



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería Mecánica Eléctrica

**DISEÑO DE UN SISTEMA DE IDENTIFICACIÓN ESTUDIANTIL POR
RADIOFRECUENCIA PARA EL LABORATORIO DE ELECTRÓNICA DE LA
ESCUELA DE INGENIERÍA MECÁNICA ELÉCTRICA**

José Manuel Márquez Alemán

Asesorado por el Ing. Byron Odilio Arrivillaga Méndez

Guatemala, noviembre de 2012

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**DISEÑO DE UN SISTEMA DE IDENTIFICACIÓN ESTUDIANTIL POR
RADIOFRECUENCIA PARA EL LABORATORIO DE ELECTRÓNICA DE LA
ESCUELA DE INGENIERÍA MECÁNICA ELÉCTRICA**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

JOSÉ MANUEL MÁRQUEZ ALEMÁN

ASESORADO POR EL ING. BYRON ODILIO ARRIVILLAGA MÉNDEZ

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO ELECTRÓNICO

GUATEMALA, NOVIEMBRE DE 2012

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

| | |
|------------|-------------------------------------|
| DECANO | Ing. Murphy Olympto Paiz Recinos |
| VOCAL I | Ing. Alfredo Enrique Beber Aceituno |
| VOCAL II | Ing. Pedro Antonio Aguilar Polanco |
| VOCAL III | Inga. Elvia Miriam Ruballos Samayoa |
| VOCAL IV | Br. Juan Carlos Molina Jiménez |
| VOCAL V | Br. Mario Maldonado Muralles |
| SECRETARIO | Ing. Hugo Humberto Rivera Pérez |

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

| | |
|------------|--|
| DECANO | Ing. Murphy Olympto Paiz Recinos |
| EXAMINADOR | Ing. José Aníbal Silva de los Ángeles |
| EXAMINADOR | Ing. Luis Fernando García Cienfuegos |
| EXAMINADOR | Ing. Marvin Marino Hernández Fernández |
| SECRETARIO | Ing. Hugo Humberto Rivera Pérez |

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

DISEÑO DE UN SISTEMA DE IDENTIFICACIÓN ESTUDIANTIL POR RADIOFRECUENCIA PARA EL LABORATORIO DE ELECTRÓNICA DE LA ESCUELA DE INGENIERÍA MECÁNICA ELÉCTRICA

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería Mecánica Eléctrica, con fecha 12 de abril de 2012.


José Manuel Márquez Alemán



Guatemala, 5 de octubre de 2012.

Ingeniero
Carlos Eduardo Guzmán Salazar
Coordinador del Área de Electrónica
Escuela Mecánica Eléctrica
Facultad de Ingeniería
USAC

Ingeniero Guzmán:

De acuerdo con la designación efectuada por la Dirección de Escuela, me permito informarle que he procedido a asesorar el Trabajo de Graduación titulado: DISEÑO DE UN SISTEMA DE IDENTIFICACIÓN ESTUDIANTIL POR RADIOFRECUENCIA PARA EL LABORATORIO DE ELECTRÓNICA DE LA ESCUELA DE INGENIERÍA MECÁNICA ELÉCTRICA, desarrollado por el estudiante JOSÉ MANUEL MÁRQUEZ ALEMÁN, carne 2006-11108 y, encontrándolo satisfactorio en su contenido y resultados, me permito dar aprobación al mismo, remitiéndolo a esa Coordinación para el tramite pertinente, en el entendido que el Autor y este Asesor somos responsables del contenido y conclusiones del Trabajo.

Agradeciendo la atención a la presente, me es grato suscribirme, deseándole éxitos en sus labores cotidianas.

Atentamente

Ing. Byron Arrivillaga Méndez
C/115217

Ing. Byron Odilio Arrivillaga Méndez
Colegiado 5217
ASESOR



Ref. EIME 50.2012.

Guatemala, 11 de OCTUBRE 2012.

Señor Director
Ing. Guillermo Antonio Puente Romero
Escuela de Ingeniería Mecánica Eléctrica
Facultad de Ingeniería, USAC.

Señor Director:

**Me permito dar aprobación al trabajo de Graduación titulado:
"DISEÑO DE UN SISTEMA DE IDENTIFICACIÓN
ESTUDIANTIL POR RADIOFRECUENCIA PARA EL
LABORATORIO DE ELECTRÓNICA DE LA ESCUELA DE
INGENIERÍA MECÁNICA ELÉCTRICA", del estudiante José
Manuel Márquez Alemán, que cumple con los requisitos
establecidos para tal fin.**

Sin otro particular, aprovecho la oportunidad para saludarle.

Atentamente,
ID Y ENSEÑADA TODOS

Ing. Carlos Eduardo Guzmán Salazar
Coordinador Área Electrónica



CEGS/sr



FACULTAD DE INGENIERIA

REF. EIME 56.2012.

El Director de la Escuela de Ingeniería Mecánica Eléctrica, después de conocer el dictamen del Asesor, con el Visto Bueno del Coordinador de Área, al trabajo de Graduación del estudiante: JOSÉ MANUEL MÁRQUEZ ALEMÁN titulado: "DISEÑO DE UN SISTEMA DE IDENTIFICACIÓN ESTUDIANTIL POR RADIOFRECUENCIA PARA EL LABORATORIO DE ELECTRÓNICA DE LA ESCUELA DE INGENIERÍA MECÁNICA ELÉCTRICA", procede a la autorización del mismo.

Ing. Guillermo Antonio Puente Romero



GUATEMALA, 17 DE OCTUBRE 2012.



DTG. 610.2012

El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería Mecánica Eléctrica, al trabajo de graduación titulado: **DISEÑO DE UN SISTEMA DE IDENTIFICACIÓN ESTUDIANTIL POR RADIOFRECUENCIA PARA EL LABORATORIO DE ELECTRÓNICA DE LA ESCUELA DE INGENIERÍA MECÁNICA ELÉCTRICA**, presentado por el estudiante universitario **José Manuel Márquez Alemán**, autoriza la impresión del mismo.

IMPRÍMASE:

A handwritten signature in black ink, appearing to read 'Alfredo Beber'.

Ing. Alfredo Enrique Beber Aceituno
Decano en Funciones

Guatemala, 22 de noviembre de 2012.



/gdech

ACTO QUE DEDICO A:

- Dios** Gracias Padre por este maravilloso día que me has regalado, y porque ha sido tu inmenso amor el motor que me ha impulsado a alcanzar tan ansiada meta.
- Mis padres** José Márquez y Doris Alemán, porque son ellos todo mi ejemplo de vida, de arduo trabajo, de humildad y gran amor. Gracias por todo su amor, ayuda, comprensión y enseñanzas en cada día de mi vida.
- Mis hermanos** Lucía Márquez, Ana Márquez y Alberto Márquez, por todo el apoyo que siempre me han dado, y porque quiero inspirarlos a que sigan sus sueños y alcancen las metas que se propongan.
- Mis amigos** Mario Morales, Luis Castillo, Oscar Guerra, Luis Rodríguez, Julio Xocoy, Daniel Camargo, Carlos Coguox, Fredy González, William García, Dámaris Gómez, Claudia Contreras, María José Pivaral y Brenda Cermeño, por todo su apoyo incondicional, porque es en los momentos difíciles que amigos como ustedes valen su peso en oro.
- Mis compañeros de estudios** Porque de una u otra manera compartimos la alegría de vivir la etapa de la universidad, y el sueño de convertirnos en grandes profesionales.

AGRADECIMIENTOS A:

| | |
|------------------------------------|--|
| Laboratorio de Electrónica | Por su entera colaboración, prestando el equipo y las instalaciones para el desarrollo del presente trabajo de graduación. |
| Ing. Byron Arrivillaga | Por su apoyo y asesoría en este trabajo de graduación, depositando su confianza en mi persona para elaborarlo. |
| Ing. Guillermo Puente | Por su amistad y confianza que han sido un regalo para mí, así como sus enseñanzas y consejos en la culminación de mi carrera. |
| Departamento de Matemáticas | Porque durante el tiempo en que elaboré este trabajo de graduación, me dio la oportunidad de trabajar como auxiliar de cátedra, junto con personas que compartí muy buenos momentos. |

ÍNDICE GENERAL

| | |
|--|-------|
| ÍNDICE DE ILUSTRACIONES | VII |
| LISTA DE SÍMBOLOS | IX |
| GLOSARIO | XI |
| RESUMEN | XIX |
| OBJETIVOS | XXI |
| INTRODUCCIÓN | XXIII |
| | |
| 1. ASPECTOS GENERALES DE UN SISTEMA RFID | 1 |
| 1.1. Introducción | 1 |
| 1.2. Principio de funcionamiento | 2 |
| 1.2.1. Frecuencias de radio | 3 |
| 1.2.2. Modulación | 4 |
| 1.2.3. Bandas de frecuencia para sistemas RFID | 5 |
| 1.3. Componentes de un sistema RFID | 6 |
| 1.3.1. Lector de etiquetas de radiofrecuencia | 7 |
| 1.3.2. Etiquetas de radiofrecuencia | 8 |
| 1.3.2.1. Etiquetas activas | 9 |
| 1.3.2.2. Etiquetas pasivas | 9 |
| 1.3.3. Antenas | 9 |
| 1.3.4. Computadora | 10 |
| | |
| 2. MICROCONTROLADOR | 13 |
| 2.1. Introducción al microcontrolador PIC | 14 |
| 2.1.1. Familias del microcontrolador PIC | 14 |
| 2.2. Arquitectura | 16 |

| | | |
|----------|--|----|
| 2.3. | Procesador CPU | 17 |
| 2.4. | Configuraciones del oscilador | 18 |
| 2.4.1. | Multiplicador de frecuencia con PLL | 20 |
| 2.5. | Memoria..... | 21 |
| 2.5.1. | Memoria de programa | 21 |
| 2.5.2. | Memoria RAM | 22 |
| 2.5.3. | Memoria EEPROM | 22 |
| 2.6. | Puertos digitales de E/S | 23 |
| 2.6.1. | Puertos de comunicación | 24 |
| 2.6.1.1. | Comunicación SPI | 24 |
| 2.7. | Interrupciones | 26 |
| 2.8. | Microcontrolador PIC18F4520..... | 27 |
| 2.8.1. | Descripción del patillaje | 28 |
| 2.9. | Programación del microcontrolador..... | 29 |
| 2.9.1. | Escritura del código | 29 |
| 2.9.2. | Compilador <i>MikroBasic</i> | 32 |
| 2.9.2.1. | Librerías de funciones | 33 |
| 2.9.3. | Carga del código al PIC..... | 34 |
| 3. | MÓDULOS PERIFÉRICOS ADICIONALES..... | 35 |
| 3.1. | Módulo <i>RFid Reader</i> de <i>MikroElektronika</i> | 35 |
| 3.1.1. | Componentes y diagrama circuital..... | 37 |
| 3.1.2. | Circuito integrado transceptor EM4095..... | 38 |
| 3.1.3. | Señales de datos y control | 39 |
| 3.1.4. | Flujo de señales en la operación de lectura..... | 39 |
| 3.2. | Tarjetas <i>RFid card</i> | 41 |
| 3.2.1. | Componentes | 42 |
| 3.2.2. | Circuito integrado transpondedor EM4100 | 42 |
| 3.2.3. | Protocolo del código de la tarjeta..... | 44 |

| | | |
|----------|--|----|
| 3.2.4. | Codificación Manchester | 46 |
| 3.3. | Módulo <i>Serial Ethernet Board</i> de <i>MikroElektronika</i> | 46 |
| 3.3.1. | Componentes y diagrama circuital | 47 |
| 3.3.2. | Controlador de red <i>Ethernet</i> ENC28J60..... | 49 |
| 3.4. | Módulo LCD 2x16..... | 51 |
| 3.4.1. | Componentes | 52 |
| 3.4.2. | Patillaje | 52 |
| 3.4.3. | Controlador Hitachi HD44780U..... | 53 |
| 3.4.4. | Señales de datos y control para desplegar texto..... | 55 |
| 4. | CONEXIÓN A BASE DE DATOS EN SERVIDOR LINUX | 57 |
| 4.1. | <i>Software</i> libre | 58 |
| 4.2. | Servidor <i>web</i> Apache..... | 59 |
| 4.3. | Servidor base de datos MySQL..... | 59 |
| 4.3.1. | Gestor gráfico de base de datos PhpMyAdmin..... | 60 |
| 4.4. | Lenguaje de programación web PHP | 60 |
| 4.4.1. | Aplicación en páginas <i>web</i> | 61 |
| 4.4.2. | Aplicación en conexión a base de datos MySQL..... | 62 |
| 4.4.3. | Aplicación de <i>sockets</i> de comunicación | 63 |
| 4.5. | Comunicación por LAN sobre <i>Ethernet</i> | 64 |
| 4.5.1. | Arquitectura Cliente – Servidor | 65 |
| 4.5.1.1. | Esquema..... | 65 |
| 4.5.1.2. | Interacción de servicios..... | 66 |
| 4.5.2. | Modelo de red OSI y modelo TCP/IP | 67 |
| 4.5.3. | Estándar <i>Ethernet</i> de la capa de acceso a la red..... | 72 |

| | | |
|----------|---|----|
| 4.5.4. | Protocolo IP de capa de red | 74 |
| 4.5.5. | Protocolos de la capa de transporte | 76 |
| 4.5.5.1. | Protocolo TCP | 77 |
| 4.5.5.2. | Protocolo UDP | 78 |
| 4.5.6. | Funciones de la capa de aplicación | 78 |
| 5. | DISEÑO Y DESARROLLO DEL SIDERF | 79 |
| 5.1. | Esquema de bloques del SIDERF | 79 |
| 5.1.1. | Capacidades del SIDERF | 80 |
| 5.2. | Diagrama de conexiones del microcontrolador PIC18F4520 | 81 |
| 5.2.1. | Circuito de alimentación | 82 |
| 5.2.2. | Conexión de los módulos | 82 |
| 5.2.3. | Otros componentes | 84 |
| 5.2.4. | Diseño del circuito impreso en placa | 85 |
| 5.3. | Programa del microcontrolador PIC18F4520 | 86 |
| 5.3.1. | Diagrama de flujo | 86 |
| 5.3.2. | Algoritmo del código del programa | 88 |
| 5.3.3. | Descripción de comandos y librerías | 89 |
| 5.3.4. | Parámetros de compilación en <i>MikroBasic</i> | 91 |
| 5.3.5. | Grabación del programa | 91 |
| 5.4. | Configuración de la computadora | 92 |
| 5.4.1. | Requerimientos | 92 |
| 5.4.2. | Instalación del <i>software</i> | 93 |
| 5.5. | Conexión de la red microcontrolador – computadora | 95 |
| 5.5.1. | Configuración de la red | 95 |
| 5.5.2. | Procedimiento de intercambio de datos | 96 |
| 5.5.3. | <i>Script</i> de comunicación de la computadora | 96 |
| 5.5.4. | Descripción de instrucciones | 98 |

| | | |
|----------|--|-----|
| 5.6. | Creación de la base de datos en MySQL | 100 |
| 5.6.1. | Tablas y campos..... | 100 |
| 5.6.1.1. | Tabla Estudiantes | 101 |
| 5.6.1.2. | Tabla IngresoRF | 102 |
| 5.7. | Interfaz de administrador en página <i>web</i> | 103 |
| 5.7.1. | Páginas que componen la interfaz | 105 |
| 5.7.1.1. | Página inicial..... | 105 |
| 5.7.1.2. | Página principal..... | 106 |
| 5.7.1.3. | Página de registrar usuarios..... | 107 |
| 5.7.1.4. | Página para modificar datos..... | 108 |
| 5.7.1.5. | Página de tablas y datos | 110 |
| 5.7.1.6. | Página de último ingreso | 111 |
| 5.7.1.7. | Página de búsqueda de carné..... | 112 |
| 5.7.1.8. | Página de búsqueda de tarjeta..... | 113 |
| 5.7.2. | Autenticación y seguridad | 114 |
| 5.7.2.1. | <i>Script</i> de autenticación de sesión | 114 |
| 5.7.2.2. | <i>Script</i> de verificación de sesión | 115 |
| 5.7.2.3. | <i>Script</i> de cierre de sesión..... | 115 |
| 5.7.3. | Consulta a base de datos MySQL..... | 115 |
| 5.7.3.1. | <i>Script</i> de insertar datos | 116 |
| 5.7.3.2. | <i>Script</i> de modificar datos | 116 |
| 5.7.3.3. | <i>Script</i> de eliminar datos..... | 116 |
| 5.8. | Funcionamiento final del SIDERF | 116 |
| 5.8.1. | Inicialización y apagado del sistema | 117 |
| 5.8.2. | Procedimiento de identificación por RF..... | 118 |
| 5.8.3. | Operaciones del administrador <i>web</i> | 119 |
| 5.9. | Inversión económica..... | 122 |
| 5.9.1. | Cotización del sistema propuesto | 122 |
| 5.9.2. | Sistemas disponibles en el mercado | 124 |

CONCLUSIONES 125
RECOMENDACIONES 127
BIBLIOGRAFÍA 129

ÍNDICE DE ILUSTRACIONES

FIGURAS

| | | |
|-----|---|----|
| 1. | Forma de onda sinusoidal en el tiempo..... | 3 |
| 2. | Componentes de un sistema RFID..... | 7 |
| 3. | Categorías y familias de los PIC..... | 15 |
| 4. | Estructura de la arquitectura Harvard..... | 16 |
| 5. | Proceso de segmentación de instrucciones..... | 18 |
| 6. | Conexión SPI..... | 25 |
| 7. | Diagrama de pines del PIC18F4520..... | 28 |
| 8. | Proceso de programación de un microcontrolador..... | 29 |
| 9. | Módulo RFid Reader..... | 36 |
| 10. | Diagrama circuital del módulo RFid Reader..... | 37 |
| 11. | Ejemplo de señal electromagnética en la operación de lectura..... | 40 |
| 12. | Señales de la operación de lectura..... | 41 |
| 13. | RFid cards de 125 KHz..... | 42 |
| 14. | Representación de la tasa de transmisión de <i>bits</i> | 44 |
| 15. | Estructura del código del transpondedor EM4100..... | 45 |
| 16. | Ejemplo de codificación Manchester..... | 46 |
| 17. | Módulo Serial Ethernet Board..... | 47 |
| 18. | Diagrama circuital del módulo Serial Ethernet Board..... | 48 |
| 19. | Módulo LCD 2x16..... | 52 |
| 20. | Flujo de señales de escritura en pantalla LCD..... | 56 |
| 21. | Estructura cliente-servidor..... | 65 |
| 22. | Servicio entre el lector y la computadora..... | 67 |
| 23. | Servicios web y base de datos..... | 67 |

| | | |
|-----|--|-----|
| 24. | Comunicación según el modelo OSI..... | 70 |
| 25. | Comparación de modelos de red..... | 71 |
| 26. | Conector RJ45 en conexión cruzada..... | 73 |
| 27. | Diagrama de bloques del SIDERF..... | 79 |
| 28. | Diagrama circuital de la placa del PIC..... | 85 |
| 29. | Diagrama del circuito impreso de la placa del PIC..... | 86 |
| 30. | Diagrama de flujo del programa del PIC..... | 87 |
| 31. | Secuencia de intercambio de datos..... | 97 |
| 32. | Estructura de base de datos..... | 103 |
| 33. | Mapa del sitio de administración web..... | 104 |
| 34. | Página index.php..... | 106 |
| 35. | Página rfsesion.php..... | 107 |
| 36. | Página veringresodatos.php..... | 108 |
| 37. | Página vermodificardatos.php..... | 109 |
| 38. | Página mostrardatos.php..... | 110 |
| 39. | Página ultimoingreso.php..... | 111 |
| 40. | Página buscarcarnet.php..... | 112 |
| 41. | Página buscartarjeta.php..... | 113 |

TABLAS

| | | |
|-------|---|-----|
| I. | Espectro electromagnético..... | 4 |
| II. | Librerías utilizadas en el código del PIC..... | 34 |
| III. | Conexión PIC y RFid Reader..... | 83 |
| IV. | Conexión PIC y Serial Ethernet Board..... | 83 |
| V. | Conexión PIC y el módulo LCD 2x16..... | 83 |
| VI. | Listado de componentes electrónicos del circuito del PIC..... | 122 |
| VII. | Listado de componentes electrónicos del circuito de LCD..... | 123 |
| VIII. | Listado de módulos periféricos adicionales..... | 123 |

LISTA DE SÍMBOLOS

| Símbolo | Significado |
|----------------|-----------------------------|
| GB | <i>Gigabyte</i> |
| GHz | <i>Gigahertz</i> |
| Hz | <i>Hertz</i> |
| KHz | <i>Kilohertz</i> |
| Mbps | <i>Megabits por segundo</i> |
| MB | <i>Megabyte</i> |
| MHz | <i>Megahertz</i> |

GLOSARIO

| | |
|---------------------------|--|
| AM | Amplitud modulada. |
| Antena | Dispositivo de los aparatos emisores o receptores que, con formas muy diversas, sirve para emitir o recibir ondas electromagnéticas. |
| Apache | Servidor web HTTP de distribución libre y de código abierto. |
| ARP | Protocolo de resolución de direcciones. |
| ASCII | Código estándar estadounidense para el intercambio de información. |
| <i>Basic</i> | Familia de lenguajes de programación de alto nivel. |
| <i>Bit</i> | Unidad básica de información equivalente a uno de entre dos estados lógicos igualmente probables. |
| <i>Byte</i> | Conjunto de información formado por 8 <i>bits</i> . |
| Circuito integrado | Pastilla pequeña de material semiconductor sobre la que se fabrican gran cantidad de circuitos electrónicos. |

| | |
|--------------------------------|---|
| Cliente | Computadora, o proceso dentro de ella, que solicita servicios a un servidor, en una red informática. |
| CMOS | Metal óxido semiconductor complementario. |
| Codificación | Proceso por el cual la información de una fuente es convertida en otra secuencia de información, agregando datos de control de errores, para ser comunicada. |
| Codificación Manchester | Método de codificación eléctrica de una señal binaria en el que en cada tiempo de <i>bit</i> hay una transición entre dos niveles de señal. |
| Compilador | Programa que convierte el lenguaje informático de alto nivel empleado por el usuario en lenguaje binario. |
| Cotejamiento | Tipo de codificación de idioma usado en los datos almacenados en la base de datos. |
| CSMA/CD | Detección de portadora con acceso múltiple y detección de colisiones. |
| Datagrama | Conjunto de datos independiente que transporta información suficiente para ser encaminado desde una computadora de origen hacia otra computadora de destino, sin tener que depender de que se haya producido anteriormente tráfico alguno entre ambos y la red de transporte. |

| | |
|------------------------|---|
| Demodulación | Extraer la información contenida en una señal modulada. |
| DIP | Encapsulado en línea dual. |
| EEPROM | Memoria de solo lectura, programable y borrable eléctricamente. |
| <i>Ethernet</i> | Estándar de la capa de acceso a la red más popular para la tecnología LAN, definida por la IEEE 802,3. |
| <i>Flash</i> | Tecnología de almacenamiento, derivada de la memoria EEPROM, que permite la lectura y escritura de múltiples posiciones de memoria en la misma operación. |
| <i>Host</i> | Dispositivo conectado a una red informática. |
| HTML | Lenguaje de marcado de hipertexto. |
| HTTP | Protocolo de transferencia de hipertexto. |
| IEEE | Instituto de ingenieros eléctricos y electrónicos. |
| Interfaz | Conexión física y funcional entre dos aparatos o sistemas independientes. |
| IP | Protocolo de Internet. |

| | |
|-------------------------|---|
| ISO | Organización internacional para la normalización. |
| JPEG | Grupo conjunto de expertos en fotografía. |
| LAN | Red de área local. |
| LCD | Pantalla de cristal líquido. |
| LED | Diodo emisor de luz. |
| LSB | <i>Bit</i> menos significativo. |
| MAC | Control de acceso al medio. |
| Microcontrolador | Circuito integrado, de alta escala de integración, que incorpora la mayor parte de los elementos que configuran un controlador. |
| Microchip | Circuito integrado, en miniatura, que realiza numerosas funciones en dispositivos electrónicos. |
| Modulación | Acción de variar el valor de la amplitud, frecuencia o fase de una onda portadora, de alta frecuencia, en función de una señal de información de baja frecuencia. |
| MSB | <i>Bit</i> mas significativo. |
| MSSP | Puerto serial síncrono maestro. |

| | |
|------------------------|--|
| MySQL | Servidor y sistema de gestión de bases de datos estructurales y relacionales, de código abierto. |
| <i>Nibble</i> | Conjunto de información formado por 4 <i>bits</i> . |
| NIC | Tarjeta de interfaz de red. |
| OSI | Interconexión de sistemas abiertos. |
| PHP | Preprocesador de hipertexto. |
| PhpMyAdmin | Programa escrito en PHP con la intención de manejar la administración de MySQL a través de páginas web. |
| PIC | Controlador de interfaz periférico. |
| PLL | Lazo de seguimiento de fase. |
| Protocolo | Conjunto de normas y procedimientos útiles para la transmisión de datos, conocido por el emisor y el receptor. |
| Radiofrecuencia | Cada una de las frecuencias de las ondas electromagnéticas, entre 3 KHz y 300 GHz. |
| RAM | Memoria de acceso aleatorio. |
| RFID | Identificación por radiofrecuencia. |

| | |
|-----------------------|---|
| RISC | Computadora con juego de instrucciones reducido. |
| ROM | Memoria de solo lectura. |
| <i>Script</i> | Conjunto de instrucciones, generalmente almacenadas en un archivo de texto, que deben ser interpretados línea a línea en tiempo real para su ejecución. |
| Servidor | Computadora, o proceso dentro de ella, que ofrece servicios a múltiples usuarios clientes, en una red informática. |
| SIDERF | Sistema de identificación estudiantil por radiofrecuencia. |
| Sincronización | Hacer que coincidan en el tiempo la señal de información y el mecanismo de muestreo de dicha señal. |
| <i>Socket</i> | Aplicación de software por el cual dos programas pueden intercambiar datos por red en el modelo TCP/IP. |
| SPI | Interfaz serial periférica. |
| SQL | Lenguaje de consulta estructurado. |
| TCP | Protocolo de control de transmisión. |

| | |
|----------------------|--|
| Transceptor | Dispositivo transmisor y receptor de señales para estaciones de radiofrecuencia. |
| Transpondedor | Dispositivo transmisor/respondedor, que transmite una señal de radiofrecuencia en respuesta ante la excitación externa de otra señal de radiofrecuencia. |
| Ubuntu | Distribución libre de Linux que ofrece un sistema operativo enfocado a computadoras de escritorio. |
| UDP | Protocolo de datagrama de usuario. |
| Web | Red informática. |

RESUMEN

El presente trabajo de graduación tiene la principal finalidad de proporcionar el diseño de un sistema de identificación estudiantil, el cual facilite el control de acceso de estudiantes a las instalaciones del Laboratorio de Electrónica de la Facultad de Ingeniería.

El sistema propuesto se basa en la identificación del estudiante, quien desea ingresar al laboratorio, por medio de la lectura de una tarjeta de radiofrecuencia de carácter personal, con la cual se pueda verificar, en una base de datos electrónica, la información de dicho estudiante, y así generar un registro de la asistencia al laboratorio.

El primer capítulo describe los aspectos generales de un sistema de identificación por radiofrecuencia. El segundo y tercer capítulo describen las características del equipo que conforman a la estación lectora. El cuarto capítulo describe la configuración y programación de la computadora que presta el servicio de base de datos.

El capítulo final presenta el desarrollo del sistema de identificación utilizando el equipo aportado por el Laboratorio de Electrónica. Por una parte, se desarrolla la conexión y configuración de una estación lectora de tarjetas de radiofrecuencia, junto con el programa que ejecuta las operaciones de lectura y transmisión de datos. Por otra parte se desarrolla, en una computadora, la comunicación por red con la estación lectora, y una interfaz *web* para la administración de las bases de datos asociadas a la información de los estudiantes y de los registros de asistencia.

OBJETIVOS

General

Diseñar y desarrollar un sistema de identificación estudiantil por medio de tecnología de radiofrecuencia, para utilizarse en el Laboratorio de Electrónica, con fin de mejorar el control de acceso a las instalaciones.

