muy pequeño
67	**00**	Longitud especificada mas grande que la longitud del registro o mayor a 32 - invalido
69	**85**	Ningún archivo seleccionado

5.4.3. ESCRITURA DE UN REGISTRO

Escribe un número de bytes (máximo la longitud del registro) a un registro en el archivo actualmente seleccionado.

Comando:

CLA	INS	P1	P2	P3	Datos
80	D2	Reg. No.	00	Long	Byte1……byte N

Registro No. Número del registro a leer

longitud Número de bytes a ser escritos al Registro No.

Byte1……byte N Datos a ser escritos en los primeros *long* bytes del registro

Respuesta:

SW1 SW2
Estado

Códigos de Estado Específicos:

SW1	SW2	SIGNIFICADO
69	**82**	Condiciones de seguridad no satisfechas
6A	**83**	Archivo no encontrado – archivo muy pequeño
67	**00**	Longitud especificada más grande que la longitud del registro o mayor a 32 - Invalido
69	**85**	Ningún archivo ha sido seleccionado

5.5. PERSONALIZACIÓN DE LA TARJETA

Esta sección describe el procedimiento general en la personalización de la tarjeta inteligente ACOS2. Mientras la personalización puede ser llevada a cabo en procedimientos separados, el proceso general de personalización generalmente requiere la ejecución de los pasos descritos a continuación.

Para la personalización de una tarjeta inteligente ACOS2 nueva se sugiere llevar a cabo la siguiente secuencia:

1. Apagar y resetear la tarjeta
2. Enviar el código ID por defecto (ese código es el puesto por el fabricante)

3. Seleccionar el Archivo de Personalización (ID = FF02H) y escribir las propiedades requeridas en el *Registro de Opciones* y el parámetro N_OF_FILE. Precaución: no fijar un '1' en el Bit de Personalización y no cambie el *Registro de Opciones de Seguridad* en este paso.
4. Realice un reseteo de la tarjeta. Luego del reset, la tarjeta ACOS2 lee el Archivo de Personalización y acepta el nuevo valor de N_OF_FILE y los bits de opciones son guardados en el *Registro de Opciones.*
5. Enviar el código ID por defecto.
6. Seleccionar el Archivo de Direcciones de Archivos del Usuario (ID= FF04H) y escriba el Archivo de Bloques de Definiciones para los Archivos de usuarios requeridos (comando WRITE RECORD) con los atributos de seguridad con libre acceso.
7. Seleccione los Archivos de Usuario individuales e inicialice los datos en los archivos requeridos (comando WRITE RECORD).
8. Seleccionar el Archivo de Direcciones de Archivos del Usuario (ID= FF04H) y escriba los atributos de seguridad para todos los Archivos de Usuarios (comando WRITE RECORD). Verifique los contenidos del el Archivo de Direcciones de Archivos del Usuario (comando READ RECORD). Precaución: no cambie accidentalmente ningún otro parámetro del Archivo de Bloques de Definición.

9. Realice un reseteo de la tarjeta. El ciclo de vida de la tarjeta es indicado en el ATR y debe estar en ‘Etapa de Usuario’.
10. La correcta personalización puede ser verificada, enviando los correctos códigos secretos y las llaves programadas en la tarjeta y leyendo o escribiendo datos en la tarjeta ejecutando los comandos de lectura y escritura.

5.6. CÓDIGOS DE ESTADO

SW1	SW2	SIGNIFICADO
90	**00**	Correcto.
91	**Nn**	El archivo de datos de Usuario ha sido seleccionado. El correspondiente bloque de definición esta almacenado en el registro No. Nn
67	**00**	Error en el valor de P3.
69	**66**	Comando no disponible.
69	**82**	Condiciones de seguridad no satisfechas.
69	**83**	La llave secreta ha sido bloqueada.
6A	**82**	El archivo no existe.
6A	**83**	Registro no encontrado – archivo muy pequeño.
6A	**86**	P1 o P2 incorrecto.
6D	**00**	Instrucción desconocida.
6E	**00**	Clase inválida.

Tabla 5.3. Códigos de estado de la tarjeta ACOS2.

CAPÍTULO 6

6. ETHERNET

OBJETIVOS:

- Dar una introducción acerca de la pila de Microchip usada en la implementación del proyecto de tesis.
- Describir la tarjeta de comunicación SBC45EC, sus características, configuración e interfaces.

6.1. DESCRIPCIÓN DE LA PILA TCP/IP DE MICROCHIP

6.1.1. INTRODUCCIÓN

La pila de microchip es un conjunto de programas que proveen servicios basados en el estándar TCP/IP (servidor HTTP, cliente de correo, etc.), que pueden ser usados en un aplicación especifica basada en TCP/IP.

Los usuarios de esta pila no necesitan conocer toda la complejidad de las especificaciones TCP/IP para poder usarla, por tal motivo, en el presente documento no se pretende discutir los protocolos TCP/IP[8].

6.1.2. ARQUITECTURA

Muchas implementaciones de aplicaciones TCP/IP siguen una arquitectura de software referida al “modelo de Referencia TCP/IP”. El software basado en este modelo se divide en múltiples capas, donde las capas están apiladas una sobre otra (de allí el nombre “Pila TCP/IP”) y cada capa accede a servicios de una o más capas directamente debajo de ella. Una simple versión del modelo es el mostrado en la Figura 6.1.
Según especificaciones, muchas de las capas TCP/IP están “vivas”, en el sentido de que no sólo actúan cuando un servicio es requerido, sino también cuando ocurren eventos como el tiempo de espera o el arribo de un nuevo paquete.

Un sistema operativo con suficiente memoria de datos y memoria de programa puede fácilmente incorporar estos requerimientos. Un sistema operativo de multitareas puede proveer facilidad adicional y su implementación puede ser modular. Las tareas se vuelven difíciles cuando el sistema usa un microcontrolador de 8 bits, con algunos cientos de bytes de memoria en la RAM y una limitada memoria de programa. Además, sin el acceso a un sistema operativo de múltiples tareas, el usuario debe poner mucha atención para hacer la pila independiente del

programa principal. La pila TCP/IP está firmemente integrada y puede ser muy eficiente en el uso del espacio. Pero una pila especializada puede plantear problemas únicos al integrar más y nuevas aplicaciones.

La pila está escrita en el lenguaje de programación ‘C’, tanto para los compiladores de Microchip C18 y HI-TECH PICC 18. Dependiendo de cuál se use, los archivos fuentes harán los respectivos cambios. La pila de Microchip esta diseñada para que compile en los microcontroladores de la familia 18 (PIC18) solamente [8].

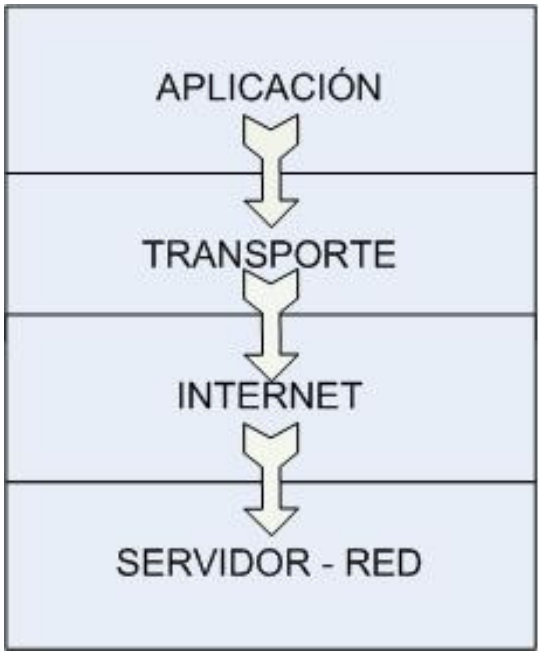


Figura 6.1. Capas del modelo referencial TCP/IP de Microchip

6.1.3. CONFIGURACIÓN DE LA PILA

Multitareas cooperativas permiten a la aplicación del usuario llevar a cabo sus propias tareas sin necesidad de manejar la pila TCP/IP de Microchip la cual también tiene el formato de multitarea.

Además del diseño de multitareas cooperativas, el usuario debe comprender primero algunos detalles de configuración básicos.

Para facilitar el proceso de configuración, la pila usa la instrucción del compilador 'C', "defines" para habilitar o deshabilitar un parámetro en particular, el usuario puede cambiar uno o más de estos "defines". La mayoría de éstos son definidos en el archivo de encabezado "StackTsk.h". Algunos de los **defines* que están definidos en otros archivos se muestran en el nombre correspondiente del archivo. Una vez que estos archivos hayan sido modificados, el usuario debe recompilar el proyecto de aplicación para incluir los cambios realizados. Los "defines" utilizados en el proyecto son listados en la tabla siguiente:

Define	Valores	Usado por	Propósito
CLOCK_FREQ	Frecuencia del Oscilador (Hz)	Tick.c	Define la frecuencia de oscilación del sistema para determinar el valor del contador tick
TICK_PER_SECONDS	10 -255	Tick.c	Calcula un segundo
TICK_PRESCALE_VALUE	2, 4, 8, 16, 32, 64, 128, 256	Tick.c	Determina el valor del contador tick
STACK_USE_UDP	N/A	UDP.c StackTask.c	Comente este módulo si no se usa. Este módulo va ha ser automáticamente habilitado si hay al menos un módulo de alto nivel que requiere UDP.
STACK_CLIENT_MODE	N/A	ARP.c	Código relativo al cliente a ser habilitado

Tabla 6.1. “Defines” utilizados en el proyecto[1].

[1] Para mayor información de todos los “defines” disponibles en la pila TCP/IP de Microchip por favor referirse a la nota de aplicación AN833 de Microchip.

6.2. TARJETA DE INTERFASE CON ETHERNET

6.2.1. INTRODUCCIÓN

La tarjeta que se selecciono para realizar la comunicación con la red es la SBC45EC. Esta diseñada para comunicarse con Ethernet a una velocidad de 10Mbs. Posee un controlador de gama alta el PIC18F452 y un controlador de red RTL8019. En la Figura 6.2. se describe en forma general el funcionamiento de la tarjeta SBC45EC. El microcontrolador es el encargado de controlar al dispositivo de red, este le provee del bus de datos y de direcciones, con los cuales se tiene acceso al mismo [7].

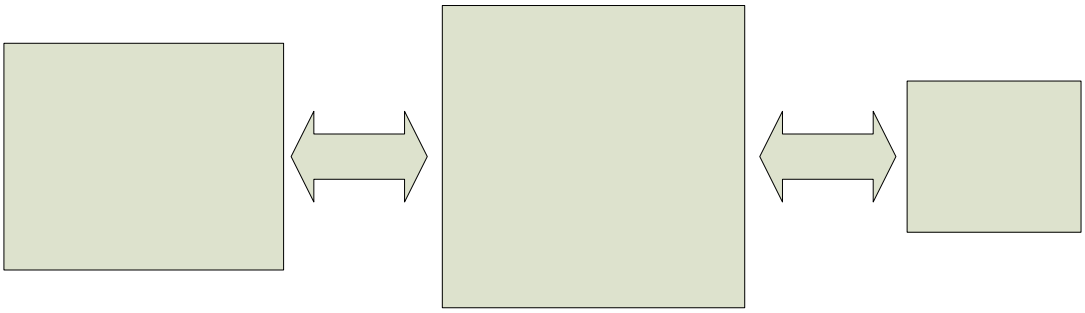


Figura 6.2. Diagrama de Bloques de la tarjeta SBC45EC

En la Figura 6.3. se describe el funcionamiento interno del RTL8019, el cual consta de un identificador LAN que es una dirección física que define la unicidad de la tarjeta. Internamente esta implementado un algoritmo de

reconocimiento *“*plug and play*”* que requiere de un hardware que el integrado le provee internamente, este algoritmo realiza la secuencia de reconocimiento de los comandos.

Finalmente el integrado posee un *transceiver que es el bus de datos de recepción y de transmisión hacia la red.

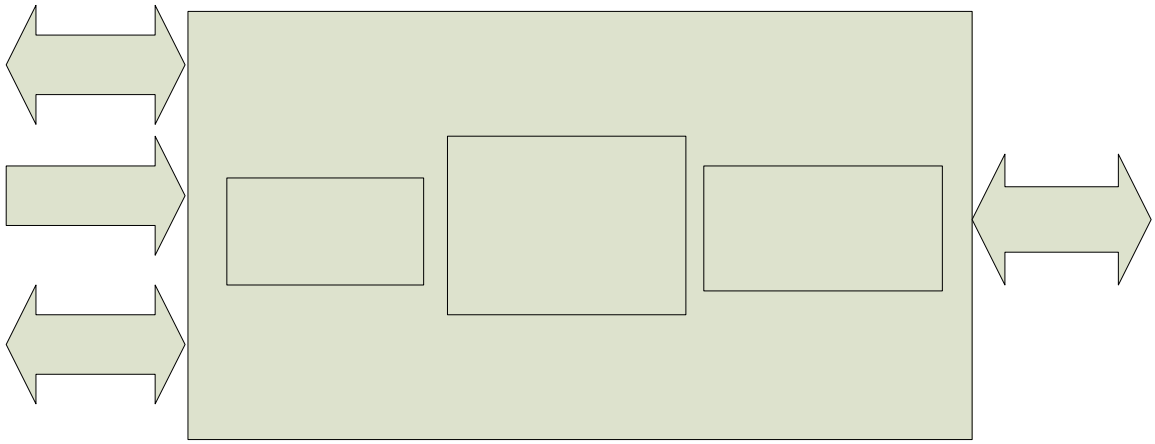


Figura 6.3. Diagrama de Bloques del RTL8019.

