**UNIVERSIDAD NACIONAL DE INGENIERIA**

**RECINTO UNIVERSITARIO SIMÓN BOLÍVAR**

**FACULTAD DE ELECTROTECNIA Y COMPUTACIÓN**

# “Implementación de un Sistema Electrónico para el Control de Acceso Utilizando Tecnología RFID para el Programa UNI-Online”

TRABAJO MONOGRÁFICO PARA OPTAR AL TÍTULO DE

INGENIERO ELECTRONICO

**Elaborado por:**

Br. Carlos Eberto Tróchez Zúniga

**Tutor:**

Ing. Juan Manuel Martínez Toribio

Managua, Julio del 2017

# DEDICATORIA

De manera muy especial dedico este trabajo monográfico, con mucho cariño, aprecio y gratitud a mis padres y a mi hermana, por todo el sacrificio, y apoyo incondicional quienes al día de hoy me han logrado sacar adelante a pesar de todos los obstáculos que hemos tenido durante todo este trayecto.

A mi tía Gloria Castillo y familia, por el apoyo económico que me dieron para que yo finalizara con éxito mi proyecto monográfico.

A Reynerio Flores por ser un gran amigo de la familia y por ofrecernos su apoyo en los momentos más difíciles.

A Gabriel Delgadillo Fernández por ser un gran amigo y ofrecerme su apoyo incondicional.

A mis compañeros, quienes fueron un apoyo en los momentos más difíciles de la carrera.

A mi tutor Ing. Juan Manuel Martínez Toribio y al profesor Ing. José Manuel Arcia Salmerón por el apoyo que me dieron con mi trabajo monográfico, por brindarme su tiempo y conocimientos para concluir con éxito mi proyecto.

## RESUMEN

El presente trabajo monográfico tiene el propósito de brindar una alternativa de solución a los problemas de seguridad que afecta a la UNI-Online, la cual radica en la inseguridad de equipos informáticos y documentos académicos entre otros, que se encuentran en el local.

Como solución se decidió implementar un sistema de control de acceso utilizando tecnología RFID, debido a su bajo costo y velocidad de identificación.

El sistema está compuesto por un Arduino Mega, cuya función principal es controlar de forma autónoma el acceso a personal autorizado mediante la apertura de una cerradura electromagnética. El mecanismo de identificación es a través de tarjetas de identificación por radiofrecuencia (RFID por sus siglas en ingles), que al ser energizadas por el campo magnético producido por el modulo lector MFRC522, envían al lector un código de identificación. El Arduino procesa el número de identificación comparándolo con una base de datos en el sistema, para posteriormente ejecutar las acciones necesarias. Para la configuración del sistema y los usuarios es mediante un interfaz de un Display LCD y teclado, que permite al usuario definir los parámetros del sistema.

## INDICE DE CONTENIDOS

## LISTA DE FIGURAS

## LISTA DE TABLAS

# I. INTRODUCCION

El presente trabajo monográfico consiste en brindar una alternativa de solución a problemas existentes en el programa UNI[1]-Online, la cual radica en la inseguridad de equipos informáticos (PC[2], UPS[3], mouse, etc.) y documentos académicos (libros) entre otros, que se encuentran en el local. Cabe destacar que el programa UNI-Online consta de dos espacios: Una para oficina de dirección y otra para laboratorio de cómputo.

Los equipos antes mencionados han sido asignados al personal, sin embargo, en los últimos años han sufrido muchas pérdidas de bienes, lo cual les ha generado desconfianza. Esto provoca que el personal opte por cargar con sus bienes.

Debido al carácter público del programa UNI-Online y sus alrededores, frecuentemente hay personas haciendo gestiones, como consultas académicas y trámites administrativos, por lo tanto, es necesario dotar un sistema de seguridad capaz de prevenir el acceso a personal no autorizado. Muchas entidades laborales utilizan diferentes tipos de tecnologías que controlan el acceso de personal, a diferentes áreas de sus instalaciones, una de las tecnologías más utilizadas es el control de **identificación por radio frecuencia** (RFID).

La tecnología de radio frecuencia (RFID) permite la identificación de objetos de forma inalámbrica, sin necesidad de que existiera entre el lector y el objeto contacto o línea de visión directa, requisito indispensable para otras tecnologías como por ejemplo la lectura láser de códigos de barra. Esta identificación se realiza mediante la incorporación o fijación de un transpondedor (“tag”[4]) al objeto, el cual transmite los datos que contiene cuando detecta que está siendo interrogado por un lector RFID.

[1] UNI: Universidad Nacional de Ingeniería
[2] PC: Personal Computer
[3] UPS: Uninterrupted Protection System
[4] Tags: Son las etiquetas RFID también llamada transpondedor (transmisor y receptor)

Las tecnologías basadas RFID son muy utilizadas para el control de acceso, logística, entre otras aplicaciones. Las características que lo hacen ser uno de los más utilizados masivamente van desde su bajo costo hasta la velocidad de identificación y la lectura de código emitido a grandes distancias. Su característica inalámbrica permite la identificación de usuarios sin necesidad de interactuar directamente con el sistema, evitando la molestia de introducir códigos manualmente.

En este documento abordaremos el desarrollo de un sistema electrónico para el control de acceso utilizando la tecnología de identificación por radio frecuencia (RFID), enfocándonos principalmente en nuestro plan de implementación, el cual pretende controlar el acceso a personal no autorizado en el programa UNI-Online. Haremos una descripción estructurada de las características del sistema en cuanto a los dispositivos que lo conforman.

# II. OBJETIVOS

## Objetivo General

Desarrollar un sistema electrónico mediante la implementación de la tecnología de identificación por radiofrecuencia (RFID) para el control de acceso en el programa UNI-Online

## Objetivos Específicos

- Diseñar la estructura de un sistema para el control de acceso de rápida acción, robusta y de alta tecnología
- Seleccionar la estructura que permita la construcción y el diseño tecnológico del sistema, cumpliendo con los estándares que rigen el desarrollo de este tipo de tecnología
- Construir un código de programación capaz de llevar a cabo el accionamiento de todas las tareas para la cual estará diseñada el sistema
- Implementar la estructura de un sistema para el control de acceso mediante el uso de un sistema de identificación por radio frecuencia (RFID) en el programa UNI-Online

## III. JUSTIFICACION

La falta de seguridad en las instalaciones universitaria es la causa principal de pérdidas de equipos y documentos en el programa UNI-Online. Para eso será necesario desarrollar un sistema capaz de controlar el acceso de personas, control que puede ser llevado a cabo por un sistema electrónico.

Existen diversas gamas de tecnologías que ofrecen seguridad, tales como los sistemas biométricos, códigos de barra, sistemas magnéticos, infrarrojos, identificación por radiofrecuencia entre otros. Como ingeniero electrónico tengo la obligación y el deber de buscar la solución a la problemática que se presente.

La elección de un sistema de control de acceso robusto, daría como resultado una confortable seguridad, y proporcionaría un ambiente de trabajo seguro al personal qué utilice este local. De esta manera, se evitará la incómoda situación de que el personal cargue a diario con sus bienes, y se generaría un grado de protección a los equipos que se encuentran en la UNI-Online.

# IV. MARCO TEORICO

## CAPITULO 1. TECNOLOGÍAS DE AUTOIDENTIFICACIÓN

En el transcurso de los años se ha venido desarrollando distintas técnicas para la identificación, que desde los inicios fue de gran relevancia las credenciales de identificación, los cuales fueron precursores para lograr la identificación de personales en las áreas laborales, en la actualidad estos sistemas de identificación han venido desarrollándose gracias a los avances tecnológicos, siendo estos de mayor fidelidad que los antiguos sistemas de identificación.

En el presente capítulo se hace un análisis de las tecnologías de identificación existentes, presentando sus principales características, así como sus ventajas y desventajas.

### 1.1 Tipos de Sistemas de Identificación

Los sistemas de identificación se clasifican conforme a la tecnología mediante la cual han sido desarrolladas, los cuales se clasifican en (Finkenzeller, 2010, págs. 2-6): sistemas biométricos, tarjetas magnéticas, código de barras, RFID (Identificación por Radio Frecuencia), entre otros.

#### 1.1.1 Sistemas de Identificación Biométricos (Control de Acceso)

La tecnología biométrica consta de métodos automatizados mediante el cual se analiza rasgos físicos o determinadas características humanas, a continuación clasificaremos algunas de las técnicas de identificación biométrica (Finkenzeller, 2010, pág. 5):

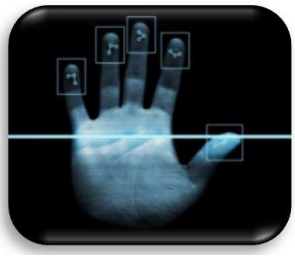
- Reconocimiento de iris
- Reflexión retinal

- Geometría de la mano
- Geometría facial
- Termografía mano, facial
- Huellas dactilares
- Patrón de la voz

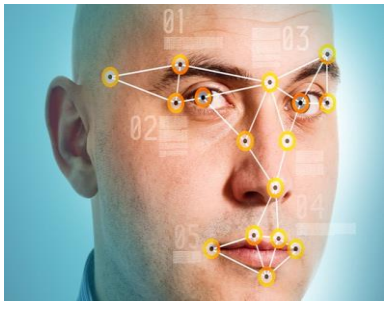
Estos sistemas analizan los rasgos físicos por medio de un lector biométrico para compararlo con una muestra almacenada en la memoria del sistema, si la correlación entre estas es coherente el lector envía el número de identificación del usuario al panel. (Núñez Zeledón & Zeledón Espinoza, 2013, pág. 8)

Estos sistemas proveen al usuario una gran confidencialidad, sus principales desventajas está en su alto costo y escasa aplicación, la cual es limitada debido a su rango de identificación el cual requiere contacto directo o de distancia mínima con el modulo identificador. En la **Figura 1** se muestra los diferentes tipos de autentificación biométrica.

A. Huella dactilar



B. Geometría de la mano



C. Reconocimiento facial

D. Reconocimiento de retina

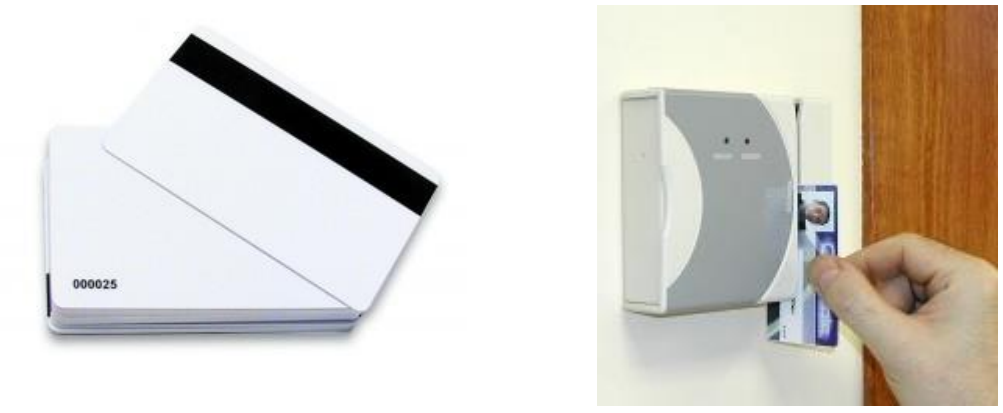
**Figura 1. Autentificación biométrica**

### 1.1.2 Sistemas de Identificación con Tarjetas Magnéticas

Utilizadas masivamente para la fabricación de tarjetas de crédito, y el control de apertura de cerraduras electrónicas, el funcionamiento de este tipo de sistema es la de la lectura de una banda magnética, el cual requiere de contacto físico directo con el modulo lector.

La banda magnética se presenta como una cinta de color negro, estas bandas están compuestas por partículas ferromagnéticas incrustadas en una matriz de resina, las cuales guardan cierta información, mediante determinada codificación que polariza dichas partículas. Este tipo de tecnología son sensibles a grandes interferencias electromagnéticas y su uso continuo provoca un desgaste en las bandas debido a la fricción al momento de hacer contacto con el modulo validador (lector).

En la **Figura 2** se muestra los componentes necesarios para llevar a cabo este tipo de identificación.