Específicos

1. Diseñar la estación lectora de las tarjetas de radiofrecuencia, utilizando los módulos proporcionados por el Laboratorio de Electrónica.
2. Diseñar el código de programación del microcontrolador que gobierne las operaciones de la estación lectora.
3. Diseñar el proceso de comunicación e intercambio de información entre la estación lectora y la computadora, dentro de una red.
4. Diseñar la configuración necesaria del software en la computadora para el sistema de identificación y registro.
5. Diseñar la configuración básica de las bases de datos que contenga la información de los estudiantes y que lleven el registro de asistencia al laboratorio.
6. Diseñar la interfaz web de administración de las base de datos.

INTRODUCCIÓN

En la actualidad el Laboratorio de Electrónica, perteneciente a la Escuela de Ingeniería Mecánica Eléctrica, de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, ha tomado la decisión, y presentado la propuesta, de modernizarse en cuanto a infraestructura, instalaciones y equipo para satisfacer las necesidades de aprendizaje de los estudiantes que acuden al laboratorio, y desarrollo de tecnología para el país; demandas que han ido creciendo en el transcurso de los años y que merecen ser atendidas urgentemente con innovación y calidad.

Entre los planes de modernización del Laboratorio de Electrónica se tiene el diseño de un sistema de identificación estudiantil por radiofrecuencia, que permita llevar un mejor control de los estudiantes que ingresan y trabajan en proyectos dentro de las instalaciones del laboratorio.

El presente trabajo tiene la misión de proponer una solución ante la necesidad del Laboratorio de Electrónica, con el diseño de un sistema de identificación por radiofrecuencia. Dicho diseño plantea todo el conjunto de componentes físicos, circuitos y módulos, así como de programación, conectividad y comunicación, y el manejo de bases de datos.

Por su parte, el Laboratorio de Electrónica aporta el equipo y componentes necesarios para conformar el sistema, los cuales son un módulo de lectura por radiofrecuencia y tarjetas, un módulo de comunicación *Ethernet*, una pantalla LCD, un microcontrolador, y una computadora. Con la disposición de este equipo, el diseño fue guiado y orientado a utilizar y optimizar dichos recursos.

1. ASPECTOS GENERALES DE UN SISTEMA RFID

El presente capítulo describe los aspectos generales de un sistema de RFID, siglas en inglés para *Radio Frequency Identification*, en español se refiere a identificación por radiofrecuencia, que es una tecnología que permite transmitir un código único de identidad de un objeto mediante ondas de radio con el fin de identificar dicho objeto.

1.1. Introducción

Un sistema RFID es un sistema que engloba el proceso de recuperación de datos a través de una estación lectora de radiofrecuencia, compuesto por una antena que se comunicará con una serie de dispositivos denominados etiquetas o transpondedores RFID que contienen el código único de identificación, con el fin de utilizar la información recuperada en procedimientos de control. La tecnología RFID tiene varias ventajas como la rapidez de lectura, la capacidad de almacenamiento, tamaño, seguridad, así como su utilización a distancia por medio de ondas de radio frecuencia.

La tecnología RFID tiene su origen en la II Guerra Mundial, época en la que el uso del radar permitía la detección de aviones a kilómetros de distancia, y su objetivo era discriminar rápidamente entre la propia flota de aviones y los escuadrones enemigos.

Los sistemas de radar y de comunicaciones por radiofrecuencia avanzaron en las décadas de los años 50 y 60 en las que los científicos de los países más avanzados trabajaban para explicar cómo identificar objetos remotamente.

Las primeras patentes para dispositivos RFID fueron solicitadas en los Estados Unidos, concretamente en enero de 1973 cuando Mario W. Cardullo se presentó con una etiqueta RFID activa que contenía una memoria reescribible. El mismo año, Charles Walton recibió la patente para un sistema RFID pasivo que abría las puertas sin necesidad de llaves, una tarjeta con una etiqueta RFID que comunicaba una señal al lector de la puerta, que cuando validaba la tarjeta, desbloqueaba la cerradura.

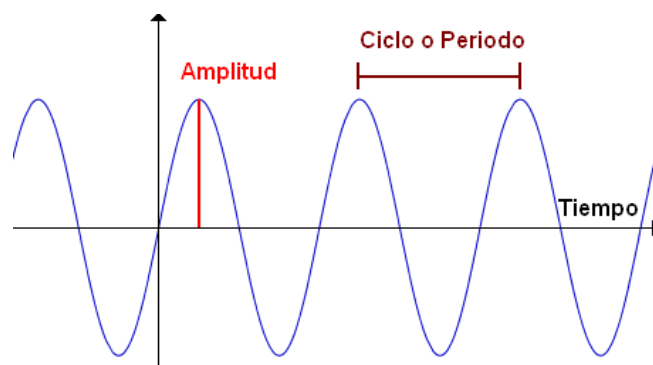
La tecnología RFID fue avanzando su desarrollo hasta que en 1999 se nació la principal responsable del desarrollo e implantación de esta tecnología, la entidad *Auto-ID Center*, una sociedad constituida por un centenar de empresas de vanguardia, universidades y centros de investigación.

1.2. Principio de funcionamiento

La tecnología RFID utiliza la comunicación de información por radiofrecuencia, que físicamente se basa en la radiación de ondas electromagnéticas, portadoras de información entre el lector y una etiqueta a través de antenas incorporadas en cada uno. Dichas ondas electromagnéticas poseen ciertas características, especialmente la de frecuencia define que se le llamen ondas de radiofrecuencia.

El concepto de radiación de ondas de radio se asocia a la perturbación eléctrico-magnética en un espacio físico, moviéndose de forma oscilatoria a través del espacio y oscilando también en el tiempo. La forma básica de una onda generada por un transmisor es la de una onda sinusoidal. Pero una onda que se radia al espacio puede conservar o no las características de una onda sinusoidal, dependiendo del tipo de modulación que se aplique.

Figura 1. **Forma de onda sinusoidal en el tiempo**



Fuente: elaboración propia.

La primera característica de una onda sinusoidal es la amplitud, como se observa en la figura 1 es el valor máximo que dista desde el valor medio de la onda. Otras características surgen cuando la onda sinusoidal se repite con la misma forma y amplitud. El ciclo de la onda es el intervalo fundamental que se repite indefinidamente y con ello se define también el período de la onda como el valor de tiempo en que se realiza un ciclo completo.

La frecuencia de una onda se define como el número de ciclos de una onda sinusoidal que se completa en un segundo. La unidad de medida de la frecuencia es el *Hertz* (Hz), que se refiere a ciclos por segundo.

1.2.1. Frecuencias de radio

Las frecuencias entre 3 KHz y 300 GHz se denominan radiofrecuencias porque usualmente se utilizan en comunicaciones de radio. La tabla 1 muestra la distribución, clasificación y aplicaciones de las ondas electromagnéticas de acuerdo a su frecuencia. Esta distribución es llamada espectro electromagnético.

1.2.2. Modulación

Para utilizar las ondas electromagnéticas como medio para enviar información, debe adecuarse la frecuencia de transmisión de dichas ondas. Para ello la señal de corriente con que se alimenta a la antena transmisora debe ser modulada con la información.

La modulación se define como el proceso de superponer una señal de información de baja frecuencia sobre una señal portadora de frecuencia más alta. La modulación en amplitud es utilizada en los sistemas RFID, y consiste en variar la amplitud de la señal portadora proporcionalmente a las variaciones de amplitud de la señal de información.

Tabla I. **Espectro electromagnético**

| Frecuencia | Designación | Aplicaciones |
|-------------------|-----------------------------|--|
| 3 Hz – 30 KHz | Frecuencia muy baja (VLF) | Audio, telefonía, radionavegación |
| 30 KHz – 300 KHz | Frecuencia baja (LF) | Radioayudas, radiofaros, transmisión por portadora |
| 300 KHz – 3 MHz | Frecuencia media (MF) | Radiodifusión comercial, defensa civil, radioaficionados |
| 3 MHz – 30 MHz | Frecuencia alta (HF) | Radioaficionados, radiotelefonía móvil, comunicaciones militares |
| 30 MHz – 300 MHz | Frecuencia muy alta (VHF) | Televisión VHF, radio FM, control de tránsito aéreo, policía |
| 300 MHz – 3 GHz | Frecuencia ultra alta (UHF) | Televisión UHF, telemetría espacial, comunicaciones militares, banda ciudadana |

Continuación de la tabla I.

| | | |
|------------------|-----------------------------|--|
| 3 GHz – 30 GHz | Frecuencia super alta (SHF) | Radar, comunicaciones por satélite, radioenlaces de microondas |
| 30 GHz – 300 GHz | Frecuencia extra alta (EHF) | Radioastronomía, comunicaciones ópticas |

Fuente: BRICEÑO, José. Principio de las comunicaciones. p. 567

Diferentes sistemas de comunicación utilizan diferentes rangos de frecuencia para transmitir, como se indica en la tabla I. También cada aplicación utiliza distintos tipos de modulación de la información, adaptándola para obtener mayor confiabilidad y más efectivo intercambio de información.

1.2.3. Bandas de frecuencia para sistemas RFID

Existen actualmente diversos sistemas de RFID operando en distintas frecuencias. Las bandas establecidas para la tecnología RFID son:

- 125 KHz, operando en la banda de LF (*Low Frequency*), su velocidad de comunicación es baja, lo que lo hace deficiente para operar en entornos donde haya más de una etiqueta presente en el campo de la antena. Su rango máximo de lectura no supera los 50 centímetros y su aplicación más común es en controles de accesos e identificación de animales.
- 13,56 MHz, operando en la banda de HF (*High Frequency*), la velocidad de comunicación es aceptable para sistemas estáticos o de baja velocidad, su rango máximo de lectura es alrededor de un metro, sus aplicaciones son en librerías e identificación de contenedores.

- 868 - 928 MHz, opera en la banda de UHF (*Ultra High Frequency*), sus principales inconvenientes se encuentran en la interferencia provocada por metales y líquidos. Otro punto negativo es la dificultad de estandarizar la frecuencia, dado que cada país legisla esta banda con distintas limitaciones. Entre sus puntos positivos está el rango de lectura (que alcanza hasta 9 metros), su velocidad de lectura (1 200 etiquetas por segundo). Sus principales aplicaciones se encuentran en la cadena de abastecimientos, peajes e identificación de equipajes.
- 2,4 - 5,8 GHz, trabaja en la banda de UHF, si bien su velocidad de transmisión es buena, su rango de lectura no es mayor a 2 metros. Este tipo de sistemas no se encuentran muy difundidos y su aplicación principal se encuentra en sistemas de peaje.

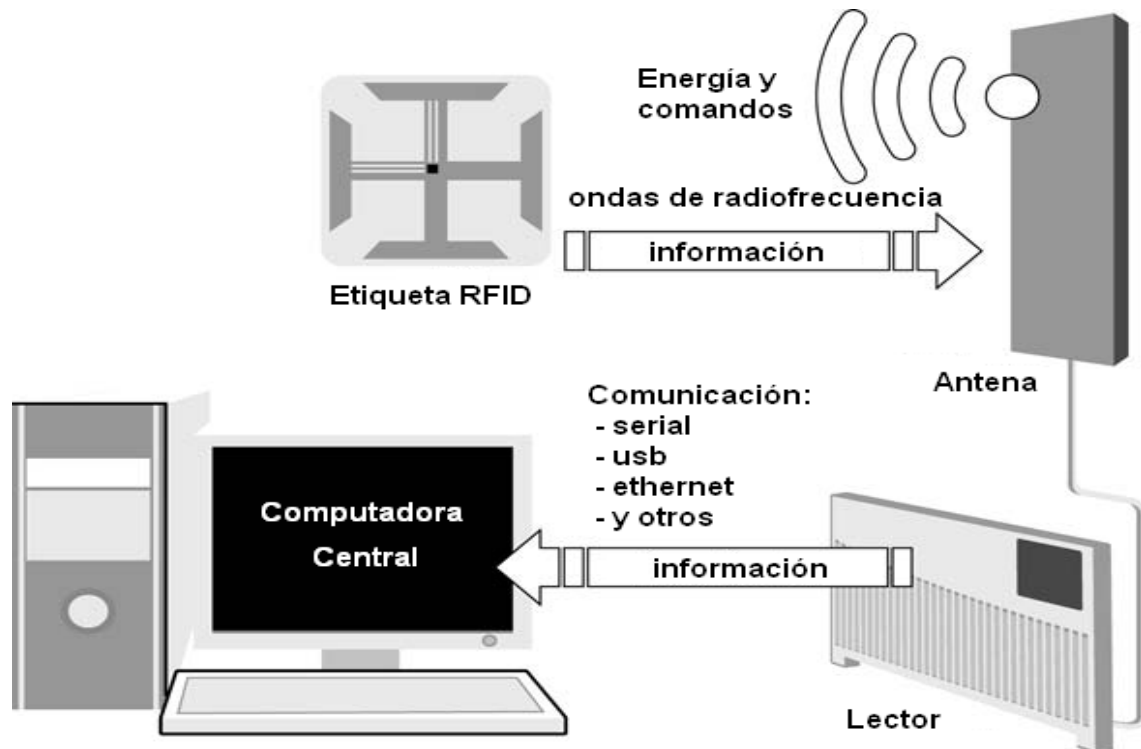
1.3. Componentes de un sistema RFID

Un sistema típico de RFID está constituido por cuatro componentes principales:

- Lector de etiquetas de radiofrecuencia
- Etiquetas de radiofrecuencia
- Antenas
- Computadora o servidor

La figura 2 muestra a los componentes anteriores interactuando para formar un sistema de identificación por radiofrecuencia. La computadora intercambia información con el lector a través de una interfaz de comunicación. A su vez, el lector gobierna el funcionamiento de la antena, detectando la presencia de una etiqueta de RF y decodificando el número de identificación.

Figura 2. **Componentes de un sistema RFID**



Fuente: GIDEKEL, Alan. Introducción a la identificación por radio frecuencia. p. 7

1.3.1. **Lector de etiquetas de radiofrecuencia**

El lector es utilizado para leer la información contenida en la etiqueta de radiofrecuencia y posiblemente escribir la información siempre que la etiqueta posea la capacidad de ser reescrita.

El lector consta de una antena que utiliza para enviar información digital codificada a través de ondas de radiofrecuencia con una modulación determinada. Un circuito receptor en la etiqueta es capaz de detectar el campo modulado, decodificar la información y usar su propia antena para enviar una señal más débil a modo de respuesta.

Un lector puede configurarse para operar en forma constante realizando lecturas fijas y acumulando listas de etiquetas en su memoria. Una computadora hará una petición al lector para que le sea enviada la información adquirida en la lectura. También el lector puede configurarse para operar de forma interactiva con la computadora a la que se conecta, esto es, realizando la acción que la computadora le indique. Una vez completado el comando instruido por la computadora, el lector espera hasta recibir la siguiente instrucción.

El lector debe contar con un microprocesador que controle las actividades de lectura de etiquetas, codificación y decodificación de la información, así como el control del módulo de comunicación con la computadora para el envío de datos. Una memoria para el almacenamiento de datos. Una antena para la transmisión y recepción de información con las etiquetas RFID. Un módulo de comunicación con un protocolo que sea reconocido por la computadora, que puede ser serial RS-323, USB, *Ethernet*, entre otros.

1.3.2. Etiquetas de radiofrecuencia

Una etiqueta RFID está compuesta por un microchip y una antena flexible instalada sobre una superficie plástica. El microchip contiene grabada la información electrónica en un código para su transmisión y lectura.

Una forma de clasificar etiquetas es por su habilidad para almacenar datos. Mientras unos microchips permiten sólo la lectura, otros admiten la lectura y escritura de información. Una etiqueta de sólo lectura está programada de fábrica, y la información nunca puede ser cambiada. Por otro lado, una etiqueta reescribible tiene la capacidad de borrar y cambiar la información guardada en el microchip, usando un lector o impresora RFID.

Otra forma de clasificar las etiquetas de radiofrecuencia es por la fuente de energía que utilice dicha etiqueta, ésta puede ser denominada como etiqueta activa o etiqueta pasiva.

1.3.2.1. Etiquetas activas

Las etiquetas activas poseen su propia fuente de poder. Una batería incorporada a bordo energiza el circuito del microchip y el transmisor. Las etiquetas activas son capaces de recibir y transmitir señales en largas distancias. Son ideales para aplicaciones donde pueden ser instaladas y mantenidas en forma permanente.

1.3.2.2. Etiquetas pasivas

Por otro lado, las etiquetas pasivas no poseen batería. Estas etiquetas obtienen la energía desde el lector para funcionar. Las ondas electromagnéticas transmitidas desde el lector inducen una corriente en la antena de la etiqueta. La etiqueta utiliza esa energía para responder al lector. Las etiquetas pasivas son menores en tamaño, más livianas en peso, tienen una vida útil más prolongada y están sujetas a menos regulaciones que las etiquetas activas.

1.3.3. Antenas

El intercambio de datos entre las etiquetas y los lectores se realiza por medio de las antenas que generan las ondas electromagnéticas. Existen diferentes diseños de antenas optimizadas para varias aplicaciones. Las antenas pueden ser fabricadas de aluminio, cobre u otros materiales conductores.

La cantidad de material conductivo utilizado y el tamaño de la antena determinan la sensibilidad de una etiqueta. La sensibilidad es crucial para obtener buenos rangos de lectura y minimizar la influencia de los materiales a los que son aplicadas las etiquetas.

A pesar de que los chips son pequeños, las antenas no lo son, éstas necesitan ser lo suficientemente grandes como para captar la señal emitida por el lector. La antena permite que una etiqueta pueda leerse a una distancia de 3 metros o más, incluso a través de distintos materiales. El tamaño de la antena tiende a determinar el tamaño de una etiqueta RFID.

1.3.4. Computadora

La computadora se encargará de utilizar la información proveniente de las etiquetas de radiofrecuencia, para procedimientos de control como registros de productos, animales, personas, en una base de datos, así como validaciones de identidad de personas, transacciones, entre otros.

Existen varias tecnologías de comunicación soportadas por una computadora. Entre las más comunes y actuales están: la comunicación serial RS-232, USB, red *Ethernet*, red *Wifi*, entre otras.

Últimamente se ha adoptado que la computadora y el lector se comuniquen a través del modelo TCP/IP de comunicaciones, sobre una red *Ethernet*, formando una red llamada cliente-servidor. La forma de operar de una red de este tipo es bastante simple, se tiene una máquina cliente que requiere un servicio de una máquina servidor, y éste último realiza la función para la que está programado, enviando una respuesta o resultado a la máquina cliente.

En un sistema de red más amplio, cada máquina puede cumplir la función de servidor para algunas tareas y la función de cliente para otras. Incluso dentro de una misma máquina, hay procesos que son considerados clientes y otros considerados servidores. De esta forma la computadora podrá ejecutar procesos de cliente para realizar peticiones al lector, como también peticiones a procesos servidores dentro de la misma. Usualmente los procesos de servidor usados en el sistema RFID son validación de información y base de datos.

2. MICROCONTROLADOR

En el ámbito de la electrónica, el término controlador lo recibe el dispositivo, o conjunto de ellos, que se emplea para el gobierno de uno o varios procesos. Aunque este concepto de controlador ha permanecido invariable en el tiempo, su implementación física ha cambiado y disminuido en tamaño.

El controlador ha ido evolucionando su construcción, desde el uso exclusivo de componentes de lógica discreta hasta el empleo de microprocesadores conectados a chips de memoria y periféricos de comunicación. En la actualidad, todos los elementos del controlador se han podido incluir en un chip, el cual recibe el nombre de microcontrolador.

Entonces se puede definir al microcontrolador como un circuito integrado, de alta escala de integración, que incorpora la mayor parte de los elementos que configuran un controlador. Así, con su reducido tamaño es posible montar el microcontrolador en el propio dispositivo al que gobierna.

Actualmente un microcontrolador es construido para incorporar en su interior un procesador CPU, memoria ROM para instrucciones de programa, memoria RAM de datos, líneas de entrada y salida digitales, módulos para el control de periféricos, y un generador de pulsos de reloj.

Los microcontroladores están siendo empleados en multitud de sistemas presentes en nuestra vida diaria. Conteniendo básicamente la estructura de una computadora, aumentan las prestaciones de un producto reduciendo su tamaño y costo, además de mejorar la fiabilidad en su funcionamiento.

Muchas son las aplicaciones como pueden ser televisores, computadoras, impresoras, módems, juguetes, horno microondas, entre otros productos de consumo. También son utilizados en aplicaciones más complejas como instrumentación electrónica, control de sistemas en un automóvil, etc.

2.1. Introducción al microcontrolador PIC

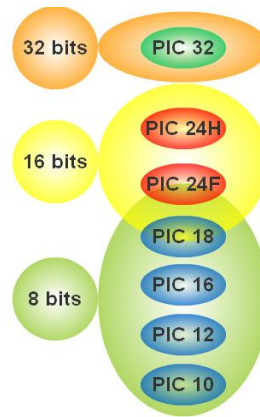
Con el avance de la tecnología de manufactura, muchas empresas como Intel, Motorola, Toshiba, Microchip, y otras, han optado por fabricar microcontroladores en forma masiva para abastecer la creciente demanda. De particular mención se tiene el microcontrolador PIC, fabricado por la empresa *Microchip Technology Inc.*

El microcontrolador PIC (*Peripheral Interface Controller*), en español se traduce como controlador de interfaz periférico, es un circuito integrado digital monolítico que posee la capacidad de ser programable con un código que hace uso de sus recursos para gobernar un proceso. Este microcontrolador presenta muchas razones para ser escogido en el desarrollo de proyectos, como el bajo numero de instrucciones de programación, amplia documentación, bajo precio, y muchas herramientas de *software* y *hardware* para la programación y prueba.

2.1.1. Familias del microcontrolador PIC

La clasificación más importante entre los microcontroladores PIC es la capacidad de los registros del procesador interno, esta capacidad puede ser de 8, 16 o 32 *bits*. Dentro de cada categoría también se subdivide en familias con sus características y recursos distintivos entre ellos.

Figura 3. **Categorías y familias de los PIC**



Fuente: elaboración propia.

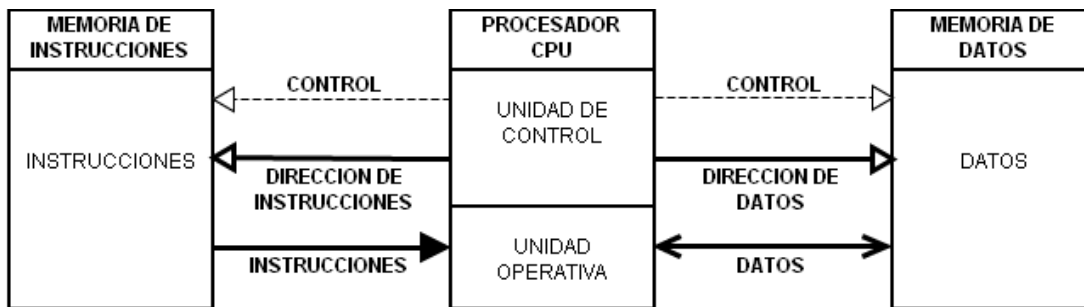
Dentro de la categoría de 8 *bits*, que es la más popular, *Microchip* ha dividido a sus microcontroladores en tres gamas particulares, tomando como parámetro de clasificación el tamaño de las instrucciones programación del procesador CPU. Esta clasificación es:

- Gama baja. En esta gama se encuentran todos los PIC que cuentan con instrucciones de 12 *bits*. Éstos cuentan hasta con 33 instrucciones. Dentro de esta gama se encuentran los PIC12CXX y PIC16CX5.
- Gama media. En esta gama están todos los PIC de instrucciones de 14 *bits*. Éstos cuentan con un juego de 35 instrucciones o más. Dentro de esta gama se encuentran los PIC16CXXX y los PIC16FXXX.
- Gama alta. En esta gama están los PIC que poseen instrucciones de 16 *bits*. Éstos poseen una mayor capacidad de memoria que cualquier otra gama, además de un juego de 60 o más instrucciones. Dentro de esta gama se encuentran los PIC17CXX, los PIC18CXXX y los PIC18FXXX.

2.2. Arquitectura

Los microcontroladores PIC son diseñados de acuerdo a la arquitectura *Harvard*. Dicha arquitectura presenta dos secciones de memoria independientes dentro del microcontrolador. Una sección de memoria contiene sólo instrucciones del programa a ejecutar. La otra sección es utilizada para almacenar datos usados en el proceso. Ambas disponen de sus respectivos sistemas de buses de acceso y es posible realizar operaciones de lectura o escritura, simultáneamente en ambas memorias.

Figura 4. **Estructura de la arquitectura *Harvard***



Fuente: elaboración propia.

Por otro lado, la estructura interna del PIC está orientada al esquema de bancos de registros. Esto quiere decir que todos los objetos dentro del microcontrolador, como los puertos de entrada/salida, comparadores, unidad de conversión analógico/digital, temporizadores, los sectores de memoria y otros, están implementados físicamente como registros.

A cada registro dentro del microcontrolador se le asigna una dirección dentro del mapa de los bancos de registros. Es más, los registros con funciones especiales se les otorgan un nombre específico con el que se identifica su

función. La cantidad de bancos de registros depende de las características y funciones que incluya de fábrica el modelo de PIC que se adquiera.

Al final, con entender que toda unidad dentro del microcontrolador es catalogada como un registro, se facilita el acceso de lectura o escritura a cada unidad y sector de memoria, además que se logra una mejor configuración de la forma de operar del PIC al manipular directamente los *bits* de cada registro.

2.3. Procesador CPU

El procesador es la parte primordial de todo microcontrolador y determina sus principales características, tanto a nivel *hardware* como *software*. Se encarga de direccionar la memoria de instrucciones, decodificar la instrucción y ejecutar de la operación que implica la instrucción, así como también la búsqueda de origen de los datos a operar y el almacenamiento del resultado.

El procesador de un microcontrolador PIC se destaca por ser diseñado bajo la filosofía RISC (*Reduced Instruction Set Computer*), la cual se refiere a una computadora con juego de instrucciones reducido. Esta arquitectura del procesador presenta la capacidad de ejecutar un repertorio reducido de instrucciones, lo cual simplifica la programación del código y agiliza la ejecución de la instrucción en un solo ciclo de máquina.

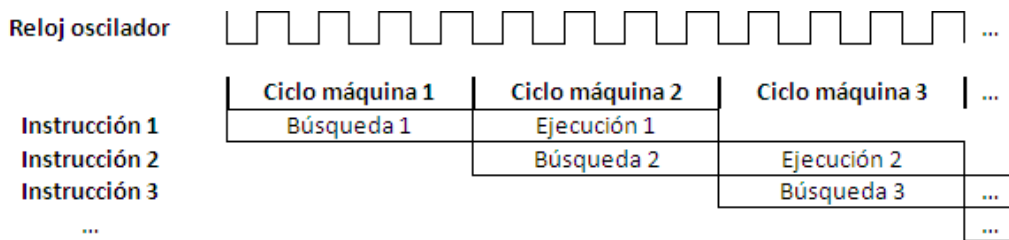
El formato de todas las instrucciones es de la misma longitud. Todas las instrucciones de los PIC de la gama baja tienen una longitud de 12 *bits* y disponen de un repertorio de 33 instrucciones. Las instrucciones de la gama media tienen 14 *bits* y un repertorio de 35 instrucciones. Las instrucciones de la gama alta tienen 16 *bits* y constan de un repertorio de 60 o más instrucciones dependiendo del modelo de PIC.

La rapidez y sencillez de este juego de instrucciones permiten optimizar el *hardware* y el *software* del procesador; es por ello que los microcontroladores PIC son tan versátiles y fáciles de programar independientemente del modelo.

Otro aspecto importante que cabe destacar del procesador de un PIC es el desempeño en la ejecución de cada instrucción. El procesador utiliza la técnica de segmentación en la ejecución de las instrucciones.

La segmentación permite al procesador realizar al mismo tiempo la ejecución de una instrucción y la búsqueda del código de la siguiente. De esta forma se puede ejecutar cada instrucción en un ciclo de máquina. Un ciclo de máquina se define como 4 ciclos del reloj oscilador.

Figura 5. **Proceso de segmentación de instrucciones**



Fuente: elaboración propia.

2.4. Configuraciones del oscilador

Todos los microcontroladores disponen de un circuito oscilador que genera una onda cuadrada de alta frecuencia, que configura los impulsos de reloj usados en la sincronización de todas las operaciones del sistema. Generalmente, el circuito de reloj está incorporado en el microcontrolador y sólo se necesitan unos pocos componentes exteriores para seleccionar y estabilizar

la frecuencia de trabajo. Dichos componentes suelen consistir en un cristal de cuarzo junto a elementos pasivos o bien un resonador cerámico.