6.2.2. CARACTERÍSTICAS

- Diodo de protección
- Amplio rango de voltaje de operación 7 – 35 V.

BUS DE DATOS

BUS DE DIRECCION

IDENTIFICADOR LAN

PL

- Conector RJ45 con dos LEDs. Un LED verde para la indicación de comunicación, y un LED amarillo para indicar de que se encuentra en actividad.
- Posee un conector para programación serial dentro del circuito ICSP.
- Puerto estándar para TCP y UDP
- Fácil adaptación de microcontroladores de la gama 18.
- Diseño modular.
- Interfase RS-232 a través de 3 pines en el panel frontal.

Posee conectores frontales que pueden ser usados como puertos de expansión para añadirle mayores funcionalidades. Contienen los pines de los puertos de comunicación para los protocolos de comunicación I2C, SPI, RS-232. Los conectores frontales se detallan en la figura y tabla siguientes.

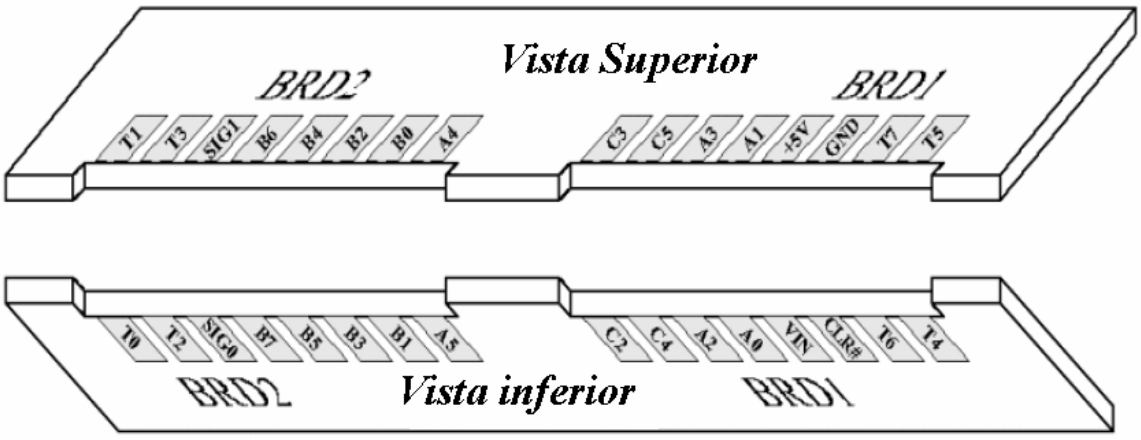


Figura 6.4. Conector frontal de la tarjeta SBC45EC

CONECTOR FRONTAL BRD2			CONECTOR FRONTAL BRD1		
Nombre	**Pin**	**Señal**	**Nombre**	**Pin**	**Señal**
T0	2	N.C.	T4	2	N.C.
T1	1	N.C.	T5	1	N.C.
T2	4	N.C.	T6	4	Pin del pic RC0
T3	3	Tierra	T7	3	Pin del pic RC1
SIG0	6	señal de recepción RS232	GND	5	Tierra
SIG1	5	señal de transmisión RS232	+5V	7	Fuente de poder regulada +5v a 0.5A
B0	13	Pin del PIC RB0	VIN	8	Voltaje de entrada no regulada
B1	14	Pin del PIC RB1	CLR	6	Pin del PIC /MCLR
B2	11	Pin del PIC RB2	A0	10	N.C.
B3	12	Pin del PIC RB3	A1	9	N.C.
B4	9	Pin del PIC RB4	A2	12	N.C.
B5	10	Pin del PIC RB5	A3	11	N.C.
B6	7	Pin del PIC RB6- también usado por ICP	C2	16	Pin del PIC RC2
B7	8	Pin del PIC RB6- también usado por ICP	C3	15	Pin del PIC RC3- también usado por I2C
A4	15	Pin del PIC RA4	C4	14	Pin del PIC RC4- también usado por I2C
A5	16	N.C.	C5	13	Pin del PIC RC5

Tabla 6.2. Conectores Frontales de la Tarjeta SBC45EC.

6.2.3. INTERFACES

6.2.3.1. ETHERNET

La tarjeta SBC45EC tiene un puerto de Ethernet de 10Mbs. El conector RJ45 es de acuerdo al estándar IEEE 802.3. El conector RJ45 tiene dos LEDs internos, un LED verde para la indicación de comunicación, y un LED amarillo para indicar que se encuentra en actividad.

6.2.3.2. RS232

La tarjeta tiene una comunicación USART con protecciones de +-15V de polarización inversa. Las señales del USART están disponibles en el panel frontal de conectores. Cuatro **jumpers* en la parte posterior de la tarjeta son usados para configurar si las señales USART tienen niveles de voltaje RS232 o TTL. En la Figura 6.5. se encuentra el diagrama de conexiones de la tarjeta usada. Los **jumpers* soldados predeterminadamente son SJ3 y SJ4, que configuran el USART para niveles de voltaje RS232. Soldando los **jumpers* SJ1 y SJ2 y abriendo SJ3 y SJ4, los pines del USART pueden ser configurados con niveles de voltaje TTL.

6.2.3.3. CONECTOR ICSP

La tarjeta SBC45EC tiene un conector de este tipo (Programación serial en el circuito). Este conector permite programar el microcontrolador.

6.2.4. CONFIGURACIÓN

La tarjeta SBC45EC puede ser configurada por medio de los *jumpers SJ1 a SJ5. Los *jumpers SJ1 a SJ4 son usados para seleccionar las señales RS232 o las señales TTL del USART.

6.2.5. MEMORIA EXTERNA

La tarjeta SBC45EC tiene una memoria EEPROM como la 24LC256 (32Kbtes) que puede ser expandida a una EEPROM 24LC512 (64 Kbytes). La memoria 24LC256 tiene 32 kbytes de memoria no volátil, la cual es lo suficientemente grande como para albergar datos de las aplicaciones del usuario como por ejemplo algunas páginas Web que incluyan pequeñas figuras.

6.2.6. ESPECIFICACIONES

Rangos Máximos Absolutos

Ítem	Símbolo	Min	Tipo	Max	Unidad
Temperatura de Operación	Top	0		70	^{0}C
Temperatura del Almacenamiento	Tst	-65		140	^{0}C

Tabla 6.3. Rangos Máximos Absolutos de la Tarjeta SBC45EC.

Especificaciones eléctricas

Ítem	Símbolo	Condición	Min	Tipo	Max	Unidad
Voltaje DC de alimentación	Vdd	-	7		35	V
Corriente de Operación	Idd	Vdd = 5 V		50		mA
Conector RJ45 Ethernet RX/TX		T = 25^0C		0.35		Ω
Inductancia del conector RJ45 de Ethernet		T = 25^0C		0.3		uH
Capacitancia del conector RJ45 de Ethernet		T = 25^0C		12		pF

Tabla 6.4. Especificaciones eléctricas de la Tarjeta SBC45EC.

El conector RJ45 cumple con el estándar de la norma IEEE 802.3.

DIAGRAMA DE CONEXIONES DE LA TARJETA SBC45EC:

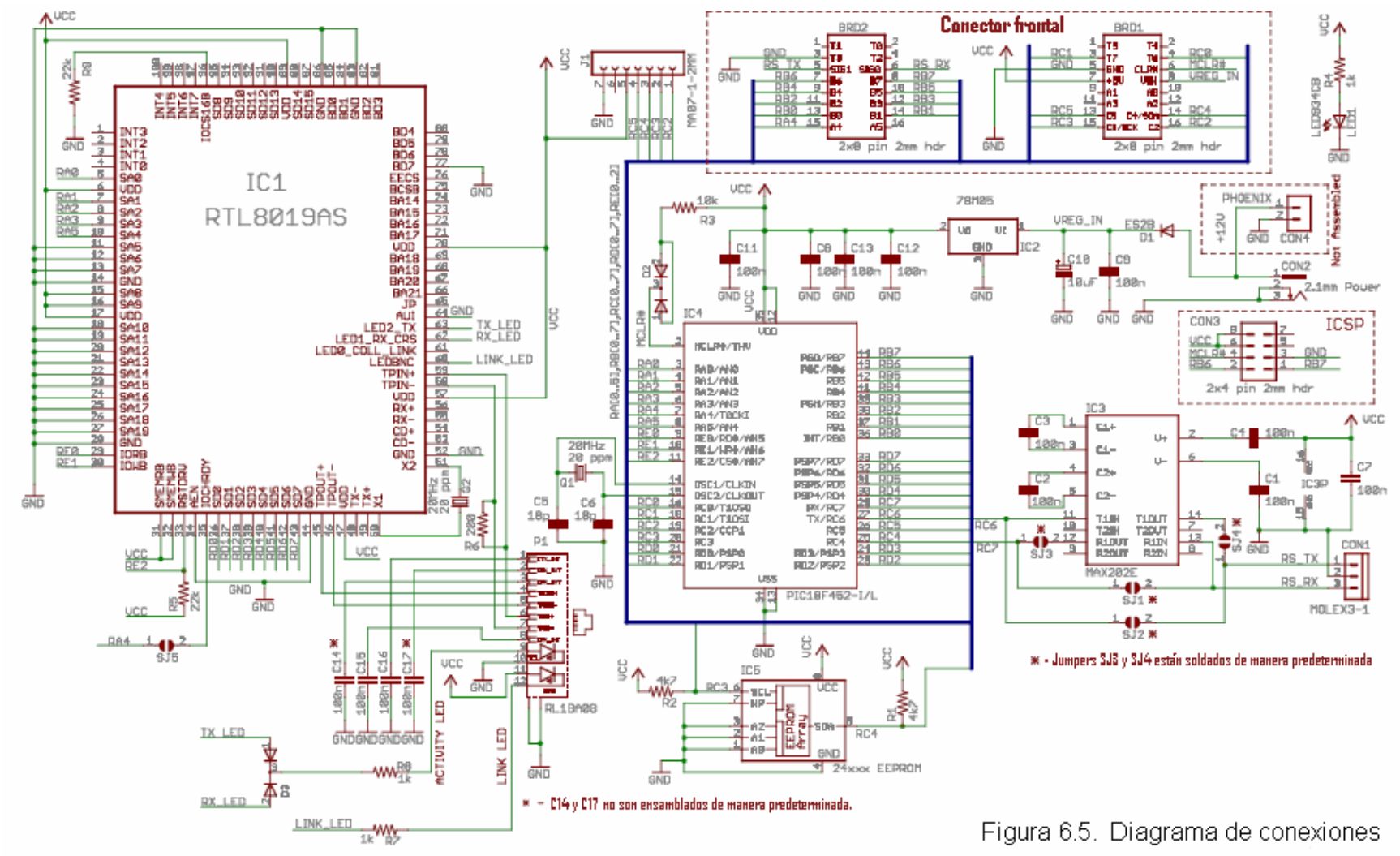


Figura 6.5. Diagrama de conexiones

La tarjeta SBC45EC conforma una tarjeta madre compacta con las dimensiones que se muestran en la Figura 6.6.

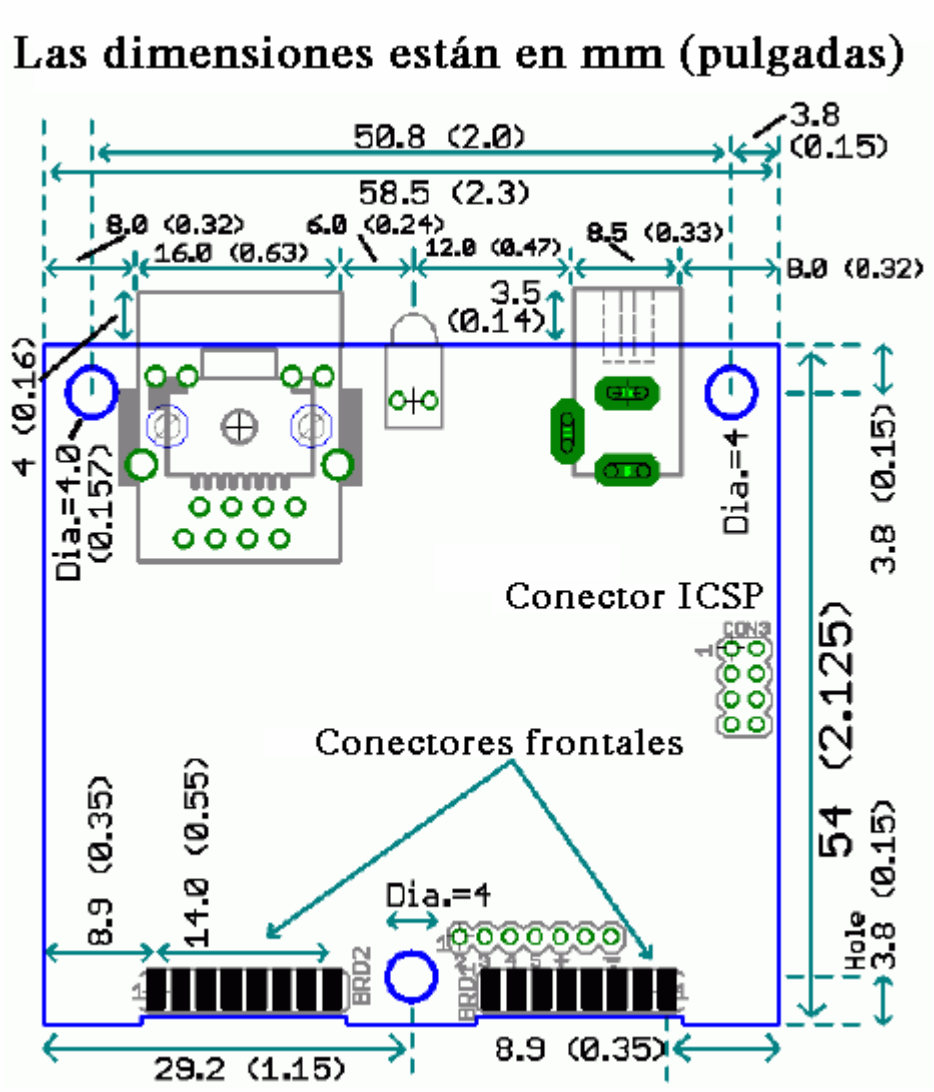


Figura 6.6. Dimensiones de la tarjeta SBC45EC.