**A. Tarjetas Magnéticas**

**B. Lector Magnético**

**Figura 2. Identificación Magnética**

### 1.1.3 Sistemas de Identificación con Código de Barra

Esta tecnología es explotada masivamente por el sector del comercio, utilizada para la identificación de los productos en venta, sus estructuras de codificación están presentes en forma de barras o rayas de color negro espaciadas estratégicamente para la representación de caracteres, estos códigos son leídos por equipos especiales de lectura óptica, las cuales se encargan de llevar información a un ordenador para su debido procesamiento. No requiere de contacto físico directo, aunque la distancia de lectura del módulo lector es bastante corta, se han inventado alrededor de 270 diferentes simbologías para soportar requerimientos específicos, Cada una de estas simbologías cae dentro de alguna de las siguientes tres categorías (Sweeney II, 2005, págs. 34-37):

- **Lineal**. Consiste en líneas verticales, de diferentes anchos, con espacios blancos que separan dos líneas adyacentes. El máximo número de caracteres que pueden ser codificados, mediante esta metodología, son 50

- **Dos dimensiones**. Esta simbología tiene la mayor capacidad de almacenamiento, el máximo número de caracteres que pueden ser codificados es de 3,750

- **Tres dimensiones (Bumpy).** Este tipo de código de barras es leído, utilizando el relieve de las barras, es decir, no depende del contraste entre barras obscuras y espacios, por lo tanto, puede ser embebidos directamente en los productos como por ejemplo en llantas o en partes plásticas directamente desde el molde. La ventaja de estos códigos es que pueden ser utilizados en ambientes de uso rudo

Como principal desventaja en comparación con las demás tecnologías; es que son fáciles de falsificar y alterar. **En Figura 3**. Podemos observar las diferentes etiquetas de código de barras.


Figura 3. Tipos de etiquetas de Códigos de Barras

### 1.1.4 Sistemas de Identificación por Radio Frecuencia (RFID)

Los sistemas de identificación basados en la tecnología de radiofrecuencia, son en la actualidad uno de los más utilizados por las diferentes industrias, su principal aplicación se encuentra en el sector de la seguridad; control de acceso mediante la fabricación de credenciales, control de acceso vehicular, identificación del ganado, en la logística, identificación de equipajes en los aeropuertos, entre otras de las muchas aplicaciones a la cual está disponible esta tecnología.

La tecnología RFID está estructurada por tres componentes fundamentales: un “transponder” o tarjeta transmisora, esta porta un código de identificación, un “lector”, llamado también validador ya que este se encarga de procesar los códigos emitidos por el “transponder”. Y un “sistema de cómputo” que se encarga de procesar la información obtenida de los lectores. La tecnología RFID es completamente inalámbrica y su distancia de lectura es relativamente amplia, razón por la cual posee cierta superioridad en comparación a las demás tecnologías de identificación; la distancia de lectura depende completamente de la tecnología de los transponder.

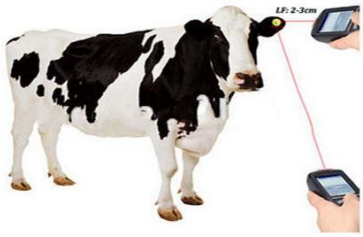
Está tecnología es muy utilizada para el control de acceso e identificación de equipajes en los aeropuertos, estas aplicaciones son muy interesantes ya que pueden ser aplicados para el control de acceso en laboratorios y monitoreo de equipos, control de acceso de personal en los bancos, control de acceso vehicular y estudiantes en centro de estudios universitarios.

En el **Capítulo 2**, abordaremos con más detalle el funcionamiento, así como su estructura, y las ventajas que ofrece este tipo de tecnología.

En la **Figura 4** se aprecia algunas aplicaciones que tiene la tecnología RFID.

**A. Control de Acceso**

**B. Acceso vehicular**



**C. Identificación del ganado bovino**

**D. Identificación de los equipajes en los aeropuertos**

**Figura 4. Aplicaciones de la Tecnología RFID**

## 1.2 Comparación entre tecnologías de Radiofrecuencia y Código de Barras

La identificación por radiofrecuencia ha tenido un gran auge en los últimos años, su origen se remonta a la necesidad de buscar mejores alternativas de identificación automáticas que los sistemas de códigos de barras no ofrecen debido a su ineficiencia.

A continuación, se describirá las diferencias entre ambos sistemas (Finkenzeller, 2010, págs. 3-9):

Una de las ventajas de la identificación por radiofrecuencia con respecto al código de barra, es que no necesita línea directa de visión entre etiqueta y lector para poder ser identificada y, dependiendo de la tecnología que se utilice, la distancia puede variar desde un par de centímetros hasta varios metros. Lo que lo hace adecuada para muchas aplicaciones en las que no se puedan utilizar códigos de barras.

Otra ventaja que ofrece esta tecnología es que se pueden identificar productos del mismo tipo como únicos, es decir que puede diferenciar productos iguales atreves de una etiqueta RFID, a diferencia de los códigos de barras que para productos iguales los identifica como el mismo. Una etiqueta de RFID es mucho más difícil de clonar que un código de barras que puede ser igualado por medio de una fotocopia.

Un código de barras no puede ser modificado, una vez que se ha impreso, por lo tanto, es una tecnología de solo lectura. A diferencia de las etiquetas RFID pueden tener la capacidad de lectura y escritura, ya que cuentan con una memoria interna que puede ser modificada miles de veces durante su ciclo de vida. Esta capacidad hace de RFID una tecnología muy robusta.

Otro problema del código de barras es que sólo puede identificar un solo producto a la vez, a diferencia de la tecnología RFID que puede realizar múltiples lecturas simultáneas, ofreciendo una alta velocidad y una gran precisión.

Y finalmente una etiqueta de RFID tiene una mayor durabilidad y un menor desgaste, debido a que, si un código de barras sufre de desgaste o roturas, ya no podrá ser leído.

El único punto a favor del código de barras es que su precio puede llegar a ser insignificante.

Habiendo detallado las características de cada sistema por separado, se puede resumir lo expuesto en la **Tabla 1**.

**Tabla 1. Diferencias entre los Sistemas de Auto-identificación**

*Fuente: Sistema de Control de Acceso con RFID*

| | Código de Barras | Banda Magnética | Sistemas Biométricos | RFID Pasivo | RFID activo |
|---|---|---|---|---|---|
| **Modificación de la información** | No Modificable | Modificable | No Modificable | Modificable | Modificable |
| **Seguridad de los Datos** | Mínima | Media | Alta | Variable (baja a alta) | Alta |
| **Capacidad de Almacenamiento de datos** | -Lineales (8 - 30 caracteres)<br><br>-2 hasta 7.200 caracteres | Hasta 128 bytes | No aplica | Hasta 64 KB | Hasta 8MB |
| **Precio** | Bajo | Medio-Bajo | Alto | Medio | Muy Alto |
| **Distancia de Lectura** | Línea de vista y (hasta 1.5m) | Requiere contacto | Depende del biométrico | No requiere línea de vista ni contacto Hasta 10m | No requiere línea de vista ni contacto Hasta 100 m. y mayores |
| **Interferencia Potencial** | Cualquier modificación en las barras y objetos entre el código y el lector | Bloqueo del contacto | Puede ser bloqueo del contacto, o bloqueo de línea de vista e inclusive el ruido | Ambientes o campos que afecten la transmisión de radio frecuencia | La interferencia es muy limitada, debido a la potencia de transmisión |

# CAPITULO 2. TECNOLOGÍA DE IDENTIFICACIÓN POR RADIOFRECUENCIA

Un sistema de RFID (Radio Frequency Identification) es la tecnología inalámbrica que nos permite, básicamente, la comunicación entre un lector y una etiqueta. Estos sistemas permiten almacenar información en sus etiquetas mediante comunicaciones de radiofrecuencia. Esta información puede ir desde un Bit hasta KBytes, dependiendo principalmente del sistema de almacenamiento que posea el transponder. (Ciudad Herrera & Casanovas , pág. 8)

La tecnología de identificación por radiofrecuencia ha tenido un crecimiento notable en los últimos años, la explotación de esta tecnología por parte de las empresas y las industrias han llevado a ser una de las más usadas en el sector comercio.

Nace como una alternativa de identificación automática de productos u objetos, similar a la lectura de códigos de barras que parece ser ya obsoleta e ineficiente. Comparando ambos casos, RFID no sólo tiene la ventaja de facilitar la creación de sistemas que almacenen mucha más información, sino que también permite identificar un producto u objeto como único, aunque sea de una misma clase, en contraparte, la lectura del código de barras considera un solo código de identificación por cada clase. (Herrera Lozada, Pérez Romero, & Marciano Melchor, 2009, pág. 57)

Este trabajo monográfico se basa en el diseño e implementación de un sistema de seguridad electrónica usando tecnología RFID. Para tener una mejor comprensión de los resultados de este proyecto en el presente capítulo se abordará fundamentos teóricos sobre la tecnología RFID, así como sus principales características y las ventajas que ofrece esta tecnología.

## 2.1 Arquitectura

Un sistema RFID consta de los siguientes tres componentes (Wikipedia, 2016):

- **Etiqueta RFID o transpondedor:** compuesta por una antena, un transductor radio y un material encapsulado o chip. El propósito de la antena es permitirle al chip, el cual contiene la información, transmitir la información de identificación de la etiqueta. Existen varios tipos de etiquetas. El chip posee una memoria interna con una capacidad que depende del modelo y varía de una decena a millares de bytes. Existen varios tipos de memoria:
  a. *Solo lectura***:** el código de identificación que contiene es único y es personalizado durante la fabricación de la etiqueta.
  b. *De lectura y escritura***:** la información de identificación puede ser modificada por el lector.
  c. *Anticolisión:* Se trata de etiquetas especiales que permiten que un lector identifique varias al mismo tiempo (habitualmente las etiquetas deben entrar una a una en la zona de cobertura del lector).

- **Lector de RFID o transceptor**: compuesto por una antena, un transceptor y un decodificador. El lector envía periódicamente señales para ver si hay alguna etiqueta en sus inmediaciones. Cuando capta una señal de una etiqueta (la cual contiene la información de identificación de esta), extrae la información y se la pasa al subsistema de procesamiento de datos.

- **Sistema de cómputo**. Por lo general, es un repositorio de datos que se utiliza para procesar la información obtenida por los lectores. En la **Figura 5** se muestra un sistema básico de RFID.

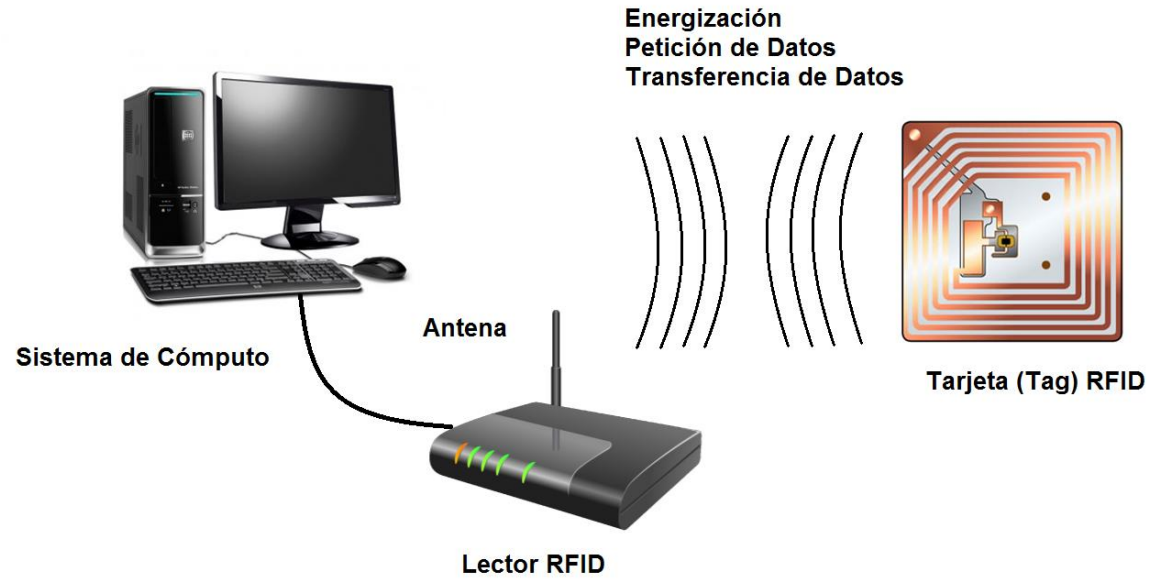



**Figura 5. Sistema básico de RFID**

*Fuente: Elaboración Propia*

## 2.2 Principios Básicos de Funcionamiento

El modo de funcionamiento de los sistemas RFID es sencillo. Podemos observar la **Figura 6**, el lector RFID genera un pequeño campo de radiofrecuencia que estimula e induce una antena en miniatura contenida en el encapsulado de la tarjeta, generándose en ésta una corriente eléctrica que permite que un microcircuito sea capaz de transmitir sus datos al lector. Así, cuando el lector hace una petición de datos, la tarjeta responde a dicha solicitud. Los datos extraídos por el lector RFID pueden ser almacenados en una base de datos para realizar alguna consulta; en realidad, el sistema de cómputo se adecuará a las necesidades específicas de la aplicación. (Want, 2006, págs. 27, 28)