Los microcontroladores PIC de la gama alta pueden ser operados en diez diferentes modos de oscilador, modificando los *bits* de configuración del registro del oscilador. Los modos de operación son los siguientes:

- LP – cristal de bajo poder
- XT – cristal o resonador
- HS – cristal o resonador de alta velocidad
- HSPLL – cristal o resonador de alta velocidad con PLL activado
- RC – red de resistor y capacitor
- RCIO – red externa de resistencia y capacitor, sin pulso de reloj de salida
- INTIO1 – oscilador interno, con pulso de reloj de salida en el pin RA6
- INTIO2 – oscilador interno, sin pulso de reloj de salida
- EC – señal de reloj externa de otro sistema
- ECIO – señal de reloj externa de otro sistema, sin pulso de reloj de salida

En los modos LP, XT, HS o HSPLL se utiliza un cristal oscilador o un resonador cerámico conectado a los pines de entrada del oscilador. En el modo LP se utiliza un cristal o resonador con frecuencia de 32 KHz. En el modo XT se utilizan frecuencias entre 1 MHz a 4 MHz. En el modo HS se utilizan frecuencias entre 4 MHz hasta 25 MHz.

Los modos de RC y RCIO requieren de un resistor y un capacitor para generar los pulsos de reloj con el proceso de carga y descarga del capacitor. La frecuencia del oscilador es determinada por la red de resistencia y capacitancia.

Algunos modelos de PIC contienen osciladores internos que no requieren de componentes externos. En los modos INTIO1 e INTIO2 se configura la utilización del oscilador interno. La fuente principal del oscilador interno genera un pulso de reloj de frecuencia de 8 MHz. El microcontrolador puede utilizar este valor de frecuencia directamente en sus operaciones, o puede escalar el pulso de reloj al rango de frecuencias entre 31 KHz y 4 MHz, dependiendo de la configuración especificada por el programador.

Los modos EC y ECIO permiten la conexión de una señal de reloj proveniente de un sistema externo. Estos modos son convenientes para la sincronización y operación conjunta con otros dispositivos ajenos al microcontrolador PIC.

2.4.1. Multiplicador de frecuencia con PLL

Un circuito PLL (*Phase Locked Loop*), que significa lazo de seguimiento de fase, es un circuito capaz de realizar la multiplicación de la frecuencia de los pulsos de reloj.

Algunos modelos de microcontrolador PIC integran un circuito PLL para elevar la frecuencia del oscilador utilizado. Aumentar la frecuencia de reloj supone disminuir el tiempo en que se ejecutan las instrucciones pero lleva aparejado un incremento del consumo de energía.

El circuito PLL únicamente puede ser activado cuando se configura el oscilador en el modo HSPLL, en el cual la frecuencia del cristal o resonador es aumentada por cuatro veces para la operación del PIC.

2.5. Memoria

En los microcontroladores la memoria de instrucciones y datos está integrada en el propio chip. Una parte debe ser no volátil, tipo ROM, y se destina a contener el programa de instrucciones que gobierna la aplicación. Otra parte de memoria será tipo RAM, volátil, y se destina a guardar las variables y los datos.

Los diseñadores con microcontroladores trabajan con capacidades de ROM comprendidas entre 512 *bytes* y 8 *kilobytes* y de RAM comprendidas entre 20 y 512 *bytes*. Como el microcontrolador sólo se destina a una tarea en especial, en la memoria ROM sólo hay que almacenar un único programa de trabajo.

2.5.1. Memoria de programa

Esta es la memoria utilizada para almacenar las instrucciones del programa a ejecutar en un microcontrolador PIC. Esta sección de memoria consta de memorias tipo *flash*, que se trata de una memoria no volátil, de bajo consumo, que se puede escribir y borrar. Funciona como una memoria ROM y una RAM pero consume menos energía y es más pequeña.

A diferencia de la ROM, la memoria *flash* es programable en el circuito. Las memorias *flash* son muy útiles al permitir que los microcontroladores que las incorporan puedan ser reprogramados en circuito, es decir, sin tener que sacar el chip de la tarjeta de trabajo.

2.5.2. Memoria RAM

Este tipo de memoria se considera volátil, esto significa que los datos se perderán cuando se desconecta la fuente de poder al microcontrolador. La RAM en estos dispositivos es de poca capacidad pues sólo debe contener las variables y los cambios de información que se produzcan en el transcurso del programa.

Por otra parte, como sólo existe un programa activo, no se requiere guardar una copia del mismo en la RAM pues se ejecuta directamente desde la ROM.

2.5.3. Memoria EEPROM

Se trata de memoria de sólo lectura, programable y borrable eléctricamente EEPROM (*Electrical Erasable Programmable Read Only Memory*). Tanto la programación como el borrado, se realizan eléctricamente desde el propio grabador y bajo el control programado de una computadora. Es muy cómoda y rápida la operación de grabado y la de borrado.

Los microcontroladores dotados de memoria EEPROM permiten almacenar datos que no se borran cuando se retire la fuente de alimentación. El número de veces que puede grabarse y borrarse una memoria EEPROM es finito, por lo que no es recomendable una reprogramación continua.

La memoria EEPROM en un microcontrolador PIC está separada de la memoria de programa y de la memoria RAM. La cantidad de esta memoria es limitada, y cada dirección permite el almacenamiento de un *byte*.

2.6. Puertos digitales de E/S

La principal utilidad de los pines que posee la cápsula que contiene un microcontrolador es soportar las líneas de entrada y salida que comunican al computador interno con los periféricos exteriores. Según los controladores de periféricos que posea cada modelo de microcontrolador, las líneas de E/S se destinan a proporcionar el soporte a las señales de entrada, salida y control.

Dependiendo del modelo seleccionado, el microcontrolador PIC puede contener hasta 5 puertos digitales disponibles. Cada puerto tiene longitud de un *byte*, es decir, 8 pines físicos que representan los 8 *bits*. Las líneas de cada puerto son bidireccionales, lo que permite ser configurados como líneas de entrada o bien líneas de salida.

Cada puerto utiliza tres registros para su configuración y operación. El registro TRIS para configurar la dirección de las líneas, indicando si son entradas o salidas. El registro PORT para leer los estados de las líneas de entrada durante la ejecución del programa. El registro LAT que permite la escritura de señales en las líneas de salida.

Los puertos son identificados por letras del abecedario iniciando con la letra A, y así hasta la cantidad de puertos existentes. Los registros relacionados a los puertos también se identifican, en su última letra, por la letra del puerto.

Algunos pines de los puertos cumplen con más de una función y deben ser configurados en el código del programa, para su operación en uno u otro modo.

2.6.1. Puertos de comunicación

Estos puertos permiten al microcontrolador la posibilidad de comunicarse con otros dispositivos externos, otros buses de microprocesadores, buses de sistemas, buses de redes y con otros elementos bajo otras normas y protocolos para el control, ejecución de acciones e intercambio de información.

Algunos modelos disponen de recursos que permiten directamente establecer la comunicación bajo un protocolo determinado. Entre estos recursos se destacan el adaptador de comunicación serie asíncrona UART, el puerto paralelo esclavo para conectarse con los buses de otros microprocesadores, el módulo de bus serie universal USB para conexión con computadoras, la interfaz serie de dos hilos en modalidad I²C y en comunicación SPI, entre otros.

2.6.1.1. Comunicación SPI

De especial interés en este trabajo es la comunicación SPI. Algunos microcontroladores PIC poseen el módulo de puerto serial síncrono maestro, MSSP, que permite realizar la conexión y comunicación por SPI.

La comunicación por interfaz serial periférica, SPI (*Serial Peripheral Interface*), consiste en un bus de tres líneas, sobre el cual se transmiten paquetes de información de 8 *bits*. Cada una de estas tres líneas transporta información entre los dispositivos conectados al bus.

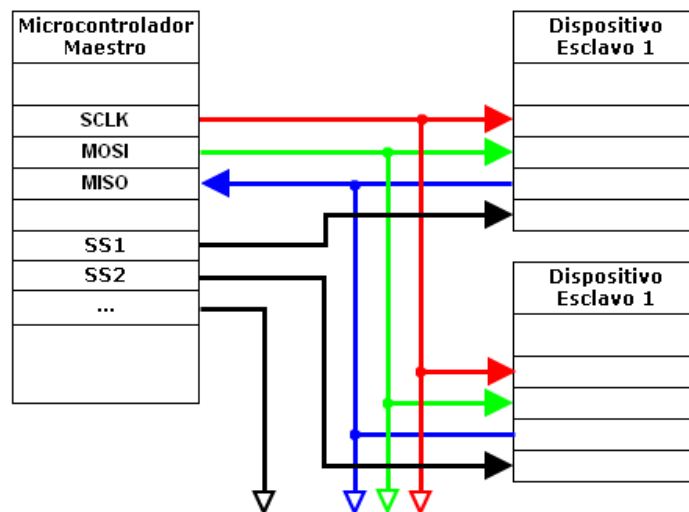
Cada dispositivo conectado al bus puede actuar como transmisor y receptor al mismo tiempo, por lo que este tipo de comunicación serial es *full dúplex*, es decir, completamente bidireccional simultáneamente. Los dispositivos involucrados son definidos como maestros y esclavos. Un maestro

es aquel que inicia la transferencia de información sobre el bus y genera las señales de reloj y control, para hacer llegar la información a un esclavo seleccionado por una línea específica.

Todas las líneas del bus transmite la información en una sola dirección. La señal sobre la línea de reloj (SCLK) es generada por el maestro y sincroniza la transferencia de datos. La línea MOSI (*Master Out Slave In*) transporta los datos del maestro hacia el esclavo. La línea MISO (*Master In Slave Out*) transporta los datos del esclavo hacia el maestro. Y finalmente la línea SS (*Slave Select*) selecciona el dispositivo esclavo con el cual se comunica.

La figura 6 presenta el bosquejo de la conexión de dispositivos a través de la interface SPI.

Figura 6. **Conexión SPI**



Fuente: elaboración propia.

2.7. Interrupciones

Uno de los propósitos del microcontrolador consiste en reaccionar ante los cambios o eventos que ocurren en su entorno, interrumpiendo y suspendiendo su operación normal y dedicándose a realizar una acción específica. Una vez el microcontrolador ha terminado de atender dicho evento, retoma su operación normal en el punto exacto donde se interrumpió. La señal que informa al procesador central acerca de dicho acontecimiento se denomina interrupción.

En el microcontrolador PIC hay dos tipos de interrupciones posibles, una es mediante una acción externa y la otra es interna. Las interrupciones externas se realizan a través de pines específicos del PIC, mientras que las interrupciones internas pueden ser realizadas por el mismo programa en ejecución o por la detección de anomalías en operaciones internas.

Una interrupción externa se detecta por el cambio del nivel de tensión en los pines INT del microcontrolador. Este cambio de tensión se le denomina flanco y puede ser de subida, cuando la tensión cambia de un estado lógico 0 a un estado lógico 1, o bien de bajada, cuando la tensión cambia de un estado lógico 1 a un estado lógico 0. La tensión de un estado lógico 1 es mayor a la tensión de un estado lógico 0.

Los pines INT utilizados para detectar interrupciones externas se encuentran en el puerto B de un PIC, y deben ser habilitados y configurados como entradas para cumplir correctamente la función de interrupción.

Una interrupción interna se detecta por el cambio de estado de ciertos *bits*, denominados banderas de interrupción, que indican si hubo alguna anomalía en la operación de un registro, por ejemplo el desbordamiento de la capacidad de

un registro puede activar una interrupción. También se puede especificar en el código del programa de ejecución que se realice una interrupción.

La acción que realiza el microcontrolador al detectar una interrupción se le llama rutina de servicio de interrupción, la cual consta de un conjunto de instrucciones que el microcontrolador debe ejecutar. Es muy importante resaltar que dentro de dicha rutina de servicio de interrupción se debe desactivar la bandera de detección de interrupción, para que al terminar el PIC retome nuevamente la operación que estaba ejecutando anteriormente.

2.8. Microcontrolador PIC18F4520

La familia de microcontroladores PIC18 ofrece características convenientes de comunicación con otros dispositivos y equipos, implementando detección de interrupciones y protocolos de comunicación.

Entre los recursos internos del microcontrolador PIC18F4520, necesarias para la conexión con los módulos adicionales de la estación lectora, se tienen:

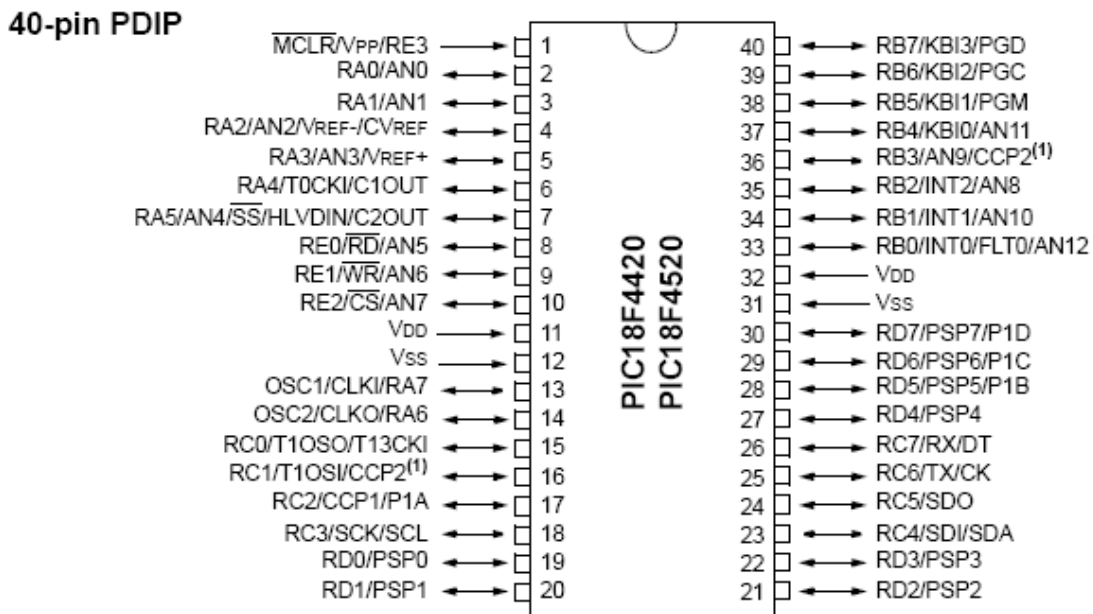
- Tres pines para interrupciones externas programables, que se utilizarán para sincronización y el control de lectura del código de la etiqueta de RF.
- Posibilidad de utilizar un oscilador con frecuencias hasta los 40 MHz, suficientes para sincronizar y detectar señales de alta frecuencia.
- Multiplicador de frecuencia con PLL. Multiplicará la frecuencia de entrada del cristal o resonador por 4, permitiendo alcanzar la velocidad necesaria para la lectura de las etiquetas de radiofrecuencia.

- Módulo de MSSP, puerto serial síncrono maestro, con soporte de comunicación SPI, para la conexión del microcontrolador con el módulo *Serial Ethernet Board*, y así comunicarse dentro de una red *Ethernet*.
- 36 pines digitales de entrada/salida

2.8.1. Descripción del patillaje

El microcontrolador PIC18F4520 posee en total una distribución de 40 pines divididos en tres puertos digitales de ocho pines cada uno, un puerto digital de seis pines, otro puerto digital de tres pines, cuatro pines de alimentación de voltaje, dos pines para el oscilador externo y un pin de reinicio. La figura 7 muestra el encapsulado tipo DIP del microcontrolador PIC18F4520.

Figura 7. Diagrama de pines del PIC18F4520



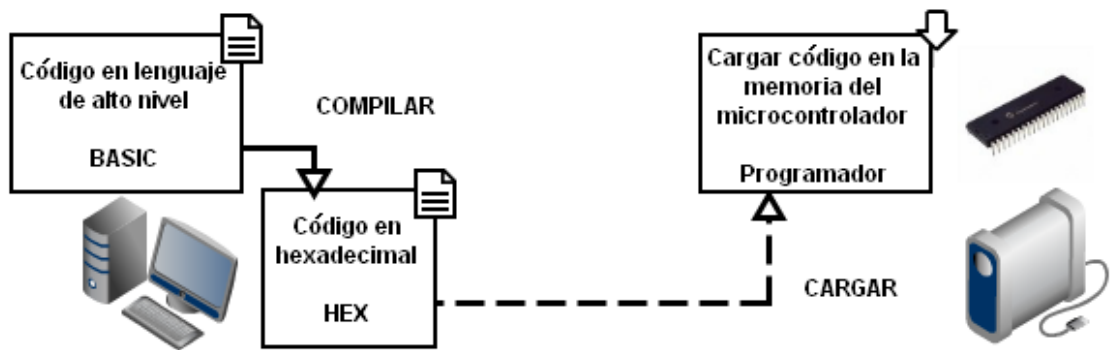
Fuente: Microchip. PIC18F4520 Data Sheet. p. 3

2.9. Programación del microcontrolador

El proceso de programación de un microcontrolador se puede separar en la siguiente serie de pasos:

- Escribir las sentencias e instrucciones en un lenguaje de programación
- Compilar las sentencias e instrucciones a código binario
- Cargar el código binario a la memoria del microcontrolador

Figura 8. Proceso de programación de un microcontrolador



Fuente: elaboración propia.

2.9.1. Escritura del código

El microcontrolador ejecuta las instrucciones del programa cargado en su memoria *flash*. Dicho programa se denomina como código ejecutable y está compuesto por una serie de ceros y unos que indican las operaciones que el microcontrolador debe realizar.

Dependiendo de la arquitectura del microcontrolador, el código binario está compuesto por palabras de 12, 14 o 16 *bits* de anchura. Cada palabra es interpretada por el procesador como una instrucción a ser ejecutada.

Es más compacto trabajar con el sistema de numeración hexadecimal por lo que el código ejecutable se representa con frecuencia como una serie de los números hexadecimales denominada código hexadecimal (HEX). La escritura de un código ejecutable en hexadecimal es muy complicada y tediosa, además que requiere de tiempo y de conocimiento detallado del microcontrolador.

Para facilitar la escritura del código ejecutable se desarrolló el lenguaje ensamblador (ASM). Las instrucciones en ensamblador consisten en las abreviaturas con significado. Un programa denominado ensamblador instalado en una computadora compila las instrucciones del lenguaje ensamblador a código hexadecimal. La ventaja principal del lenguaje ensamblador es su simplicidad y el hecho de que a cada instrucción corresponde una localidad de memoria. Además permite el control detallado de todos los procesos.

En la búsqueda y desarrollo de un lenguaje de programación que tuviera mayor similitud al lenguaje utilizado en el habla cotidiana, fueron creados los lenguajes de programación de alto nivel, y entre ellos el lenguaje *Basic*. La ventaja principal de estos lenguajes es la simplicidad de escribir un programa.

Similar al lenguaje ensamblador, un programa especializado e instalado en la computadora se encarga de compilar un programa a código máquina. A diferencia del ensamblador, los compiladores para los lenguajes de programación de alto nivel crean un código ejecutable que no es siempre tan corto como el código escrito en ensamblador.

El lenguaje *Basic* es un lenguaje de programación simple y fácil de entender. Para utilizarlo correctamente, basta con conocer sólo unos pocos elementos básicos en los que consiste cada programa. Estos son:

- **Identificadores.** Los identificadores son los nombres arbitrarios asignados a los objetos básicos del lenguaje tales como constantes, variables, funciones, procedimientos etc.
- **Comentarios.** Los comentarios son utilizados para aclarar las instrucciones de programa y para proporcionar más información.
- **Operadores.** Un operador es un símbolo que denota una operación aritmética, lógica u otra operación particular.
- **Instrucciones.** Una instrucción es un comando que especifica la operación a realizar.
- **Constantes.** Las constantes son los números o caracteres cuyo valor no puede ser cambiado durante la ejecución de programa.
- **Variables.** Una variable es un objeto nombrado capaz de contener un dato que puede ser modificado durante la ejecución de programa.
- **Símbolos.** Un símbolo es un identificador con un nombre especificado por el programador, que puede representar una constante, un registro, una instrucción o función, etc.
- **Etiquetas.** Se utilizan para denotar líneas particulares en el programa donde se debe ejecutar la instrucción de salto y la subrutina apropiada.

- Procedimientos y funciones. Las funciones y los procedimientos, denominados bajo el nombre común de rutinas, son subprogramas (bloques de sentencias autónomos) que ejecutan ciertas tareas a base de un número de los parámetros de entrada. Las funciones devuelven un valor después de la ejecución, mientras que los procedimientos no devuelven un valor.

2.9.2. **Compilador *MikroBasic***

El compilador *MikroBasic* es un *software* desarrollado por la empresa *MikroElektronika*, el cual consta de una amplia variedad de librerías que permiten una programación efectiva y fácil acceso al usuario. Entre unas de las capacidades que tiene este compilador se pueden mencionar:

- Permite escribir el programa en lenguaje *Basic*
- Utilización de librerías incluidas para el mejoramiento y rapidez de adquisición de datos, memoria, conversiones y comunicaciones.
- Monitorear la estructura del programa, variables, y funciones utilizando un explorador de código.
- Generación de archivo ensamblador (ASM) y hexadecimal (HEX) estándar compatible para cargar el código a la memoria del microcontrolador.
- Un depurador para inspeccionar el flujo del programa paso a paso
- Mapas de visualización de memoria ROM y RAM
- Amplia variedad de modelos de microcontroladores PIC

Lo primero a realizar al iniciar el compilador es crear un proyecto que contendrá todas las especificaciones y parámetros como el modelo de microcontrolador PIC a utilizar y frecuencia de operación del oscilador, librerías de funciones y otros. El proceso de crear y ejecutar un proyecto contiene los siguientes pasos:

- Crear un proyecto (nombre de proyecto, configuración de proyecto, dependencias entre archivos).
- Editar un programa
- Depurar el código, ejecutando el programa paso a paso para asegurarse de que se ejecutan las operaciones deseadas.
- Compilar el programa y corrección de errores
- Programar un microcontrolador (cargar el archivo HEX generado por el compilador en el microcontrolador utilizando el programador *PICflash*).

2.9.2.1. Librerías de funciones

Las librerías son archivos independientes que contienen un gran número de funciones listas para ser utilizadas en la escritura de otro código. Las librerías en *MikroBasic* proporcionan muchas facilidades para escribir programas para los microcontroladores PIC. El compilador debe conocer todas las dependencias del archivo fuente de *MikroBasic* para poder compilarlo.

Las librerías a utilizar en la programación del código de la estación lectora, del sistema de identificación estudiantil por radiofrecuencia, son las mencionadas en la tabla II.

Tabla II. **Librerías utilizadas en el código del PIC**

| Librería | Descripción |
|-----------------------------|---|
| <i>Conversions Library</i> | Rutinas para conversiones de números en cadenas y BCD/decimal. |
| <i>Lcd Library</i> | Utilizada para las operaciones con el LCD de 2x16 caracteres. |
| <i>SPI Library</i> | Utilizada para las operaciones con el módulo SPI incorporado. |
| <i>SPI Ethernet Library</i> | Utilizada para la comunicación SPI con el módulo <i>Serial Ethernet Board</i> . |

Fuente: elaboración propia.

2.9.3. Carga del código al PIC

El programador *PICflash* es una herramienta diseñada para programar todos los tipos de microcontroladores PIC. Está compuesto por dos partes:

- La parte *hardware* se utiliza para poner en el búfer el código hexadecimal y para programar el microcontrolador por medio de niveles de voltaje específicos. Durante el proceso de la programación, un nuevo programa se escribe en la memoria *flash* del microcontrolador, mientras que el programa anterior es borrado.
- La parte de *software* se encarga de enviar el programa (archivo HEX) a la parte *hardware* del programador por medio de un cable USB. Antes de enviar el código, es posible modificar algunas configuraciones del programador y controlar el funcionamiento de la parte *hardware* como cargar un archivo, escribir el código en la memoria del PIC, limpiar la memoria y otras.

3. MÓDULOS PERIFÉRICOS ADICIONALES

Los módulos periféricos son circuitos electrónicos independientes, que son diseñados para realizar una función específica. Los módulos periféricos se conectan a las líneas de entrada/salida del microcontrolador PIC para interactuar y operar conjuntamente.

El microcontrolador gobierna la operación de la estación lectora. La información de una etiqueta de radiofrecuencia esta contenida en una *RFid card*, y se adquiere a través del módulo *RFid Reader*. La información se envía a la computadora, para ser registrada y almacenada, a través del módulo *Serial Ethernet Board*. Por último, el módulo LCD muestra indicaciones de uso, información y control al usuario de la estación lectora.

3.1. Módulo *RFid Reader* de *MikroElektronika*

Este módulo es utilizado para leer *RFid cards*, tarjetas pasivas de identificación por radiofrecuencia, por medio de ondas de radio a una frecuencia de 125 KHz. El módulo consta de un circuito integrado transmisor y receptor, una antena y pines de comunicación.

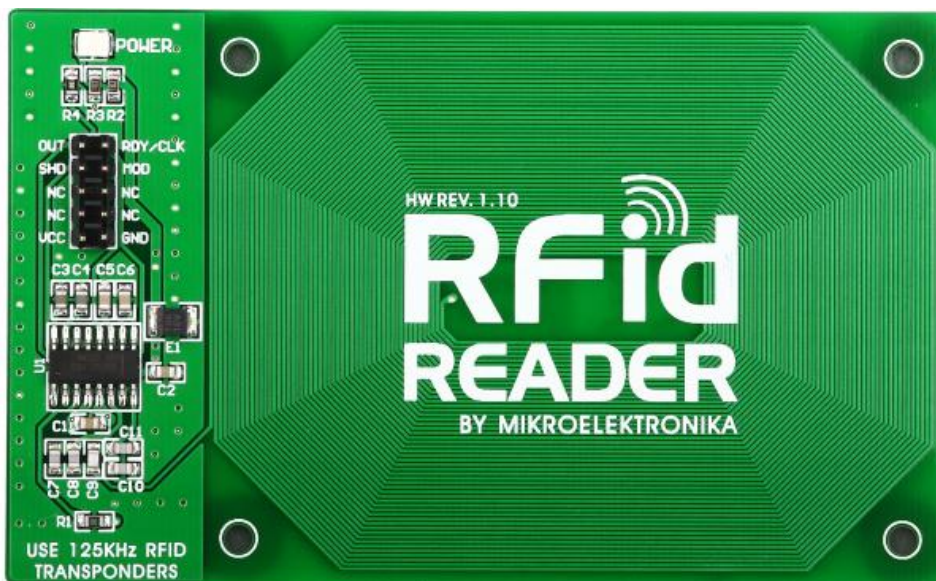
El circuito integrado transmisor y receptor es el dispositivo EM4095, que controla las señales moduladas en amplitud de 125 KHz que son enviadas a la antena en la transmisión, o que son recibidas de la antena en la recepción, realizando la operación de lectura del código único de la tarjeta pasiva.

La antena emite un campo electromagnético para activar la etiqueta radiofrecuencia e iniciar la lectura del código. Dicho campo no excede los 7 cm en la dirección principal de radiación. Los pines de comunicación transportan las señales de información hacia las líneas de entrada del microcontrolador.

El módulo *RFid Reader* cumple con tres funciones principales que son:

- Energizar la etiqueta de radiofrecuencia para que remita una respuesta
- Demodular la señal de radio recibida
- Decodificar la información de la etiqueta de radiofrecuencia

Figura 9. **Módulo *RFid Reader***



Fuente: www.mikroe.com/eng/products/view/185/rfid-reader-board/. Consulta: 13 de mayo de 2012.