El layout de la tarjeta se muestra en la Figura 6.7.

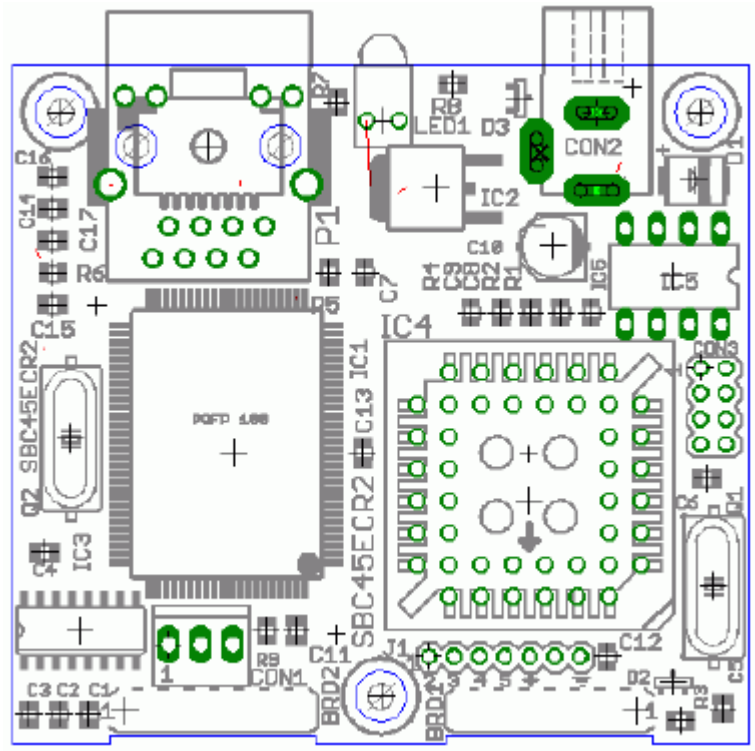


Figura 6.7. Layout de la tarjeta madre SBC45EC usada en el desarrollo del proyecto.

CAPÍTULO 7

7. SISTEMA DE ADMINISTRACION Y CONTROL DE ACCESOS (SAC)

OBJETIVOS:

- Describir del software desarrollado para este proyecto de tesis y sus distintos módulos.
- Detallar el análisis de los requerimientos presentados en la elaboración del sistema.
- Definir los parámetros de diseño utilizados en el sistema.
- Revisión de las pruebas de funcionamiento realizadas para garantizar la calidad del sistema.

Descripción General del sistema

Sistema que permite la asignación de permisos de ingreso sobre distintos puntos de acceso[1] en una edificación a usuarios/clientes, utilizando como llave de acceso una tarjeta inteligente y teniendo la funcionalidad de ser ejecutado en un *browser.

[1] **Punto de Acceso (**hardware): dispositivo creado para la lectura de tarjetas inteligentes ACOS2 reconocidos como nodos de una red Ethernet y que provee el mecanismo de apertura de una puerta.

Este sistema permite realizar monitoreo de ingresos de los usuarios/clientes, así como también intentos fallidos de acceso a lugares no autorizados. Dentro de este sistema pueden ser llevados controles adicionales como disponibilidad de habitaciones por áreas o días.

Dentro del funcionamiento de sus módulos se permite la atención de requerimientos de ingreso de todos los puntos conectados a la red y su validación contra la lógica de permisos que se encuentra en una base de datos, para luego proceder con el envío de la respuesta de confirmación positiva o negativa.

7.1. Especificación de la plataforma de implementación

7.1.1. Motor de base de datos

El motor de base de datos utilizado para el almacenamiento es SQL Server 2000 [11], el cual fue escogido por su innata compatibilidad con tecnología Windows, tecnología sobre la cual esta desarrollado el *driver de conexión del dispositivo USB ACR30U[1].

[1] **Dispositivo USB ACR30U** dispositivo para la lectura y grabación de tarjetas Smart Card ACOS2.

Entre las ventajas y características de SQL Server 2000 [11] encontramos:

- Configuración y manejo de seguridades en base de datos como encriptación de columnas.
- El retorno de documentos XML[1] para su manipulación como resultado de transacciones SQL.
- Alto desempeño en transacciones utilizando servicios de la WEB, ejecutándose bajo IIS[2].
- Escalabilidad y disponibilidad pudiendo tener múltiples instancias ejecutándose a la vez.
- Desarrollo de grandes bases soportando capacidades en terabytes como tamaño de base de datos.
- Altos niveles de disponibilidad requerida para aplicaciones WEB.
- Manejo de múltiples usuarios trabajando en múltiples bases de datos.

7.1.2. Lenguaje de programación

La elección del lenguaje de programación a utilizar se basó en los especificaciones técnicas dadas para la lectura/grabación de las tarjetas Smart Card ACOS2 [2] y la creación de la interfase para controlar el dispositivo USB ACR30U que permite leer y grabar las tarjetas inteligentes, estas especificaciones son:

---

[1] **XML (eXtensible Markup Language)** lenguaje Web estándar para etiquetar datos y crear otros lenguajes.

[2] **ISS (Internet Information Server) s**ervidor Web propietario de Microsoft.

- Driver de conexión dado por el proveedor con soporte para tecnología Windows.
- Kit de desarrollo para aplicaciones utilizando el dispositivo USB ACR30U, contiene guías para el desarrollo implementadas en C++, C#, VB y VB.NET [12].

Por lo cual se escogió como lenguaje de programación C# utilizando tecnología ASP 2.0 y el *entorno de trabajo Visual estudio 2005 para el desarrollo del sistema [13].

7.2. Análisis de requerimientos y alcances

7.2.1. Definición de requerimientos del sistema

En esta sección se especifican los requerimientos que el sistema SAC va a resolver para el cumplimiento del desarrollo del proyecto de tesis. Los cuales han sido clasificados como funcionales y no funcionales para su mejor entendimiento y análisis.

7.2.1.1. Requerimientos funcionales.

- Reconocimiento de los puntos de accesos como nodos de una red Ethernet.

- Atención de los requerimientos de ingreso a puertas enviados por todos los puntos de accesos conectados a la red mediante protocolo UDP.
- Enviar respuesta de apertura de puerta positiva o negativa al punto de acceso que envía un requerimiento.
- Creación de llaves utilizando tarjetas inteligentes la cual pueda dar acceso a puerta(s) de acuerdo a la demanda o lógica del negocio donde será implantado el proyecto.
- Proveer una lógica que permita dar accesos a usuarios/clientes a ubicaciones específicas o a grupos de puertas si fuese necesario.
- Proveer mecanismo para dar accesos a usuarios/clientes restringidos por horarios.
- Control de disponibilidad de habitaciones con asignación única como habitaciones de hotel.
- Monitoreo y registro de accesos exitosos y no exitosos de los usuarios/clientes.
- Manejo de creación de usuarios para administración/operación del sistema manipulado por medio de perfiles.

7.2.1.1. Requerimientos no-funcionales.

- Asegurar el acceso de los usuarios/clientes sólo a los lugares permitidos de acuerdo a lo definido por un operador.
- Creación de llaves para distintos puntos de acceso con la posibilidad de reutilizar una tarjeta inteligente.
- Crear un mecanismo seguro de duplicación de llaves y de desactivación de las mismas en caso que fuese necesario.
- Eliminación efectiva de los accesos a puertas o ubicaciones que tienen los usuarios/clientes

7.2.2. Definición de alcances

A continuación detallaremos cual es el alcance y límites del sistema que proponemos para cubrir con las necesidades de este proyecto de tesis y sus requerimientos.

- Al recibir un requerimiento de ingreso de parte de cualquier punto de acceso en la red, el sistema mediante su módulo de servidor UDP tomará la dirección IP de donde se esta generando el requerimiento verificará si esta se encuentra registrada como validable para un punto de acceso, posteriormente tomará los datos del usuario/cliente e identificador único de tarjeta que son enviados mediante paquetes UDP para validar si tiene permiso al lugar solicitado y si la hora está

autorizada para el ingreso. Para finalmente enviar la contestación de apertura de puerta o la negación del acceso según corresponda y registrar el evento en la base de datos [14].

- Existirá un módulo de operación del sistema el cual será habilitado cuando un usuario de aplicación tenga asignado el perfil de OPERADOR, con el cual podrá realizar las siguientes funciones:
    - Creación de usuarios con su respectiva tarjeta inteligente para que sean utilizadas como llaves de acceso.
    - Creación de accesos para un usuario a puertas, áreas con la posibilidad de definición de días y horarios de autorización de ingreso.
    - Inactivación de accesos y de usuarios.
    - Consultas referentes a disponibilidad de habitaciones y registro de accesos exitosos o fallidos de un usuario específico.

- Existirá un módulo de administración del sistema el cual será habilitado cuando un usuario de aplicación tenga asignado el perfil de ADMINISTRADOR, en el cual podrá realizar las siguientes funciones:
    - Creación de usuarios para la aplicación SAC y la asignación del perfil teniendo como consideración que para la creación debe tener disponible una tarjeta inteligente para el acceso.
    - Agregar y modificar áreas así como también puntos de acceso.

- Creación de usuarios con su respectiva tarjeta inteligente para que sean utilizadas como llaves de acceso.
- Creación de accesos para un usuario a puertas y/o áreas con la posibilidad de definición de días y horarios de autorización de ingreso.
- Inactivación de accesos y de usuarios.
- Consultas referentes a disponibilidad de habitaciones y registro de accesos exitosos o fallidos de un usuario específico.
- Consultas referentes a disponibilidad de habitaciones y registro de accesos exitosos o fallidos de un usuario especifico

7.2.3. Especificación de actores y casos de Uso

Los actores son entidades que interactúan con el sistema de forma directa o indirecta como por ejemplo una persona u otro sistema. Éstos pueden ser considerados como primarios o secundarios.

Un actor primario es aquel que inicia un proceso y para el cual el sistema esta construido; y, un actor secundario participa dentro del proceso de forma activa o pasiva para apoyar los objetivos de los actores primarios.

Actores Primarios

Operador. Persona responsable de atender a los usuarios/clientes, asignarles las tarjetas inteligentes y los respectivos accesos. Entre las transacciones que el puede realizar se encuentran:

- Crear usuarios/clientes y asignarles una tarjeta inteligente.
- Crear los accesos a puertas y/o áreas a los usuarios, con fechas de inicio, fin y los horarios de autorización.
- Revisar los registros de ingresos exitosos o fallidos.
- Inactivar usuarios.
- Eliminar accesos a puertas y/o áreas a los usuarios.
- Verificar disponibilidad de habitaciones.

Administrador. Persona responsable de administrar los puntos de acceso, usuarios de la aplicación y adicionalmente puede realizar las funciones de operador si fuese necesario. Entre las transacciones que el puede realizar se encuentran:

- Creación de usuarios de aplicación y asignación de perfil.
- Creación de áreas
- Modificación de áreas
- Creación de puntos de acceso.

- Modificación de puntos de acceso.
- Crear usuarios/clientes y asignarles una tarjeta inteligente.
- Crear los accesos a puertas y/o áreas a los usuarios, con fechas de inicio, fin y los horarios de autorización.
- Revisar los registros de ingresos exitosos o fallidos.
- Inactivar usuarios.
- Eliminar accesos a puertas y/o áreas a los usuarios.
- Verificar disponibilidad de habitaciones.

Punto de acceso. Hardware creado para la lectura de tarjetas inteligentes ubicado en las puertas de acceso de una edificación que se encarga del envío de datos por medio de protocolo UDP y de la apertura de la puerta en caso de recibir una confirmación acceso exitoso por parte del servidor.

Actores Secundarios

Servidor UDP. Servidor que atiende las peticiones de ingreso de todos los puntos de acceso en la red, el cual extrae los datos de CI del usuario y el identificador único de las tarjetas inteligentes, consultando al servidor de base de datos los accesos del usuario y envía la respuesta de acceso exitoso o fallido al punto de acceso que envió el requerimiento.