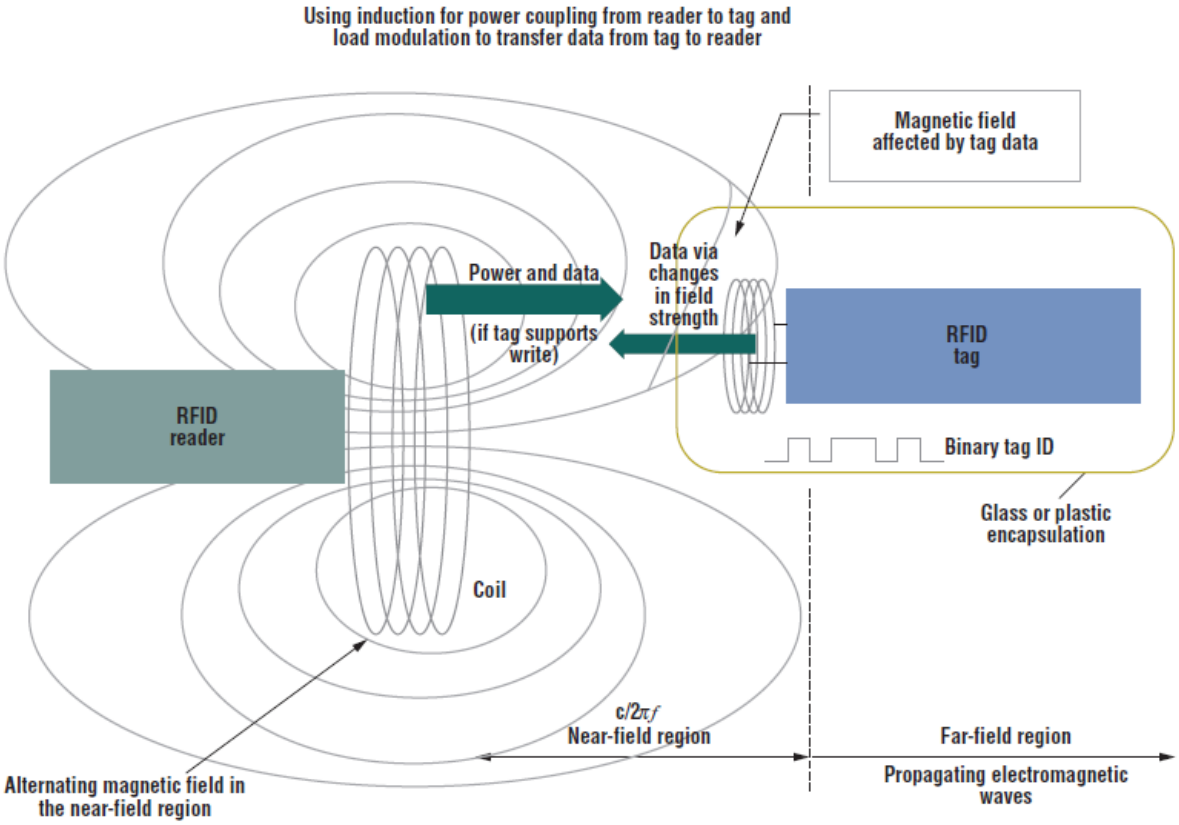



**Figura 6. Mecanismo de alimentación/comunicación de campo cercano para las etiquetas RFID que operan a menos de 100 MHz**

*Fuente: An Introduction to RFID Technology*

En este contexto, se dispone de tarjetas pasivas (sin alimentación interna, menor tamaño, menor coste) o tarjetas activas (alimentación interna, mayor almacenamiento). En las de tipo pasivo, la alimentación se obtiene de la misma frecuencia de trabajo y el sistema funciona mediante la técnica de modulación digital por frecuencia: ASK (*Modulación por desplazamiento de amplitud*), FSK (*Modulación por desplazamiento de frecuencia*) y PSK (*Modulación por desplazamiento de fase*), con la que se facilita la adquisición, pero está limitada en la distancia entre el lector y la tarjeta (10 centímetros máximo) y en el número de lecturas que se pueden realizar. En las tarjetas activas de RFID, se utiliza comúnmente la alimentación por batería, propiciando alcances mayores en la proximidad (de 50 centímetros hasta 25 metros). (Finkenzeller, 2010, pág. 186)

Los datos dentro de cada tarjeta se guardan en una memoria. Cada objeto a identificar tiene un código único y puede extraerse a distancia y sin tocarlo mediante el lector. Esta información puede ir desde un Bit hasta KBytes, dependiendo principalmente del sistema de almacenamiento que posea el transponder.

La comunicación entre el lector y la etiqueta se realiza mediante señales de radiofrecuencia a una determinada frecuencia que generan las antenas de lector y etiqueta, estas frecuencias pueden ser iguales o pueden ser armónicos. La comunicación entre ellas tiene unas determinadas características de alcance, velocidad y seguridad según el rango de frecuencia, el tipo de antenas utilizadas, el tipo de etiquetas y demás parámetros que se pueden configurar para una aplicación u otra.

En equipos RFID nos podemos encontrar con sistemas anticolisión que permiten leer varias tarjetas al mismo tiempo. En caso de que varias tarjetas estén en el rango de alcance del interrogador y dos o más quieran transmitir al mismo tiempo, se produce una colisión. El interrogador detecta la colisión y manda parar la transmisión de las tarjetas durante un tiempo. Después irán respondiendo cada una por separado por medio de un algoritmo bastante complejo. Obviamente a mayor capacidad de la etiqueta y el lector, más efectivos serán estos algoritmos. (Ciudad Herrera & Casanovas , pág. 9)

Este tipo de comunicación no necesita de ninguna licencia debido a que el campo magnético utilizado por la NFC (Comunicación de campo cercano) tiene una frecuencia de 13,56 MHz, que no implica riesgo para la salud y no requiere la regulación de ningún organismo, lo que es una gran ventaja (Ver **Anexo C.2.1**). (Cacuango Guachalá & Zapata Narváez, 2015, pág. 5)

## 2.3 Estructura de los Sistemas de Control de Acceso Basados en la Tecnología RFID

Un sistema de control de acceso basado en RFID consta principalmente de cuatro partes:

### 2.3.1 Transponder (Transmite y Responde)

El transpondedor es el dispositivo que va embebido en una etiqueta o tag y contiene la información asociada al objeto al que acompaña, transmitiéndola cuando el lector la solicita.

Está compuesto principalmente por un microchip y una antena (Ver **Figura 7**). Adicionalmente puede incorporar una batería para alimentar sus transmisiones o incluso algunas etiquetas más sofisticadas pueden incluir una circuitería extra con funciones adicionales de entrada/salida, tales como registros de tiempo u otros estados físicos que pueden ser monitorizados mediante sensores apropiados como sensores de temperatura, humedad, etc.

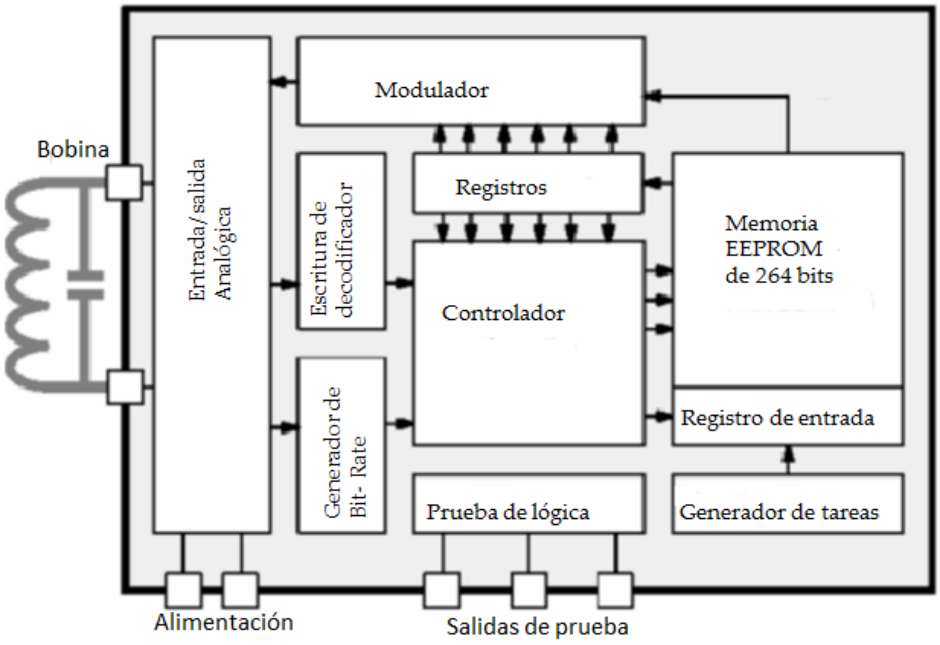



**Figura 7. Esquema de un transponder de RFID**

El microchip incluye (Portillo García, Bermejo Nieto, Bernardos Barbolla, & Martínez Salles, págs. 35, 36, 37):

- Una circuitería analógica que se encarga de realizar la transferencia de datos y de proporcionar la alimentación.

- Una circuitería digital que incluye:
  - La lógica de control
  - La lógica de seguridad
  - La lógica interna o microprocesador

- Una memoria para almacenar los datos. Esta memoria suele contener:

  - Una **ROM** (*Read Only Memory*) o memoria de sólo lectura, para alojar los datos de seguridad y las instrucciones de funcionamiento del sistema

  - Una **RAM** (*Random Access Memory*) o memoria de acceso aleatorio, utilizada para facilitar el almacenamiento temporal de datos durante el proceso de interrogación y respuesta

  - **Una memoria de programación no volátil**. Se utiliza para asegurar que los datos están almacenados, aunque el dispositivo esté inactivo. Típicamente suele tratarse de una EEPROM (Electrically Erasable Programmable ROM). Este tipo de memorias permite almacenar desde 16 bytes hasta 1 Mbyte, posee un consumo elevado, un tiempo de vida (número de ciclos de escritura) limitado (de entre 10.000 y 100.000) y un tiempo de escritura de entre 5 y 10 ms. Como alternativa aparece la FRAM (Ferromagnetic RAM) cuyo consumo es 100 veces menor que una EEPROM y su tiempo de escritura también es menor, de aproximadamente 0.1 µs, lo que supone que puede trabajar prácticamente en tiempo real. En

sistemas de microondas se suelen usar una SRAM (Static RAM). Esta memoria posee una capacidad habitualmente entre 256 bytes y 64 kbytes (aunque se puede llegar a 1Mbyte) y su tiempo de escritura es bajo, pero en contrapartida necesita una batería adicional para mantener la información

- **Registros de datos (buffers)** que soportan de forma temporal, tanto los datos entrantes después de la demodulación como los salientes antes de la modulación. Además, actúa de interfaz con la antena

La información de la etiqueta se transmite modulada en amplitud (ASK, *Amplitude Shift Keying*), frecuencia (FSK, *Frequency Shift Keying*) o fase (PSK, *Phase Shift Keying*). Es decir, para realizar la transmisión se modifica la amplitud, frecuencia o fase de la señal del lector. Típicamente la modulación más utilizada es la ASK debido a su mayor sencillez a la hora de realizar la demodulación. La frecuencia utilizada por el transpondedor, en la gran mayoría de los casos, coincide con la emitida por el lector. Sin embargo, en ocasiones se trata de una frecuencia subarmónica (submúltiplo de la del lector) o incluso de una frecuencia totalmente diferente de la del lector (no armónica).

Existen dos mecanismos por los cuales es posible transferir la potencia de la antena del lector a la antena de la etiqueta, para que ésta transmita su información: acoplamiento inductivo y propagación por ondas electromagnéticas. Estos dos tipos de acoplamiento dependen de si se trabaja en campo cercano o en campo lejano. (Portillo García, Bermejo Nieto, Bernardos Barbolla, & Martínez Salles, pág. 37). En la **Tabla 2** se resumen las principales características de ambos modos

Tabla 2. Principales características de los modos de propagación

| Propagación/acoplamiento inductivo | Propagación por ondas EM |
|---|---|
| Trabaja en el campo cercano: cobertura baja. | Trabaja en el campo lejano: cobertura mayor. |
| Hay que considerar la orientación de la antena. | La orientación de la antena es indiferente. |
| Suele trabajar a bajas frecuencias. | Suele trabajar a altas frecuencias. |
| Suele utilizar etiquetas pasivas. | Suele utilizar etiquetas activas. |
| Es muy sensible a las interferencias electromagnéticas. | Necesita regulación. |

*Fuente: Tecnología de Identificación por Radiofrecuencia: Aplicaciones en el ámbito de la salud.*

#### 2.3.1.1 Etiquetas Pasivas

Son transponders que no necesitan baterías adicionales, ya que únicamente se alimentan de la energía del campo generado por el lector. Para las etiquetas pasivas, la energía que necesitan para transmitir la información que contienen, proviene en su totalidad de la señal generada por el lector (Ciudad Herrera & Casanovas , pág. 15), razón por la cual necesitan estar por lo menos unos 10 cm.