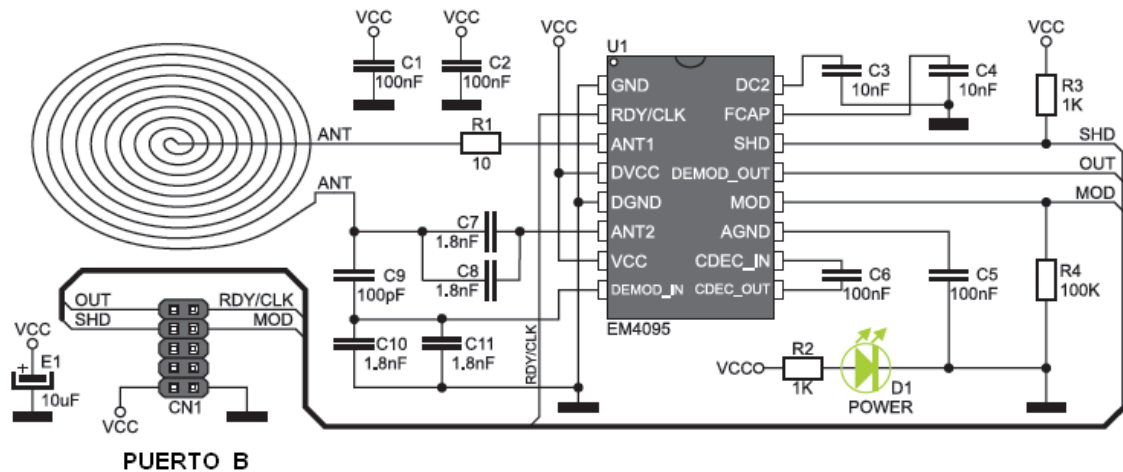
La figura 9 muestra el *hardware* que compone el módulo *RFid Reader* diseñado por la empresa *MikroElektronika*.

3.1.1. Componentes y diagrama circuital

Los componentes principales que contiene este módulo son:

- Un circuito integrado transceptor EM4095
- Una antena
- Pines para conexión al puerto B del microcontrolador PIC
- Elementos pasivos como resistores y capacitores

Figura 10. Diagrama circuital del módulo *RFid Reader*



Fuente: www.mikroe.com/eng/downloads/get/1240/rfid_manual_v100.pdf. Consulta: 13 de mayo de 2012.

El circuito integrado transceptor se encarga de generar las señales que alimentan a la antena, para crear el campo electromagnético. También es el encargado de transmitir las señales de información y reloj de sincronización al microcontrolador PIC. Se utiliza las líneas de entrada del puerto B del microcontrolador, porque éstas se configuran para ser utilizadas como entradas de interrupción.

3.1.2. Circuito integrado transceptor EM4095

Es un circuito integrado de tecnología CMOS, llamado transceptor porque cumple la función de transmisor y receptor de señales para estaciones de radiofrecuencia. Sus funciones principales son:

- Control de señales en la antena
- Modulación en amplitud del campo electromagnético para escribir en los chips de las etiquetas de radiofrecuencia.
- Demodulación en amplitud de la señal detectada en la antena, inducida por la lectura del chip de una etiqueta de radiofrecuencia.
- Generación de las señales de sincronización y de datos, para la comunicación con el microcontrolador u otro dispositivo de control.

El chip transceptor EM4095 está diseñado para ser conectado a un circuito de antena y a un microcontrolador. En su interior contiene los sistemas necesarios para operar, por lo que son pocos los componentes externos requeridos, los cuales son utilizados para acoplar el filtraje de radiofrecuencia.

El transceptor EM4095 presenta las siguientes características:

- Sistema de PLL integrado para acoplar la frecuencia de la señal portadora con la frecuencia de resonancia de la antena.
- Un rango de frecuencia portadora entre 100 a 150 KHz
- Controlador directo de alimentación de la antena
- Transmisión de datos en AM con 100% de modulación

- Compatibilidad con múltiples protocolos de chips de etiquetas de radiofrecuencia, como lo son: EM4102, EM4200, EM4205 y EM4450.

3.1.3. Señales de datos y control

Los pines de comunicación al exterior, presentan las siguientes señales:

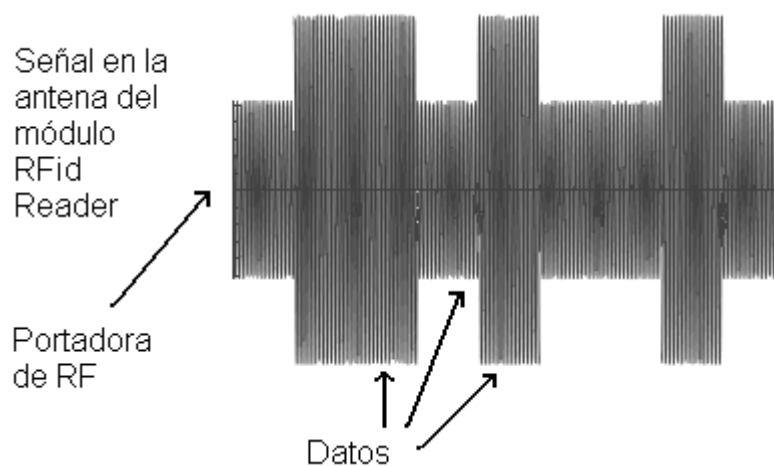
- OUT: Señal de salida digital del circuito integrado EM4095, que transmite el código extraído de la tarjeta de radiofrecuencia.
- RDY/CLK: Señal de salida digital con la frecuencia de reloj para la sincronización con el microcontrolador en la transmisión de datos.
- SHD: Señal de entrada digital para el control del estado de actividad o inactividad del módulo.
- MOD: Señal de entrada digital para activar la emisión de las ondas electromagnéticas en la antena.

3.1.4. Flujo de señales en la operación de lectura

La operación de lectura del código de una etiqueta de radiofrecuencia es la función primordial del módulo *RFid Reader*. La señal recibida en la antena por medio de ondas electromagnéticas es modulada en amplitud con datos digitales. El transceptor detecta las variaciones de amplitud de la señal portadora, con frecuencia de 125 KHz, para extraer los datos digitales del código. La figura 11 muestra un ejemplo de la señal de radiofrecuencia recibida.

Para que el módulo *RFid Reader* opere en modo de lectura, éste debe ser configurado utilizando señales específicas en las entradas digitales SHD y MOD. La señal en la entrada SHD debe permanecer en estado lógico bajo, al igual que la señal en la entrada MOD.

Figura 11. **Ejemplo de señal electromagnética en la operación de lectura**

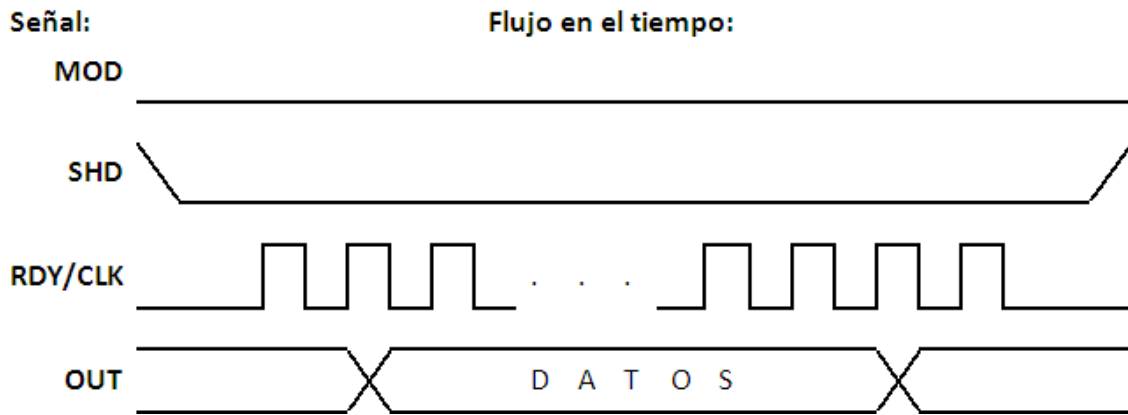


Fuente: elaboración propia.

El módulo transmite el código leído de una etiqueta de radiofrecuencia por medio de las salidas digitales OUT y RDY/CLK. La salida digital RDY/CLK emite una señal de reloj para la sincronización en el muestreo de los datos digitales provenientes de la salida OUT.

El microcontrolador debe detectar la señal de reloj y sincronizarse. Una vez sincronizado puede empezar a muestrear los *bits* consecutivos de los datos del código.

Figura 12. Señales de la operación de lectura



Fuente: elaboración propia.

3.2. Tarjetas *RFid card*

Las tarjetas *RFid cards* utilizan etiquetas pasivas, es decir que no poseen una fuente de alimentación propia. Este tipo de tarjetas debe utilizar la frecuencia de 125 KHz para que sean compatibles con el lector *RFid Reader*.

Para lograr la lectura de la información de la tarjeta, la antena del lector debe emitir un campo electromagnético que energice el circuito de la etiqueta, para ello la tarjeta debe colocarse en el espacio físico dentro de dicho campo, que tiene un rango aproximado de 7 centímetros desde el lector *RFid Reader*.

La figura 13 muestra la imagen de las tarjetas de radiofrecuencia, fabricadas en material plástico, en cuyo interior se encuentran un circuito integrado y una antena.

Figura 13. **RFid cards de 125 KHz**



Fuente: www.mikroe.com/img/development-tools/componentes/rfid-card/rfid_thumb.gif. Consulta: 13 de mayo de 2012.

3.2.1. Componentes

Una tarjeta consta de un circuito integrado, que contiene el código de identificación único, y una antena. El circuito integrado es el dispositivo EM4100 que al ser energizado emite una señal, modulada en amplitud, con el código almacenado en su memoria hacia la antena, que lo convierte en ondas de radio con una frecuencia de 125 KHz.

3.2.2. Circuito integrado transpondedor EM4100

Es un circuito integrado de tecnología CMOS, llamado transpondedor porque realiza la operación de responder, es decir, transmite una señal de radiofrecuencia de respuesta cuando se le coloca dentro de un campo electromagnético externo.

El circuito integrado EM4100 consta de las siguientes características:

- Memoria con capacidad de 64 *bits*

- Rectificador de onda completa, para extraer el voltaje de alimentación
- Resonador de capacitor, para extraer la frecuencia del campo electromagnético externo.

- Codificador de los datos del número de identificación
- Modulador de los datos a transmitir

El transpondedor es energizado por un voltaje inducido en la antena desde un campo electromagnético proveniente de la estación lectora de etiquetas de radiofrecuencia. El voltaje, de corriente alterna, inducido es rectificado para proveer una alimentación de voltaje de corriente directa.

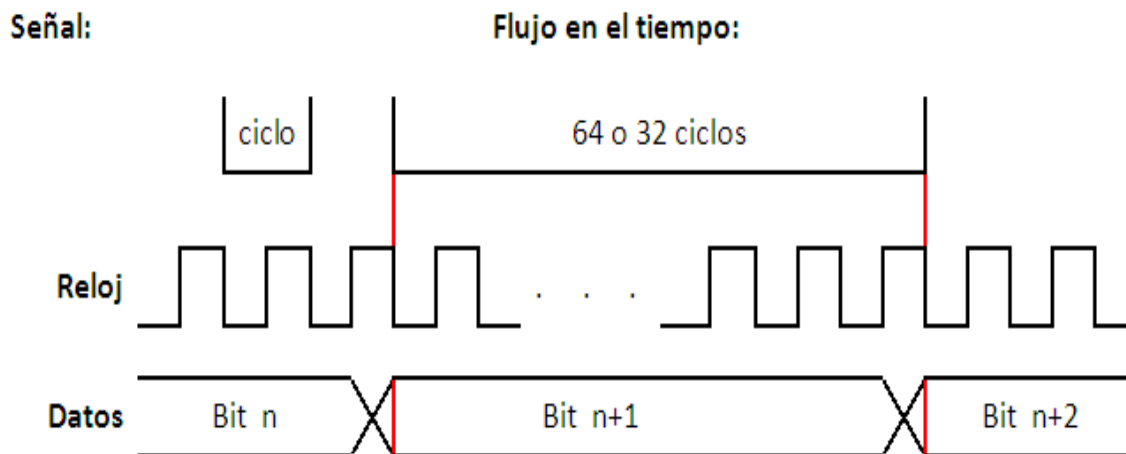
El transpondedor utiliza un resonador para extraer la frecuencia del campo electromagnético externo, para generar propiamente una señal de reloj. La señal de respuesta emitida es modulada con la misma frecuencia extraída. El rango de frecuencias aceptadas es desde 100 KHz hasta 150 KHz.

La memoria tiene la capacidad de almacenar 64 *bits*, grabados desde su fabricación. Los datos deben ser codificados para la transmisión. Los tipos de codificación posibles son Manchester, Bifásico, y PSK. Para el presente trabajo se especifica la utilización de chips con codificación Manchester.

La emisión de los datos codificados por radiofrecuencia ocurre encendiendo y apagando la corriente de modulación de acuerdo al valor del *bit* a transmitir. Con un valor de *bit* igual a uno lógico, se enciende la modulación, y con un valor de cero lógico, se apaga. Una vez los 64 *bits* han sido emitidos, la secuencia se repite continuamente hasta que la fuente de energía desaparece.

La tasa de transmisión, de cada *bit* de datos codificados, puede ser de 64 o 32 ciclos de la señal de reloj, según la especificación desde su fabricación. Para el presente trabajo se especifica la utilización de chips con transmisión de un *bit* cada 64 ciclos de la señal de reloj.

Figura 14. **Representación de la tasa de transmisión de *bits***



Fuente: elaboración propia.

La figura 14 muestra la representación en el tiempo de un *bit* transmitido desde el circuito integrado EM4100 por radiofrecuencia.

3.2.3. **Protocolo del código de la tarjeta**

La información de las etiquetas de radiofrecuencia se transmite utilizando el estándar de codificación Manchester. El código de transpondedor consta de 64 *bits*, entre los cuales 40 *bits* son de información de identificación, 9 *bits* son de inicio, 14 *bits* son de control de paridad par, y un *bit* de finalización cero.

La transmisión de los *bits* se realiza en forma serial, un *bit* tras otro. En dicha transmisión todos los *bits* de información y de control están combinados. Para interpretar el código leído, se debe agrupar los *bits* formando la estructura mostrada en la figura 15. Los *bits* de control se determinan por paridad par a partir de los *bits* de fila o de columna, según sea su posición.

Figura 15. Estructura del código del transpondedor EM4100

Reconocimiento del código:

| | | | | | | | | | | |
|---|-----|-----|-----|-----|----|---|---|---|---|---------------------------------|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | Encabezado |
| Dígitos de identificación del propietario | D00 | D01 | D02 | D03 | P0 | | | | | |
| | D10 | D11 | D12 | D13 | P1 | | | | | |
| 8 dígitos del código de la tarjeta en hexadecimal | D20 | D21 | D22 | D23 | P2 | | | | | Bits de paridad de cada fila |
| | D30 | D31 | D32 | D33 | P3 | | | | | |
| | D40 | D41 | D42 | D43 | P4 | | | | | |
| | D50 | D51 | D52 | D53 | P5 | | | | | |
| | D60 | D61 | D62 | D63 | P6 | | | | | |
| | D70 | D71 | D72 | D73 | P7 | | | | | |
| | D80 | D81 | D82 | D83 | P8 | | | | | |
| | D90 | D91 | D92 | D93 | P9 | | | | | |
| Bits de paridad de cada columna | PC0 | PC1 | PC2 | PC3 | 0 | | | | | Bit final "0" |

Fuente: elaboración propia.

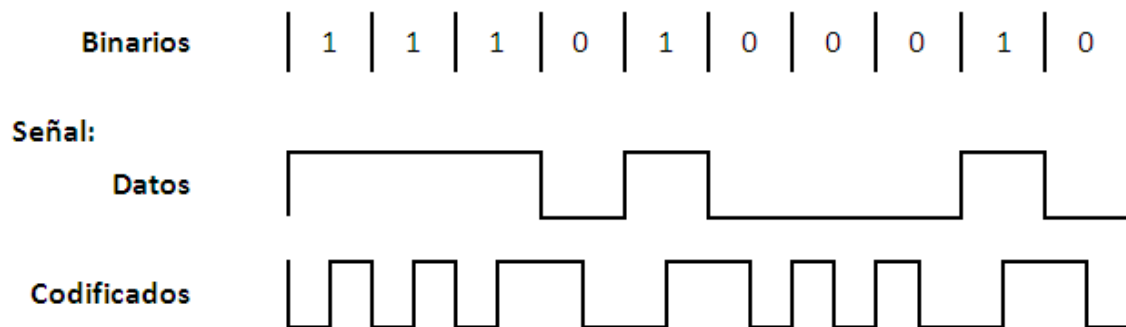
El encabezado está compuesto por 9 *bits* programados en estado lógico uno. El encabezado es seguido por 10 grupos, compuestos cada uno por 4 *bits* de datos y un *bit* de paridad par de fila. El último grupo de *bits* consiste en 4 *bits* de paridad par de columna y el *bit* de finalización en estado lógico cero.

3.2.4. Codificación Manchester

En este tipo de codificación, los datos de salida sufren una transición de estado lógico al medio del período de cada *bit* de datos de entrada. Cada *bit* de entrada se representa por dos *bits* en la salida. Un *bit* uno lógico se representa por dos *bits*, un cero lógico y uno lógico en dicho orden. Un *bit* cero lógico se representa por dos *bits*, un uno lógico y un cero lógico en dicho orden.

Otro efecto que produce la codificación es que los datos de salida se duplican en cantidad y en frecuencia a comparación con los datos de entrada. La figura 16 muestra un ejemplo de la codificación Manchester.

Figura 16. **Ejemplo de codificación Manchester**



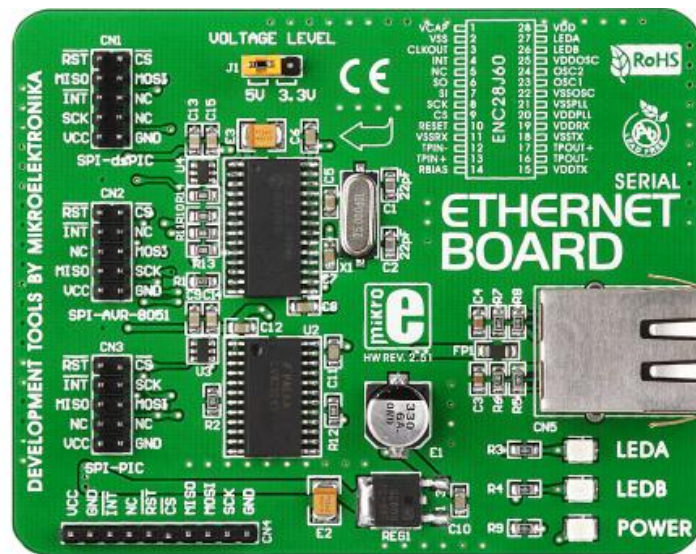
Fuente: elaboración propia.

3.3. Módulo *Serial Ethernet Board* de *MikroElektronika*

Este módulo es utilizado para conectar un microcontrolador a una red *Ethernet* mediante el sistema de comunicación SPI. *Ethernet* es la tecnología más utilizada en redes LAN (*Local Area Network*), red de área local, hoy en día.

La figura 17 muestra el *hardware* que compone el módulo *Serial Ethernet Board* diseñado por la empresa *MikroElektronika*.

Figura 17. **Módulo *Serial Ethernet Board***



Fuente: www.mikroe.com/eng/products/view/14/serial-ethernet-board/. Consulta: 13 de mayo de 2012.

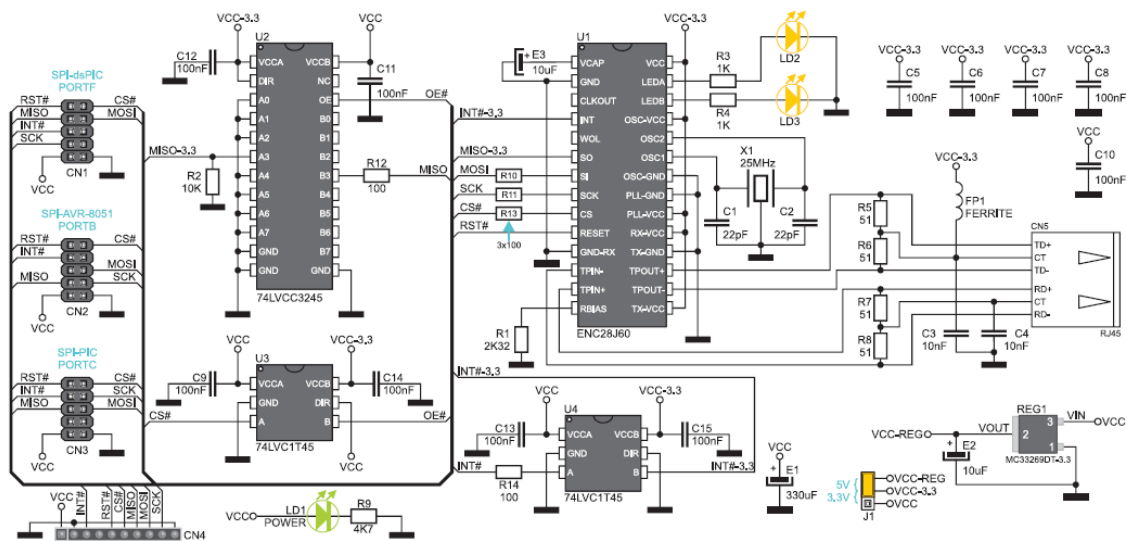
Esta tecnología comunica físicamente estaciones enviando paquetes de datos. Cada estación *Ethernet* posee una única dirección física de 48 *bits*, que es usada para especificar la estación fuente y la estación destino de cada paquete de datos.

3.3.1. Componentes y diagrama circuital

El módulo consta de un circuito integrado controlador de red *Ethernet*, un conector RJ45, un circuito regulador de tensión de 3,3 voltios, un circuito integrado convertidor de niveles de tensión de señales entre 5 voltios y 3,3

voltios, y tres LED (*Light Emitting Diode*), diodos emisores de luz, indicadores de estado. La figura 18 muestra el circuito y los componentes que constituyen el módulo *Serial Ethernet Board*.

Figura 18. Diagrama circuital del módulo *Serial Ethernet Board*



Fuente: www.mikroe.com/eng/downloads/get/45/serial_ethernet_manual_v100.pdf. Consulta: 14 de mayo de 2012.

El circuito integrado controlador de red *Ethernet* es el dispositivo ENC28J60 que contiene una dirección física de control y acceso, *MAC Address (Media Access Control Address)*, también un búfer de memoria RAM de 8 KB y un circuito de comunicación SPI para conectar un microcontrolador.

El dispositivo ENC28J60 permite la comunicación de un microcontrolador con una computadora, u otros dispositivos que también puedan conectarse a la red *Ethernet*.

El dispositivo ENC28J60 requiere una tensión de alimentación de 3,3 voltios, para ello se requiere del regulador de tensión. Unos de sus pines de entrada son tolerantes ante señales con tensión de 5 voltios, pero los pines de salida envían únicamente señales con tensión de 3,3 voltios, que no son compatibles con los niveles de tensión requeridos por un microcontrolador. El dispositivo 74HCT245 se utiliza para incrementar el nivel de tensión de 3,3 voltios a 5 voltios.

Hay tres indicadores luminosos que cumplen las siguientes funciones:

- Indicador POWER: indica si el módulo esta encendido o no
- Indicador LEDA: indica el estado de conexión a la red *Ethernet*. Este muestra si el cable de red esta insertado en el conector RJ45.
- Indicador LEDB: indica si hay actividad en la red, es decir, indica el envío o recepción de paquetes de datos.

3.3.2. Controlador de red *Ethernet* ENC28J60

Es un circuito integrado diseñado para efectuar la función de controlador *Ethernet* de forma independiente, que posee un módulo de comunicación SPI. Este dispositivo es diseñado para actuar como una interfaz entre una red *Ethernet* y un microcontrolador equipado con SPI.

El controlador ENC28J60 cumple con las especificaciones del estándar IEEE 802,3, estándar dedicado a la tecnología *Ethernet*, e incorpora una serie de esquemas de filtrado de paquetes para limitar el número de paquetes entrantes.

La comunicación con el microcontrolador, servidor, se realiza mediante el bus SPI a una velocidad de hasta 10 Mbps y dos pines especiales capaces de provocar interrupciones en el ENC28J60. Otros dos pines sirven para manejar LED que indiquen la conexión y la actividad de la red.

El ENC28J60 tiene siete bloques funcionales importantes:

- Interfaz SPI que se encarga de la comunicación con el microcontrolador
- Registros que se emplean para monitorear y controlar al ENC28J60
- Un búfer RAM doble, para los paquetes recibidos y enviados
- Un módulo encargado de los accesos al búfer RAM mediante la técnica DMA (*Direct Memory Access*), acceso directo a memoria.

- Un bus que interpreta los datos y comandos recibidos vía SPI
- Un módulo MAC (*Media Access Control*) que cumple con las especificaciones del estándar IEEE 802,3.

- Un módulo PHY (*Physical Layer*) que codifica y decodifica los datos analógicos presentes en el cable par trenzado.

Además de ello, el dispositivo cuenta con otros bloques de soporte, como el oscilador, regulador de voltaje interno, adaptadores de nivel para proveer líneas de entrada y salida de 5 voltios y lógica de control.

Toda la memoria disponible en el dispositivo ENC28J60 es del tipo RAM estática. Se divide en tres bloques principales:

- Registros de control
- Búfer *Ethernet*

- Registros del módulo PHY

El primer bloque se emplea para controlar y obtener información sobre el estado del ENC28J60. Todos estos registros pueden ser accedidos, en lectura o escritura, mediante la interfaz SPI.

El segundo bloque tiene un tamaño que puede ser ajustado por el usuario y tiene como función servir de memoria temporal para los datos que se envían a la red o que se reciben. Esta zona de memoria también puede ser accedida mediante la interfaz SPI a través de los comandos apropiados.

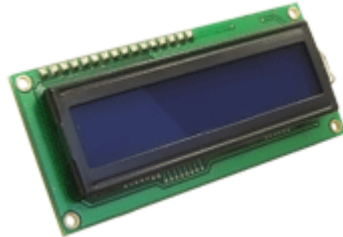
El tercer bloque se emplea para configurar y controlar el módulo PHY. Esta zona no puede ser accedida vía SPI. El acceso sólo puede realizarse mediante una comunicación a través del módulo MAC.

3.4. Módulo LCD 2x16

Es un módulo que contiene una pantalla de cristal líquido que presenta caracteres alfanuméricos. Este módulo es capaz de representar 2 líneas de 16 caracteres y además cada carácter se forma por una matriz de 5x7 puntos.

El carácter ASCII que se desea visualizar se envía mediante 8 líneas de datos, así como ciertos códigos de control que permiten realizar diferentes efectos de visualización, entre los cuales se tienen: desplazamiento de caracteres, hacia la izquierda y hacia la derecha, parpadeo del cursor, especificación de la posición de cada carácter, etc.

Figura 19. **Módulo LCD 2x16**



Fuente: www.mikroe.com/img/development-tools/components/2x16-lcd/lcd_thumb.gif. Consulta: 14 de mayo de 2012.

3.4.1. Componentes

El módulo consta de un circuito integrado controlador de la pantalla, la pantalla de cristal líquido de 2 líneas con 16 caracteres cada una, y una unidad de luz de fondo. El circuito integrado controlador de la pantalla es el dispositivo Hitachi HD44780U.

Los pines de conexión de este módulo incluyen un bus de datos de 8 *bits*, un pin de habilitación (E), un pin de selección, que indica que el dato es una instrucción o una carácter del mensaje (RS) y un pin de selección de escritura o lectura en el módulo (R/W).

3.4.2. Patillaje

El módulo LCD utiliza 16 pines para la conexión externa. Los pines son los siguientes:

- Vss. Es la patilla de tierra de alimentación
- Vdd. Es la patilla de alimentación de 5 voltios

- Vo. Es la patilla de contraste de la pantalla de cristal líquido. Normalmente se conecta a un potenciómetro a través del cual se aplica una tensión variable entre 0 y 5 voltios que permite regular el contraste de la pantalla.
- RS. Es la patilla de selección del registro de control/registro de datos. Cuando la señal es cero lógico se selecciona el registro de control, y cuando la señal es uno lógico se selecciona el registro de datos.
- R/W. Es la patilla que selecciona el modo de lectura o escritura. Cuando la señal es cero lógico el modo es de escritura, y cuando la señal es uno lógico el modo es de lectura.
- E. Es la patilla de habilitación del módulo. Cuando la señal es cero lógico el módulo es deshabilitado y no funciona, cuando la señal es uno lógico el módulo es habilitado para funcionar.
- D0-D7. Bus de datos de bidireccional. A través de estas líneas se realiza la transferencia de información entre el módulo y el microcontrolador.