Base de datos. Repositorio en donde será almacenada y consultada la información relacionada a los usuarios, accesos y los registros de ingresos fallidos o exitosos.

Especificación de casos de uso

A continuación definimos los principales casos de uso que han sido considerados para el sistema SAC.

Nombre	**1. Punto de acceso envía requerimiento**
Descripción:	Una persona ingresa su tarjeta inteligente en algún punto de acceso y es enviada la solicitud de requerimiento de ingreso.
Notas:	• El punto de acceso extrae de posiciones de memoria específicas de la tarjeta inteligente los datos de identificación del usuario y de código único de tarjeta. • El punto de acceso coloca los datos en paquete UDP para su envío al servidor UDP.

Valor medible: El requerimiento es enviado o no.

Escenarios:

1.1. El requerimiento de ingreso se envía correctamente.

1.2. El requerimiento de ingreso no se envía correctamente

Nombre **2. <u>Punto de acceso recibe confirmación</u>**

Descripción: El punto de acceso recibe mediante paquete UDP confirmación de ingreso exitoso o fallido.

Notas:

- UDP Server verifica los accesos del usuario y toma la decisión del envío de confirmación.
- El punto de acceso recibe paquete UDP con valor 1(apertura de puerta) o 0 (mantener cerrada).

Valor medible: Se recibe el valor de la confirmación o no.

Escenarios:

2.1. Se recibe confirmación sea 1 o 0 durante los siguientes 5 segundo luego del envío del requerimiento

2.2. No se recibe confirmación sea 1 o 0 luego de 5 segundos.

Nombre **3. <u>Creación de acceso</u>**

Descripción: Un usuario de aplicación autorizado (operador o administrador) crea el acceso para un usuario/cliente.

Notas:

- Debe ser ingresado cedula o RUC del usuario/cliente.
- Los accesos pueden ser creados para un nuevo usuario o para usuarios existentes.
- Se puede crear una duplicación de tarjeta inteligente.

Valor medible: Se crea el acceso para un usuario o no.

Escenarios:

3.1. Se crean accesos para un usuario existente.

3.2. Se crea una duplicación de llave.

3.3. Se crean accesos por habitación para un usuario.

3.4. Habitación asignada no se encuentra disponible

3.5. Se crean accesos por área para un usuario.

Nombre	**4. <u>Consulta de accesos de un usuario/cliente</u>**
Descripción:	Un usuario de aplicación autorizado (operador o administrador) genera reporte de accesos fallidos o exitosos de un usuario/cliente.
Notas:	• Debe ser ingresado cedula/RUC o leer datos de tarjeta inteligente del usuario/cliente.
Valor medible:	Se presenta la información de los accesos o no tiene accesos registrados.
Escenarios:	4.1. Se presenta la información de los accesos del usuario de manera correcta. 4.2. No existe la información de cedula/RUC para consultar. 4.3. La lectura de la tarjeta inteligente no se realizad de manera correcta.

Nombre	**5. <u>Consulta de disponibilidad de habitaciones</u>**
Descripción:	Un usuario de aplicación autorizado (operador o administrador) consulta disponibilidad de habitaciones.

Notas:

- El reporte de disponibilidad presenta mediante calendario los días en los cuales no se encuentra disponible una habitación.

Valor medible: Se presenta la disponibilidad de la habitación o no.

Escenarios:

5.1. Se presenta la información de la disponibilidad de la habitación de manera correcta.

5.2. No se presenta la información de la disponibilidad de la habitación de manera correcta.

Nombre **6. <u>Creación de usuario de aplicación Web SAC</u>**

Descripción: Un usuario de administrador crea usuarios para la aplicación y le asigna un perfil.

Notas:

- Debe ser leídos datos de tarjeta inteligente para la creación de usuario de aplicación.

Valor medible: Se crear el usuario de aplicación o no.

Escenarios:

6.1. Valor ingresado en el campo USUARIO es un valor no disponible.

6.2. Se crea el usuario de la aplicación de manera correcta.

Nombre **7. <u>Creación/modificación de área y punto de acceso</u>**

Descripción: Un usuario de administrador crea/modifica área y/o puntos de acceso.

Notas:

- En el formulario se da la posibilidad de realizar la creación o modificación de áreas y puntos de acceso.

Valor medible: Se crea/modifica el área o punto de acceso o no.

Escenarios:

7.1. Se escoge la opción de nueva área y se crea con éxito.

7.2. Se escoge la opción de nuevo punto de acceso y se crea con éxito.

7.3. Se escoge la opción de nueva área y se produce error en la creación.

7.4. Se escoge la opción de nuevo punto de acceso y se produce error en la creación.

7.5. Se modifican con éxito los datos de área seleccionada.

7.6. Se modifican con éxito los datos de punto de acceso seleccionado.

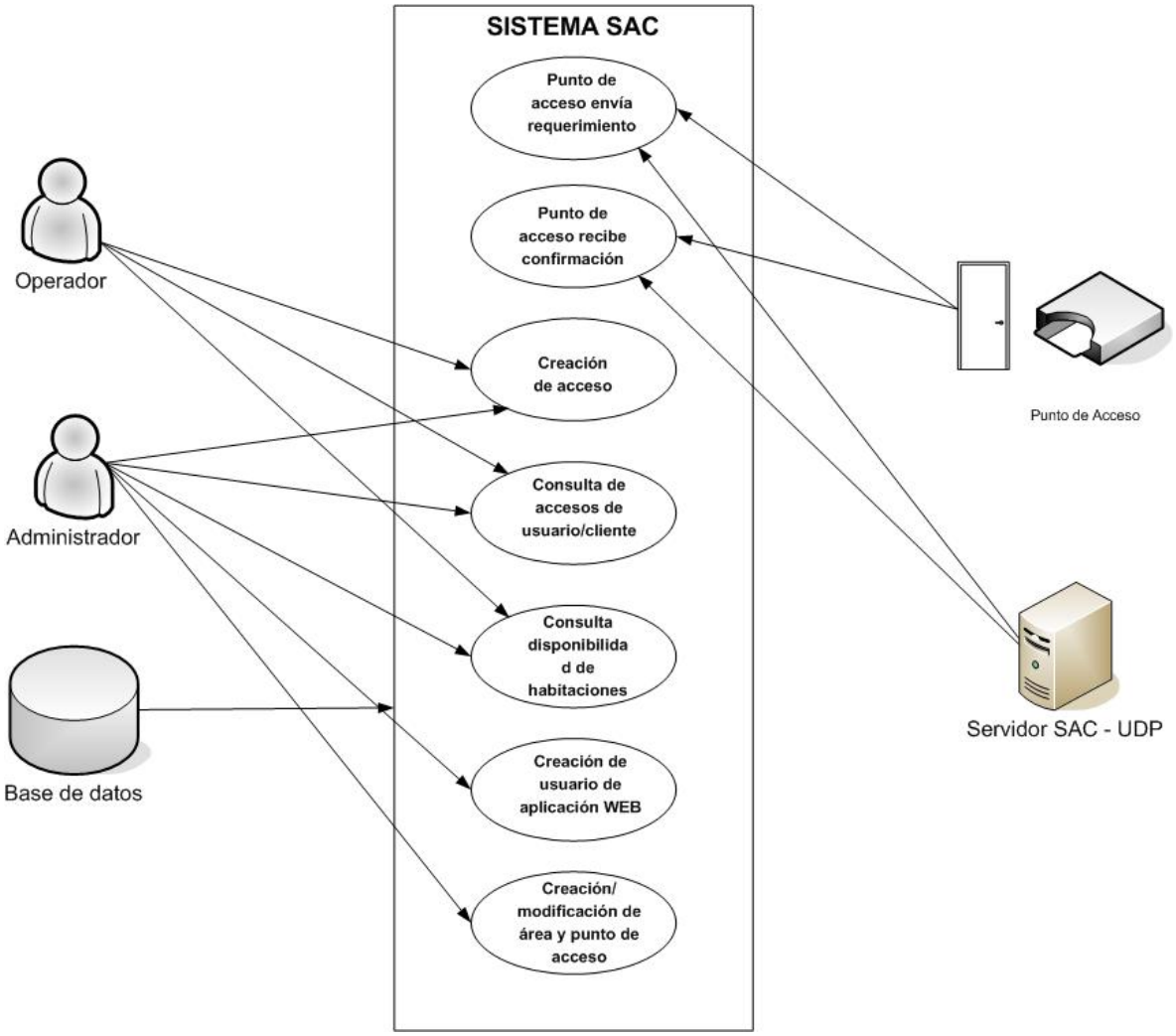


Figura 7.1 – Casos de uso

7.3. Diseño del Sistema

Esta sección esta constituida para definir las consideraciones tomadas en el diseño del sistema y el prototipo presentado en nuestro proyecto de tesis así como también mencionar cada uno de los componentes de la arquitectura.

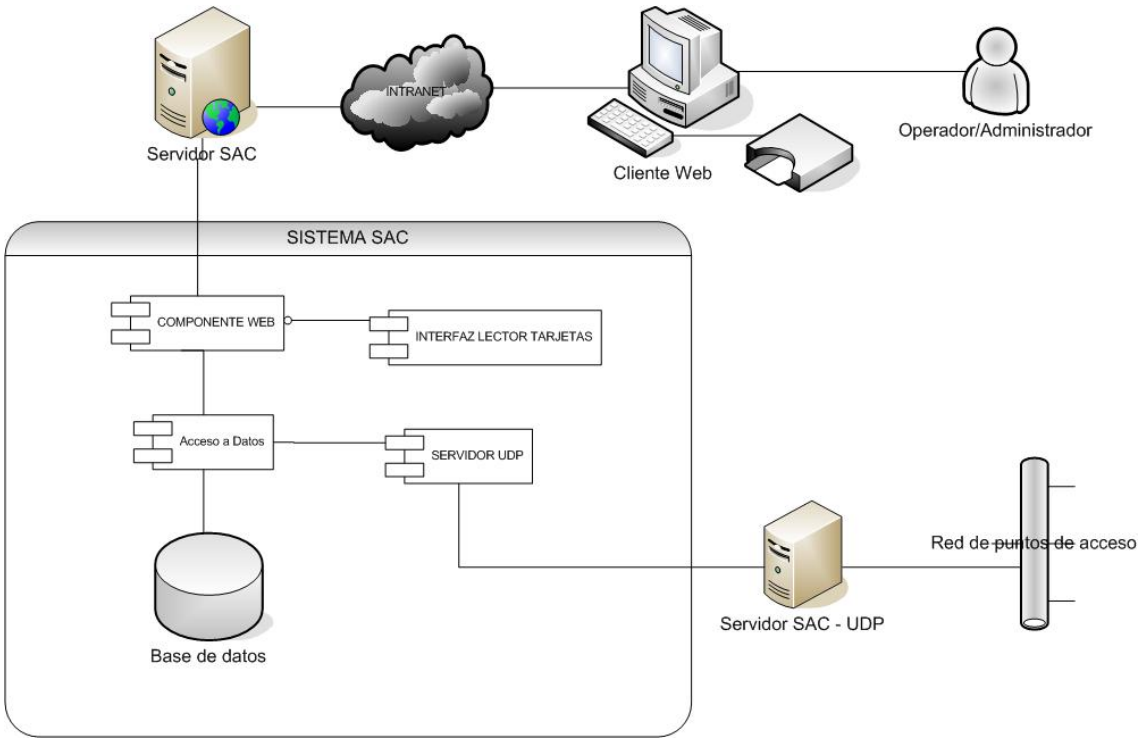


Figura 7.2 – Arquitectura del sistema SAC

Como se puede apreciar en la figura anterior el sistema SAC esta integrado por cuatro componentes esenciales los cuales son:

- Componente Web.
- Interfase Lector de Tarjetas.
- Servidor UDP.

- Acceso a datos.

Para el desarrollo del prototipo de nuestra tesis hemos configurado una PC bajo el sistema operativo Windows en el cual se encuentra el servidor Web IIS, el componente de servidor UDP como un servicio levantado permanentemente; y, el motor de base de datos SQL Server 2000.

7.3.1. Diseño de Interfase gráfica utilizada

La interfase gráfica de los módulos de administrador y operador están creadas para que sean predecibles, definiendo como criterio principal del diseño la usabilidad.

El diseño de la interfaces tiene como características principales un *banner de título en la parte superior, sector izquierdo en el cual se encuentran las opciones agrupadas por categorías dependiendo del perfil del usuario y sector central para la presentación de los contenidos de las páginas correspondientes a las opciones escogidas

La distribución de la información así como las características principales anteriormente mencionadas se detallan a continuación con la presentación

de algunas pantallas en las cuales se identifican su partes de acuerdo al módulo utilizado.

7.3.1.1 Interfase de módulo de administración

- Pantalla de inicio de sesión: inicio de sesión en el cual se ingresa usuario y contraseña.