La ausencia de batería provoca que los transpondedores pasivos sean mucho más ligeros, pequeños, flexibles y baratos que los activos, hecho que redunda en que puedan ser diseñados en una amplia gama de formas. Además, ofrecen un tiempo de vida prácticamente ilimitado. Como contrapartida, poseen unos radios de cobertura menores y requieren más cantidad de energía procedente del interrogador para poder transmitir los datos. También poseen restricciones a la hora de almacenar los datos y no funcionan demasiado bien en ambientes con interferencias electromagnéticas. Asimismo, su sensibilidad y

orientación están limitadas por la potencia disponible. (Portillo García, Bermejo Nieto, Bernardos Barbolla, & Martínez Salles, pág. 38)

Sin embargo, a pesar de estas limitaciones, las etiquetas pasivas ofrecen mejores ventajas en términos de coste y longevidad. En la **Figura 8** podemos observar los diferentes tipos de transponders pasivos existente en el mercado.


**Figura 8. Las diferentes características externas de un transponders RFID pasivo**

Existe un tipo especial de etiqueta pasiva que sí incorpora una batería, pero la misión de ésta es alimentar la circuitería interna del microchip con el propósito de obtener mayor rango de lectura. Nunca se utiliza esa energía para transmitir. Este tipo de etiquetas dependen del módulo lector para que este pueda activarse y posteriormente transmitir los códigos. En la **Figura 9** se ilustran las características externas de un transponder semipasivos.


**Figura 9. Características externas de los transponders RFID semipasivos**

### 2.3.1.2 Etiquetas Activas

Son transponders que necesitan el apoyo de baterías adicionales, ya que no tienen suficiente energía con la que proporciona el lector. Este tipo de etiqueta tiene la ventaja de poseer un alcance mayor de comunicación e incluso no necesitan que el lector sea quién inicie la comunicación. Además, permiten habitualmente procesos de lectura y reescritura enviando previamente instrucciones al lector y la utilización de memorias más grandes (existen etiquetas con 1Mb de memoria). Por el contrario, ofrecen una vida útil limitada (menos de diez años), dependiendo del tipo de batería y de las temperaturas a las que opera. También hay que destacar que su coste es bastante elevado, su precio suele ser 5 veces más alto. De esta forma aparecen nuevas aplicaciones para sistema RFID gracias a este tipo de etiquetas alimentadas por baterías. (Ciudad Herrera & Casanovas , pág. 15)

Una ventaja adicional que presentan frente a las etiquetas pasivas es que pueden usarse parar gestionar otros dispositivos, como pueden ser los sensores.

En términos generales las etiquetas RFID activas permiten un radio de cobertura mayor, mejor inmunidad al ruido y tasas de transmisión más altas cuando se trabaja a alta frecuencia.

Existen dos tipos de etiquetas activas (Portillo García, Bermejo Nieto, Bernardos Barbolla, & Martínez Salles, pág. 38):

- Aquellas que normalmente se encuentran desactivadas (modo reposo) y se activan (despiertan) cuando un lector las interroga. De esta forma se ahorra batería
- Aquellas que periódicamente envían señales, aunque un lector no las interrogue. Operan a frecuencias más bajas y a menores tasas de transferencias, para ahorrar batería

En la **Figura 10** se ilustran las presentaciones externas de un transponder activo.


**Figura 10. Características externas de los transponders RFID de estado activo**

Resumimos la comparativa de las principales características en la siguiente tabla:

**Tabla 3. Etiquetas activas vs Etiquetas activas**

| | Etiquetas Activas | Etiquetas Pasivas |
|---|---|---|
| Incorporan Batería | Sí | No |
| Costo | Mayor | Menor |
| Tiempo de Vida | Limitado | Casi Ilimitado |
| Cobertura | Mayor | Menor |
| Capacidad de Datos | Mayor | Menor |

*Fuente: Tecnología de Identificación por Radiofrecuencia: Aplicaciones en el ámbito de la salud*

### 2.3.2 Lector (Receptor)

Un lector o interrogador es el dispositivo que proporciona energía a las etiquetas, lee los datos que le llegan de vuelta y los envía al sistema de información (Ver **Figura 11**). Asimismo, también gestiona la secuencia de comunicaciones con el lector. Con el fin de cumplir tales funciones, está equipado con un módulo de radiofrecuencia (transmisor y receptor), una unidad de control y una antena. Además, el lector incorpora un interfaz que permite enviar los datos del transpondedor al sistema de cómputo. (Portillo García, Bermejo Nieto, Bernardos Barbolla, & Martínez Salles, pág. 46)

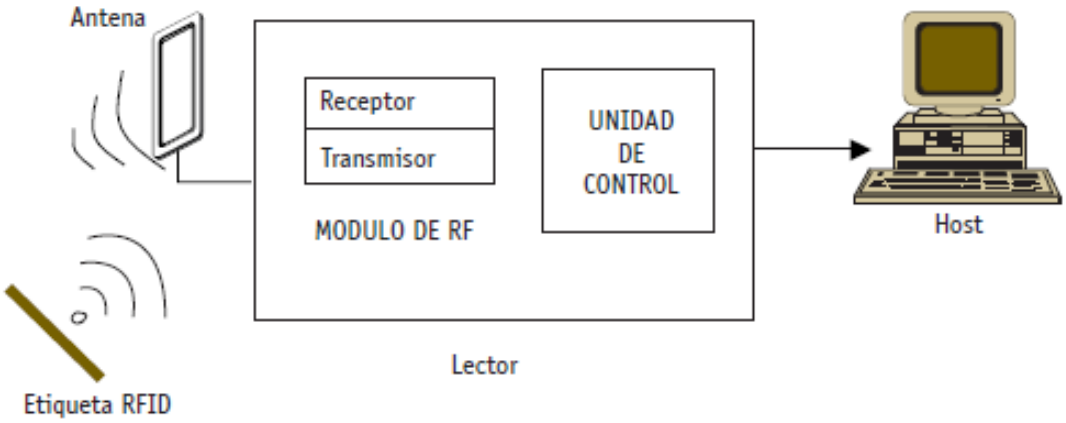



**Figura 11. Esquema de un lector de RFID**

*Fuente: Tecnología de Identificación por Radiofrecuencia: Aplicaciones en el ámbito de la salud*

El lector puede actuar de tres modos (Portillo García, Bermejo Nieto, Bernardos Barbolla, & Martínez Salles, págs. 46, 47):

- Interrogando su zona de cobertura continuamente, si se espera la presencia de múltiples etiquetas pasando de forma continua
- Interrogando periódicamente, para detectar nuevas presencias de etiquetas
- Interrogando de forma puntual, por ejemplo, cuando un sensor detecte la presencia de una nueva etiqueta

Los componentes del lector son, como podemos ver en la **Figura 11**, el módulo de radiofrecuencia (formado por receptor y transmisor), la unidad de control y la antena. A continuación, se procede a describir un poco más cada uno de estos elementos.

- El **módulo de radiofrecuencia**, que consta básicamente de un transmisor que genera la señal de radiofrecuencia y un receptor que recibe, también vía radiofrecuencia, los datos enviados por las etiquetas. Sus funciones por tanto son:
  - Generar la señal de radiofrecuencia para activar el transpondedor y proporcionarle energía
  - Modular la transmisión de la señal para enviar los datos al transpondedor
  - Recibir y demodular las señales enviadas por el transpondedor

- La **unidad de control**, constituida básicamente por un microprocesador. En ocasiones, para aliviar al microprocesador de determinados cálculos, la unidad de control incorpora un circuito integrado ASIC (Application Specific Integrated Circuit), adaptado a los requerimientos deseados para la aplicación. La unidad de control se encarga de realizar las siguientes funciones:

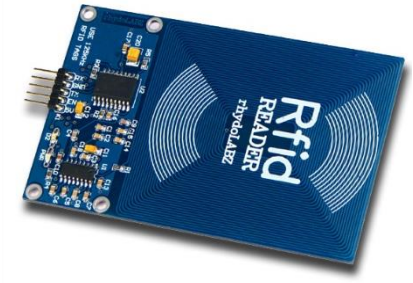
  - Codificar y decodificar los datos procedentes de los transpondedores
  - Verificar la integridad de los datos y almacenarlos
  - Gestionar el acceso al medio: activar las etiquetas, inicializar la sesión, autenticar y autorizar la transmisión, detectar y corregir errores, gestionar el proceso de multilectura (anticolisión), cifrar y descifrar los datos, etc
  - Comunicarse con el sistema de cómputo, ejecutando las órdenes recibidas y transmitiéndole la información obtenida de las etiquetas

Una de las funciones más críticas que debe realizar la unidad de control es gestionar el acceso al medio. Cuando se transmite información mediante una tecnología que no requiere contacto físico, existe la posibilidad de que aparezcan interferencias que provoquen cambios indeseados a los datos transmitidos y, en consecuencia, errores durante la transmisión. Para evitar este problema se utilizan procedimientos de comprobación (checksum). Los más comunes son la comprobación de bits de paridad, comprobación de redundancia longitudinal (LRC, Longitudinal Redundancy Check) y comprobación de redundancia cíclica (CRC, Cyclic Redundancy Check).

El número de etiquetas que un lector puede identificar en un instante de tiempo depende de la frecuencia de trabajo y del protocolo utilizado. Por ejemplo, en la banda de Alta Frecuencia suele ser de 50 tags por segundo, mientras que en la banda de Ultra Alta Frecuencia puede alcanzar las 200 tags por segundo.

- La **antena** del lector es el elemento que habilita la comunicación entre el lector y el transpondedor.

Muchos fabricantes ofrecen sistemas lectores con tecnología adicionada para la validación de los datos recepcionados (Ver **Figura 12.B**), aunque algunos fabricantes ofrecen únicamente los receptores, para su respectiva implementación (Ver **Figura 12.A**).




**A. Módulo lector RFID integrado con antena y dispositivos de decodificación**


**B. Validador RFID (contiene en su interior todo el sistema necesario para la lectura y validación de transponders)**

**Figura 12. Lector RFID**

### 2.3.3 Sistema de Computo

#### 2.3.3.1 Microcontrolador

Un microcontrolador es un circuito integrado, **Figura 13**, que incorpora en su interior los bloques básicos para formar un sistema embebido o una PC en menor escala, es decir el microcontrolador es un chip en cuyo interior encontramos una CPU, Memoria, Reloj, Puertos de Comunicación y Módulos Periféricos de E/S (Entradas y salidas).

Cada uno de estos bloques internos, cumple una función específica y permite al diseñador un mejor control de los procesos del sistema, el CPU[5] se dice que es un microprocesador en pequeño y de menor potencia, la Memoria que sirve para almacenar el programa a ejecutar, el Reloj provee una señal de sincronización para todas las tareas del sistema, los Puertos de comunicación le permiten al microcontrolador tener comunicación bi-direccional con otros microcontroladores o un PC y los Módulos Periféricos de E/S que permiten el intercambio de información de tipo digital o análoga con el exterior del sistema. (Flores Cortez, 2009, pág. 1)

Para explotar sus funciones para la cual está diseñada, es necesario utilizar un lenguaje de programación; estos lenguajes van desde el más bajo nivel (lenguaje máquina) hasta el más alto (lenguaje C), mediante el cual se lograr activar sus diferentes módulos que llevan integrados en su interior.

[5] CPU: Unidad Central de Procesamiento.