A pesar de que el bus de datos es formado por ocho líneas de entrada y salida, se puede configurar el módulo para que utilice únicamente las cuatro líneas superiores, de D4 a D7. Con esta configuración se disminuye la cantidad de líneas a usar en el microcontrolador, pero se utilizan más ciclos de escritura.

3.4.3. Controlador Hitachi HD44780U

Este dispositivo es el encargado de recibir los datos de caracteres y señales de control provenientes del microcontrolador, y también realiza la

visualización del texto y símbolos deseados en la posición correspondiente dentro de la pantalla de 2x16 caracteres.

El controlador HD44780U permite realizar las siguientes funciones de visualización:

- Presentar caracteres ASCII
- Recibir y mostrar caracteres personalizados
- Desplazar los caracteres hacia la derecha así como hacia la izquierda
- Cambiar el aspecto y movimiento del cursor
- Direccionar la posición de visualización de los caracteres

El controlador HD44780U contiene una memoria no volátil, llamada CGROM (*Character Generator ROM*), la cual contiene almacenada, en matrices de 5x7 pixeles, la visualización de los caracteres predefinidos y soportados por la pantalla.

También es posible diseñar y programar caracteres personalizados a una memoria volátil del controlador HD44780U, llamada CGRAM (*Character Generator RAM*), con capacidad de 64 *bytes* que permite almacenar hasta 8 caracteres nuevos para ser desplegados en la pantalla.

Cuando el microcontrolador envía los datos de los caracteres que se desean mostrar en la pantalla, dichos caracteres ASCII se almacenan previamente en una memoria llamada DDRAM (*Display Data RAM*), y después pasan a ser visibles en la pantalla. La memoria DDRAM, tiene capacidad de almacenar hasta 80 caracteres en total. Por cada línea, la memoria DDRAM puede soportar 40 caracteres, aunque para el caso de la pantalla 2x16 únicamente se visualicen 16 caracteres simultáneamente.

3.4.4. Señales de datos y control para desplegar texto

Del bus de datos solamente se utiliza la mitad cuando se trabaja en modo de 4 *bits*. En el modo de 4 *bits* se debe leer o escribir dos veces, en la primera transmisión se envía el *nibble* con la parte alta del *byte* de datos o instrucción (MSB) y en la segunda el *nibble* con la parte baja del *byte* (LSB) del dato o instrucción.

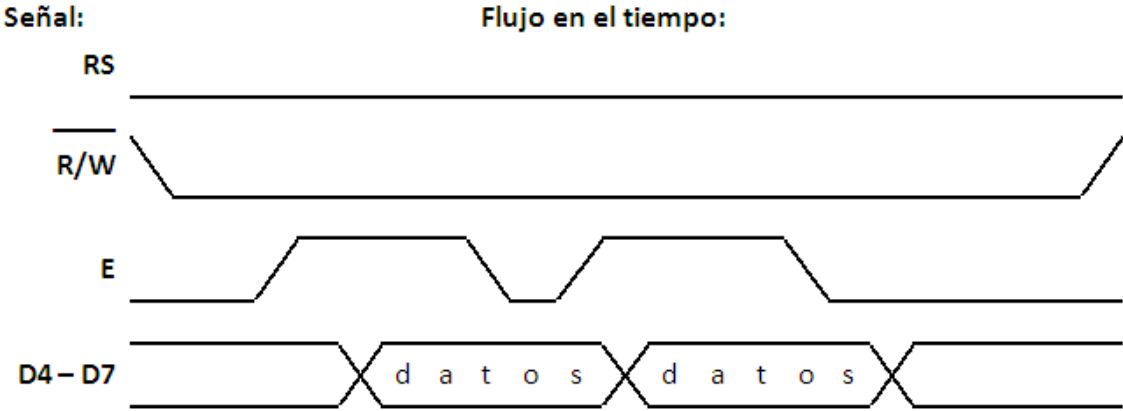
En el presente trabajo se utiliza únicamente el modo de escritura en el módulo LCD de 2x16, por lo que a continuación se describe el flujo de señales que permiten la transmisión y visualización de caracteres.

Para transmitir el código ASCII del carácter que se desea visualizar, el microcontrolador debe colocar la señal R/W en estado lógico cero para que se seleccione el modo de escritura. Después la señal E se coloca en estado lógico uno para habilitar el módulo. Lo siguiente es enviar el código del carácter a través de las líneas de datos, en modalidad de 4 *bits*.

Dado que se utiliza el modo de 4 *bits*, la escritura completa del código de una instrucción o de un carácter requiere de 2 ciclos de de transmisión de datos. En el modo de escritura, no importa el estado lógico de la señal RS, por lo que se puede colocar a estado lógico cero de forma permanente.

La figura 20 muestra el flujo de señales para la escritura de caracteres en pantalla.

Figura 20. Flujo de señales de escritura en pantalla LCD



Fuente: elaboración propia.

4. CONEXIÓN A BASE DE DATOS EN SERVIDOR LINUX

Para el desarrollo del presente trabajo se requiere de una computadora para el almacenamiento de la información de los estudiantes dentro de una base de datos. También dicha computadora debe comunicarse por LAN sobre estándar *Ethernet* con la estación lectora de tarjetas de radiofrecuencia para llevar el registro de entrada y salida de los usuarios del laboratorio que posean la tarjeta de radiofrecuencia.

Para lograr que la computadora ejecute los servicios de control y comunicación, requiere de los siguientes programas de *software* libre instalados:

- Sistema operativo Linux, distribución de Ubuntu 11,10 llamada *Oneiric Ocelot*.
- Servidor *web* Apache
- Servidor de base de datos MySQL
- Gestor gráfico de base de datos PhpMyAdmin
- Compilador del lenguaje de programación PHP

Ubuntu es un sistema operativo que utiliza un núcleo Linux. Ubuntu está compuesto de múltiple software distribuido bajo una licencia libre o de código abierto, es decir que todo el mundo puede acceder al código de programación sin restricciones y tarifas de pago.

La versión 11,10 *Oneiric Ocelot*, lanzada en octubre de 2011, es la versión de Ubuntu utilizada durante el desarrollo del presente trabajo. Este sistema operativo es posible descargarlo de forma gratuita desde Internet, para que posteriormente sea grabado en un CD o en una memoria USB, y así ser instalado en la computadora.

4.1. Software libre

El *software* libre, o de código abierto (*open source*), es la denominación del *software* que expande la libertad de los usuarios sobre su producto adquirido. Dicha libertad se concreta cuando se cumplen las siguientes cuatro premisas

- Libertad de usar el programa, con cualquier propósito
- Libertad de estudiar cómo funciona el programa y modificarlo, adaptándolo a las necesidades propias.
- Libertad de distribuir copias del programa
- Libertad de mejorar el programa y hacer públicas esas mejoras a los demás, de modo que toda la comunidad se beneficie.

Las empresas de *software* libre por lo general dejan que sus programas sean gratis, pero cobran por servicios añadidos, tales como el trabajo de instalación del *software*, ofrecer soporte técnico, generar actualizaciones al *software*, y otros.

Bajo la filosofía del *software* libre se orienta el presente trabajo, por lo que todos los programas instalados pueden ser utilizados y configurados sin la necesidad de adquirir licencias.

4.2. Servidor web Apache

Apache es un servidor web HTTP (*Hypertext Transfer Protocol*) de código abierto, usado principalmente para transferir páginas *web* estáticas y dinámicas en la *World Wide Web*, es decir en la red amplia mundial conocida como Internet. Muchas aplicaciones *web* están diseñadas asumiendo como ambiente de implantación a Apache, o que utilizarán características propias de este servidor *web*.

Apache también es usado para muchas otras tareas donde el contenido necesita ser puesto a disposición de un cliente en una forma segura y confiable.

Apache presenta, entre otras características altamente configurables, bases de datos de autenticación y negociado de contenido.

4.3. Servidor base de datos MySQL

MySQL es la base de datos de código abierto más popular del mundo. MySQL ingresó en el mercado de las bases de datos en un momento en el que Apache es el producto de código abierto dominante en el mercado de servidores *web* y en el que la presencia de varios sistemas operativos de código abierto, como Linux, es cada día mas notable en el mercado de servidores.

MySQL es un sistema de administración de bases de datos relacional, es decir, que almacena datos en tablas separadas. Se trata de un programa capaz de almacenar una enorme cantidad de datos de gran variedad y de distribuirlos para cubrir las necesidades de cualquier tipo de organización, desde pequeños establecimientos comerciales a grandes empresas y organismos administrativos.

MySQL utiliza el lenguaje de consulta estructurado (SQL). Se trata del lenguaje utilizado por todas las bases de relacionales. Este lenguaje permite crear bases de datos, así como agregar, manipular y recuperar datos en función de criterios específicos.

4.3.1. Gestor gráfico de base de datos PhpMyAdmin

PhpMyAdmin es una herramienta escrita en lenguaje PHP con la intención de manejar la administración de MySQL a través de una página *web*. Actualmente puede crear y eliminar bases de datos, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar cualquier sentencia SQL, administrar claves en campos, administrar privilegios y exportar datos en varios formatos como hojas de cálculo o bien en pdf.

4.4. Lenguaje de programación web PHP

PHP, acrónimo de *PHP: Hypertext Preprocessor*, pero que en sus inicios se conocía como *Personal Home Page Tools (PHP Tools)*, es un lenguaje de código abierto interpretado de alto nivel, especialmente orientado para desarrollos *web* y el cual puede ser incrustado en páginas HTML. El objetivo de este lenguaje es permitir, a los creadores de páginas *web*, escribir páginas dinámicas de una manera rápida y fácil, aunque se puede hacer mucho más con PHP.

Se usa principalmente para la interpretación del lado del servidor ante peticiones *web* de un cliente, pero actualmente también puede ser utilizado en la creación de otros tipos de programas que se ejecutan desde una interfaz de línea de comandos o en aplicaciones con interfaz gráfica.

El lenguaje PHP es libre, por lo que se presenta como una alternativa de fácil acceso para todos, además de poseer una amplia documentación en su sitio *web* oficial. En el ámbito de bases de datos, PHP tiene la capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.

4.4.1. Aplicación en páginas *web*

En los inicios de Internet, las páginas *web* más sencillas fueron desarrolladas en el lenguaje HTML, que hizo posible presentar información de forma ordenada. Lo que se visualiza en una página en Internet es la interpretación que hace el navegador del código HTML.

HTML es la abreviatura de *HyperText Mark-up Language*, es decir, lenguaje de marcado de hipertexto. El HTML consta de varios componentes vitales, entre ellos los elementos y sus atributos, tipos de datos y la declaración de tipo de documento.

Los elementos proporcionan la estructura al documento HTML e indican al navegador la forma de presentación de la página *web*. Por lo general, los elementos están formados por una etiqueta de inicio, el contenido, y una etiqueta de cierre. Las etiquetas suministran atributos y propiedades al contenido.

Un documento HTML puede ser creado y editado con cualquier editor de textos básico, como puede ser Gedit en Linux, el Bloc de notas de Windows, o cualquier otro editor que admita texto sin formato como GNU Emacs, Microsoft Wordpad, entre otros.

Actualmente, las páginas *web* presentan muchas más aplicaciones, y no solamente texto e imágenes estáticas. En este contexto, las funciones de PHP proporcionan una presentación dinámica e interactiva a las páginas *web*, permitiendo el envío de información a través de formularios, la realización de consultas a bases de datos, autenticación de usuarios e ingreso a páginas *web* exclusivas, y otras más.

La sintaxis de las funciones de PHP se incluye dentro de la sintaxis de elementos de lenguaje HTML, formando así la estructura de la página *web* con las funciones dinámicas deseadas. Para que la página *web* sea interpretada con todas las funciones, el documento debe ser creado como un documento PHP, que se le conoce comúnmente como *script* de PHP. Cuando una página *web* creada en PHP es requerida por un navegador *web*, el intérprete de PHP del servidor *web* traduce el *script* PHP de dicha página, antes de enviarla y presentarla en el navegador.

4.4.2. Aplicación en conexión a base de datos MySQL

Una de las principales ventajas que presenta el trabajar con páginas dinámicas del lado del servidor es el poder trabajar con contenidos que están alojados en bases de datos. De esta forma, se puede organizar, actualizar y buscar la información de una manera mucho más simple.

El lenguaje PHP ofrece interfaces para el acceso a la mayoría de las bases de datos existentes. Existen funciones de PHP para bases de datos de código abierto, como MySQL, otras para comerciales propietarias como Oracle y además tiene librerías para acceso a datos por ODBC (*Open DataBase Connectivity*), lo que permite comunicarse con todas las bases de datos posibles en sistemas Microsoft, como Access o SQL Server.

Gracias al conjunto de funciones existentes para cada sistema gestor de base de datos, se puede realizar cualquier acción con los datos que se necesiten para el desarrollo de una variada gama de aplicación *web*. De esta manera PHP se relaciona con servidores *web* y base de datos.

4.4.3. Aplicación de *sockets* de comunicación

El término *socket* designa un concepto abstracto por el cual dos programas, posiblemente situados en computadoras distintas, pueden intercambiar cualquier flujo de datos, generalmente de manera fiable y ordenada sobre una LAN.

Los *sockets* de Internet forman el mecanismo para la entrega de paquetes de datos provenientes de la tarjeta de red a los procesos correspondientes de un programa. Un *socket* queda definido por las direcciones IP fuente y destino, un protocolo de transporte y los números de puerto fuente y destino.

Los *sockets* permiten implementar una arquitectura cliente-servidor entre máquinas. Un *socket* es un proceso, existente en una máquina cliente y en una máquina servidor, que sirve para que el programa servidor y el cliente envíen y reciban información entre ellos. Esta información será transmitida por las diferentes capas de red.

Hoy en día, los *sockets* están implementados como librerías de funciones de programación para multitud de sistemas operativos, simplificando el trabajo de escribir programas capaces de comunicarse entre sí. En PHP la extensión *socket* implementa una interfaz a bajo nivel para las funciones de comunicación de *sockets*, proporcionando la posibilidad de actuar tanto como servidor o como cliente de *socket*.

4.5. Comunicación por LAN sobre *Ethernet*

Una red de área local, LAN, es un sistema de comunicación de datos que permite a un determinado número de dispositivos independientes comunicarse en un área local.

Entre las características principales de una red de área local se tienen:

- Las comunicaciones son confinadas a un área geográfica moderada
- Por lo general son redes de propiedad privada
- La comunicación opera a velocidades moderada y alta, entre 1 y 100 Mbps, con bajo porcentaje de errores.
- Permite a las estaciones comunicarse directamente, empleando un medio físico común sobre enlace de punto a punto sin requerir de ningún nodo de conmutación intermedio.
- Permite la compatibilidad entre equipos de diferentes fabricantes

Ethernet es el estándar de la capa de acceso a la red más popular para la tecnología LAN usada actualmente. Otros tipos de LAN incluyen *Token Ring*, *Fast Ethernet*, FDDI, ATM y *LocalTalk*. *Ethernet* es popular porque permite un buen equilibrio entre velocidad, costo y facilidad de instalación.

Estos puntos fuertes, combinados con la amplia aceptación en el mercado y la habilidad de soportar virtualmente todos los protocolos de red populares, hacen a *Ethernet* la tecnología ideal para la red de la mayoría los usuarios de la informática actual.

El estándar *Ethernet* fue definida por el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) como IEEE *Standard* 802,3. Adhiriéndose a la norma de IEEE, los equipos y protocolos de red pueden comunicarse eficazmente entre ellos.

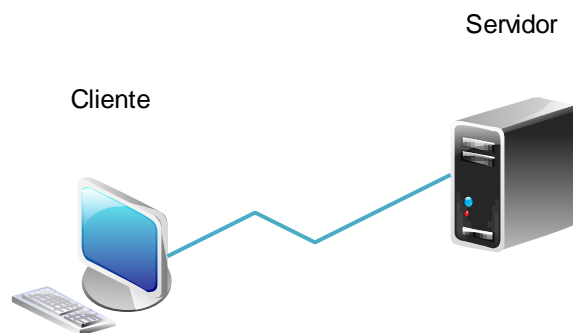
4.5.1. Arquitectura cliente – servidor

En general, las redes de comunicación LAN modernas utilizan la estructura de cliente-servidor. La arquitectura cliente-servidor se basa en la asignación de roles a las entidades que participan en la comunicación, donde un cliente pide información, y otro la sirve, denominado servidor.

4.5.1.1. Esquema

En el modelo cliente-servidor, el cliente envía un mensaje solicitando un determinado servicio a un servidor, es decir, hace una petición, y el servidor envía uno o varios mensajes con la respuesta, cumpliendo con proveer el servicio.

Figura 21. **Estructura cliente-servidor**



Fuente: elaboración propia.

En un sistema distribuido cada máquina puede cumplir el rol de servidor para algunas tareas y el rol de cliente para otras. Es más, dentro de una misma máquina pueden existir procesos que sean clientes y otros que sean servidores, interactuando entre ellos, dependiendo de si piden información o si la otorgan.

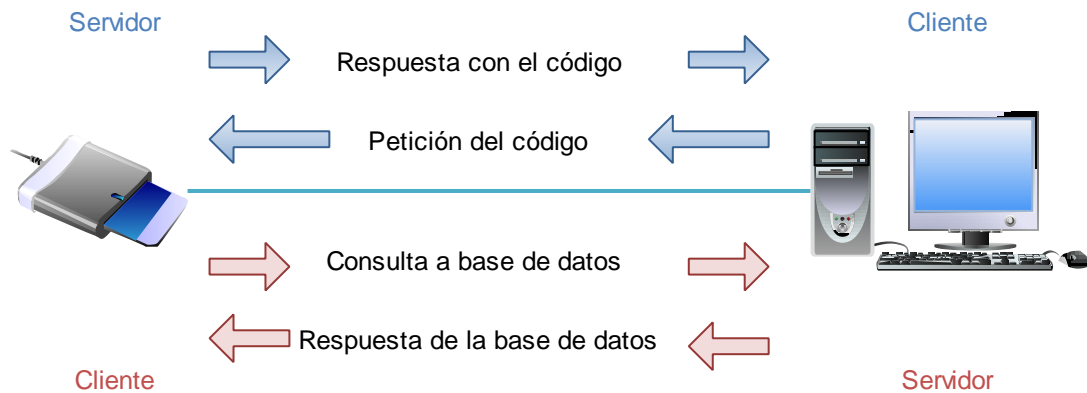
4.5.1.2. Interacción de servicios

Para el desarrollo de este trabajo se especifica la arquitectura cliente-servidor para el intercambio de información entre la estación lectora de etiquetas de radiofrecuencia y la computadora. Incluso, se aplica la arquitectura cliente-servidor entre los procesos de páginas *web* y base de datos que interactúan con la interfaz de administración *web*.

La interacción de servicios necesarios, para el correcto funcionamiento de los procesos requeridos por el presente trabajo, se describe a continuación:

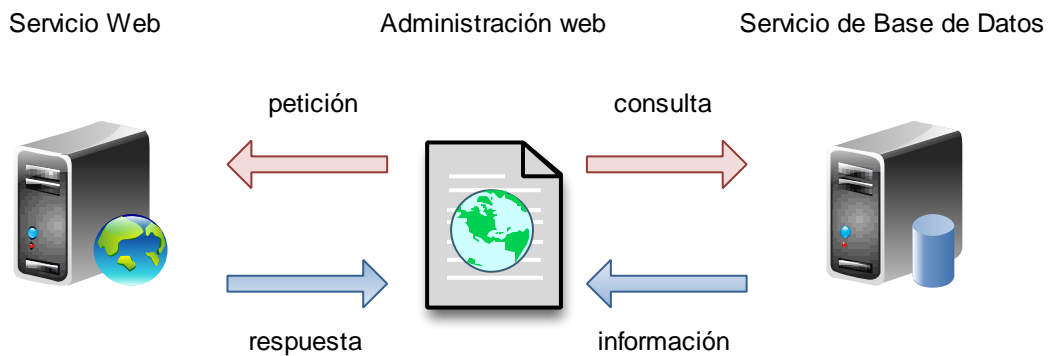
- Comunicación entre el lector y la computadora. En el intercambio del código de la etiqueta de radiofrecuencia, la estación lectora adquiere el rol de servidor y espera la petición de la computadora, que adquiere el rol de cliente. En el una consulta a base de datos, los roles se invierten. La figura 22 presenta una idea de la interacción mencionada.
- Interacción del servidor *web* y base de datos con la interfaz de administración *web*. El proceso cliente y el proceso servidor se ejecutan sobre la computadora. Los programas de servidor *web* Apache y de base de datos MySQL adquieren los roles de servidores dado que proveen información, por una parte la pagina *web* y por otro datos almacenados, ante la petición de la interfaz de administración *web* para desplegarlos al usuario. La figura 23 presenta un bosquejo de esta interacción.

Figura 22. **Servicio entre el lector y la computadora**



Fuente: elaboración propia.

Figura 23. **Servicios web y base de datos**



Fuente: elaboración propia.

4.5.2. Modelo de red OSI y modelo TCP/IP

La comunicación en red requiere de estándares que permitan alcanzar la interoperabilidad entre diversos sistemas. La Organización Internacional de Normalización (ISO) desarrolló un modelo conceptual para la conexión en red

denominado *Open Systems Interconnection Reference Model*, o bien modelo de referencia de interconexión de sistemas abiertos, en resumen modelo de referencia OSI.

El modelo de referencia OSI divide la comunicación de datos, a través de una red, en siete niveles o capas, donde cada capa se encarga de ejecutar una determinada parte del proceso global. Cada capa agrega *bits* de encabezado a los datos que pasan a través ella, para que pueda ser identificado por la respectiva capa en el sistema receptor.

El modelo OSI proporciona las normas básicas para una serie de procesos distintos de conexión en red:

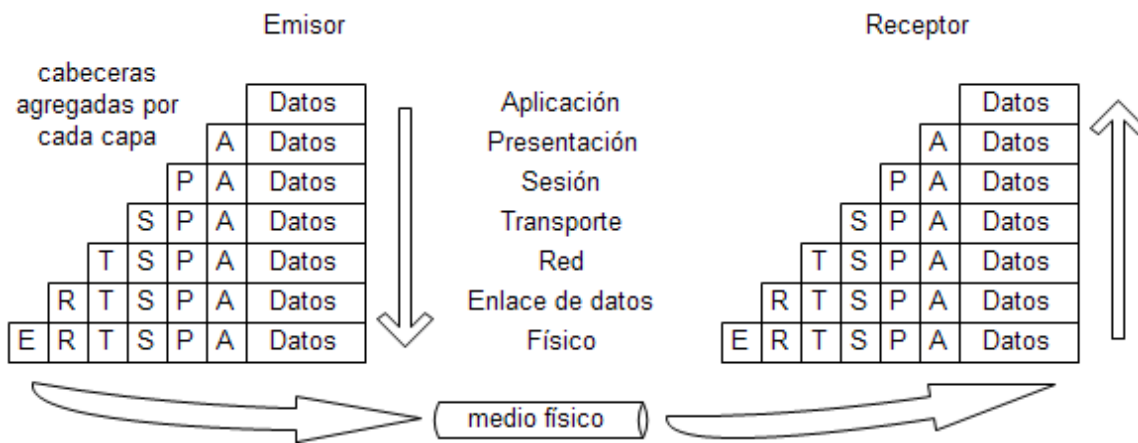
- El modo en que los datos se traducen a un formato apropiado para la arquitectura de red que se utiliza. Distintos datos de información requieren distintos formatos para ser transmitidos.
- El modo en que las computadoras y otros dispositivos de red se comunican. El mecanismo que proporcione un canal de comunicación entre el remitente y el destinatario.
- El modo en que los datos se transmiten entre los distintos dispositivos y la forma en que se resuelve la secuencia de paquetes y comprobación de errores. El conjunto de reglas que controlen la forma en que los datos se intercambian.
- El modo en que los paquetes son direccionados, tanto en nivel lógico como en nivel físico, para la correcta comunicación.

Las capas que define el modelo de referencia OSI, para la conexión de red son las siguientes:

- Capa 1 – capa física. Define la transmisión de *bits* a través de un medio. Presenta las especificaciones de instalación y mantenimiento de las conexiones físicas, y las especificaciones mecánicas y eléctricas para usar el medio de transmisión.
- Capa 2 – capa de enlace de datos. Organiza los *bits* en estructuras de información denominados tramas. Dirige la información al receptor usando la dirección asociada al *hardware*. Crea la información de seguridad y realiza el análisis de errores resultados de la transmisión.
- Capa 3 – capa de red. Establece la comunicación y determina el camino que tomarán los datos en la red, basadas en direcciones lógicas.
- Capa 4 – capa de transporte. Segmenta los datos originados en el emisor. También se encarga de asegurar que el receptor reciba exactamente la misma información que ha querido enviar el emisor. Multiplexa y demultiplexa segmentos de mensajes de distintas aplicaciones.
- Capa 5 – capa de sesión. Establece, administra y finaliza la comunicación entre las aplicaciones del emisor y el receptor. Permite a un mismo usuario, realizar y mantener diferentes conexiones a la vez.
- Capa 6 – capa de presentación. Regula la sintaxis y la semántica de la presentación de datos. Se encarga de encriptar y desencriptar los datos.

- Capa 7 – capa de aplicación. Proporciona la interfaz y servicios que soportan las aplicaciones de usuario.

Figura 24. **Comunicación según el modelo OSI**



Fuente: elaboración propia.

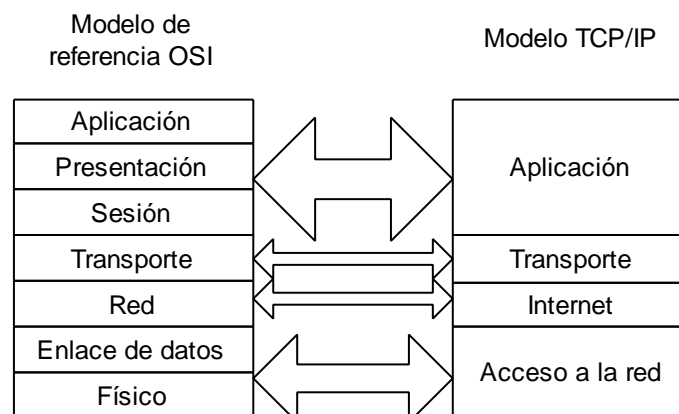
La figura 24 muestra el esquema teórico de comunicación del modelo OSI a través de las distintas capas de dos dispositivos conectados por medio de canal o medio físico.

El modelo TCP/IP es un conjunto de protocolos encaminados que pueden ejecutarse en distintas plataformas de *software* y casi todos los sistemas operativos de red lo soportan como protocolo de red predeterminado. Las siglas TCP significan *Transmisión Control Protocol* y las siglas IP significan *Internet Protocol*.

El modelo TCP/IP es un modelo aplicado al entorno de redes, y se constituye por cuatro capas que se relacionan con las capas del modelo de referencia OSI. Las capas del modelo TCP/IP son las siguientes:

- Capa 1 – capa de acceso a la red. Esta capa corresponde a la capa física, de transferencia de *bits*, y a la de enlace del modelo OSI, y realiza las funciones de formación de tramas, control y detección de errores.
- Capa 2 – capa de red. También llamada capa de Internet, se basa en el protocolo IP que define el direccionamiento lógico de los paquetes de datos. Esta capa encapsula los datos en unidades llamadas datagramas.
- Capa 3 – capa de transporte. Esta capa brinda una comunicación de principio a fin entre dos programas de aplicación. Se basa en el protocolo TCP, para transmisiones orientadas a la conexión, o bien en el protocolo UDP, para transmisiones no orientadas a la conexión.
- Capa 4 – capa de aplicación. Esta capa corresponde a las capas 5 al 7 en el modelo OSI, y se ocupa de las funciones de sintaxis, formato y encriptación de los datos.

Figura 25. **Comparación de modelos de red**



Fuente: elaboración propia.

En las siguientes secciones se describe en mayor detalle las especificaciones y protocolos utilizados en el desarrollo del presente trabajo.

4.5.3. Estándar *Ethernet* de la capa de acceso a la red

El comité de la IEEE 802,3 definió el estándar *Ethernet*, especificando el cableado a usar y el máximo de distancia alcanzable en este tipo de red. El estándar describe una LAN un método de acceso al medio llamado CSMA/CD.