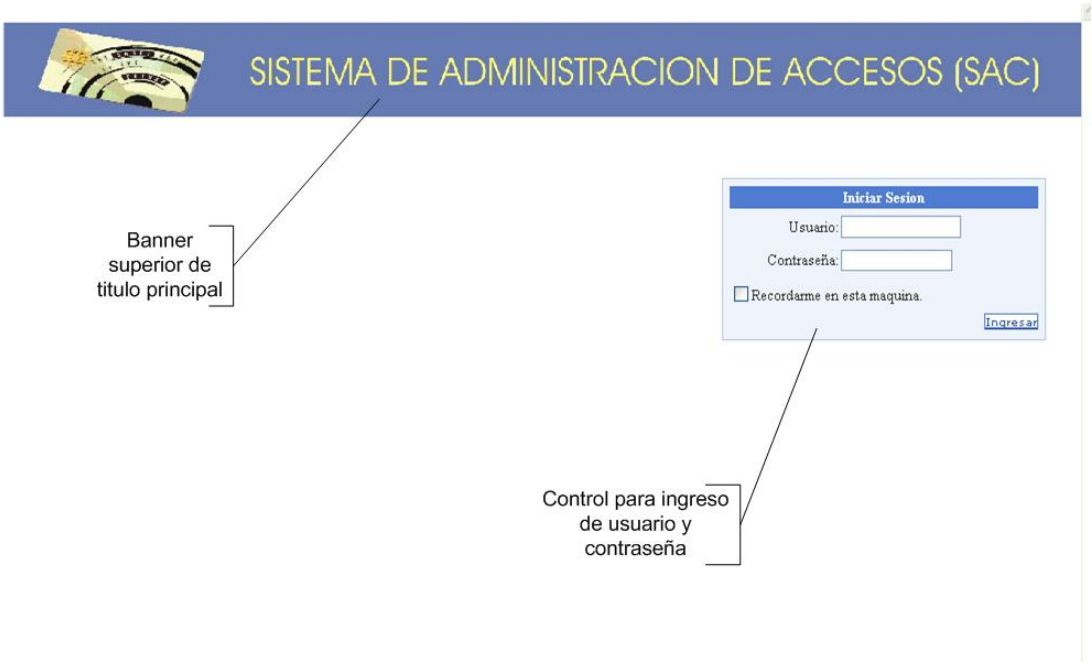


Figura 7.3 – Inicio de sesión en el sistema

- Pantalla Principal: pantalla direccionada luego del correcto inicio de sesión en el cual encontramos un banner principal de titulo, al lado izquierdo encontramos el listado de opciones agrupados mediante las categorías de Accesos y Mantenimiento SAC.

Figura 7.4 – Pantalla principal del sistema

- Pantalla creación de acceso: utilización de formato de formulario para el ingreso de datos, combos de elección y links para la ejecución de procesos.

Figura 7.5 – Pantalla de creación de acceso

7.3.1.2 Interfase de módulo operador

- Pantalla Principal: pantalla direccionada luego del correcto inicio de sesión en el cual encontramos un banner principal de titulo, al lado izquierdo encontramos el listado de opciones agrupados mediante las categorías de Accesos y Consultas SAC.

Figura 7.6 – Pantalla principal del sistema de módulo operador

- Pantalla disponibilidad de habitaciones: en el cual encontramos el banner principal de titulo característico del estilo de la aplicación, encontramos también presentación de reportes en formato visual para su fácil entendimiento y comprensión para cualquier tipo de usuario.

Figura 7.7 – Pantalla disponibilidad de habitaciones

7.3.2. Diseño de los módulos del sistema

Dado que el sistema de administración y control de accesos cuenta con varios componentes corriendo concurrentemente para su completo y correcto funcionamiento, presentamos a continuación la descripción de los módulos y/o componentes que lo integran y la descripción de la interacción entre ellos.

7.3.2.1. Descripción de los módulos/componentes del sistema

- Componente Web: provee la interfase necesaria para la presentación de formularios de creación de accesos, usuarios y consultas de accesos para los administradores y operadores.

- Interfase Lector de Tarjetas: provee las funciones necesarias para la que el componente Web puede realizar la manipulación del dispositivo USB ACR30U y realizar la lectura o grabado de datos en las tarjetas inteligentes

- Servidor UDP: provee las funciones necesarias para la recepción de los requerimientos de ingreso enviados desde los nodos de la red mediante protocolo UDP.

- Acceso a datos: provee las funcionalidades necesarias para acceso a los procedimientos de bases datos necesarios para la comunicación de todos los componentes con la base de datos.

7.3.2.2. Interacción entre los módulos/componentes del sistema

Como se muestra la figura 7.2 de la arquitectura del sistema SAC los módulos y componentes del sistema se encuentran directamente relacionados para el desarrollo integral de las funcionalidades del sistema, en esta sección se detallan la interacción entre todos los componentes.

- Componente Web: utiliza el componente interfase lector de tarjetas para la creación de accesos, registro de accesos e inactivación de usuarios, en la creación de la tarjeta inteligente y la lectura de datos de la misma. El componente acceso a datos para grabar los datos de usuarios y accesos en las tablas de base de datos según la lógica implementada.

- Servidor UDP: utiliza el componente de acceso a datos para grabar los datos de usuarios y registros de ingresos exitosos o fallidos de las peticiones enviadas por los puntos de accesos ubicados en la red.

7.3.3. Diseño de la base de datos

Para el sistema SAC será utilizada como motor de base de datos relacional SQL Server 2000 y será creada una base llamada “TESIS”.

7.3.3.1. Diseño del modelo lógico

Dentro de los parámetros del diseño de base de datos se define una nomenclatura para los nombres de las tablas la cual permitirá su fácil identificación, esta nomenclatura se define de la siguiente manera:

- Tres primeros caracteres siglas del proyecto en nuestro caso SAC.
- Tres siguientes caracteres la definición si es una tabla de almacenamientos de datos utilizamos TBD, si es una tabla relacional utilizamos TBR, o en caso de ser una tabla transaccional utilizamos TBT.
- Los caracteres restantes el nombre que se le proporcionara a la tabla.

A continuación encontrara el modelo de entidad relación de la base de datos TESIS diseñada para el sistema SAC y una breve descripción de las tablas y sus relaciones.

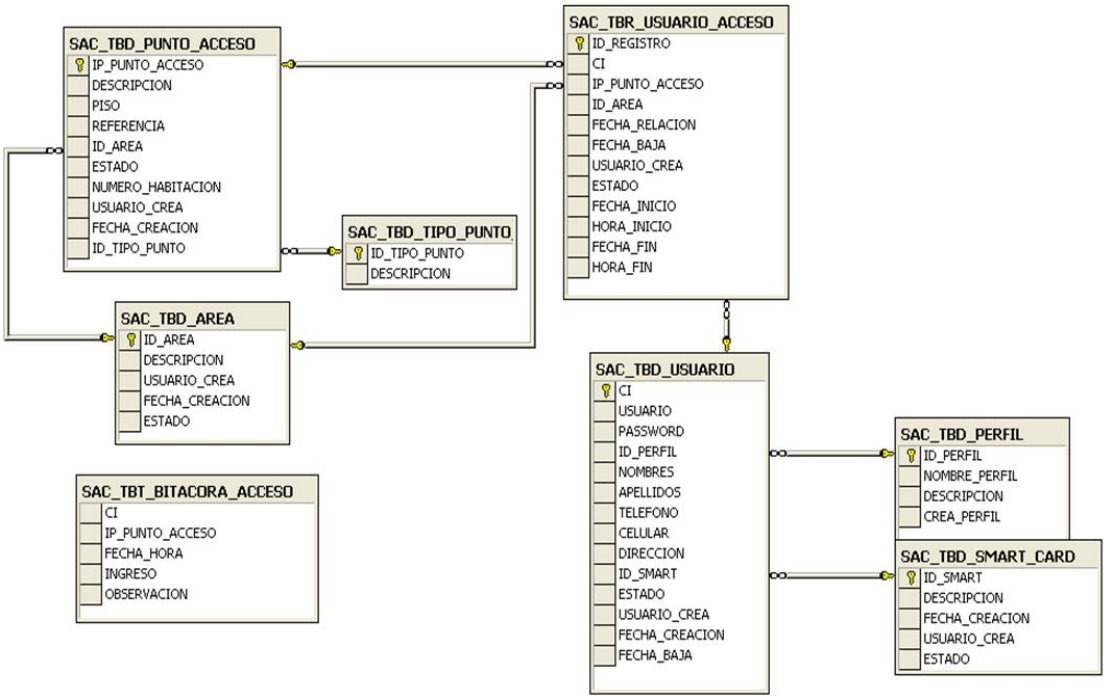


Figura 7.8 – Modelo de entidad relación

Nombre de tabla: SAC_TBD_PUNTO_ACCESO

IP_PUNTO_ACCESO: clave primaria que identifica la dirección IP del punto de acceso en la red.

DESCRIPCION: breve descripción del punto de acceso.

PISO: piso en el cual se encuentra ubicado el punto de acceso.

REFERENCIA: referencia para la fácil ubicación del punto de acceso en un edificio espacio físico.

ID_AREA: clave foránea de la tabla SAC_TBD_AREA.

ESTADO: estado del punto de acceso si se encuentra activo o inactivo.

NUMERO_HABITACION: número de habitación asignada al punto de acceso.

USUARIO_CREA: usuario que realiza la creación del punto de acceso por medio del sistema SAC.

FECHA_CREACION: fecha de creación del punto de acceso por medio del sistema SAC.

ID_TIPO_PUNTO: clave foránea que relaciona con la tabla SAC_TBD_TIPO_PUNTO_ACCESO.

Nombre de tabla: SAC_TBD_AREA

ID_AREA: clave primaria que identifica al área.

DESCRIPCION: breve descripción del área.

USUARIO_CREA: usuario que realiza la creación del área por medio del sistema SAC.

FECHA_CREACION: fecha de creación del área por medio del sistema SAC.

ESTADO: estado del área si se encuentra activa o inactiva.

Nombre de tabla: SAC_TBD_TIPO_PUNTO_ACCESO

ID_TIPO_PUNTO: clave primaria que identifica el tipo de punto de acceso.

DESCRIPCION: breve descripción del tipo de punto de acceso la cual esta definida como habitación y entrada general.

Nombre de tabla: SAC_TBD_USUARIO

CI: clave primaria, cedula de identidad que identifica de manera única a un usuario.

USUARIO: usuario utilizado para el ingreso al módulo WEB de administrador de accesos.

PASSWORD: contraseña utilizada para el ingreso al módulo WEB de administrador de accesos, valores se encuentran encriptados en la base.

ID_PERFIL: clave foránea de la tabla SAC_TBD_PERFIL.

NOMBRES: nombres del usuario.

APELLIDOS: apellidos del usuario.

TELEFONO: teléfono convencional del usuario.

CELULAR: teléfono celular del usuario.

DIRECCION: dirección del usuario.

ID_SMART: clave foránea de la tabla SAC_TBD_SMART_CARD.

ESTADO: estado del usuario si se encuentra activa o inactiva.

USUARIO_CREA: usuario autorizado que realiza la creación del usuario por medio del sistema SAC.

FECHA_CREACION: fecha de creación del usuario por medio del sistema SAC.

FECHA_BAJA: fecha de inactivación del usuario por medio del sistema SAC.

Nombre de tabla: SAC_TBD_SMART_CARD

ID_SMART: clave primaria, identificación única de las tarjetas inteligentes asignadas a cada usuario.

DESCRIPCION: breve descripción de la tarjeta inteligente asignada.

ESTADO: estado de la tarjeta de acceso si se encuentra activa o inactiva.

USUARIO_CREA: usuario que realiza la creación de la tarjeta inteligente por medio del sistema SAC.

FECHA_CREACION: fecha de creación de la tarjeta inteligente por medio del sistema SAC.

Nombre de tabla: SAC_TBD_PERFIL

ID_PERFIL: clave primaria que identifica al perfil.

NOMBRE_PERFIL: nombre asignado al perfil.

DESCRIPCION: breve descripción del perfil.

USUARIO_CREA: usuario que realiza la creación del perfil por medio del sistema SAC.

Nombre de tabla: SAC_TBR_USUARIO_ACCESO

ID_REGISTRO: clave primaria que identifica al registro.

CI: clave foránea de la relación con la tabla SAC_TBD_USUARIO.

IP_PUNTO_ACCESO: clave foránea de la relación con la tabla SAC_TBD_PUNTO_ACCESO.

ID_AREA: clave foránea de la relación con la tabla SAC_TBD_AREA.

FECHA_RELACION: fecha de creación del acceso al usuario por medio del sistema SAC.

FECHA_BAJA: fecha de inactivación del acceso al usuario por medio del sistema SAC.

USUARIO_CREA: usuario que realiza la creación del acceso por medio del sistema SAC.

ESTADO: estado del acceso si se encuentra activa o inactivo.

FECHA_INICIO: fecha de inicio de la asignación de una habitación o área.

HORA_INICIO: hora de inicio de la asignación de una habitación o área para los días indicados.

FECHA_FIN: fecha de fin de la asignación de una habitación o área.

HORA_FIN: hora de fin de la asignación de una habitación o área para los días indicados.

Nombre de tabla: SAC_TBT_BITACORA_ACCESO

CI: cedula o identificación enviada en el requerimiento de ingreso.

IP_PUNTO_ACCESO: IP de donde se genera el requerimiento de ingreso.

FECHA_HORA: hora de la solicitud de requerimiento de ingreso.

INGRESO: resultado si la confirmación de ingreso para el punto de acceso es exitoso o fallido.

OBSERVACION: observación de la confirmación de ingreso, motivo por el cual se niega el acceso.

7.3.3.2. Diseño del modelo conceptual.

Para realizar un modelo lógico eficiente y que cumpla con todas las especificaciones del proyecto de tesis es necesario entender la lógica del negocio/problema a resolver. A continuación se encuentra el modelo conceptual del sistema SAC.