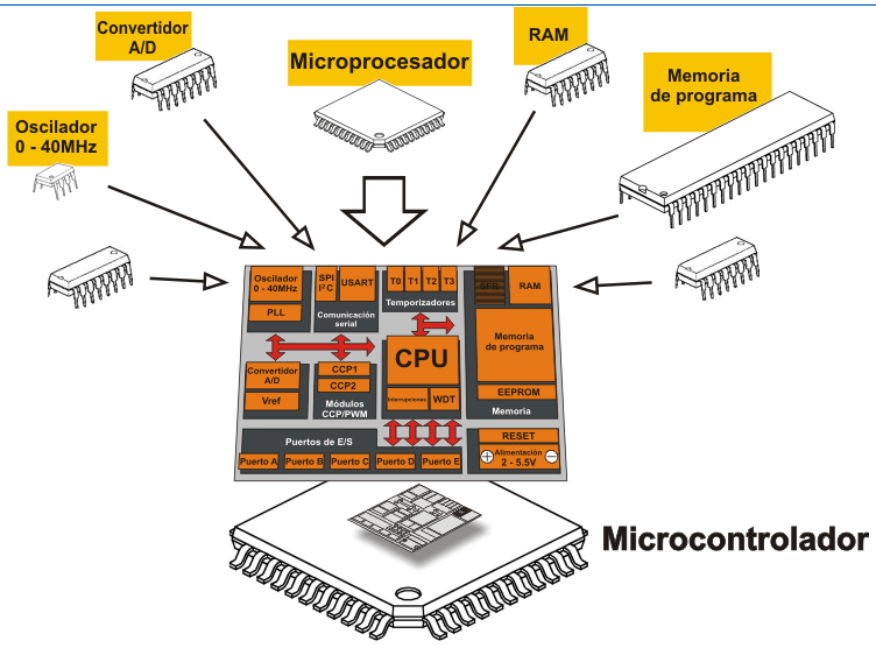



**Figura 13. Estructura de un Microcontrolador**

*Fuente: MikroElektronika*

### 2.3.4 Fuente de poder

Este componente es el encargado de suministrar energía eléctrica a los componentes que conforman el sistema para mantener su funcionamiento. En la **Figura 14** observamos los componentes que conforman un sistema RFID básico.

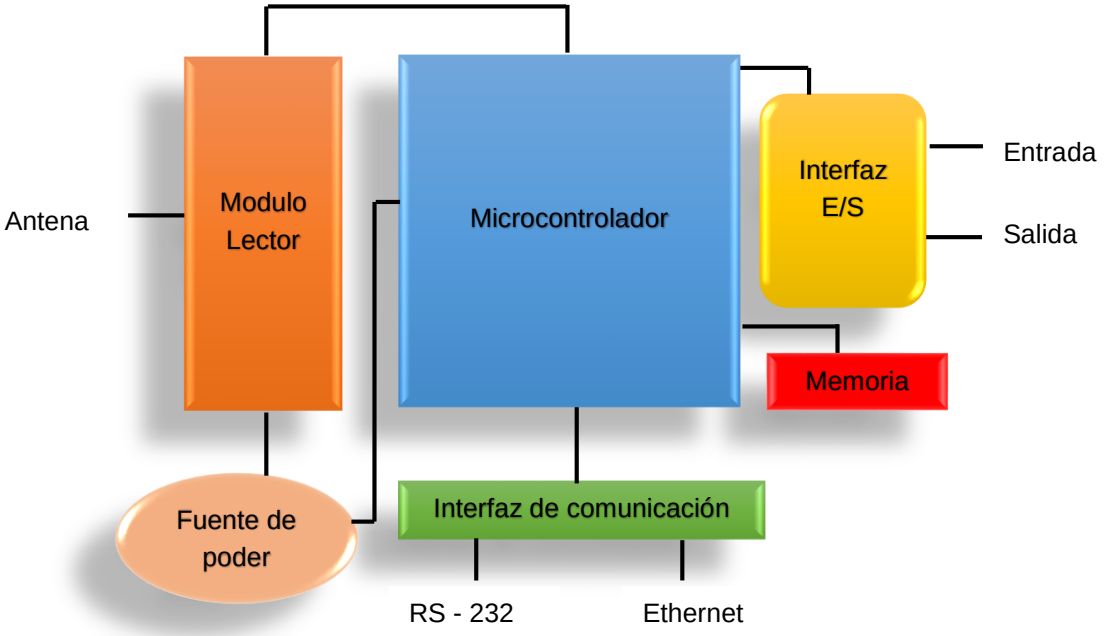



**Figura 14. Estructura y módulos que conforman un sistema RFID**

## 2.4 Estándares

La tecnología RFID debe cumplir con estándares creados por organizaciones como ISO y EPC (Weinstein, 2005, págs. 28, 29):

a) ISO[6]

ISO tiene 3 estándares para RFID: ISO 14443 (para sistemas sin contacto), ISO15693 (para sistema de proximidad) e ISO 18000 (para especificar la interfaz aérea para una variedad de aplicaciones)

b) EPC

EPC global es una organización sin fines de lucro que ha desarrollado una amplia gama de estándares para la identificación de productos. Mediante este estándar podemos seleccionar tags como si estuviéramos accediendo a una base de datos. Esto implica que, si tengo una producción completa de millones de objetos ya empaquetados, y necesito localizar una parte de mi producción pequeña, por ejemplo, podré ordenar al reader que sólo seleccione dichos tags. Luego, un tag no será más que un objeto en una base de datos que podremos leer, escribir, es decir, modificarlo de estado

c) ANSI[7]

Se trata de un organismo privado con fines no lucrativos que administra y coordina el organismo de estándares americano

d) ONS

EPC global ha desarrollado un sistema llamado ONS (Object Naming Service) que es similar al DNS (Domain Name Service) utilizado

[6] ISO: International Organization for Standarization
[7] ANSI: American National Standards Institute

en Internet. ONS actúa como un directorio para las organizaciones que desean buscar números de productos en Internet

## 2.5 Frecuencias

El fenómeno físico que aplica en el funcionamiento de las tecnologías de RFID es la transmisión y recepción de ondas electromagnéticas que contienen información del objeto a identificar. En la **Figura 15** se muestra las frecuencias utilizadas en RFID en el espectro radioeléctrico.

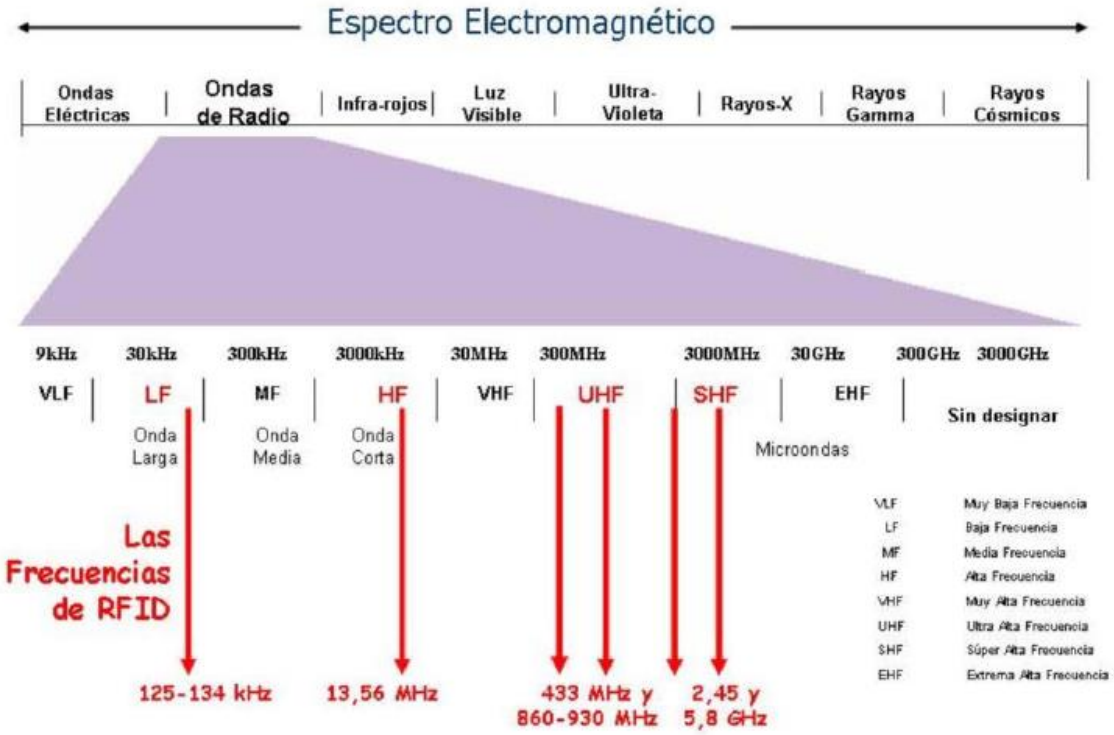



**Figura 15. Representación del Espectro Electromagnético**

La tecnología RFID opera en las bandas de frecuencias bajas (LF), altas (HF), ultra altas (UHF), y súper altas (SHF). Las longitudes de onda para las frecuencias bajas y altas son del orden de centímetros y para las frecuencias ultra y súper altas son en metros.

Las frecuencias de RFID pueden ser divididas en 4 rangos (Portillo García, Bermejo Nieto, Bernardos Barbolla, & Martínez Salles, págs. 57, 58, 59, 60, 61, 62):

1. **Frecuencia de 135 KHz**, Los sistemas RFID de baja frecuencia suelen emplear etiquetas pasivas y cubre distancias menores de 0.5 metros. Su funcionamiento se basa en el acoplamiento inductivo.

   Principales características:

   - La capacidad de datos es baja, de alrededor de 64 bits
   - Las tasas de transferencia de datos son bajas, típicamente entre 200 bps y 1 kbps
   - Al tratarse de un sistema inductivo, el campo magnético decrece muy rápidamente con la distancia. Las etiquetas pasivas suelen poseer una cobertura pequeña, que alcanza como mucho los 0.5 metros, aunque depende también de la potencia disponible en la etiqueta. Las etiquetas activas pueden superar los 2 metros
   - La penetración en materiales no conductores es buena, pero no funcionan bien con materiales conductores
   - Aptas para aplicaciones que requieran leer poca cantidad de datos y para pequeñas distancias. Por ejemplo: control de accesos, identificación de animales, gestión de bienes, identificación de vehículos y contenedores, y como soporte a la producción

2. **Alta Frecuencia (13.56 MHz).** Esta frecuencia es muy popular y cubre distancias de 1cm a 1.5 m. Típicamente las etiquetas que trabajan en esta frecuencia son de tipo pasivo, su principio de funcionamiento básico, al igual que en baja frecuencia, se basa en el acoplamiento inductivo.

   Principales características:

- Las etiquetas (pasivas) suelen poseer capacidades que van desde 512 bits hasta 8 kbits
- Típicamente la velocidad de datos suele ser de unos 25 Kbps hasta 100 Kbps
- Típicamente las etiquetas pasivas poseen un radio de cobertura de alrededor de 1 metro
- Posee una buena penetración en materiales y líquidos no conductores. Sin embargo, no funciona bien cuando existen materiales metálicos en la zona de lectura
- Al igual que en bajas frecuencias, los sistemas de alta frecuencia son aptos para aplicaciones que requieran leer poca cantidad de datos y a pequeñas distancias. Es el caso de la gestión de maletas en aeropuertos, bibliotecas y servicios de alquiler, seguimiento de paquetes y aplicaciones logísticas en la cadena de suministros

3. **Ultra Alta Frecuencia (433 MHz, 860 MHz, 928 MHz)**. Los sistemas RFID que trabajan a Ultra Alta Frecuencia basan su funcionamiento en la propagación por ondas electromagnéticas para comunicar los datos y para alimentar la etiqueta en caso de que ésta sea pasiva.

   Principales características:

   - Están disponibles etiquetas activas y pasivas con capacidades típicas desde los 32 bits hasta los 4 Kbits
   - La velocidad de transferencia de datos está típicamente alrededor de 28 kbps pero también están disponibles velocidades mayores
   - Las etiquetas de UHF pasivas pueden alcanzar una cobertura de 3 ó 4 metros. Trabajando con etiquetas activas y a la frecuencia más baja, 433 MHz, la cobertura puede alcanzar los 10 metros

- Estas frecuencias no pueden penetrar el metal ni los líquidos a diferencia de las bajas frecuencias, pero pueden trasmitir a mayor velocidad y por lo tanto son buenos para leer más de una etiqueta a la vez
- Es apta para aplicaciones que requieran distancias de transmisión superiores a las bandas anteriores, como en la trazabilidad y seguimiento de bienes y artículos, y logística de la cadena de suministros

4. **Microondas (2.45-5.8GHz).** Cuenta con una velocidad de trasferencia de datos de 100 kbps a 1 Mbps. La ventaja de utilizar un intervalo tan amplio de frecuencias es su resistencia a los fuertes campos electromagnéticos, producidos por motores eléctricos, por lo tanto, estos sistemas son utilizados en líneas de producción de automóviles. Sin embargo, estas etiquetas requieren de mayor potencia y son más costosas, pero es posible lograr lecturas a distancias de hasta 1 y 2 metros para dispositivos pasivos y hasta 15 metros o más, para dispositivos activos. Esta frecuencia apta para aplicaciones que requieran alta cobertura y velocidades de transmisión elevadas. Por ejemplo: automatización en la fabricación, control de accesos, peaje de carreteras, logística de la cadena de suministros y aplicaciones logísticas militares.