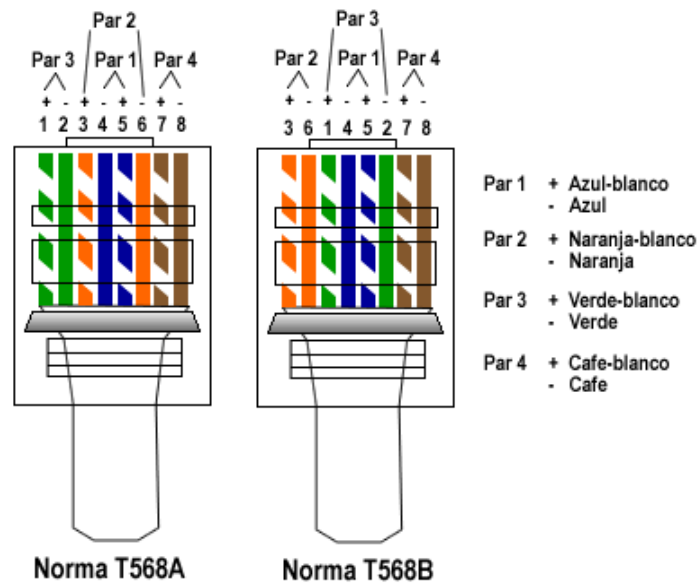
La velocidad de transmisión de datos en *Ethernet* es de 10 Mbps en las configuraciones habituales pudiendo llegar a ser de 100 Mbps en las especificaciones *Fast Ethernet*.

Al principio, sólo se utilizaba cable coaxial con una topología en bus, sin embargo se ha cambiado y ahora se utilizan nuevas tecnologías como el cable de par trenzado (10 Base-T), fibra óptica (10 Base-FL) y las conexiones a 100 Mbits/s (100 Base-X o *Fast Ethernet*).

Para el presente trabajo se utiliza el cable de par trenzado, denominado 10 Base-T, que consta de 4 pares de cables. En cada pareja van trenzados entre sí un cable de color y un cable blanco marcado con el mismo color. Los colores que se usan habitualmente son el naranja, el verde, el azul y el café. En las terminales del cable de par trenzado se utilizan conectores RJ45. Este cable es capaz de transmitir a 10 Mbps.

El cable de par trenzado, en configuración de cable cruzado, se rige por dos tipos de normas o configuraciones. Éstas son: La EIA/TIA-568A (T568A) y la EIA/TIA-568B (T568B). La diferencia entre ellas es el orden de los colores de los pares a seguir para el conector RJ45.

Figura 26. **Conector RJ45 en conexión cruzada**



Fuente: http://media.wilkinsonpc.com.co/free/img/cable-de-red-normas-t568a-t568b_1.gif.

Consulta: 28 de junio de 2012.

Para una conexión directa entre dos máquinas, se debe utilizar un cable cruzado que, en vez de conectar hilo a hilo, cruza entre sí las señales RX y TX cambiando los cables verdes por los cables naranjas.

Por otra parte, para lograr el enlace entre los dispositivos de una red *Ethernet* es necesaria la integración de una tarjeta de red, o NIC (*Network Card Interface*), en los dispositivos, con la cual se puedan enviar y recibir paquetes de datos, empleando un protocolo para su comunicación y convirtiendo a esos datos a un formato de *bits*.

A cada tarjeta de red le es asignado un identificador único por su fabricante, conocido como dirección MAC (*Media Access Control*), que consta

de 48 *bits*. Dicho identificador permite direccionar el tráfico de datos de la red del emisor al receptor adecuado.

Ethernet está diseñado de manera que no se puede transmitir más de una información a la vez. El objetivo es que no se pierda ninguna información, y se controla con un sistema conocido como CSMA/CD (*Carrier Sense Multiple Access with Collision Detection*), es decir, detección de portadora con acceso múltiple y detección de colisiones, cuyo principio de funcionamiento consiste en que una estación, para transmitir, debe detectar la presencia de una señal portadora y, si existe, comienza a transmitir.

Dentro de la capa de acceso a red opera el protocolo ARP (*Address Resolution Protocol*), que se encarga de asociar direcciones IP con direcciones físicas MAC que utiliza *Ethernet*.

4.5.4. Protocolo IP de capa de red

IP, *Internet Protocol*, es un protocolo no orientado a la conexión, con mensajes de un tamaño fijo. Este protocolo recibe información del nivel superior y le añade la información necesaria para su gestión, formando bloques de datos llamados datagramas.

Cada datagrama se gestiona de forma independiente, por lo que dos datagramas pueden utilizar diferentes caminos para llegar al mismo destino, provocando que lleguen en diferente orden o bien duplicados. El protocolo IP no corrige los anteriores problemas, ni tampoco informa de ellos, por lo que una capa superior debe encargarse de dicha tarea.

El protocolo IP utiliza direcciones IP para encaminar los datagramas entre dos dispositivos de una red. Una dirección IP es una etiqueta numérica que identifica, de manera lógica y jerárquica, a una interfaz de red de un dispositivo dentro de una LAN.

La dirección IP de un dispositivo no es exclusiva, es decir, puede cambiar a menudo por cambios en la red o porque el dispositivo encargado dentro de la red de asignar las direcciones IP, decida asignar otra IP al dispositivo. También puede especificarse a un dispositivo que permanezca con una dirección IP fija sin que ésta sea modificada por las condiciones de la red.

Existen varias versiones del protocolo IP. IPv4 es en la actualidad la más empleada, aunque el crecimiento en el tamaño de las redes compromete cada vez más su capacidad. El número de equipos que IPv4 puede direccionar comienza a ser insuficiente. Para solucionar esta situación se ha desarrollado la versión IPv6, con una capacidad de direccionamiento muy superior a IPv4.

Las direcciones IPv4 se expresan por un número binario de 32 *bits* permitiendo un espacio de direcciones de hasta 4 294 967 296 direcciones posibles. Las direcciones IP se expresan como números de notación decimal, separando los 32 *bits* de la dirección en cuatro octetos. El valor decimal de cada octeto está comprendido en el rango de 0 a 255.

Las direcciones IP se dividen en dos secciones de *bits*, de longitud variable, siendo la primera sección de *bits* la que define la dirección de red, mientras que los *bits* restantes, llamados *bits* de *host*, identifican y distinguen a un dispositivo dentro de la red establecida.

La máscara permite distinguir los *bits* que identifican la red y los que identifican el *host* de una dirección IP. La máscara se forma poniendo a 1 los *bits* que identifican la red y a 0 los *bits* que identifican el *host*.

Los dispositivos con direcciones IP pertenecientes al mismo segmento de red y conectados físicamente en una LAN son capaces de intercambiar datos entre sí. Pero cuando uno de los dispositivos pertenece a otro segmento de red, la única manera en que pueden comunicarse es por intermedio de un enrutador de redes (*router*), que encamina los datos por las conexiones necesarias logrando una comunicación efectiva.

En la pequeña red formada por la estación lectora y la computadora, se especifican direcciones IP fijas para cada uno, las cuales pertenecen al mismo segmento de red para evitar así la necesidad de otro dispositivo enrutador.

4.5.5. Protocolos de la capa de transporte

El objetivo de la capa transporte es proporcionar un servicio eficiente y confiable a sus usuarios, que normalmente son procesos de la capa aplicación. Para lograr este objetivo, la capa transporte utiliza los servicios proporcionados por la capa de red.

Hay dos tipos de servicio en la capa transporte, el orientado y el no orientado a la conexión. En el servicio orientado a la conexión consta de tres partes: establecimiento, transferencia de datos, y liberación. En el servicio no orientado a la conexión se tratan los paquetes de forma individual. El protocolo no orientado a la conexión es el UDP y el orientado es el TCP.

La capa de transporte es responsable de hacer llegar los datos a las aplicaciones que los requieren en las capas superiores. Para ello se asocia cada aplicación a un número de 16 *bits* al que se denomina número de puerto.

Cuando un proceso desea establecer una conexión con un proceso de aplicación remoto, debe especificar a cuál número de puerto se conectará. El método que normalmente se emplea es definir direcciones de transporte en las que los procesos pueden estar a la escucha de solicitudes de conexión.

4.5.5.1. Protocolo TCP

TCP es el protocolo que proporciona un transporte fiable de flujo de *bits* entre aplicaciones. Está diseñado para poder enviar grandes cantidades de información de forma fiable, liberando al programador de aplicaciones de la dificultad de gestionar la fiabilidad de la conexión, es decir los procesos involucrados en retransmisiones, pérdidas de paquete, orden en que llegan los paquetes ,duplicados de paquetes, y otros que gestiona el propio protocolo.

Pero la complejidad de la gestión de la fiabilidad tiene un coste en eficiencia, ya que para llevar a cabo las gestiones anteriores se tiene que añadir bastante información a los paquetes a enviar. Debido a que los paquetes a enviar tienen un tamaño máximo, cuanto más información añade el protocolo para su gestión, menos información que proviene de la aplicación podrá contener ese paquete.

El protocolo TCP necesita que se establezca una conexión entre los equipos situados en ambos extremos de la misma. Antes de iniciar la transferencia de datos TCP efectúa una negociación entre los dos equipos basada en el intercambio de tres segmentos de datos.

4.5.5.2. Protocolo UDP

UDP es un protocolo que proporciona un nivel de transporte no fiable de datagramas, ya que apenas añade información al paquete que envía al nivel inferior, solo la necesaria para la comunicación extremo a extremo.

Ciertas aplicaciones prefieren utilizar en la capa de transporte el protocolo UDP aunque éste no haga corrección ni detección de errores. Como estas aplicaciones necesitan transmitir pequeñas cantidades de datos, resulta más eficaz reenviar los datagramas defectuosos que no sobrecargar cada uno con información de control en la cabecera.

4.5.6. Funciones de la capa de aplicación

Ésta es la capa superior dentro de la estructura jerárquica del protocolo TCP/IP e incluye las aplicaciones y procesos con los que intercambia datos la capa de transporte. TCP/IP tiene en esta capa protocolos que soportan servicios de conexión remota, correo electrónico y transferencia de archivos.

La capa de aplicación, de la computadora con el servicio de base de datos del sistema de identificación por radiofrecuencia, contiene el programa y proceso de consulta de la información almacenada, y el proceso que gobierna la comunicación con la estación lectora de etiquetas de radiofrecuencia, utilizando el control de *sockets*.

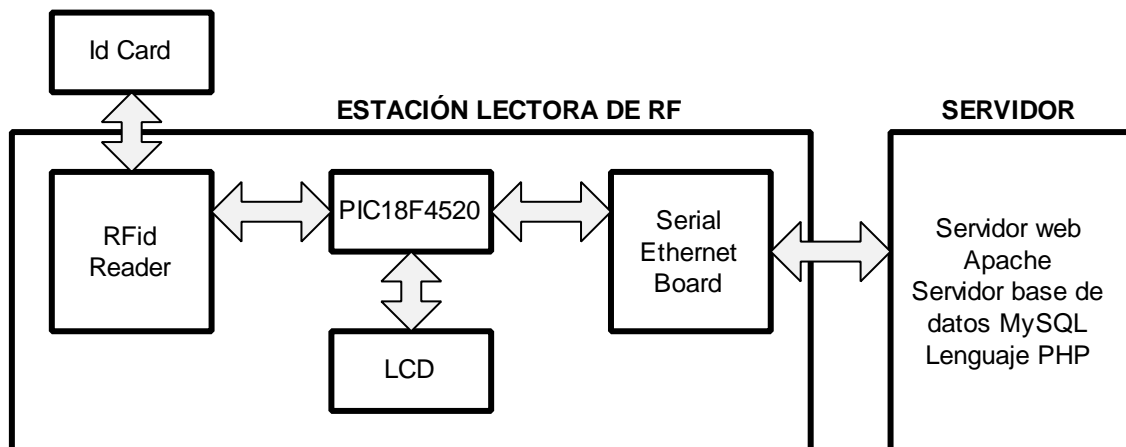
5. DISEÑO Y DESARROLLO DEL SIDERF

En el presente capítulo se describe de forma puntual el diseño y desarrollo del sistema de identificación estudiantil por radiofrecuencia, denominado SIDERF para posterior referencia, presentando los diagramas de conexión de los módulos, los algoritmos de programación, la configuración de la red y por último se explica la forma de operación del sistema y las facilidades que ofrece la administración *web* básica.

5.1. Esquema de bloques del SIDERF

El SIDERF se constituye físicamente por el conjunto formado por la estación lectora de tarjetas de radiofrecuencia, la computadora con los servicios de base de datos y la conexión física que los comunica.

Figura 27. Diagrama de bloques del SIDERF



Fuente: elaboración propia.

La figura 27 muestra un bosquejo de la conexión de los diferentes módulos periféricos, el microcontrolador y la computadora a través de un diagrama de bloques.

5.1.1. Capacidades del SIDERF

El SIDERF posee una variedad de capacidades orientadas al control de acceso estudiantil al Laboratorio de Electrónica. A continuación se enumeran las capacidades y funciones del sistema:

- El sistema interactúa con los estudiantes que posean una tarjeta de radiofrecuencia compatible y registrada en la base de datos del Laboratorio de Electrónica. Dicha interacción se realiza mediante el despliegue de instrucciones en la pantalla LCD de la estación lectora.
- La estación lectora lee el código de una sola tarjeta por vez y únicamente dentro del rango de 7 cm de la antena, con ello se logra un mejor control de la persona que tiene acceso al laboratorio.
- El sistema es capaz de acoplarse sobre una infraestructura de red Ethernet previamente instalada, puesto que tanto la computadora como la estación lectora trabajan como dispositivos independientes en la red, con la única condición que sean conectadas en el mismo segmento de direcciones IP. Por otra parte, en el Laboratorio de Electrónica se instalará una red dedicada y privada entre la computadora y la estación lectora, por temas de seguridad de los datos transmitidos.
- La estación lectora realiza consultas a la base de datos para comprobar la validez del código de la tarjeta que ha leído.

- La estación lectora es capaz de generar una señal de activación para una cerradura eléctrica acoplada a la puerta de acceso.
- La administración *web* básica de la base de datos permite registrar nuevos códigos de tarjetas, realizar consultas de toda la base de datos, realizar búsquedas de registros por número de tarjeta o bien por número de carné estudiantil y también puede presentar en pantalla la información y fotografía del último estudiante que accedió al laboratorio.
- Para una administración más avanzada de las bases de datos se tiene el *software* de PhpMyAdmin, que permite crear, modificar o borrar elementos, campos e información de las bases de datos. También es capaz de exportar la información de las bases de datos a hojas de cálculo.

El sistema puede ser expandido en capacidades gracias a la flexibilidad que ofrece la programación del código del microcontrolador y la programación de los *scripts* de PHP en la computadora.

5.2. Diagrama de conexiones del microcontrolador PIC18F4520

Cada módulo periférico es independiente físicamente y posee únicamente una serie de pines para conectarse y comunicarse con el microcontrolador, por ello la placa electrónica del microcontrolador debe ser diseñada para que cada conexión de puerto sea compatible con los conectores de cada módulo.

5.2.1. Circuito de alimentación

La placa electrónica diseñada para conectar el microcontrolador tiene implementado un circuito regulador de voltaje de 5 voltios fijos, con el fin de mantener un suministro de energía estable. Dicha fuente de alimentación también suministra energía a cada uno de los módulos periféricos adicionales que se conectan con el microcontrolador.

El circuito de alimentación consta de un puente rectificador de señal de corriente alterna a corriente directa, un dispositivo regulador 7805 y capacitores reductores de rizo.

A la entrada del circuito de alimentación se debe conectar un transformador reductor de voltaje, preferentemente que proporcione un nivel de tensión entre 9 a 12 voltios y con capacidad de una corriente máxima igual o superior a 1,5 amperios.

5.2.2. Conexión de los módulos

Los módulos periféricos se conectan a la placa del microcontrolador por medio de cables y conectores IDC10. Cada módulo se conecta a un puerto específico del microcontrolador, en el cual se realiza la interacción de señales eléctricas para el intercambio de información y el control de los procesos.

Las siguientes tablas describen la conexión de los pines entre cada módulo periférico y el correspondiente puerto del PIC y su la función que realiza.

Tabla III. **Conexión PIC y *RFid Reader***

| PIC | Módulo | Función (sobre el pin del PIC) |
|------------|---------------|---|
| RB0 | OUT | Recepción del código de la <i>RFid Card</i> . |
| RB1 | RDY/CLK | Recepción de la señal de reloj de sincronización. |
| RB2 | SHD | Control de actividad o inactividad del módulo. |
| RB3 | MOD | Activación de emisión de ondas electromagnéticas. |
| VSS | VCC | Conexión de voltaje positivo de la fuente. |
| VDD | GND | Referencia del voltaje de la fuente. |

Fuente: elaboración propia.

Tabla IV. **Conexión PIC y *Serial Ethernet Board***

| PIC | Módulo | Función (sobre el pin del PIC) |
|------------|---------------|--|
| RC0 | RST | Control del reinicio del módulo. |
| RC1 | CS | Selección del dispositivo esclavo a comunicarse. |
| RC3 | SCK | Generación de la señal de reloj de sincronización. |
| RC4 | MISO | Recepción de datos desde el módulo. |
| RC5 | MOSI | Transmisión de datos hacia el módulo. |
| VSS | VCC | Conexión de voltaje positivo de la fuente. |
| VDD | GND | Referencia del voltaje de la fuente. |

Fuente: elaboración propia.

Tabla V. **Conexión PIC y el módulo LCD 2x16**

| PIC | Módulo | Función (sobre el pin del PIC) |
|------------|---------------|---------------------------------------|
| RD0 – RD3 | D4 – D7 | Transmisión de datos. |

Continuación de la tabla V.

| | | |
|-----|-----|---|
| RD4 | RS | Selección del registro control o registro de datos. |
| RD5 | E | Habilitación de la operación del módulo. |
| VSS | VCC | Conexión de voltaje positivo de la fuente. |
| VDD | GND | Referencia del voltaje de la fuente. |

Fuente: elaboración propia.

5.2.3. Otros componentes

Para el efectivo funcionamiento del microcontrolador PIC se requiere de un cristal oscilador que proporcione una señal de reloj que sincronice las operaciones de la CPU.

Por la necesidad de alta frecuencia de la señal del oscilador, para realizar la lectura del código de las tarjetas de radiofrecuencia, se utiliza un cristal oscilador de 8 MHz de frecuencia.

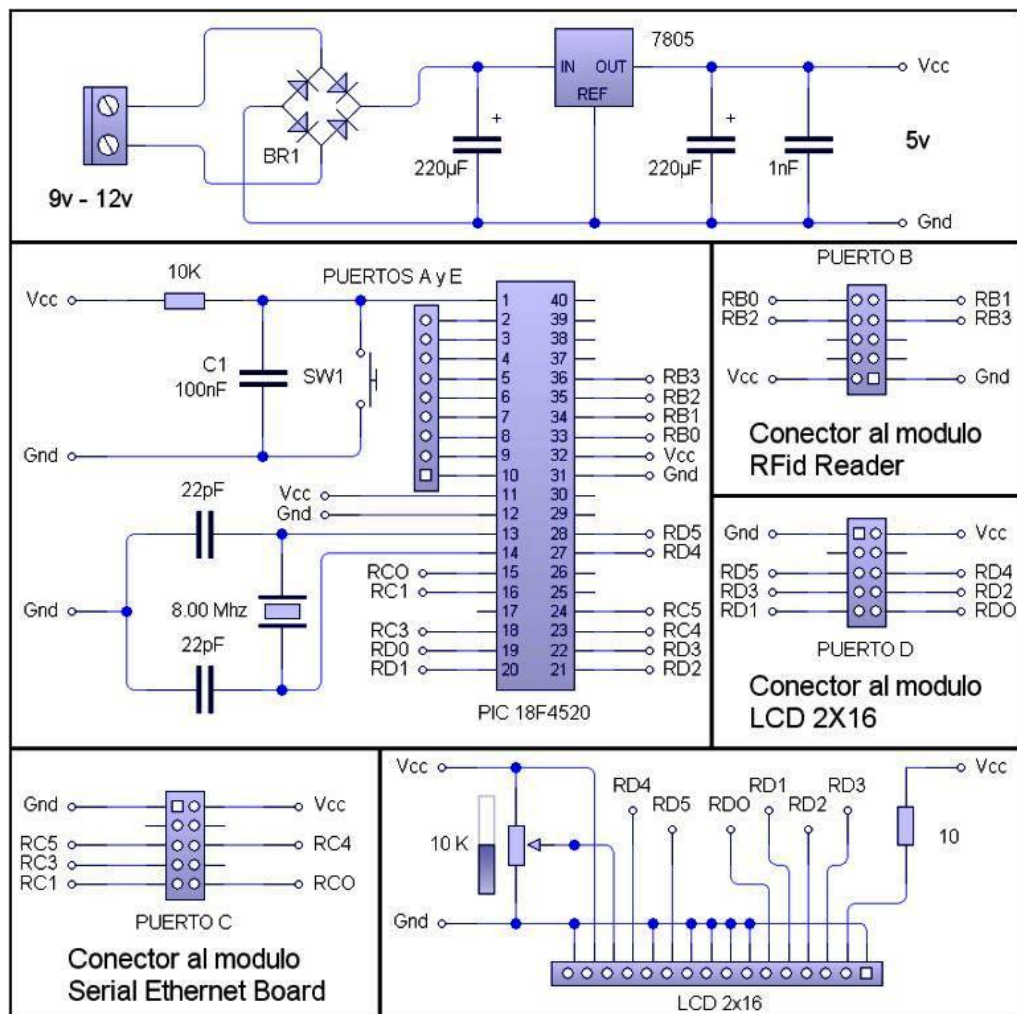
En la programación del microcontrolador se especifica la operación en modo HS del circuito oscilador y la utilización del multiplicador de frecuencia PLL para aumentar 4 veces la frecuencia, es decir, que el microcontrolador operará con una frecuencia de 32 MHz.

Otros componentes, incluidos en el circuito del microcontrolador, son los que conforman un circuito de reinicio del PIC. Para dicha función se utiliza un arreglo de una resistencia y un interruptor pulsador que alimentan con un nivel de tensión a la patilla del reinicio maestro (MCLR) del PIC.

5.2.4. Diseño del circuito impreso en placa

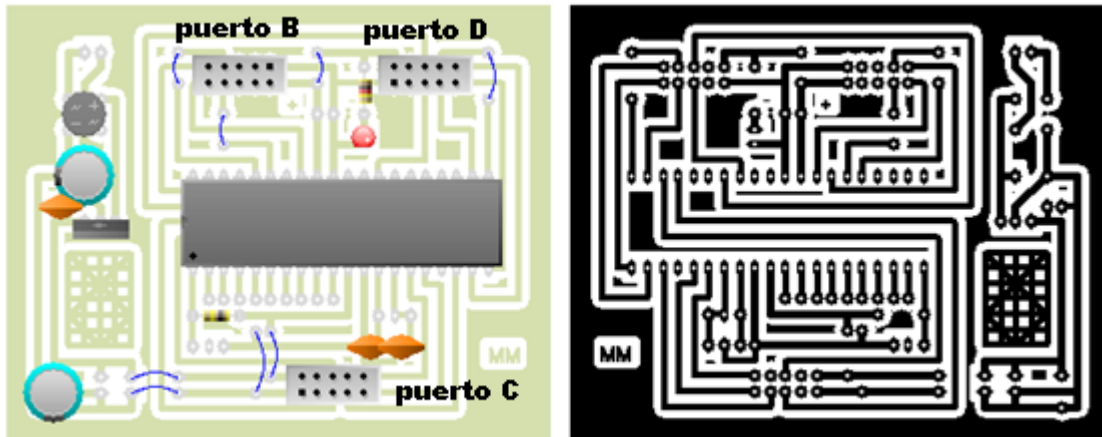
La figura 28 muestra el diagrama circuital de la conexión del microcontrolador PIC y sus circuitos auxiliares. La figura 29 muestra el diagrama del circuito impreso de la placa del microcontrolador PIC18F4520 y la composición física de los componentes instalados.

Figura 28. Diagrama circuital de la placa del PIC



Fuente: elaboración propia.

Figura 29. Diagrama del circuito impreso de la placa del PIC



Fuente: elaboración propia.

5.3. Programa del microcontrolador PIC18F4520

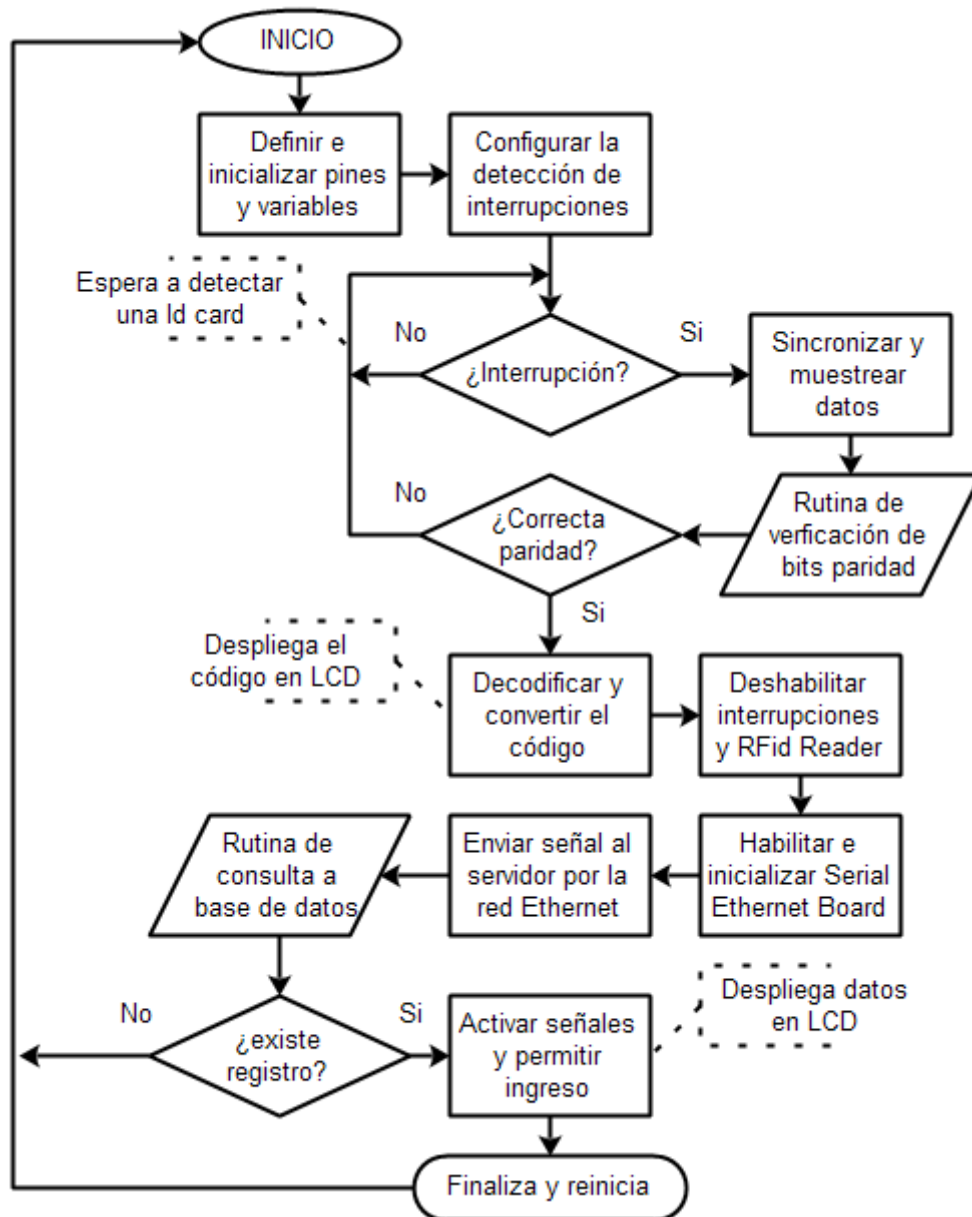
El microcontrolador es el encargado de operar las funciones de la estación lectora, realizando las acciones de lectura del código de la tarjeta de radiofrecuencia, la decodificación y extracción de la información y el envío de paquetes por la red. Todas estas operaciones se traducen a instrucciones en el código del programa que se carga en el microcontrolador.

El código del programa que gobierna las acciones del PIC18F4520 se escribe en lenguaje *Basic*. Haciendo uso del programa *MikroBasic* y las librerías de funciones se tiene gran flexibilidad en la programación y facilidades en el manejo de los módulos periféricos.

5.3.1. Diagrama de flujo

La figura 30 muestra el diagrama de flujo de operaciones del PIC.