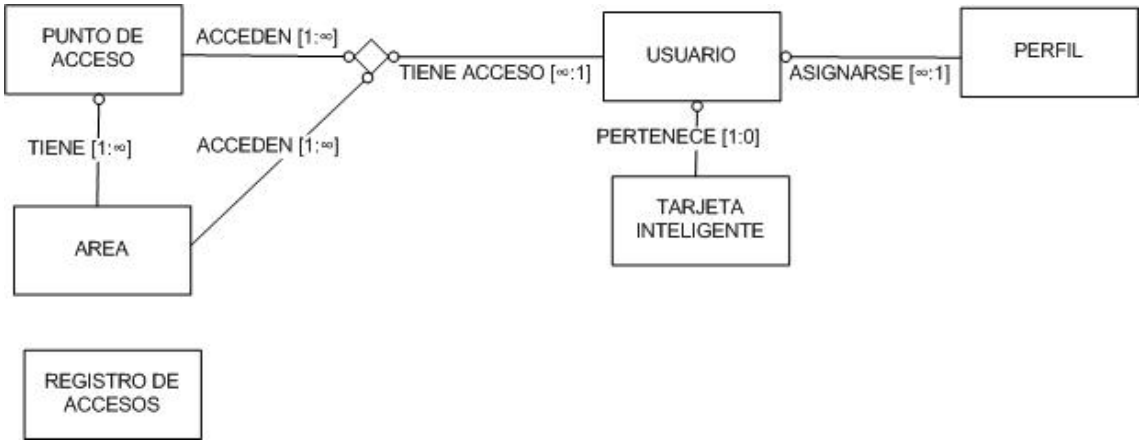


Figura 7.9 – Modelo conceptual

7.3.4. Diseño del esquema de seguridad

El esquema de seguridad diseñado para el módulo Web del sistema SAC esta basado en la autenticación del usuario, luego de que se ha autenticado se verifica el perfil asignado y se presentan las opciones correspondientes.

7.3.4.1. Autenticación

Quien tiene acceso al módulo WEB del sistema SAC posee un usuario y una contraseña el cual tiene que ingresar para que el servidor Web realice la autenticación y luego de que esta sea exitosa se obtenga el perfil asignado al usuario y el componente Web proceda a la

presentación de las opciones asignadas, la cuales están estructuradas mediante un archivo XML [15].

7.3.4.1. Manejo de sesiones

Cuando una persona accede al sistema luego de autenticarse se procede con la creación de variables de sesiones en las cuales se guarda el usuario y el rol que tiene asignado.

De tal manera que en todas las páginas del sistema se valide que las variables de sesiones se encuentren activas para continuar con el flujo de las opciones, caso contrario se deberá solicitar el inicio de sesión y se reenviará a la página de inicio del sistema.

7.3.5. Pruebas

7.3.5.1. Tipos de pruebas

El tipo de pruebas escogidas son de tipo caja negra las cuales validan al sistema comparando los resultados con la salida esperada dado un conjunto de datos ingresados.

Las pruebas deben mostrar como el sistema satisface los requerimientos funcionales como los no funcionales, garantizando que el resultado de las pruebas pueda ser medido con valores de completada con éxito o no, con lo cual mantenemos una estructura de prueba cerrada.

7.3.5.1. Diseño de pruebas

Prueba No 1: **Creación de usuario y accesos a una habitación o una área**

Pre-requisitos:

- El usuario que va a ingresar a la aplicación debe estar registrado con el perfil de operador o administrador.
- Que el acceso que se va a dar al usuario se encuentre registrado como punto de acceso y se encuentre asignado a un área.

Instrucciones de configuración:

- Ninguna.

Instrucciones de la prueba:

- El usuario ingresa con su usuario y contraseña al sistema.
- El usuario da clic en la opción “Creación de Acceso”.
- El usuario ingresa la identificación del nuevo usuario y los datos personales.
- El usuario da clic en el botón “Guardar”.
- El usuario da clic en el botón “Crear Smart”.
- Seleccionar los días de inicio, fin y el respectivo horario para dar el acceso a la habitación.
- Seleccionar la habitación
- El usuario da clic en el botón “Agregar”.
- Seleccionar los días de inicio, fin y el respectivo horario para dar el acceso al área.
- Seleccionar el área
- El usuario da clic en el botón “Agregar”.

Comportamiento aceptable:

- El sistema garantiza la correcta ejecución de la opción y la correcta navegabilidad en la página.

- El sistema presenta mensajes de procesamiento realizado con éxito en las transacciones.

Prueba No 2: **Punto de acceso envía requerimientos a servidor y recibe confirmación para apertura de puerta positiva o negativa**

Pre-requisitos:

- Reconocimiento como nodo de la red al punto de acceso que enviará el requerimiento de ingreso.
- El servicio del servidor UDP se encuentre levantado.

Instrucciones de configuración:

- La tarjeta inteligente a ser utilizada en el punto de acceso tiene que estar previamente grabada con para ingresar al punto de acceso solicitado.

Instrucciones de la prueba:

- El usuario ingresa la tarjeta inteligente en el dispositivo de acceso y la retira.
- Espera unos segundos para la apertura del relé del dispositivo.

Comportamiento aceptable:

- El sistema garantiza la correcta atención de los requerimientos de ingreso.
- El sistema presenta mensajes de procesamiento realizado con éxito a los requerimientos.
- El sistema presenta mensajes de registro en base de datos realizado con éxito.
- El sistema envía confirmación.

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

1. La transmisión de datos a través de una red Ethernet con sistemas embebidos facilita el control de dispositivos remotos con los cuales se quiere establecer comunicación.
2. El sistema puede migrar con facilidad a diferentes tipos de aplicaciones en donde se requiera el control de acceso y registro de personal.
3. El sistema administrador de accesos puede ser adaptado para que adquiera datos de distintos tipos de dispositivos que puedan comunicarse mediante una red Ethernet.
4. Con el trabajo realizado en la tesis se ve la posibilidad de actualizar varios equipos para permitir así la comunicación vía Ethernet.
5. El diseño de múltiples usuarios le brinda al sistema las condiciones de seguridad necesarias para el correcto manejo de la información generada por el mismo al mismo tiempo que se restringe el acceso a áreas determinadas.
6. En este sistema sólo se empleó una entrada consistente en la información de la tarjeta inteligente y una sola salida que es la que controla la cerradura eléctrica. Esto no impide para que sobre la misma base se puedan desarrollar nuevos proyectos con múltiples entradas y salidas.

7. El sistema de administración y control de acceso puede ser implementado en edificios pertenecientes a entidades públicas o privadas que requieran un monitoreo constante y niveles de autorización a diversas áreas. Como clientes potenciales del sistema identificamos edificios de entidades educativas como la ESPOL, ministerios de gobierno, hospitales entre otros.

RECOMENDACIONES

1. Para la comunicación de sistemas embebidos a la Web hay que tener en cuenta el modelo referencial de las capas de la pila de microchip para el desarrollo de la aplicación, ya que no se debe saltar ninguna de estas capas iniciales (ARP).
2. Para llevar el prototipo al sector comercial se requiere de una base para el controlador que sea de fácil montaje en pared y de una central UPS para la alimentación de la cerradura en el caso de falla con el sistema de alimentación del edificio.
3. Debido al aumento de los sistemas con tarjetas RFID (o sin contacto) se recomienda el desarrollo del sistema con este tipo de tarjetas teniendo en cuenta que el cambio es solo en la forma de lectura de la tarjeta más no en el protocolo de comunicación.
4. Añadir módulos de adquisición de datos al sistema para ampliar sus funcionalidades permitiendo también el monitoreo y control de otros equipos tales como motores y sensores que puedan enviar información de su estado por medio de una red Ethernet.

APÉNDICES

APÉNDICE A: TIPOS DE TARJETAS SOPORTADAS POR EL LECTOR DE TARJETAS INTELIGENTES ACR30

La siguiente tabla resume los valores que deben ser especificados en el comando SET_CARD_TYPE para un tipo particular de tarjeta a ser utilizada y cómo los bits en la respuesta a un comando GET_ACR_STAT corresponde con los respectivos tipos de tarjeta.

Código del tipo de tarjeta	Tipo de tarjeta
00_H	Auto selección del protocolo de comunicación T = 0 ó T =1
01_H	GPM103, SLE4406 SLE4436, SLE5536 (firmware 2.10 en adelante)
02_H	I2C
05_H	SLE4418, SLE4428
06_H	SLE4432, SLE4442
$0C_H$	Tarjetas microcontroladas con protocolo de comunicación T = 0
$0D_H$	Tarjetas microcontroladas con protocolo de comunicación T = 1

$C0_H$	Tarjetas SAM con protocolo de comunicación T = 0
$D0_H$	Tarjetas SAM con protocolo de comunicación T = 1

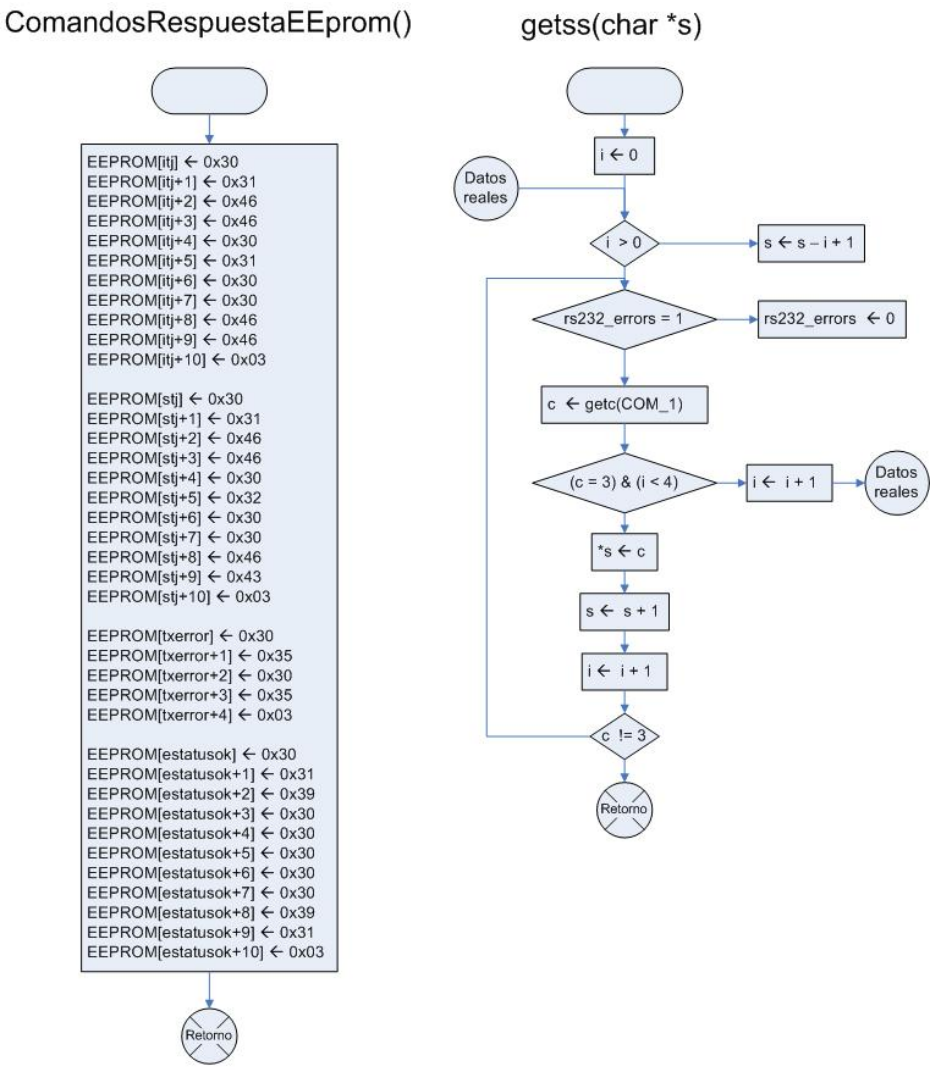
APÉNDICE B: CÓDIGOS DE ESTADO DE RESPUESTA DEL LECTOR DE TARJETAS INTELIGENTES ACR30