A continuación, en la siguiente figura se realiza una comparativa de las características de las etiquetas, dependiendo del intervalo de frecuencia de trabajo:

| *Parámetros* | *Baja frecuencia (<135 KHz)* | *Alta frecuencia (13,56 MHz)* | *Ultra alta Frecuencia (433 MHz, 860 MHz, 928 MHz)* | *Frecuencia microondas (2,45 GHz, 5,8 GHz)* |
|---|---|---|---|---|
| Cobertura | Menor | ⟷ | ⟷ | Mayor |
| Tamaño de la etiqueta | Mayor | ⟷ | ⟷ | Menor |
| Velocidad de lectura de datos | Menor | ⟷ | ⟷ | Mayor |
| Lectura en presencia de líquidos o metales | Mejor | ⟷ | ⟷ | Peor |
| Lectura en presencia de interferencias EM | Peor | ⟷ | ⟷ | Mejor |

**Figura 16. Comparativa de las características asociadas a cada rango de frecuencia**

*Fuente: Tecnología de Identificación por Radiofrecuencia: Aplicaciones en el ámbito de la salud*

## 2.6 Ventajas de la Identificación por radiofrecuencia

A continuación, se describen las principales ventajas de la tecnología de RFID en cuanto a seguridad, línea de vista, velocidad de lectura, mantenimiento, reescritura, entre otras:

**Seguridad.** Una de las ventajas que tiene esta tecnología es que debido al diseño al que esta hechas las etiquetas RFID, el código que cuentan estas tarjetas es difícil de duplicar fácilmente. Cada una posee un código distinto y no permite que varios usuarios puedan tener una tarjeta duplicada. Esta característica es ideal para situaciones de máxima seguridad y alta tecnología; lo que lo diferencia de otros sistemas como lo son los de banda magnética o código de barras, donde la duplicación de tarjetas es bastante frecuente.

**Sin necesidad de alineación o línea vista.** La flexibilidad que ofrece esta tecnología hace que sea más fácil y practico la identificación. La razón es porque no necesita que la tarjeta sea pasada por una ranura en el sentido correcto, lo que le da una mayor agilidad y practicidad de uso. Esto vendría siendo una ventaja en sistemas donde se requiere monitorear objetos en diferentes puntos.

**Alta velocidad**. Múltiples dispositivos pueden ser leídos simultáneamente a gran velocidad, esto puede ahorrar tiempo si se compara con otras tecnologías, en las que es necesario alinear los dispositivos para leerlos uno por uno.

**Tarjetas sin desgaste**. Al no tener fricción la tarjeta con el lector por lo cual no se desgasta y su vida útil es prolongada. Esto permite su reutilización tras asignarlas, al personal de nuevo ingreso. El resultado es la optimización de recursos. Las tarjetas de proximidad vienen de varias formas. La más difundida y estándar es una de plástico bastante rígido, que está preparado para que se le pueda personalizar por medio de una impresión.

**Reescribible**. Algunos tipos de etiquetas RFID, pueden ser leídas y escritas en múltiples ocasiones. En caso de que se aplique a componentes reutilizables, puede ser una gran ventaja.

## 2.7 RFID Control de Acceso

Para lograr controlar el acceso al personal autorizado, es necesario dotar de una identificación (transponder) a cada usuario, en la **Figura 17.A** se aprecia como interactúa el sistema (RFID) con el usuario. Como podemos observar no existe ningún tipo de contacto entre el usuario y el modulo lector, pero se debe de aclarar que esto se da si el transponder es del tipo activo o semipasivos. En el caso de utilizar transponder de estado pasivo, se necesitará cierto aproximamiento del usuario hacia el modulo lector (Ver **Figura 17.B**).


A B

**Figura 17. Interacción de usuario con el validador RFID**

# CAPITULO 3. ANALISIS Y PRESENTACION DE RESULTADOS

## 3.1 ANÁLISIS

En esta etapa se realiza un análisis de la problemática y los requerimientos que solicita el programa UNI Online, con el objetivo de proponer una solución electrónica que mejore las condiciones de seguridad del departamento.

### 3.1.1 Inspección Técnica

El programa UNI-Online no dispone de ningún sistema de control de acceso en sus instalaciones, por lo cual, se precisa de un sistema que garantice la protección de equipos y documentos que dispone el personal.

### 3.1.2 Requerimientos del Sistema de Seguridad

Desde hace años el programa UNI-Online recibe para el desarrollo de sus actividades de virtualización en lo educativo equipos de costo considerable y para la protección de los mismos es necesario la implementación de un sistema eficiente que cumpla con las siguientes características:

- Controlar el acceso por medio de un punto de control donde el personal autorizados deben autentificarse
- Registrar los eventos de acceso y alarma que se producen en el sistema con capacidad de visualizarlos y transferirlos a un PC
- Brindar seguridad de forma continua cuando haya cortes de energía

## 3.2 DISEÑO DEL SISTEMA DE SEGURIDAD ELECTRÓNICA

### 3.2.1 Consideraciones de Diseño

Basándose en los conocimientos anteriormente estudiados de RFID y los requerimientos de la UNI Online, el sistema de control de acceso RFID propuesto deberá tener las siguientes características con el fin de mantener la seguridad en el local:

- Controlar el acceso al personal autorizado por medio de la validación de las tarjetas RFID en el punto de acceso
- Generar alarmas locales para avisar a sus usuarios cuando se presenta un evento que viole la política de seguridad
- El sistema dispondrá de un Display LCD y un teclado para que el usuario pueda hacer la configuración del sistema a través de un menú Usuario/Administrador
- El menú del administrador tendrá el privilegio de agregar, borrar y cambiar contraseña de los usuarios, así como también agregar o eliminar los tags de cada uno. Además, contara con una opción que desactive la alarma
- El menú del usuario contara con las únicas opciones de cambiar contraseña y eliminar tag
- Ambos menús tanto usuarios y Administrador contaran con una opción que permite el ingreso al local, en caso de que pierdan u olviden la tarjeta RFID
- Registrar los eventos de acceso y alarma que se producen en el sistema
- El sistema estará diseñado para que brinde seguridad cuando haya cortes eléctricos.

### 3.2.2 Descripción General del Sistema

El sistema de control de acceso RFID está compuesto por tres subsistemas: Subsistema de control de acceso, Subsistema de comunicación y Subsistema de alarma. Estos operan en conjunto para formar un solo sistema.

Cada subsistema a su vez cuenta con derivaciones más pequeñas dedicadas a realizar tareas específicas que son necesarias para la funcionalidad del sistema completo, a continuación, se describe en términos generales su funcionamiento.

El subsistema de control de acceso permite realizar el proceso de autentificación para poder acceder al local. Su funcionamiento es sencillo, cada usuario cuenta con una etiqueta pasiva RFID, cuando quiera acceder al programa UNI-Online, el usuario necesitara cierto aproximamiento hacia el modulo lector. Una vez recibido el código este será procesado a través del microcontrolador para verificar si aparece registrado en el sistema, en caso de que este registrado se le permitirá el acceso mediante la apertura de la cerradura electromagnética; de lo contrario, el sistema mantendrá bloqueado el punto de acceso. Además, el usuario podrá realizar configuraciones del sistema (como agregar o borrar usuarios) mediante un módulo Display LCD y un teclado.

El subsistema de comunicación se encargará de comunicar el microcontrolador con la computadora, con el objetivo de enviar los eventos (entrada de los usuarios y activación de la alama) a la computadora, para que puedan ser almacenadas en una base de datos.

El subsistema de alarma se encarga de evitar que intrusos descifren las contraseñas de los usuarios autorizados. Cuando este alcance un límite de ingresos inválidos se activará la alarma. En la **Figura 18** se muestra los subsistemas por la cual estará compuesto nuestro sistema.

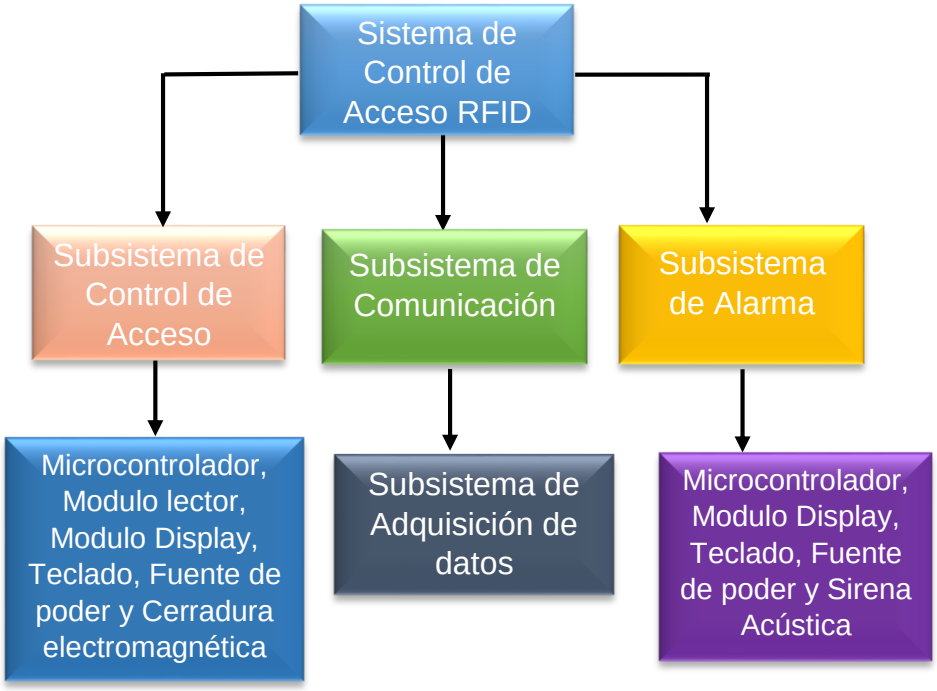



**Figura 18. Sistema de Control de Acceso**

### 3.2.3 Descripción del Diagrama de bloques del Sistema

El sistema de control de acceso RFID está compuesto por:

- Un módulo lector que se encarga de recibir los datos de las etiquetas RFID y enviarla al microcontrolador
- Un microcontrolador que se encarga de procesar la información recibida del lector, y verificar si el código se encuentra almacenado en la memoria del sistema, en caso de existir permitirá el acceso mediante la cerradura electromagnética, de lo contrario mantendrá cerrada el punto de acceso
- Una fuente de poder que se encarga de suministrar energía al sistema
- Un botón de reinicio de contraseña por si el usuario desea salir del local
- Un interfaz de comunicación que permite comunicar el microcontrolador con la computadora

- Un botón de reinicio de contraseña por si al administrador se le olvida la contraseña
- Una interfaz de un Display LCD y un teclado que permite al usuario definir parámetros como agregar o barrar usuarios, además de acceder al local en caso de que se le haya perdido la etiqueta
- Una sirena audible que permite notificar a sus ocupantes que un intruso violo las políticas de seguridad del sistema

En la **Figura 19** se aprecia el diagrama de bloques del sistema de control de acceso utilizando tecnología RFID.

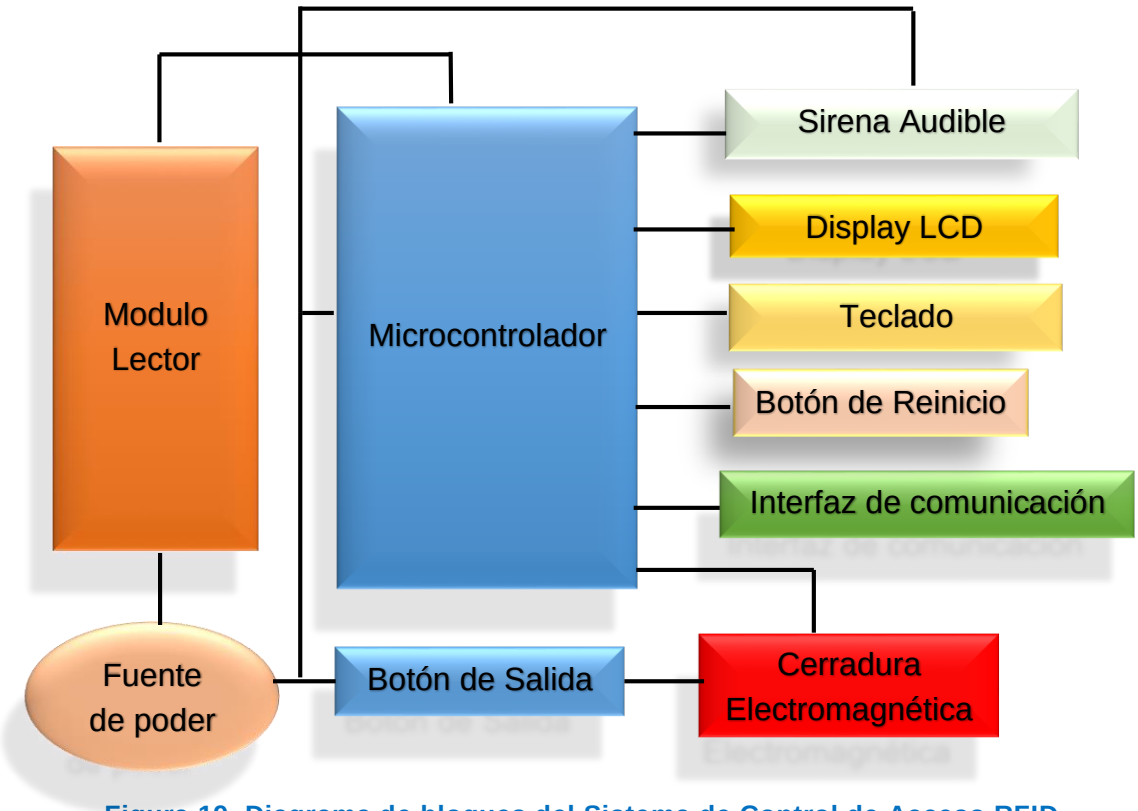



**Figura 19. Diagrama de bloques del Sistema de Control de Acceso RFID**

El subsistema de control de acceso y el subsistema de alarma comparten varios componentes, en la cual destaca:

- Microcontrolador
- Módulo Display LCD
- Teclado
- Fuente de poder

Al compartir estos recursos se logró reducir los tiempos de desarrollo y ejecución, y los costos del prototipo.