Figura 30. Diagrama de flujo del programa del PIC



Fuente: elaboración propia.

5.3.2. Algoritmo del código del programa

El código del programa del PIC consiste en la siguiente secuencia de pasos:

- Definir los pines de puerto que serán utilizados para la comunicación con los módulos externos.
- Definir e inicializar variables de almacenamiento
- Configurar la detección de interrupciones para las señales de reloj y datos provenientes del módulo *RFid Reader*.
- Esperar hasta que se detecte las señales de reloj y datos
- Sincronizar y muestrear los *bits* del código de la tarjeta
- Verificar la autenticidad del código corroborando los *bits* de control de paridad. Una respuesta negativa reinicia la detección de interrupciones. Una respuesta positiva continúa con el proceso.
- Decodificar el número de tarjeta
- Desplegar el número de tarjeta en la pantalla LCD
- Deshabilitar el funcionamiento de detección de interrupciones y del módulo *RFid Reader*.
- Habilitar el funcionamiento del módulo *Serial Ethernet Board*
- Inicializar los parámetros de red del módulo *Serial Ethernet Board*
- Enviar una señal de aviso al servidor por la red *Ethernet*
- Enviar número de tarjeta al servidor para corroborar si existe registro en la base de datos. De no existir registro, se despliega que la tarjeta es desconocida y reinicia el proceso completo de detección de tarjeta.

- Si el registro de la tarjeta existe, en la base de datos, se procede a recibir y desplegar la información de número de carné y nombre asociados.
- Permitir el acceso al laboratorio
- Finalmente se reinicia el proceso del PIC

5.3.3. Descripción de comandos y librerías

A continuación se definen y explican los comandos más importantes en el código del programa del PIC.

- Dim. Sentencia para declarar variables
- Const. Sentencia para declarar constantes
- Sub procedure. Sentencia que declara la sección de instrucciones de un procedimiento.
- Sub function. Sentencia que declara la sección de instrucciones de una función.
- If. Sentencia condicional de toma de decisión básica
- While. Sentencia condicional de bucle
- For. Sentencia condicional de repetición
- Delay_ms. Sentencia de espera de tiempo de la ejecución del programa
- Dec. Sentencia que decrementa el valor de una variable en una unidad
- Inc. Sentencia que incrementa el valor de una variable en una unidad
- Reset. Sentencia que reinicia la ejecución del programa

De la librería de funciones de conversión se utiliza el siguiente comando:

- LongWordToStrWithZeros. Sentencia que convierte el valor numérico de una variable de 32 *bits* de longitud en 10 caracteres de texto.

De la librería de funciones para la operación de la pantalla LCD 2x16 se utilizan los siguientes comandos:

- Lcd_Init. Sentencia que inicializa el control del módulo LCD
- Lcd_Cmd. Sentencia que envía un comando al registro de control de la pantalla LCD.
- Lcd_Chr. Sentencia que imprime un carácter en la pantalla LCD en una posición específica.

De la librería de funciones para la operación del módulo SPI del PIC y las funciones orientadas al estándar *Ethernet*, se utilizan los siguientes comandos:

- SPI1_Init. Sentencia que inicializa la configuración por defecto del módulo de comunicación SPI.
- SPI_Ethernet_Init. Sentencia que inicializa al controlador de red ENC28J60 del módulo *Serial Ethernet Board*.
- SPI_Ethernet_setUserHandlers. Sentencia que establece la sección por defecto de procedimientos de respuesta ante peticiones TCP y UDP.
- SPI_Ethernet_doPacket. Sentencia que procesa el recibimiento de paquetes de la red, y distingue el tipo de petición realizada.

- `SPI_Ethernet_sendUDP`. Sentencia que envía un paquete UDP por la red hacia otro dispositivo.

5.3.4. Parámetros de compilación en *MikroBasic*

Hasta el momento el código de programación está escrito en lenguaje *Basic*, pero para poder cargarlo a la memoria del microcontrolador se requiere convertirlo en un código hexadecimal. Para dicha tarea el *software MikroBasic* posee un compilador. Antes de compilar el código es necesario configurar los siguientes parámetros indispensables:

- Frecuencia del oscilador en el valor de 8 MHz
- Tipo de oscilador: `HS-PLL enabled freq=4xFosc1`
- Activar las librerías de funciones: `Conversions`, `Lcd`, `Lcd_Constants`, `SPI`, `SPI_Ethernet` y `String`.
- Depurar los errores de programación en el código

5.3.5. Grabación del programa

Para la carga del código ejecutable a la memoria del microcontrolador se utiliza la tarjeta entrenadora *EasyPIC*, que se conecta a la computadora con un cable de USB y le transmite las señales eléctricas con la información del código.

En la computadora se utiliza el *software PICflash* en el cual se configuran los parámetros de grabación del código, además de tener la capacidad de leer la información de la memoria del PIC, borrar dicha información y cargar un nuevo código.

La configuración de los parámetros en *PICflash* es la siguiente:

- Modo del oscilador: HS-PLL
- Temporizador *Watchdog: disabled*
- Protección del código: ninguno seleccionado
- *Master clear: enabled*

5.4. Configuración de la computadora

Como se mencionó en el capítulo anterior, la computadora utiliza el sistema operativo Ubuntu 11,10 *Oneiric Ocelot*, con el *software* de Apache, MySQL y PHP instalados.

5.4.1. Requerimientos

Las características del *hardware* de la computadora deben superar los requisitos mínimos, tanto del sistema operativo Ubuntu 11,10 como del *software* Apache, MySQL, PHP y PhpMyAdmin.

Tomando como base los requisitos mínimos de cada *software* a instalar, se recopila los siguientes requisitos mínimos para el SIDERF:

- Procesador de 32 *bits* o 64 *bits* a 1 GHz
- Memoria RAM de 512 MB
- Disco duro de 15 GB. Se requieren 4.5 GB para la instalación del sistema operativo, el doble de tamaño de memoria RAM para la partición de intercambio, 50 MB para la instalación de Apache, 50 MB para la instalación de MySQL y 60 MB para la instalación de PHP. El resto del

espacio se destina para el almacenamiento de información en las bases de datos.

- Tarjeta gráfica y monitor que soporte la resolución de 800x600
- Lector/Quemador de CD o DVD
- Puertos USB
- Tarjeta de red y conexión a servicio de Internet

5.4.2. Instalación del *software*

Lo primero a instalar es el sistema operativo Ubuntu 11,10 *Oneiric Ocelot*, el cual puede ser descargado gratuitamente desde la página *web* <http://releases.ubuntu.com/11.10/>. En dicha página *web* se puede descargar para sistemas de 32 *bits* o para 64 *bits*, dependiendo de las características de la computadora.

El sistema operativo debe ser grabado en un disco y luego arrancar la computadora iniciando con el disco como primera opción. Al cargar la interfaz gráfica, escoger la opción de instalar el sistema operativo en el disco duro y seguir los pasos que indique el asistente de la instalación. Para mayor detalle de la instalación estándar de Ubuntu 11,10, leer la documentación en el sitio *web* http://doc.ubuntu-es.org/11.10:Instalación_estándar.

Una vez terminada la instalación es necesario instalar las actualizaciones que para ello se requiere de conexión a Internet. Todas las siguientes instalaciones de programas se realizan desde una terminal de comandos de Ubuntu, por su facilidad y rapidez.

El primer programa a instalar es el servidor *web* Apache en versión 2. Dentro de una terminal de comandos escribir la siguiente instrucción: *sudo apt-get install apache2*, y al ejecutarla pedirá la confirmación de la contraseña de administrador del sistema operativo, que fue especificada en la instalación del mismo.

Desde Internet se descargan aproximadamente 5 MB del archivo instalador de Apache2, y al finalizar se tendrán instalado Apache2 y junto a los módulos y librerías básicas. Para comprobar la instalación, desde un navegador *web*, ir a la dirección <http://localhost/> donde aparece el texto *It works!*.

El segundo programa a instalar es el servidor de base de datos MySQL en versión 5. Dentro de una terminal de comandos escribir la siguiente instrucción: *sudo apt-get install mysql-server* y ejecutarla. Desde Internet se descargan aproximadamente 21 MB del archivo instalador.

En un punto de la instalación, se debe especificar el usuario y contraseña del servidor de base de datos. Como configuración de prueba de MySQL se utiliza el usuario *root* y la contraseña *rootmysql*.

El siguiente programa a instalar es el lenguaje PHP en versión 5 junto con las librerías de conexión a los servidores Apache2 y MySQL. Dentro de una terminal de comandos escribir la siguiente instrucción: *sudo apt-get install php5 libapache2-mod-php5 php5-mysql* y ejecutarla. Desde Internet se descargan aproximadamente 7 MB del archivo instalador.

Por último se instala el programa PhpMyAdmin en versión 4. Dentro de una terminal de comandos escribir la siguiente instrucción: *sudo apt-get install*

phpmyadmin y ejecutarla. Desde Internet se descargan aproximadamente 6 MB del archivo instalador.

Al final de la instalación se debe indicar que el servidor *web* a reconfigurar es Apache2, marcándolo con barra espaciadora. En el paso siguiente especificar que no se quiere configurar una base de datos.

Para facilitar la programación de *scripts* y de páginas *web* se sugiere instalar el programa Geany, desde el centro de *software* que tiene el sistema operativo Ubuntu. Con este programa se puede crear, abrir y guardar *scripts* de PHP como de HTML y otros lenguajes. También permite la depuración, compilación y ejecución de los *scripts*.

5.5. Conexión de la red microcontrolador – computadora

La estación lectora y la computadora se conectan a través de una red *Ethernet*, siendo configurados con direcciones IP estáticas y pertenecientes al mismo segmento de red. En la estación lectora, el microcontrolador PIC es el encargado de establecer la configuración específica en el controlador de red del módulo *Serial Ethernet Board*.

5.5.1. Configuración de la red

Como configuración de prueba para la red microcontrolador – computadora se utilizan los siguientes parámetros para la estación lectora:

- Dirección MAC: 00 1A 92 FA D1 BC
- Dirección IP: 172.10.100.25
- Máscara de subred: 255.255.255.0

Para la computadora se especifican únicamente los parámetros de dirección IP y máscara de subred, dado que la tarjeta de red ya posee una dirección MAC programada de fábrica. Los valores son:

- Dirección IP: 172.10.100.2
- Máscara de subred: 255.255.255.0

Para la comunicación por sockets entre ambos dispositivos se especifica el número de puerto 12345 para establecer la conexión.

5.5.2. Procedimiento de intercambio de datos

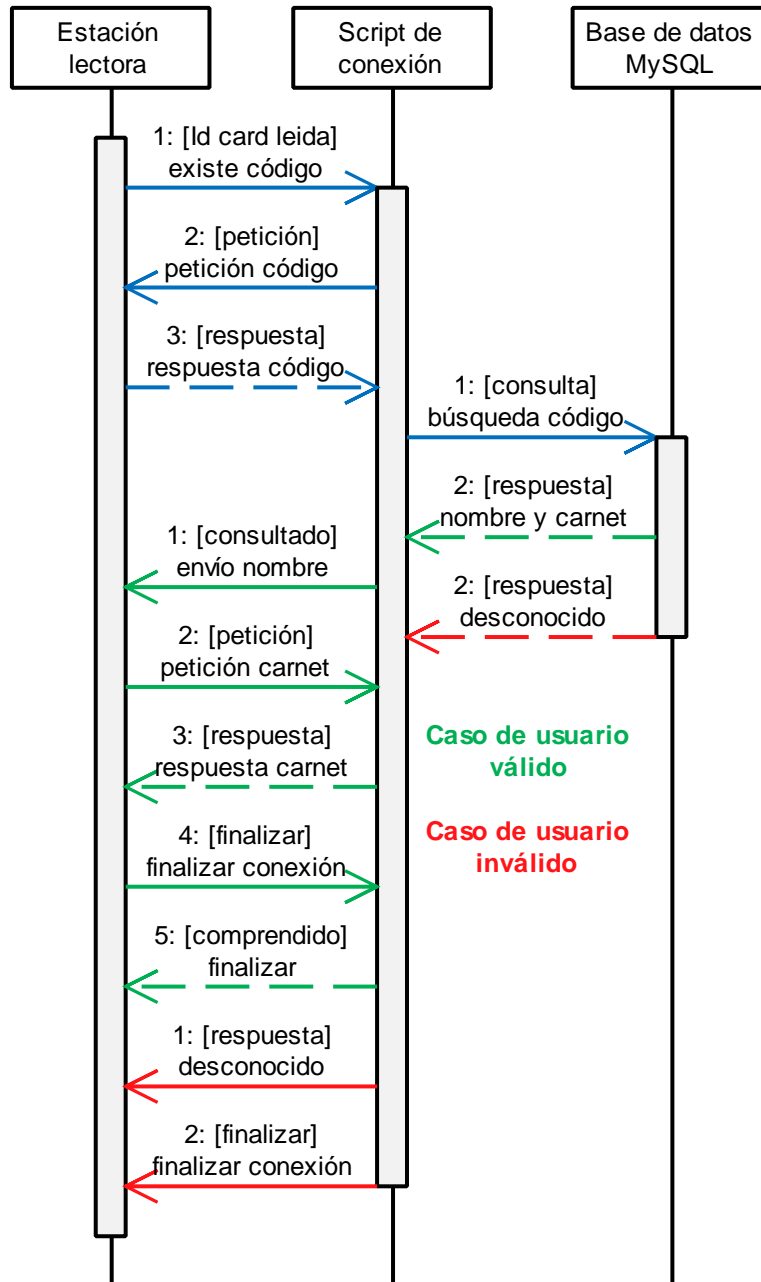
El microcontrolador PIC es programado para enviar una señal a la computadora, en anuncio de que se ha leído el código de una tarjeta de radiofrecuencia. Un *script* en la computadora se mantiene en escucha del puerto de comunicación para detectar la señal.

Cuando dicho *script* detecta la señal, se inicia una secuencia de intercambio de datos entre la computadora, la base de datos y el PIC hasta finalizar la conexión. La figura 31 presenta la secuencia del intercambio.

5.5.3. Script de comunicación de la computadora

El *script* de comunicación es escrito en lenguaje PHP y ejecutado en una terminal de comandos del sistema operativo, durante todo el tiempo de operación del sistema. Este programa es el encargado de realizar la conexión por *sockets* con la estación lectora, adquirir el número de la tarjeta de radiofrecuencia que haya sido leída y de realizar la consulta a la base de datos.

Figura 31. **Secuencia de intercambio de datos**



Fuente: elaboración propia.

Este *script* se integra tanto por funciones para el manejo de *sockets* de comunicación por red, así como de funciones para realizar consultas en la base de datos de MySQL.

5.5.4. Descripción de instrucciones

A continuación se describen los comandos más importantes en el código del *script* de comunicación.

- `#!/usr/bin/php -q`. Sentencia que define que el *scrip* de PHP es ejecutable en terminal de comandos.
- `Error_reporting(E_ALL)`. Sentencia que activa todas las notificaciones de error de PHP.
- `Set_time_limit(0)`. Sentencia que establece el tiempo de ejecución en infinito.
- `Echo`. Sentencia que muestra una o más cadenas de caracteres en la terminal.
- `Sleep()`. Sentencia que introduce una pausa en segundos de la ejecución del programa.
- `Strlen()`. Sentencia que obtiene el número de caracteres en una cadena
- `Substr()`. Sentencia que extrae caracteres dentro de una cadena
- `If`. Sentencia condicional de toma de decisión básica
- `While`. Sentencia condicional de bucle
- `Date()`. Sentencia que obtiene la fecha y hora del sistema

De la librería de funciones de operación de *sockets* se utilizan los siguientes comandos:

- `Socket_create()`. Sentencia que crea un *socket* de comunicación
- `Socket_bind()`. Sentencia que sujeta el *socket* a la escucha de paquetes de red a través de un puerto determinado.
- `Socket_recvfrom()`. Sentencia que espera a recibir información desde un *socket* a través de un puerto determinado.
- `Socket_sendto()`. Sentencia que envía información a un *socket* a través de un puerto determinado.
- `Socket_close()`. Sentencia que cierra un *socket* de comunicación

De la librería de funciones de conexión a base de datos MySQL se utilizan los siguientes comandos:

- `Mysql_connect()`. Sentencia que abre una conexión al servidor MySQL
- `Mysql_select_db()`. Sentencia que selecciona una base de datos MySQL
- `Mysql_query()`. Sentencia que envía una consulta MySQL
- `Mysql_num_rows()`. Sentencia que obtiene el número de filas de un conjunto de resultados de una consulta.
- `Mysql_free_result()`. Sentencia que libera la memoria del resultado
- `Mysql_fetch_array()`. Sentencia que obtiene los datos de una consulta en forma de un arreglo de filas y columnas.
- `Mysql_close()`. Sentencia que cierra la conexión con el servidor MySQL

5.6. Creación de la base de datos en MySQL

Para gestionar una base de datos dentro del servidor MySQL se utiliza el programa PhpMyAdmin. Este programa ofrece una interfaz gráfica para la estructuración y operación de bases de datos, tablas, campos e información dentro del servidor MySQL.

La interfaz gráfica de PhpMyAdmin se ejecuta desde un navegador *web*, ingresando a la dirección `http://localhost/phpmyadmin`. Para acceder al servidor MySQL se debe ingresar con el usuario y contraseña establecidos durante la instalación. Para este caso se utiliza el usuario *root* y la contraseña *rootmysql*.

Dentro de la sesión de PhpMyAdmin se elige la opción de crear nueva base de datos, en la cual se especifica el nombre de la base de datos, en este caso de prueba *LabRF*, y la opción de cotejamiento. El cotejamiento define el tipo de codificación de idioma usado en los datos almacenados en la base de datos. Para el idioma español se utiliza el cotejamiento *utf8_spanish_ci*.

Una base de datos está formada por una o varias tablas de datos. Las tablas son creadas dentro de la base de datos. Una tabla está formada por uno o varios campos de información.

5.6.1. Tablas y campos

En PhpMyAdmin se selecciona la base de datos y luego se elige la opción de crear nueva tabla, en la cual se especifica el nombre de la tabla y el número de columnas. Cada columna representa un campo de información. Los campos almacenan información relacionada entre sí.

Una fila de la tabla contiene los campos de información de un solo registro. Para cada campo se debe especificar los siguientes parámetros:

- Nombre del campo
- Tipo de información almacenada. Entre las opciones se tienen por ejemplo el tipo numérico entero (INT), tipo carácter de texto (VARCHAR), tipo fecha (DATE), tipo de valores concretos (ENUM) y otros más.
- Longitud y valores. Se establece la longitud máxima de caracteres o dígitos que tiene el dato, o bien la lista de valores concretos a utilizar.
- Predeterminado. Establece el valor de información por defecto del campo.
- Cotejamiento. Tipo de codificación de idioma del campo
- Atributos. Opciones para datos binarios y para fecha y hora
- Nulo. Establece si el campo acepta elementos vacíos o no
- Índice. Opción de no repetir valor, o valor único, o búsqueda avanzada
- Autoincrementar. Esta opción automáticamente aumentará en uno el valor del registro anterior en el mismo campo.
- Comentarios

5.6.1.1. Tabla Estudiantes

La primera tabla configurada para el SIDERF es la tabla que contiene los datos de los estudiantes activos a los cuales se les ha extendido una tarjeta de radiofrecuencia válida.

La tabla de datos contiene los siguientes campos y parámetros:

- Tarjeta. Contiene el número de tarjeta de radiofrecuencia, cuya longitud es de 8 caracteres.
- Carné. Contiene el número de carné del estudiante, cuya longitud es de 8 caracteres.
- Nombre. Contiene el nombre del estudiante, cuya longitud máxima es de 40 caracteres.
- Fecha. Contiene la fecha y hora en que ha sido ingresado el registro a la base de datos.
- Autor. Contiene el nombre de quien realizó el ingreso del registro a la tabla de datos.

Cada campo utiliza el tipo de dato VARCHAR, con cotejamiento *utf8_spanish_ci*, y no acepta valores vacíos. Además se configura el índice del campo tarjeta como valor único.

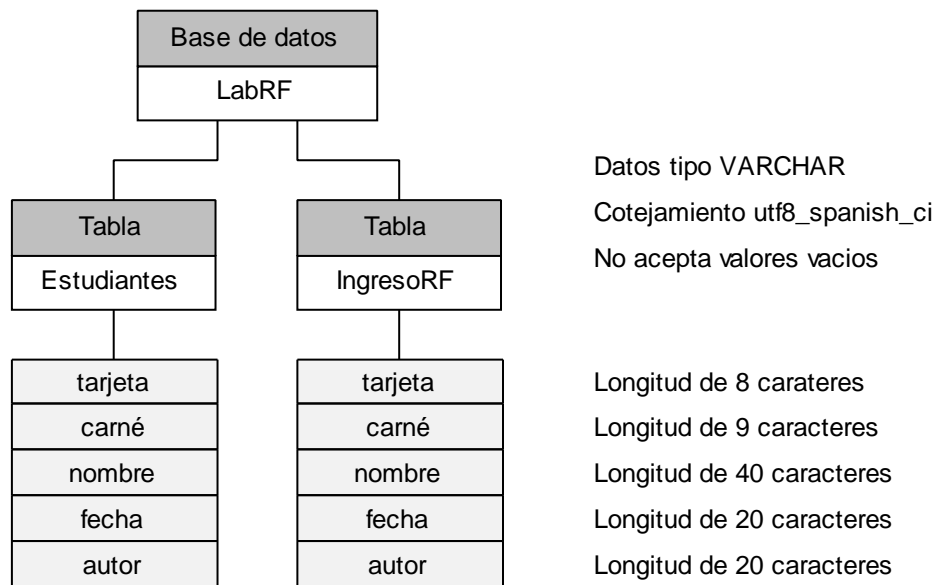
5.6.1.2. Tabla IngresoRF

La otra tabla configurada para el SIDERF es la tabla que contiene los registros de acceso al laboratorio, es decir, cada vez que un estudiante coloca su tarjeta en el lector e ingresa al laboratorio se registra su información en esta tabla.

Esta tabla contiene los mismos campos y configuración que la tabla anterior, con la única diferencia que en el campo de fecha se almacena la fecha y hora del ingreso.

La figura 32 muestra el resumen de la estructura de la base de datos mencionada anteriormente y configurada para el SIDERF para el almacenamiento de los registros de los estudiantes y los registros de ingreso al laboratorio.

Figura 32. Estructura de base de datos



Fuente: elaboración propia.

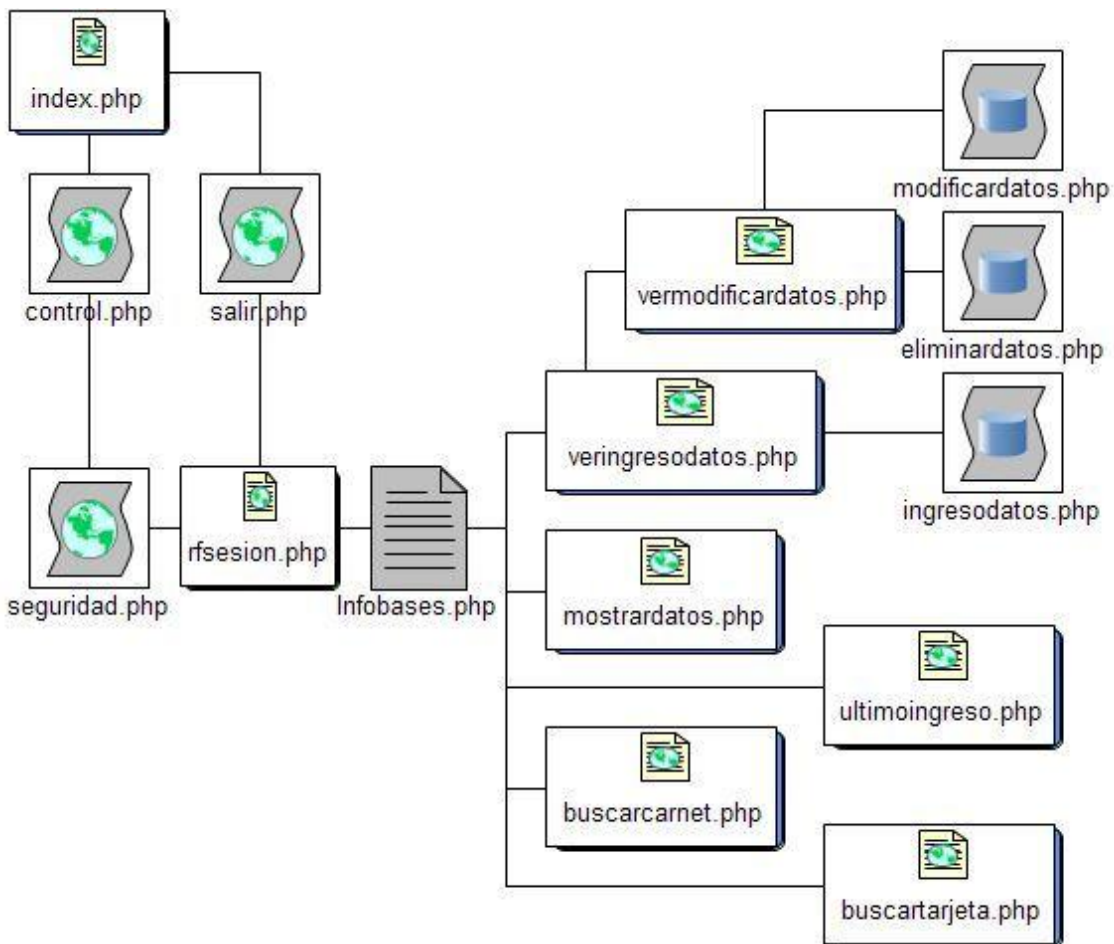
5.7. Interfaz de administrador en página web

Para la administración básica de la información de la base de datos se crea un conjunto de páginas web en las cuales se accede a opciones como la

validación de nuevos usuarios, modificación de datos de usuarios, presentación del contenido de las tablas y búsquedas por número de carné y tarjeta.

Las páginas *web* son creadas en combinación de códigos HTML y PHP, permitiendo una interacción dinámica en un ambiente gráfico. Algunas páginas poseen formularios para el ingreso de datos y consultas a la base de datos.

Figura 33. **Mapa del sitio de administración *web***



Fuente: elaboración propia.

La figura 33 muestra el esquema de relación de las páginas *web* y *scripts* que componen la interfaz de administración.

Estas páginas son almacenadas en el directorio `/var/www` del disco duro de la computadora, desde donde el servidor *web* Apache las transfiere al navegador *web* por HTTP en la dirección IP de la computadora.

5.7.1. Páginas que componen la interfaz

La interfaz se compone de ocho páginas *web*, tres *scripts* de PHP para la autenticación y control de sesión dentro de la interfaz, un *script* con la información de la base de datos y tres *scripts* para realizar consultas a la base de datos.

A continuación se describen las páginas que componen el ambiente de la interfaz de administración básica del SIDERF.

5.7.1.1. Página inicial

La página inicial presenta un formulario solicitando la autenticación del usuario y contraseña establecidos para iniciar una sesión en la administración *web*. La figura 34 muestra el entorno de la página inicial en la dirección `http://localhost/index.php`.

Los datos ingresados en los campos de usuario y contraseña son enviados a un *script* que verifica la validez de los mismos para poder iniciar sesión e ingresar a las opciones de administración de la base de datos.