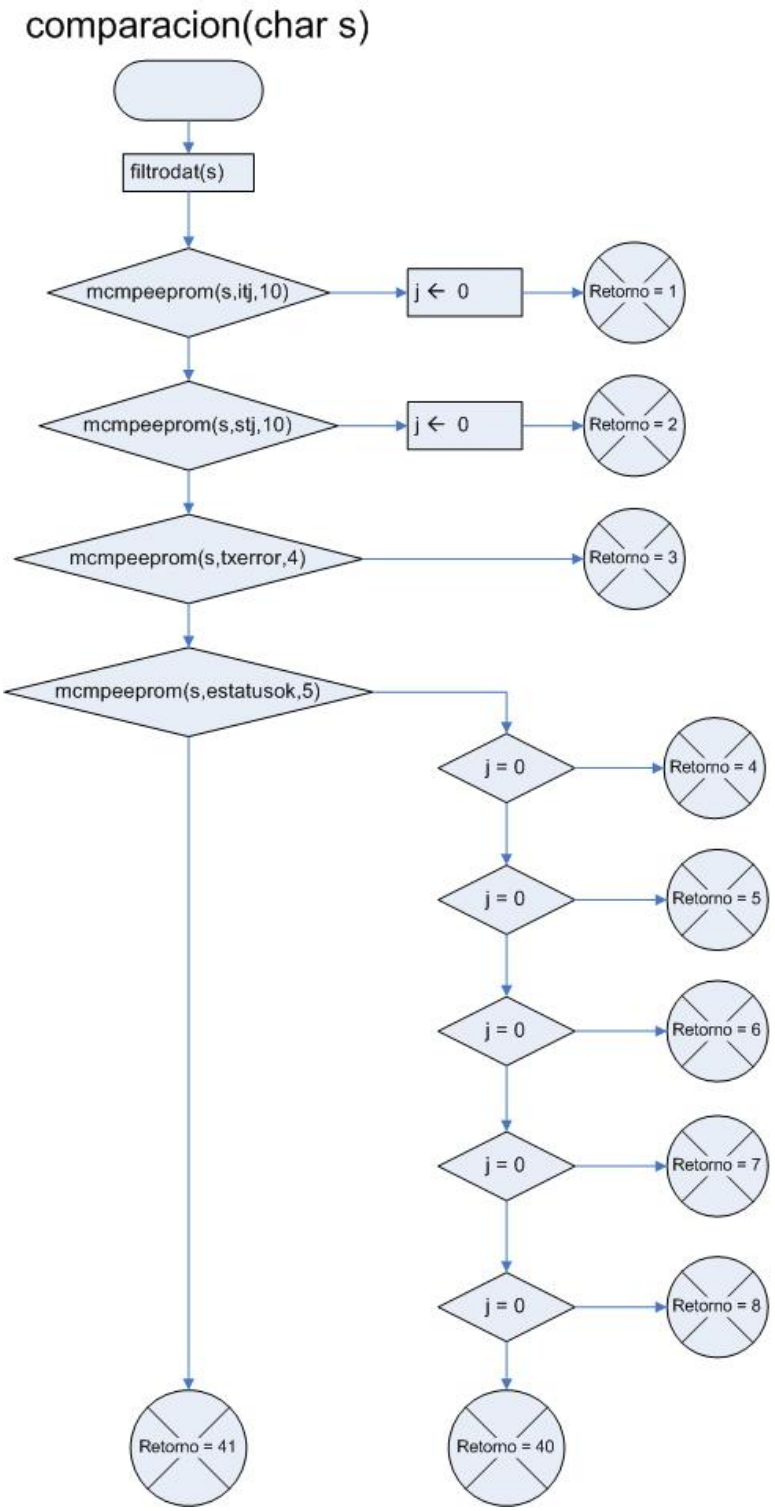
La siguiente tabla resume los posibles códigos de estado de los bytes SW1, SW2 retornados por el lector ACR30.

SW1	SW2	Estado
90	00	Comando ejecutado exitosamente.
90	01	Comando ejecutado exitosamente con protocolo T = 1 (Sólo en respuesta a un comando de RESET)
90	10	Comando ejecutado exitosamente cuando un protocolo sincrónico es usado (Sólo en respuesta a un comando de RESET). El tipo de tarjeta debe ser seleccionado usando el comando SELECT_CARD_TYPE.
60	01	No se seleccionó un tipo de tarjeta.
60	02	No hay tarjeta en el lector.
60	03	Tipo de tarjeta seleccionado está equivocado.
60	04	La tarjeta no ha sido energizada.
60	05	Código de instrucción Inválido.
60	20	Falla en la tarjeta.

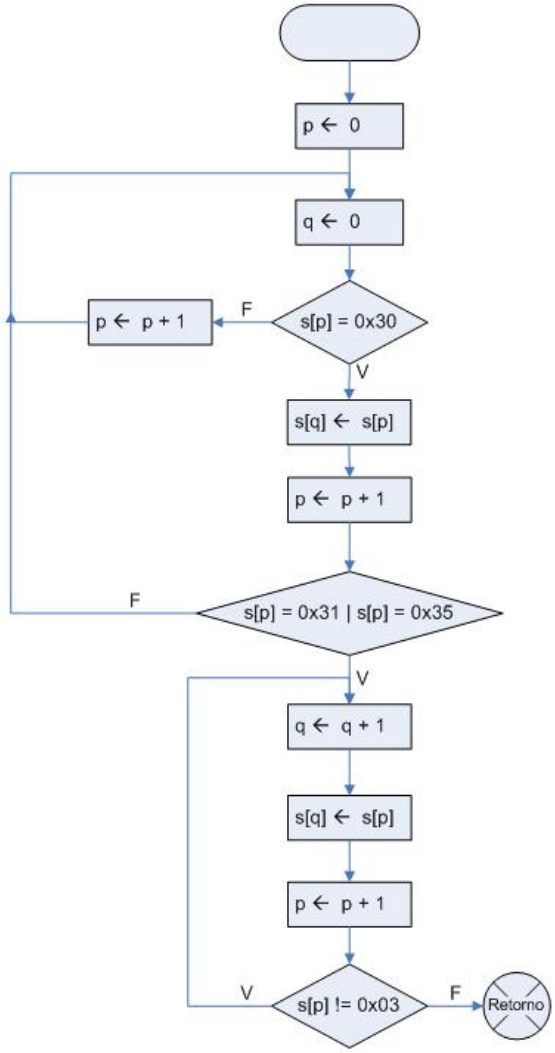
60	22	Cortocircuito en un conector de la tarjeta.
62	01	Código secreto inválido.
67	01	Comando incompatible con el tipo de tarjeta seleccionado.
67	02	Error de dirección en la tarjeta.
67	03	Error de longitud de datos.
67	04	Longitud inválida de respuesta (Con el comando READ)
67	05	Código secreto asegurado.
67	12	Comando APDU abortado.

APÉNDICE C: DIAGRAMA DE FLUJO DE LAS FUNCIONES IMPLEMENTADAS EN EL *MICROCONTROLADOR 2* –INTERFASE ENTRE EL LECTOR DE TARJETAS INTELIGENTES Y EL DISPODITIVO DE RED UTILIZADO-

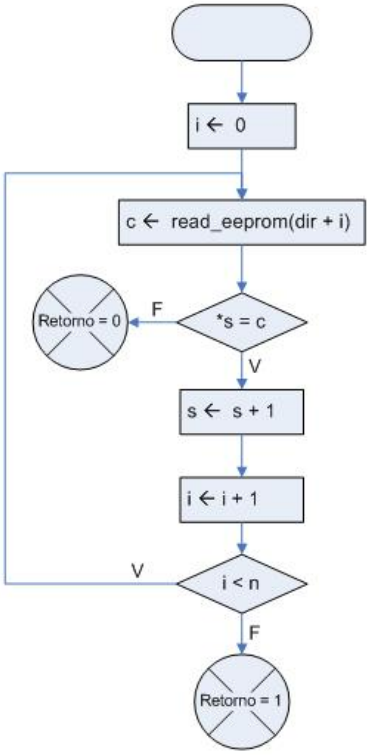



comparacion(char s)
filtrodat(s)
mcmpeeprom(s,itj,10)
j ← 0
Retorno = 1
mcmpeeprom(s,stj,10)
j ← 0
Retorno = 2
mcmpeeprom(s,txerror,4)
Retorno = 3
mcmpeeprom(s,estatusok,5)
j = 0
Retorno = 4
j = 0
Retorno = 5
j = 0
Retorno = 6
j = 0
Retorno = 7
j = 0
Retorno = 8
Retorno = 41
Retorno = 40

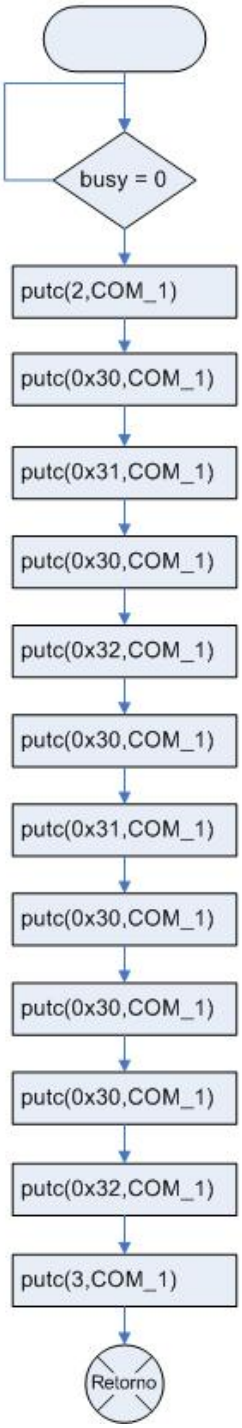
filtrodat(char *s)



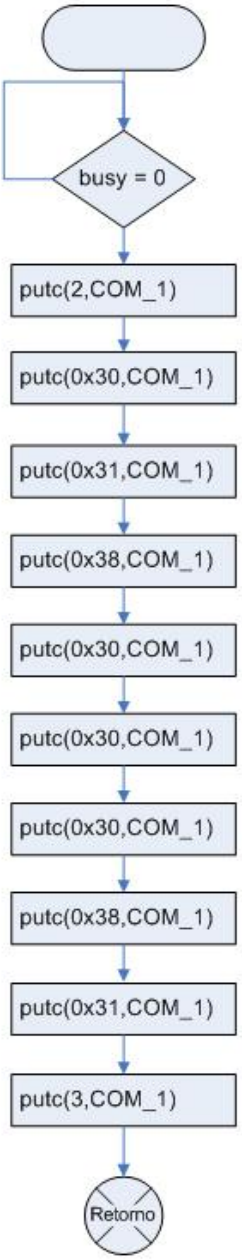
mcmpeeprom(char *s,int dir,int n)



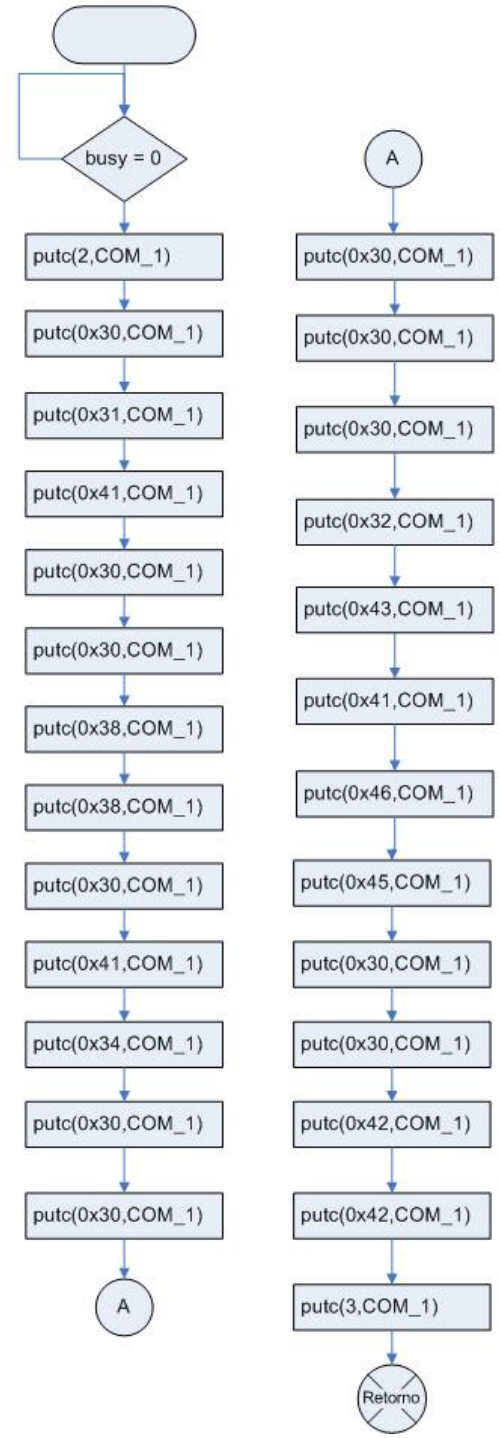
seleccionar_tarjeta()



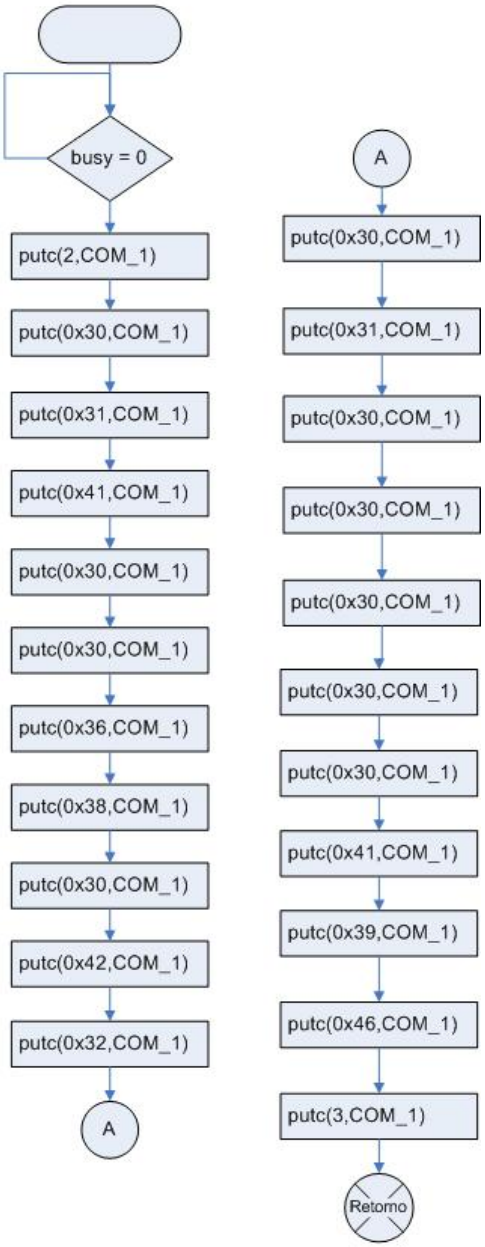
reset_card()