### 3.2.4 Diseño del Hardware

En el presente apartado se hará la selección de los componentes que conforman el sistema, seguido se procede a realizar el diseño del hardware con los compontes que fueron seleccionados.

#### 3.2.4.1 Selección de Componentes

A continuación, se describirá los componentes utilizados para desarrollar un sistema electrónico para el control de acceso utilizando tecnología de identificación por radio frecuencia (RFID), enfocándonos principalmente en nuestro plan de diseño, el cual pretende satisfacer la demanda de protección generada por los usuarios.

##### 3.2.4.1.1 Microcontrolador

Para nuestro sistema es necesario disponer de un microcontrolador que sea capaz de ejecutar todas las tareas que el sistema requiere, como lo es la comunicación con el modulo lector RFID, validación de usuarios, apertura de la cerradura electromagnética y activación de alarma.

Algunos aspectos importantes para poder seleccionar un microcontrolador:

- Se requiere un microcontrolador que disponga de comunicación SPI para poder controlar el modulo lector y comunicación serial (UART) para enviar datos a la computadora
- La velocidad de reloj del microcontrolador es importante pueda operar a mayor frecuencia; debido a que entre mayor es la frecuencia del reloj, mayor es la velocidad con que va a ejecutar las tareas programadas
- Es necesario que la capacidad de almacenamiento de datos en la memoria EEPROM sea de gran capacidad, de tal manera que pueda almacenar las credenciales de los usuarios
- Es importante que el microcontrolador disponga de gran cantidad de pines digitales para poder conectar el módulo Display LCD, el modulo lector, el teclado, la interface de la cerradura electromagnética y la alarma. En la **Tabla 4** se aprecia la cantidad de pines digitales que se necesita que disponga el microcontrolador a seleccionar

**Tabla 4. Cantidad de pines digitales que se necesita que disponga el microcontrolador a seleccionar**

| Dispositivos | Cantidad de Pines Digitales al Microcontrolador |
|---|---|
| **Módulo Display LCD** | 6 |
| **Modulo Lector** | 5 |
| **Teclado** | 8 |
| **Interfaz de la cerradura electromagnética** | 1 |
| **Interfaz de la Alarma** | 1 |
| **Botón de reinicio de contraseña** | 1 |
| **Total** | **22** |

**Nota:** No incluye los pines de alimentación de los dispositivos y de tierra común.

Antes de seleccionar un microcontrolador es importante conocer los dos tipos de microcontroladores más populares en el mercado: Los Microcontroladores PIC y Arduino.

Los PIC son una familia de microcontroladores de 8 bits fabricados por Microchip, disponen de una CPU con tecnología RISC[8] y memoria flash[9] para el almacenamiento del Firmware[10]. Mientras que, Arduino es una plataforma de hardware libre para interaccionar con elementos electrónicos. Consta de un microcontrolador de la familia de Atmel, y al igual que los microcontroladores de Microchip, también integra una CPU con tecnología RISC y memoria flash. Ambas familias cuentan con periféricos como Puertos Digitales, Convertidor Analógico-Digital (ADC), PWM, entre otros.

Basado en la descripción anterior, podemos observar que, tanto los microcontroladores PIC, como los microcontroladores Atmel que integra Arduino tienen la misma arquitectura, desde un punto de vista de estructura general. Sin embargo, podemos encontrar algunas diferencias como: Lenguaje de programación, IDE, interfaces para la programación, costo, etc. que se puede resumir en la siguiente tabla:

**Tabla 5. Diferencias entre Arduino y PIC**

| | Arduino | PIC |
|---|---|---|
| **Leguaje de programación** | Lenguaje C | Ensamblador, Lenguaje C (Copiladores de lenguaje de alto nivel) |
| **Copiladores de lenguaje de alto nivel** | Descarga gratis | Pagado |
| | Puerto Serial. | Puerto Serial. |

[8] RISC: Computador con Conjunto de Instrucciones Reducidas.
[9] Memoria Flash: Es una memoria basada en semiconductores, no volátil y reescribible.
[10] Firmware: es un software que maneja físicamente al hardware.

| | | |
|---|---|---|
| **Hardware de programación** | | Lo que lo diferencia de Arduino, es que su programación se realiza con alto voltaje (>5Vdc), lo que hace necesario el uso de circuitos externos que realizan esta conversión de niveles y por lo tanto incrementan la complejidad del circuito programador. |
| **Costo** | Un poco más elevado que el PIC | Un poco más bajo que el Arduino |

**Nota:** Arduino es un sistema que viene ya incorporado un microcontrolador, un interfaz de comunicación USB, su cristal, sus puertos acomodados para dar un uso inmediato, mientras que los microcontroladores PIC requiere de todo esto externamente.

La ventaja de Arduino con respecto a los microcontroladores PIC´s, es que dispone de un software de programación de código abierto, lo que significa que es de libre uso.

En el entorno educacativo, Arduino es más fácil de programar, por lo que, despierta fácilmente el interés. Asimismo, dispone de muy buena documentación, clara y entendible. Mientras que, los microcontroladores PIC's son excelentes pero la documentación es muy poca, incluso la de los compiladores de lenguaje de alto nivel.

Otro aspecto importante que tuve al momento de elegir Arduino, es la flexibilidad con la que se puede cargar el Firmware directamente, sin necesidad de comprar un programador, como es el caso de los PIC, debido a que ya viene incorporado una interfaz que permite programarlo directamente desde la computadora. Lo que facilito mucho a la hora de programarlo; y de esta manera, poder trabajar desde la comodidad de mi casa, sin necesidad de ir de arriba/abajo a la universidad, buscando que alguien me presté un programador para PIC.

Otra ventaja de Arduino, es la magnitud con la que se ha extendido, por lo que, existe una comunidad muy grande de desarrolladores, en la que hay un ambiente muy colaborativo. Para la mayoría de sensores y actuadores comerciales que existen, si el fabricante no ha desarrollado una librería de su dispositivo para Arduino, en la comunidad habrá alguien que la haya desarrollado.

Como principal desventaja de Arduino con respecto a los microcontroladores PIC's, es que nos ata a una arquitectura de diseño, por lo que, si al momento de realizar un proyecto donde requiera muy poca cantidad de espacio, Arduino no es el indicado.

Estas son algunas consideraciones que tome en cuenta al momento de elegir, si bien Arduino fue muy flexible para el aprendizaje y la programación, no significa que se mejor o peor que los microcontroladores PIC's; ambos pueden realizar la misma tarea y resolver un problema.

Debido a que existen múltiples modelos en el mercado con diferentes características, se realizó un estudio se realizó una investigación para preseleccionar los modelos que mejor se ajustan a los requerimientos y consideraciones de diseño del proyecto. En la **Figura 20** se muestran 5 modelos de Arduinos de los cuales se seleccionará el más apropiado.

| | Arduino One | Etherner | Leonardo | Arduino DUE | ADK |
|---|---|---|---|---|---|
| |  |  |  |  |  |
| Microcontroller | ATmega328 | ATmega328 | ATmega32U4 | Atmel SAM3U4E ARM Cortex M3 | ATmega2560 |
| Clock | 16 MHz | 16 MHz | 16 MHz | 96 MHz | 16 MHz |
| Flash Memory | 32 KB | 32 KB | 32 KB | 256 KB | 256 KB |
| SRAM | 2 KB | 2 KB | 3.3 KB | 50 KB | 8 KB |
| Digital I/O Pins | 14 | 14 (10) | 14 | 54 | 54 |
| Analog Pins | 6 | 6 | 6 | 16 (12 bit) | 16 |
| | | Wiznet W5100 Ethernet interface<br>Optional PoE Module<br>**Bring your project online!** | Onboard USB controller<br>**Build your own USB devices!** | Onboard dual-channel DAC<br>**Bringing 32 bit power to Arduino!** | Android ADK Compatible USB Host<br>**Develop your own android accessory!** |

**Figura 20. Principales Características de modelos de Arduinos propuestos**

*Fuente: Arduino*

Se seleccionó Arduino Mega (Ver **Figura 21**) porque tiene gran cantidad de pines digitales, suficientes para poder controlar el Display LCD, Teclado, el Modulo Lector RFID, activación de la cerradura y la alarma.

Si bien, se seleccionó un modelo de gama alta como lo es el Arduino Mega, para este proyecto también pudo haberse usado el Arduino Uno o Leonardo, debido a que disponen de buena capacidad de Memoria flash, Memoria SRAM y velocidad de reloj; sin embargo, no disponen de la cantidad de pines necesarios para conectar los dispositivos a controlar que requiere el sistema. Por lo que, para este proyecto, se estaría desperdiciando mucho recurso con el Arduino Mega.

Por lo mencionado anteriormente Arduino Mega, ha demostrado que cuenta con las características necesarias para el funcionamiento de nuestro sistema.

Las características técnicas del Arduino Mega se explican detalladamente en **Anexo A**.

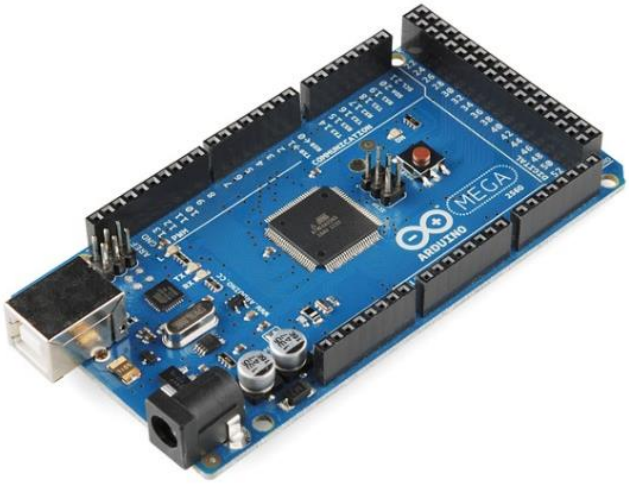



**Figura 21. Arduino Mega.**

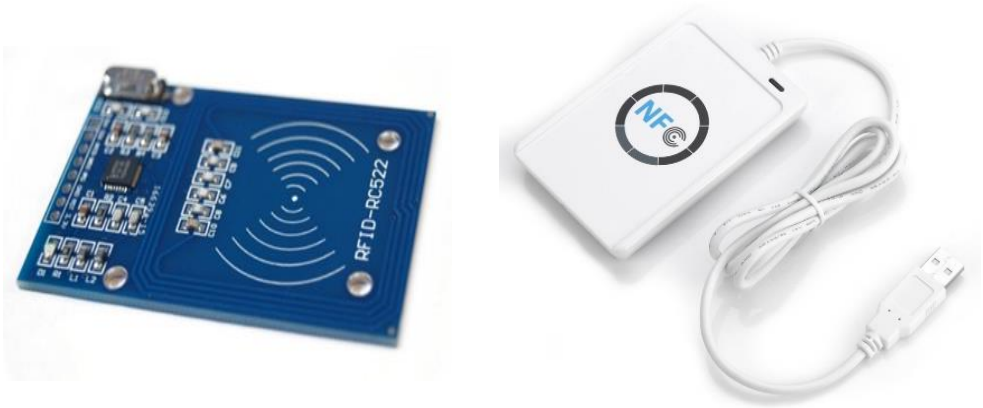
#### 3.2.4.1.2 Lector RFID

Para la selección del lector RFID es necesario que trabaje en la banda de frecuencias altas (HF) debido a su radio de cobertura que va desde algunos centímetros hasta 1m. Unas de las principales ventajas de trabajar a esa frecuencia es que se encuentra en las bandas ISM, por lo que, no necesita licencia de uso (Ver **Anexo C.2.1**).