Figura 34. **Página index.php**



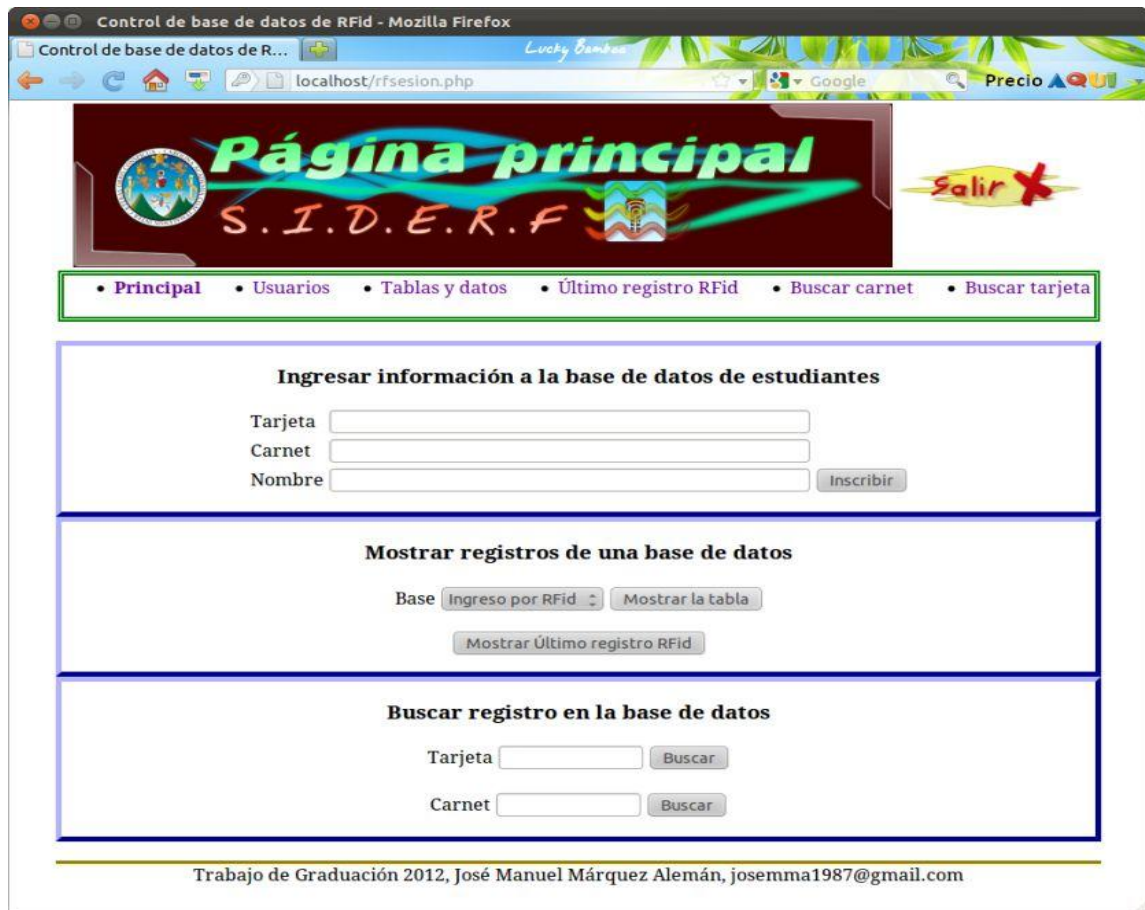
Fuente: elaboración propia.

5.7.1.2. **Página principal**

La página principal es la primera página a la cual se accede tras autenticarse. Esta página presenta el menú principal con opciones como ingresar nuevos usuarios, mostrar el listado de los registros de las tablas de datos y búsquedas de registros por número de tarjeta o por carné.

En la parte superior de cada página de la administración *web* se encuentra un conjunto de enlaces hacia las demás páginas. También se tiene una imagen con la opción de salir de sesión.

Figura 35. Página rfesion.php



Fuente: elaboración propia.

5.7.1.3. Página de registrar usuarios

El enlace de usuarios direcciona el navegador *web* a la página *veringresodatos.php* que presenta la opción de ingresar la información de nuevos usuarios. La información que se debe ingresar es el número de tarjeta de radiofrecuencia, el número de carné del estudiante y el nombre. La página indica una alerta cuando alguno de los campos no sea llenado, y también cuando ya exista registro de uno de los campos.

Figura 36. Página veringresodatos.php

Base de datos de Estudiantes - Mozilla Firefox

Base de datos de Estudiantes

localhost/veringresodatos.php

Google

Precio

Regresar

Base de Estudiantes

S.I.D.E.R.F.

Salir

- Principal
- Usuarios
- Tablas y datos
- Último registro RFid
- Buscar carnet
- Buscar tarjeta

Registros de la base Estudiantes

Ingresar información a la base de datos de estudiantes

Tarjeta

Carnet

Nombre

Tabla base de Estudiantes

| Tarjeta: | Carnet: | Nombre: | Fecha (Aa/Mm/Dd): | Opciones: |
|----------|-----------|--------------|----------------------|---------------------------|
| 12763275 | 200611108 | JOSE MARQUEZ | 2012/08/15--21:26:06 | Modificar |

Trabajo de Graduación 2012, José Manuel Márquez Alemán, josemma1987@gmail.com

Fuente: elaboración propia.

La página también muestra la tabla de estudiantes activos en el sistema y coloca un enlace hacia otra página para modificar datos o eliminar el registro indicado.

5.7.1.4. Página para modificar datos

El enlace de modificar, que aparece en la fila de cada registro, direcciona el navegador web a la página vermodificardatos.php que presenta la opción de modificar los campos de carné y nombre asociados al número de tarjeta. Si uno

de estos campos se deja en blanco, al presionar el botón de modificar, la base de datos no modifica el valor anterior, dado que no acepta campos vacíos.

Dentro de esta página web también se presenta la opción de eliminar el registro seleccionado de la tabla estudiantes.

Figura 37. **Página vermodificardatos.php**

Base de datos de Estudiantes - Mozilla Firefox

Base de datos de Estudiantes

localhost/vermodificardatos.php?tarjeta=12763275

Google

Precio

Regresar

Base de Estudiantes

S.I.D.E.R.F.

Salir

Principal • Usuarios • Tablas y datos • Último registro RFid • Buscar carnet • Buscar tarjeta

Registro de 12763275

Buscar registro en la base de datos

Tarjeta Buscar

Ingresar información a la base de datos de estudiantes

Tarjeta

Carnet

Nombre Modificar

Consulta por No. Id Card 12763275

| Tarjeta: | Carnet: | Nombre: | Fecha(Aa/Mm/Dd): | Opciones: |
|----------|-----------|--------------|---------------------|--------------------------|
| 12763275 | 200611108 | JOSE MARQUEZ | 2012/08/15-21:26:06 | Eliminar |

Fuente: elaboración propia.

5.7.1.5. Página de tablas y datos

El enlace tablas y datos direcciona el navegador *web* a la página mostrardatos.php que presenta todos los registros de una tabla de datos. La tabla a presentar puede ser la tabla Estudiantes o bien la tabla IngresoRF.

Esta página realiza una actualización de la información a cada diez segundos, mostrando el registro más reciente en la primera posición.

Figura 38. Página mostrardatos.php

The screenshot shows a web browser window with the URL `localhost/mostrardatos.php?tablaDB=IngresoRF`. The page features a header with the logo 'Registros Recientes S.I.D.E.R.F.' and navigation links: 'Principal', 'Usuarios', 'Tablas y datos', 'Último registro RFid', 'Buscar carnet', and 'Buscar tarjeta'. Below the header, the title 'Registros de la base IngresoRF' is displayed. A section titled 'Mostrar registros de una base de datos' contains a dropdown menu set to 'Ingreso por RFid' and a 'Mostrar la tabla' button. The main content area displays a table titled 'Tabla base de IngresoRF' with the following data:

| Tarjeta: | Carnet: | Nombre: | Fecha(Aa/Mm/Dd): |
|----------|-----------|--------------|----------------------|
| 12763275 | 200611108 | JOSE MARQUEZ | 2012/08/15--21:40:34 |
| 12763275 | 200611108 | JOSE MARQUEZ | 2012/08/15--21:32:20 |
| 12763275 | 200611108 | JOSE MARQUEZ | 2012/08/15--21:19:13 |
| 12763275 | 200611108 | JOSE MARQUEZ | 2012/08/15--21:18:57 |
| 12763275 | 200611108 | JOSE MARQUEZ | 2012/08/15--21:17:28 |
| 12763275 | 200611108 | JOSE MARQUEZ | 2012/08/15--21:15:55 |

At the bottom of the page, the footer text reads: 'Trabajo de Graduación 2012, José Manuel Márquez Alemán, josemma1987@gmail.com'.

Fuente: elaboración propia.

5.7.1.6. Página de último ingreso

El enlace de último ingreso RFID direcciona el navegador *web* a la página ultimoingreso.php que presenta el registro del último acceso con una tarjeta de radiofrecuencia que ha detectado la estación lectora.

Figura 39. Página ultimoingreso.php



Fuente: elaboración propia.

La página también presenta la fotografía asociada al registro. Se sugiere que la fotografía tenga un tamaño de 200x300 pixeles en formato JPEG. El

nombre de cada fotografía debe ser el número de carné del estudiante, sin espacios, y estar almacenada en el directorio /var/www/BaseFotos.

5.7.1.7. Página de búsqueda de carné

El enlace de buscar carné direcciona el navegador web a la página buscarcarnet.php que presenta todos los registros de la tabla de accesos por tarjeta de radiofrecuencia. La página solicita el número de carné del estudiante a consultar en la base de datos.

Figura 40. Página buscarcarnet.php

Busqueda por carnet de registros de RFid - Mozilla Firefox

localhost/buscarcarnet.php?carnet=00

Regresar

Búsqueda por Carnet

S.I.D.E.R.F.

Salir

• Principal • Usuarios • Tablas y datos • Último registro RFid • **Buscar carnet** • Buscar tarjeta

Registros buscados por carnet 00

Buscar registro en la base de datos

Carnet Buscar

Consulta por Carnet 00

Tarjeta: Carnet: Nombre: Fecha(Aa/Mm/Dd):

Trabajo de Graduación 2012, José Manuel Márquez Alemán, josemma1987@gmail.com

Fuente: elaboración propia.

Los registros presentados en la página son ordenados cronológicamente, colocando en la primera posición el registro más reciente y en la última posición el registro más antiguo. La columna del campo fecha, también indica la hora en que fue registrado el acceso.

5.7.1.8. Página de búsqueda de tarjeta

El enlace de buscar tarjeta direcciona el navegador *web* a la página `buscarnet.php` que presenta todos los registros de la tabla de accesos por tarjeta de radiofrecuencia. La página solicita el número de tarjeta de radiofrecuencia del estudiante a consultar en la base de datos.

Figura 41. Página `buscartarjeta.php`

Busqueda por Id Card de registros de RFid - Mozilla Firefox

localhost/buscartarjeta.php?tarjeta=00

Regresar

Búsqueda por Id Card

S.I.D.E.R.F.

Salir

- Principal
- Usuarios
- Tablas y datos
- Último registro RFid
- Buscar carnet
- Buscar tarjeta

Registros buscados por Id card 00

Buscar registro en la base de datos

Tarjeta

Consulta por No. Id Card 00

Tarjeta: Carnet: Nombre: Fecha(Aa/Mm/Dd):

Trabajo de Graduación 2012, José Manuel Márquez Alemán, josemma1987@gmail.com

Fuente: elaboración propia.

De igual manera que la página anterior, los registros presentados en la página son ordenados cronológicamente, colocando en la primera posición el registro más reciente y en la última posición el registro más antiguo. La columna del campo fecha, también indica la hora en que fue registrado el acceso.

5.7.2. Autenticación y seguridad

El soporte de sesiones en PHP consiste en una manera de guardar ciertos datos a través de diferentes accesos *web*. Esto permite crear aplicaciones más personalizadas y mejorar las características del sitio *web*.

PHP dispone de un método bastante cómodo de guardar datos en variables de sesión, y de un juego de funciones para el trabajo con sesiones y variables de sesión.

Las funciones utilizadas se describen a continuación:

- `Session_start ()`. Iniciar una nueva sesión o reanudar la existente
- `$_SESSION`. Es un array asociativo que contiene variables de sesión disponibles para el *script* actual.
- `Session_destroy()`. Destruye toda la información registrada de una sesión y finaliza la sesión.

5.7.2.1. Script de autenticación de sesión

Este *script* verifica la validez del usuario y contraseña para ingresar a la sesión dentro de la interfaz de administración. También crea la sesión de PHP para que el usuario pueda navegar por las páginas internas.

5.7.2.2. Script de verificación de sesión

Este *script* verifica que exista una sesión de PHP iniciada y activa para permitir al usuario navegar por la página solicitada. Este *script* es llamado al inicio de cada página interna de la interfaz de administración.

5.7.2.3. Script de cierre de sesión

Este *script* finaliza la sesión de PHP existente en la administración *web* y destruye todas las variables creadas.

5.7.3. Consulta a base de datos MySQL

Anteriormente ya se han descrito las funciones de PHP para la conexión a una base de datos de MySQL, pero para realizar las distintas consultas como ingreso, modificación o eliminación de registros de información, es necesario conocer las sentencias del lenguaje SQL.

Las sentencias de consulta de MySQL utilizadas en la administración *web* son las siguientes:

- INSERT. Inserta nuevos registros en una tabla existente
- SELECT. Se utiliza para buscar y recibir registros seleccionados desde una o más tablas.
- UPDATE. Actualiza columnas en registros de tabla existentes con nuevos valores.

- DELETE. Borra los registros de la tabla seleccionada que satisfacen la condición una condición determinada, y retorna el número de registros borrados.

5.7.3.1. Script de insertar datos

Este *script* realiza la consulta de insertar el registro de nuevos usuarios del sistema a la base de datos de estudiantes, antes verificando que éste no se encuentre ya registrado.

5.7.3.2. Script de modificar datos

Este *script* realiza la consulta de actualizar el registro mostrado, cambiando la información del registro en la base de datos de estudiantes.

5.7.3.3. Script de eliminar datos

Este *script* realiza la consulta de borrar un registro específico de la base de datos de estudiantes. La información borrada no puede ser recuperada.

5.8. Funcionamiento final del SIDERF

En esta sección se explican los procedimientos para la puesta en funcionamiento del sistema de identificación estudiantil por radiofrecuencia en el Laboratorio de Electrónica.

También se enumeran los pasos de operación para la identificación de una tarjeta de radiofrecuencia.

5.8.1. Inicialización y apagado del sistema

Antes de encender el SIDERF se deben cumplir los siguientes requisitos previos:

- La estación lectora y la computadora están firmemente conectadas por el cableado de red.
- La computadora tiene instalados el sistema operativo Ubuntu 11,10, el servidor *web* Apache, el servidor de base de datos MySQL y el lenguaje intérprete de PHP.
- La base de datos ha sido creada junto con sus tablas y campos correspondientes.
- El *script* de comunicación está almacenado en la computadora

Para iniciar la operación del SIDERF se deben seguir los siguientes pasos:

- Encender la computadora
- Ejecutar el *script* de comunicación en una terminal de comandos
- Encender la fuente de alimentación de la estación lectora

En este punto la estación lectora ya empieza a leer las tarjetas de radiofrecuencia y a comunicarse con el servidor.

Para detener y apagar la operación del SIDERF se deben seguir los siguientes pasos:

- Cerrar la terminal de comandos donde se ejecuta el *script* de comunicación.
- Apagar la fuente de alimentación de la estación lectora
- Apagar la computadora. Solamente si es necesario

5.8.2. Procedimiento de identificación por RF

La estación lectora presenta una serie de instrucciones, en la pantalla LCD, al estudiante para realizar la lectura de la tarjeta de radiofrecuencia. El procedimiento es el siguiente:

- Inicialmente la estación lectora está en la espera de detectar una tarjeta de radiofrecuencia. La pantalla LCD muestra un saludo de bienvenida y la solicitud de que se coloque una tarjeta en la antena.
- Un estudiante coloca su tarjeta dentro del rango de la antena
- La estación lectora presenta el mensaje del estado de lectura y el número de la tarjeta leída.
- A continuación la estación lectora envía la consulta a la base de datos del servidor.
- El servidor responde la consulta a la estación lectora, la cual presenta en pantalla el mensaje del estado de consulta y los datos del nombre y número de carné del estudiante poseedor de la tarjeta. Si no existe registro en la consulta, la estación lectora indica al usuario que la tarjeta es desconocida.

- Por último, la estación lectora permite el ingreso o lo deniega, dependiendo del resultado de la consulta.
- La estación reinicia su operación, regresando al estado de espera

En el caso que la estación lectora no opere correctamente, se tiene la posibilidad de reiniciar el microcontrolador al presionar el interruptor pulsador adjunto.

5.8.3. Operaciones del administrador *web*

Antes de realizar consultas a la base de datos desde la interfaz de administración *web* básica del SIDERF se deben cumplir los siguientes requisitos previos:

- Las páginas de la administración *web* han sido almacenadas en el directorio `/var/www`. Para poder copiar las páginas a dicha dirección se debe realizar en modo administrador del sistema operativo y otorgarles permiso de ejecución.
- En el navegador se debe haber iniciado sesión desde la página `http://localhost/index.php`.

El procedimiento para insertar un nuevo registro en la base de datos de estudiantes es el siguiente:

- En el navegador *web* ir a la página `rfesion.php`, en el enlace llamado principal. También se puede en el enlace de usuarios.

- En la sección de ingresar información en la base de datos de estudiantes, llenar todos los campos de información.
- Presionar el botón de inscribir

Si falta alguno de los campos, la consulta rechaza el ingreso del registro y alerta al usuario ante la falta de información. También, si la información de uno de los campos coincide con la del mismo campo de otro registro, la consulta rechaza el ingreso del registro y alerta al usuario sobre la existencia de información repetida.

El procedimiento para modificar un registro en la base de datos de estudiantes es el siguiente:

- En el navegador *web* ir a la página `veringresodatos.php`, en el enlace llamado usuarios.
- Buscar el registro a modificar
- Ir al enlace de modificar de dicho registro
- Ingresar la información en los campos a modificar. No es necesario llenar todos los campos. Los campos vacíos no modifican la información anterior.
- Presionar el botón de modificar

El procedimiento para eliminar un registro en la base de datos de estudiantes es el siguiente:

- En el navegador *web* ir a la página `veringresodatos.php`, en el enlace llamado usuarios.
- Buscar el registro a eliminar
- Ir al enlace de modificar de dicho registro
- Dentro de la página de modificar, ir al enlace de eliminar y con ello se borra el registro seleccionado.

El procedimiento para buscar los registros de acceso por radiofrecuencia por número de carné en la base de datos de estudiantes es el siguiente:

- En el navegador *web* ir a la página `rfesion.php`, en el enlace llamado principal. También se puede en el enlace de buscar carné.
- En la sección de buscar registro en la base de datos, escribir el número de carné.
- Presionar el botón de buscar

Para buscar los registros de acceso por radiofrecuencia por número de tarjeta, el procedimiento es similar al anterior pero con el campo de tarjeta.

Para gestionar con mayor detalle la base de datos se sugiere utilizar el programa `PhpMyAdmin`, que ofrece muchas más opciones sobre configuración, consultas y mantenimiento.

5.9. Inversión económica

La inversión económica para la elaboración del sistema consiste únicamente en el costo de los componentes de *hardware* utilizados, puesto que el *software* instalado es libre y no conlleva costos de licencias.

5.9.1. Cotización del sistema propuesto

La tabla VI enlista los componentes electrónicos del circuito del PIC.

Tabla VI. **Listado de componentes electrónicos del circuito del PIC**

| Cantidad | Descripción | Costo total en Q |
|-----------------|-------------------------------------|-------------------------|
| 1 | PIC 18F4520 | 100.00 |
| 1 | Socket DIP 40 pines | 5.00 |
| 1 | Cristal oscilador 8 Mhz | 15.00 |
| 2 | Capacitor 22 pF | 5.00 |
| 4 | 2x5 pines IDC | 10.00 |
| 1 | Pulsador normalmente abierto | 5.00 |
| 1 | Resistor 10 K-ohm, ¼ Watt | 0.70 |
| 1 | Resistor 1 K-ohm, ¼ Watt | 0.70 |
| 1 | Led 3mm | 1.00 |
| 1 | Puente rectificador | 3.50 |
| 2 | Capacitor electrolítico 220 uF, 25V | 8.00 |
| 1 | Capacitor 100 nF | 2.50 |
| 1 | Regulador de voltaje 7805 | 12.00 |
| | Total | Q 168.40 |

Fuente: Electrónica RC&H.

La tabla VI también incluye los componentes pertenecientes al circuito de alimentación de voltaje regulado.

Tabla VII. **Listado de componentes electrónicos del circuito de LCD**

| Cantidad | Descripción | Costo total en Q |
|-----------------|---------------------------------|-------------------------|
| 1 | 1x16 pines IDC | 4.50 |
| 1 | 2x5 pines IDC | 2.50 |
| 1 | Trimpot 10 K-ohm | 5.00 |
| 1 | Resistor 10 ohm, ¼ Watt | 0.70 |
| 1 | Pantalla LCD de 2x16 caracteres | 150.00 |
| | Total | Q 162.70 |

Fuente: Electrónica RC&H.

La tabla VII enlista los componentes electrónicos que conforman el circuito de conexión de la pantalla LCD.

Tabla VIII. **Listado de módulos periféricos adicionales**

| Cantidad | Descripción | Costo total en \$ |
|-----------------|------------------------------|--------------------------|
| 1 | <i>RFid Reader</i> | 24.00 |
| 1 | <i>RFid Card</i> | 1.50 |
| 1 | <i>Serial Ethernet Board</i> | 29.90 |
| | Total | \$ 55.40 |

Fuente: elaboración propia, con base a: www.mikroe.com. Consulta: 25 de agosto de 2012.

El precio de algunos componentes puede variar dependiendo del lugar de compra, pero se puede estimar que el costo total es aproximadamente 775,00 quetzales. Además, se debe considerar también la adquisición de una computadora que ejerza la función de servidor, y que cumpla con los requerimientos mínimos de funcionamiento.

5.9.2. Sistemas disponibles en el mercado

Los equipos y sistemas de identificación por radiofrecuencia en el mercado tienen un amplio rango de precios dependiendo de las prestaciones que ofrezca. Se pueden mencionar algunos de los modelos de controladores de acceso por radiofrecuencia existentes en el mercado y su respectivo precio, con el fin de tener un marco de referencia con el cual comparar el costo del sistema diseñado:

- Controlador Accesos RFID AEX4, marca LipSoft, con precio de € 158,00
- Controlador Accesos RFID AEX5 para 125kHz, marca LipSoft, con precio de € 168,00.
- Controlador Accesos RFID 125 Khz con teclado, marca LipSoft, con precio de € 254,00.
- Controlador Accesos RFID 125Khz con teclado antivandálico, marca Lipsoft, con precio de € 390,00.

El precio del sistema propuesto es bajo en comparación con los productos de identificación por radiofrecuencia en existencia en el mercado. Se debe tomar en cuenta que el sistema propuesto puede ser expandido en capacidades y prestaciones y con ello el costo se incrementará.

CONCLUSIONES

1. Se logró desarrollar un sistema con un diseño efectivo, que abarca todas las características fundamentales de un sistema de identificación por radiofrecuencia, con el cual se agilizará el proceso de acceso al Laboratorio de Electrónica, y se tendrán los registros para un mejor control de los estudiantes.
2. Se alcanzó optimizar los recursos de *hardware* que componen el sistema, minimizando la cantidad de componentes utilizados. Se puede resaltar el uso de un único microcontrolador seleccionado por sus características ideales de conectividad y operación con los 3 módulos periféricos, *RFid Reader*, LCD y *Serial Ethernet Board*, proporcionados por el Laboratorio de Electrónica.
3. Se consiguió desarrollar códigos ligeros y eficientes que gobiernan los distintos procesos del sistema de identificación, elaborados con ayuda de lenguajes de alto nivel orientados a la programación con librerías de funciones.
4. Los programas *software* libre fueron la mejor elección para el desarrollo de los servicios que provee la computadora, dado que otorgan la libertad de configuración y conexión requerido por el sistema de identificación.
5. Con el sistema de identificación propuesto se obtendrá un ahorro económico significativo, en comparación con el costo de los sistemas de control de acceso disponibles en el mercado.

RECOMENDACIONES

1. El Laboratorio de Electrónica debe planear una logística de control de acceso de estudiantes para aprovechar al máximo las capacidades del sistema de identificación propuesto, que será la herramienta central del proceso de acceso.
2. Por seguridad de la información almacenada, en la base de datos del sistema de identificación, debe utilizarse una red privada y dedicada, de uso exclusivo del personal autorizado del Laboratorio de Electrónica, con conexión física directa entre la estación lectora y el servidor, para evitar la intromisión de individuos ajenos.
3. Dos direcciones IP fijas del segmento de red a utilizar deben ser reservadas, una para la estación lectora y la otra para el servidor, dado que el sistema no fue configurado para usar el servicio DHCP de asignación automática de direcciones IP. Los cambios en los parámetros IP deben modificarse en el código del microcontrolador y en el servidor.
4. El cambio de usuario y contraseña de la cuenta de MySQL, para consultar en la base de datos del servidor, requiere que dichos parámetros también sean modificados en los *scripts* y páginas de administración *web* que lo utilicen.

BIBLIOGRAFÍA

1. ANGULO, José. *Microcontroladores PIC: diseño práctico de aplicaciones segunda parte PIC16F87X, PIC18FXXXX*. 2a ed. España: McGraw-Hill, 2006. 374 p. ISBN: 9788448146276.
2. BLACK, Uyles. *Redes de computadoras: protocolos, normas e interfaces*. México: Macrobit, 1990. 421 p. ISBN: 970-604-016-1.
3. CUENCA, Eugenio. *Microcontroladores PIC: la solución en un chip*. España: Paraninfo, 1997. 489 p. ISBN: 9788428323710.
4. Em Microelectronic. *EM4095 Data sheet*. [en línea]. <http://www.emmicroelectronic.com/webfiles/Product/RFID/DS/EM4095_DS.pdf>. [Consulta: 28 de mayo de 2012].
5. Em Microelectronic. *EM4100 Data sheet*. [en línea]. <http://www.mikroe.com/eng/downloads/get/1084/em4100_datasheet.pdf>. [Consulta: 28 de mayo de 2012].
6. GIDEKEL, Alan. *RFID Introducción a la identificación por radio frecuencia*. [en línea]. Argentina: Telectrónica Codificación S.A., 2006. <<http://www.telectronica.com/rfidtelectronica.pdf>>. [Consulta: 6 de febrero de 2012].

7. Microchip Technology Incorporated. *ENC28J60 Data Sheet*. [en línea]. Estados Unidos. <<http://ww1.microchip.com/downloads/en/DeviceDoc/39662c.pdf>>. [Consulta: 28 de mayo de 2012].
8. Microchip Technology Incorporated. *PIC18F4520 Data Sheet*. [en línea]. Estados Unidos. <<http://ww1.microchip.com/downloads/en/DeviceDoc/39631E.pdf>>. [Consulta: 13 de marzo de 2012].
9. MikroElektronika. *RFid Reader Manual*. [en línea]. <http://www.mikroe.com/eng/downloads/get/1240/rfid_manual_v100.pdf>. [Consulta: 13 de mayo de 2012].
10. MikroElektronika. *Serial Ethernet Manual*. [en línea]. <http://www.mikroe.com/eng/downloads/get/45/serial_ethernet_manual_v100.pdf>. [Consulta: 13 de mayo de 2012].
11. Oracle. *MySQL 5.0 Reference Manual*. [en línea]. Estados Unidos. <<http://dev.mysql.com/doc/mysql/en/>>. [Consulta: 10 de junio de 2012].
12. PHP Group. *Manual de PHP*. [en línea]. <<http://www.php.net/manual/es/>>. [Consulta: 4 de junio de 2012].
13. RFID magazine. *Tecnología RFID: Introducción*. [en línea]. [Barcelona, España]: 2005. <http://www.mas-rfid-solutions.com/docs/RFID_introduccion.pdf>. [Consulta: 7 de febrero de 2012].
14. SÁNCHEZ, Jesús. *Redes, Iniciación y referencia*. España: McGraw-Hill, 2000. 318 p. ISBN: 84-481-2771-4.

15. Servicios Informáticos KIFER. *Introducción a los sistemas RFID*. [en línea]. Lasarte – Oria, España: 2006. <<http://www.kifer.es/Recursos/Pdf/RFID.pdf>>. [Consulta: 7 de febrero de 2012].
16. The Apache Software Foundation. *Documentación del servidor HTTP Apache*. [en línea]. <<http://httpd.apache.org/docs/current/>>. [Consulta: 7 de junio de 2012].
17. The phpMyAdmin development team. *PhpMyAdmin 4.0.0-dev Documentación*. [en línea]. <http://www.phpmyadmin.net/localized_docs/es/Documentation.html>. [Consulta: 15 de junio de 2012].
18. VALDES, Fernando. *Microcontroladores: fundamentos y aplicaciones con PIC*. España: Marcombo, 2007. 340 p.
19. Winstar Display Company. *Specification LCD module*. [en línea]. <http://www.mikroe.com/eng/downloads/get/1664/lcd_2x16_spec.pdf>. [Consulta: 20 de mayo de 2012].