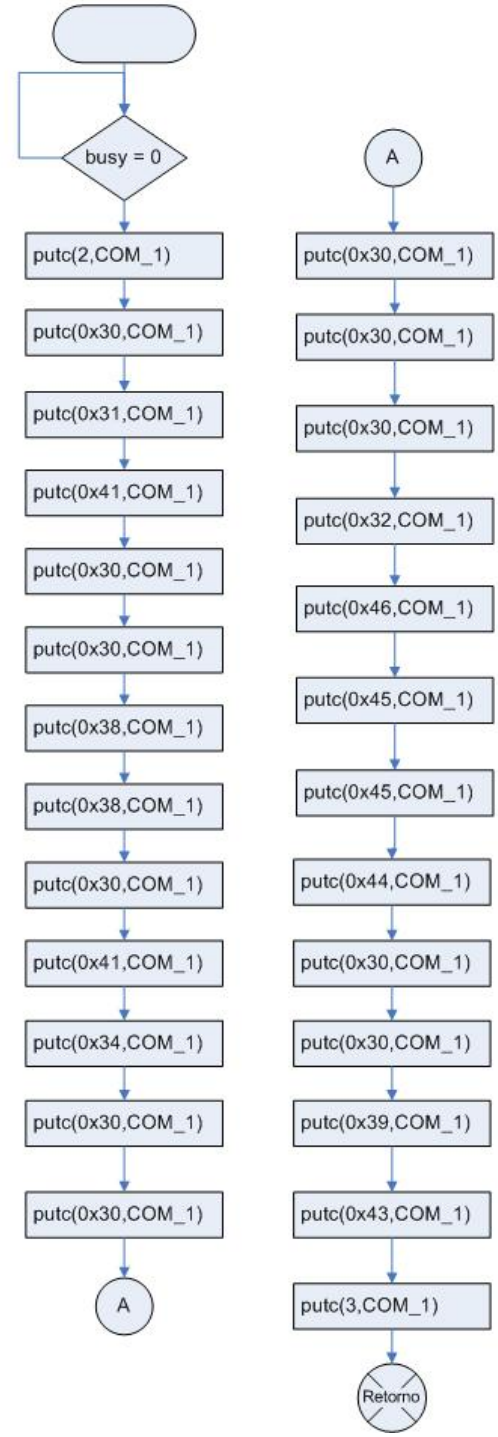
Select_file_1()



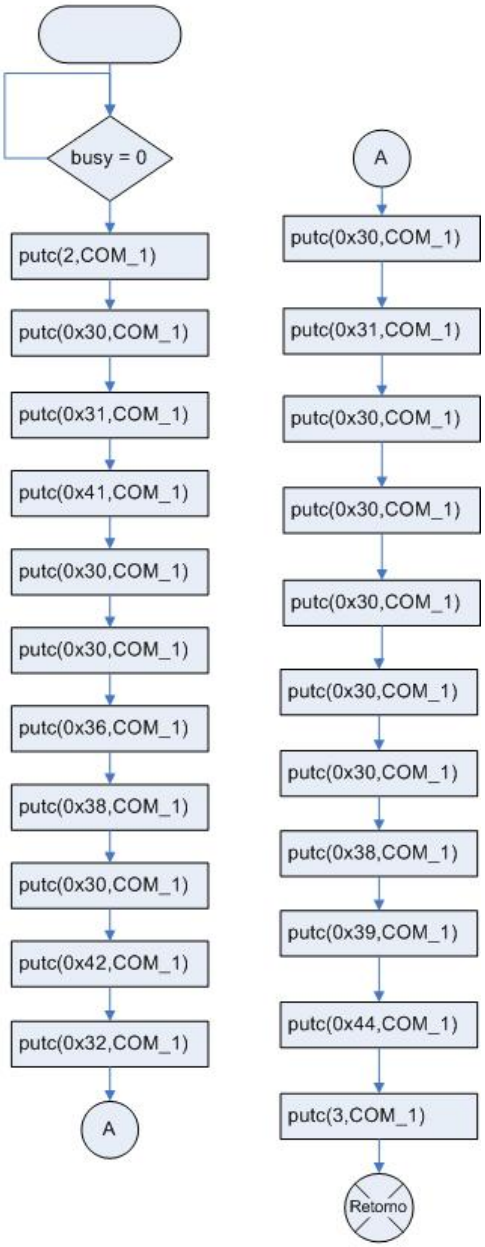
read_data_1()



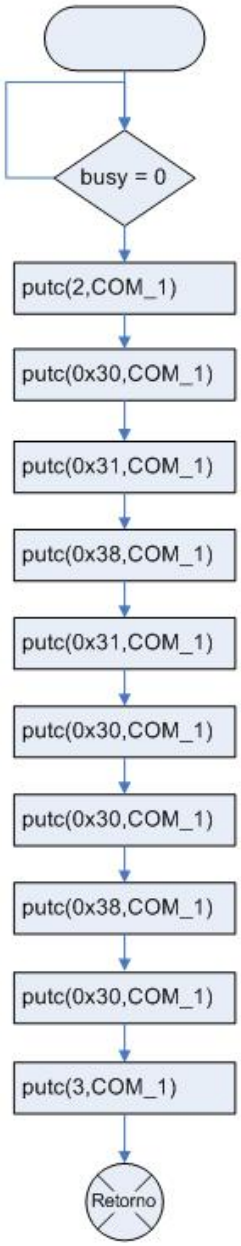
Select_file_2()



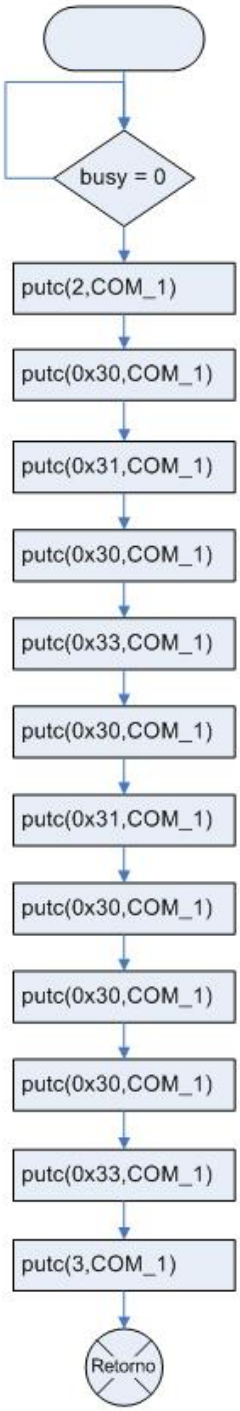
read_data_2()



poweroff()



seleccionar_retraso()



APÉNDICE D: DESCRIPCIÓN DE LAS FUNCIONES IMPLEMENTADAS EN EL MICROCONTROLADOR 2 – INTERFASE ENTRE EL LECTOR DE TARJETAS INTELIGENTES Y EL DISPODITIVO DE RED UTILIZADO-

seleccionar_tarjeta()

Esta función envía un comando que fija el tipo de tarjeta para que seleccione automáticamente el protocolo T = 0 ó T = 1. firmware dentro del ACR30 ajusta el protocolo de comunicación entre el lector y la tarjeta insertada de acuerdo con el tipo de tarjeta seleccionada.

Sintaxis

void seleccionar_tarjeta()

Parámetros

Ninguno

Valores de retorno

Ninguno

Condiciones previas

Ninguna

Observaciones

Referirse al apéndice A para cambiar el tipo de tarjeta de manera manual.

Ejemplo

```
switch (k)
    {
        case 1 : seleccionar_retraso();goto inicio;
        case 2 : seleccionar_tarjeta();goto inicio;
        case 3 : reset_card();goto inicio;
        ...
```

reset_card()

Esta función envía un comando que enciende la tarjeta insertada en el lector y realiza un reset en la tarjeta. La fuente de alimentación de la tarjeta no se apaga.

Sintaxis

void reset_card()

Parámetros

Ninguno

Valores de retorno

Ninguno

Condiciones previas

Ninguna

Observaciones

Cada tipo de tarjeta tiene un comando diferente de reset. Referirse a la bibliografía para obtener mayores detalles.

Ejemplo

```
switch (k)
    {
        case 1 : seleccionar_retraso();goto inicio;
        case 2 : seleccionar_tarjeta();goto inicio;
        case 3 : reset_card();goto inicio;
        ...
```

Select_file_x()

Selecciona un dato de un archivo para subsecuentemente ejecutar los comandos de lectura y escritura.

Sintaxis

void Select_file()

Parámetros

Ninguno

Valores de retorno

Ninguno

Condiciones previas

Haber encendido la tarjeta mediante el comando reset_card().

Observaciones

Se tiene que seleccionar un registro que se encuentre dentro del rango de archivos

Ejemplo

```
switch (k)
    {
          ...
        case 4 : reset_card();goto inicio;
        case 5 : select_file_1();goto inicio;
        case 6 : read_data( ); goto inicio;
        ...
```

read_data_x()

Lee un número de bytes (máximo la longitud del registro) de un registro en el archivo actualmente seleccionado.

Sintaxis

void read_data_x()

Parámetros

Ninguno

Valores de retorno

Ninguno

Condiciones previas

Haber seleccionado anteriormente un archivo que se realiza con el comando *select_file_x().*

Observaciones

Referirse al apéndice A para cambiar el tipo de tarjeta de manera manual.

Ejemplo

```
switch (k)
    {
        ...
        case 4 : reset_card();goto inicio;
        case 5 : select_file_1();goto inicio;
        case 6 : read_data_1( ); goto inicio;
        ...
```

poweroff()

Este comando apaga la tarjeta insertada en el lector de tarjetas inteligentes.

Sintaxis

void poweroff()

Parámetros

Ninguno

Valores de retorno

Ninguno

Condiciones previas

Haber encendido la tarjeta mediante el comando reset_card().

Observaciones

No se debe dejar encendida la tarjeta si se culmino con la lectura o escritura de la misma.

Ejemplo

```
switch (k)
    {
        ...
        case 5 : select_file();goto inicio;
        case 6 : read_data( ); goto inicio;
        case 7 : /*guardar los datos previamente*/ poweroff( ); goto inicio;
        ...
```

seleccionar_retraso()

Este comando se usa para controlar la velocidad de comunicación entre el lector ACR30 y el dispositivo servidor. La velocidad de comunicación se

controla por medio de dos factores: el factor de retraso y la tasa de baudios.

Sintaxis

void seleccionar_retraso()

Parámetros

Ninguno

Valores de retorno

Ninguno

Condiciones previas

Ninguna

Observaciones

Se debe tener en cuenta que la velocidad de transmisión sea la misma entre el lector ACR30 y el controlador.

Ejemplo

```
switch (k)
    {
        case 1 : seleccionar_retraso();goto inicio;
        case 2 : seleccionar_tarjeta();goto inicio;
        case 3 : reset_card();goto inicio;
        ...
```

BIBLIOGRAFÍA

[1] “ACR30 USB Card Reader/Writer Reference Manual”, Advanced Card Systems Ltd., 2004.

[2] “ACOS2 Smart Card with 1K/8K EEPROM Reference Manual”, Advanced Card Systems Ltd., 2004.

[3] “PIC18F1220/1320 Datasheet”, Microchip Technology Inc., 2003.

[4] “PIC18F452 Datasheet”, Microchip Technology Inc., 2003.

[5] Axelson Jan, "Embedded Ethernet and Internet Complete", Sybex, 2003.

[6] Gardner Nigel, “PICmicro MCU C. An introduction to programming The Microchip PIC in CCS C”, Bluebird Electronics, 2002.

[7] Modular Electronic Solutions, “SBC45EC Single Board computer for 44pin PLCC PICs”, 2006. Disponible en http://www.modtronix.com/products/sbc45ec/sbc45ecr2.pdf

[8] “AN833: The Microchip TCP/IP Stack”, Microchip Technology Inc., 2003.

[9] “A Multi-threaded TCP/UDP Server and Client”, 2007. Disponible en http://www.c-sharpcorner.com

[10] “Socket-level Programming”, 2007. Disponible en http://jan.netcomp.monash.edu.au

[11] “Connection strings reference list on how to connect to SQL Server”. Disponible en http://www.connectionstrings.com

[12] “Códigos ejemplo Para el desarrollo de proyectos ASP. Net.”, 2007. Disponible en http://msdn2.microsoft.com/es-es/default.aspx

[13] Price Jason, “Mastering C# Database Programming”, Sybex, 2003.

[14] “Manejo de base de datos con AJAX”, 2007. Disponible en http://ajax.schwarz-interactive.de

[15] “Asynchronous JavaScript and XML”, 2007. Disponible en http://ajax.schwarz-interactive.de