La siguiente tabla muestra las características principales de los módulos lectores propuestos para este proyecto:

**Tabla 6. Especificaciones Técnicas de los módulos lectores propuestos a) RFID-RC522. b) Yosoo NFC ACR122U.**

| Características | RFID-RC522 | Yosoo NFC ACR122U |
|---|---|---|
| **Lectura y Escritura** | Si | Si |
| **Estándar** | ISO/IEC 14443 A/MIFARE | ISO 14443 Tipo A y B, Mifare, Felica, y ISO/IEC 18092 |
| **Distancia de lectura** | 10 cm máximo (depende de la etiqueta) | 5 cm máximo (depende de la etiqueta) |
| **Comunicación** | SPI, UART, I2C | USB |
| **Velocidad de transferencia** | 10Mbit/s | 424 kbps |
| **Costo (Ya enviado a Nicaragua)** | $ 27.79 USD | $ 62.88 USD |




a) RFID-RC522

b) Yosoo NFC ACR122U

**Figura 22. Módulos lectores propuestos**

Se eligió el módulo lector RFID-RC522 (Ver **Figura 22.a**) por las siguientes razones:

- Se controla a través del protocolo SPI (Consultar el **Anexo E.1**), así como el protocolo UART[11] y I2C[12], por lo que, es compatible con casi cualquier microcontrolador, Arduino o tarjeta de desarrollo.
- Su alta velocidad de transferencia de datos que es de 10Mbit/s.
- Dispone de una librería de control creada por usuarios de la comunidad de Arduino.
- Su bajo coste

La ventaja del módulo lector RFID-RC522 es que puede ser controlado fácilmente por Arduino, por lo que, se puede adecuar sus funcionalidades según las necesidades del sistema, en cambio, el lector Yosoo NFC ACR122U que ya viene establecida su funcionalidad de fábrica, por lo que, no se puede adecuar a las necesidades del sistema.

En **Anexo B** se explica detalladamente el funcionamiento del módulo lector RFID-RC522.

[11] UART: *Transmisor-Receptor Asíncrono Universal*
[12] I2C: *circuito interintegrado*

#### 3.2.4.1.3 Tarjetas RFID

Para la selección de las tarjetas RFID es necesario que sean MIFARE y sigan el estándar ISO/IEC 14443A, para que puedan ser leídas por el modulo lector RFID-RC522.

Las tarjetas que se utilizaron son Mifare 1k S50 y dispone de las siguientes características:

- Operan en la banda de frecuencia de 13.56 MHz y siguen el estándar ISO/IEC 14443A de tarjetas de proximidad.
- Son tarjetas pasivas, por tanto, no requiere de una batería.
- La distancia típica de lectura es de hasta 10 cm.
- La velocidad de transferencia de datos es de 106 kbit/s.
- Dispone de un algoritmo anticolisión, el cual permite leer una tarjeta a la vez.
- Dispone de encriptación de datos en el canal de radio frecuencia.

Estas tarjetas son los elementos que proporcionarán la identificación por radiofrecuencia del usuario que accede al interior del inmueble. Por su construcción, proporcionan alta seguridad y no son fácilmente duplicables. Desde su fabricación es grabada con identificador único (UID) que no es posible de modificar. En la **Figura 23** se aprecia las tarjetas utilizadas para este proyecto.

**Figura 23. Tarjetas Mifare 1k S50**

En la **Tabla 7** se muestran las principales características de las tarjetas Mifare 1k S50.

**Tabla 7. Características de las Tarjetas Mifare 1k S50**

| Características | |
|---|---|
| **Modelo** | Mifare 1k S50 |
| **Frecuencia de Operación** | 13.56 MHz |
| **Capacidad** | 1kb de memoria EEPROM |
| **Ciclos de Escritura** | 100,000 ciclos |
| **Distancia de lectura** | Hasta 10 cm (Dependiendo del módulo lector) |
| **Estándar** | ISO/IEC 14443A |
| **Cifrado** | CRYPTO1 |
| **Velocidad de Transferencia de Datos** | 106 kbit/s |
| **Alta integridad de datos** | 16bit CRC, paridad, bit coding, bit counting; |
| **Anticolisión** | Si |
| **Retención de Datos** | 10 Años |
| **Otras características** | ▪ Transmisión de datos y suministro de energía sin contacto (No requiere batería).<br>▪ Memoria EEPROM DE 1KByte, organizado en 16 sectores con 4 bloques de 16 bytes cada uno.<br>▪ Número de serie único.<br>▪ Retención de datos de 10 años |

En **Anexo C** se explica su funcionamiento con más detalle.

#### 3.2.4.1.4 Módulo Display LCD

Para la selección del módulo Display LCD se requirió que alcance gran cantidad de caracteres para poder visualizar el Menú de Administrador y Usuario.

Para nuestro proyecto se utilizó el módulo Display LCD modelo RioRand que se muestra en la siguiente figura**:**

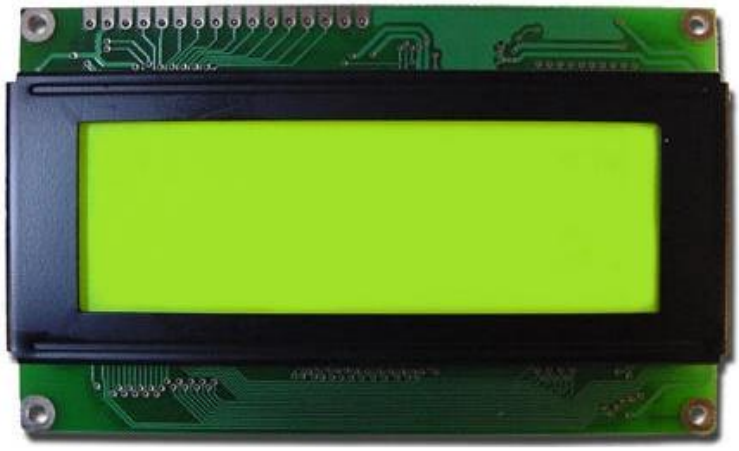


**Figura 24. Modulo LCD HD44780 20x4**

La razón por la que se seleccionó este módulo Display LCD es por lo siguiente:

- Tiene la capacidad de mostrar cualquier carácter alfanumérico, permitiendo representar la información que genera cualquier microcontrolador de una forma fácil y económica.
- La pantalla consta de una matriz de caracteres de 5x8 puntos, distribuidos en 20 columnas y 4 filas (Alcanza hasta 20 caracteres en cada fila), lo que lo hace ideal para visualizar el Menú de Usuario y Administrador de nuestro sistema.
- El proceso de visualización es gobernado por el microcontrolador HD44780 de Hitachi. El microcontrolador se encarga de gestionar el módulo Display:

polarizar los puntos de la pantalla, generar los caracteres, desplazar la pantalla, mostrar el cursor, etc. El programador se despreocupa de todos estos problemas y simplemente necesita conocer una serie de comandos o instrucciones de alto nivel (limpia Display, posiciona cursor, etc.) que le permitirán mostrar mensajes o animaciones sobre la pantalla de forma sencilla. Para comunicarse con el controlador del Display se dispone de una interfaz paralela al exterior, de fácil conexión a otros microcontroladores o microprocesadores.

- Los caracteres que se pueden representar están formados por una matriz de puntos que vienen almacenados en memoria ROM dentro del propio controlador. El fabricante reserva una pequeña zona de memoria RAM donde se pueden definir algunos caracteres especiales, como por ejemplo la letra ñ que no suele venir, o si se desean, pequeños gráficos.

En la **Tabla 8** se muestra las características principales del módulo LCD HD44780.

**Tabla 8. Características Principales del módulo LCD HD44780**

| Características | |
|---|---|
| **Modelo** | RioRand |
| **Microcontrolador** | HD44780 |
| **Columnas** | 20 |
| **Filas** | 4 |
| **Cantidad de caracteres por línea** | 20 |
| **Voltaje de operación** | 5V |
| **Corriente de operación** | 1.6 mA |
| **Otras Características** | ▪ Pantalla de caracteres ASCII, además de los caracteres |

| | |
|---|---|
| | japoneses Kanji, caracteres griegos y símbolos matemáticos.<br><br>▪ Desplazamiento de los caracteres hacia la izquierda o a la derecha<br><br>▪ Memoria de 40 caracteres por línea de pantalla, visualizándose 20 caracteres por línea<br><br>▪ Movimiento del cursor y cambio de su aspecto<br><br>▪ Permite que el usuario pueda programar 8 caracteres<br><br>▪ Pueden ser gobernados de 2 formas principales:<br>o Conexión con bus de 4 bits<br>o Conexión con bus de 8 bits |

En **Anexo D** se describe detalladamente su funcionamiento.

#### 3.2.4.1.5 Cerradura electromagnética

Se eligió la cerradura electromagnética modelo Docooler (Ver **Figura 25**) porque su fuerza de retención es de 180kgs, lo cual permite mantener la puerta cerrada de forma segura, ante cualquier intento humano de abrirla forzadamente. Además de su capacidad de liberar la puerta ante cualquier fallo de energía eléctrica lo hace adecuado para ambientes interiores.

**Figura 25. Cerradura electromagnética Docooler**

La cerradura electromagnética consta de dos piezas fundamentales:

1. Un potente electroimán que se fija en el marco de la puerta.
2. Una placa metálica montada sobre la hoja de la puerta objeto de control.

En la **Figura 26** se aprecia las partes de la cerradura electromagnética.

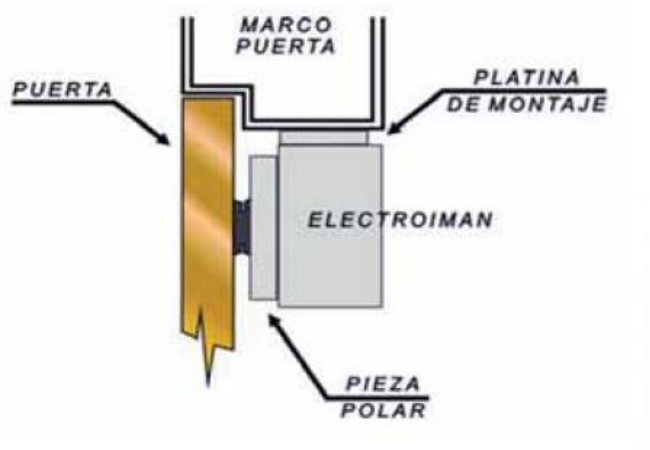



**Figura 26. Partes de la cerradura electromagnética**

El electroimán es el elemento que crea un campo magnético al proporcionarle corriente eléctrica y consta de un núcleo o barra de hierro al que se enrolla un cable barnizado de cobre, creando una bobina. Si a ésta se le suministra corriente eléctrica el núcleo se convierte en un imán capaz de atraer objetos metálicos (hierro), perdiendo sus propiedades magnéticas al cortar la corriente. Este efecto se ha aplicado a la seguridad idóneo para controlar el estado y funcionamiento de puertas, manteniéndolas cerradas de manera segura hasta que se desactiva por medio de un interruptor o de un controlador automático.

En la **Tabla 9** se aprecia las características de cerradura electromagnética Docooler.

**Tabla 9. Características de la cerradura electromagnética Docooler**

| Características | |
|---|---|
| **Modelo** | Docooler |
| **Principio** | Electroimán |
| **Presión** | 180kgs |
| **Voltaje** | DC 12V |
| **Corriente** | 380-430mA |
| **Temperatura de Trabajo** | -10 a 55 °C |

#### 3.2.4.1.6 Sirena Acústica

La selección de este dispositivo se realizó basándose en las características constructivas de la UNI-Online, las cuales determinan el modelo de propagación acústico y por consiguiente los niveles de sonido y frecuencias necesarios para establecer una condición de alarma capaz de notificar a sus ocupantes.

Se utilizó la alarma modelo Foto4easy Mini debido a que tiene los niveles de sonido y frecuencia adecuado para interiores (108 dB). En la **Figura 27** se aprecia la sirena audible Foto4easy Mini.

**Figura 27. Sirena audible Foto4easy Mini**

En la **Tabla 10** se aprecia las características principales de la sirena audible Foto4easy Mini.

**Tabla 10. Características principales del Foto4easy Mini**

| Características | |
|---|---|
| **Modelo** | Foto4easy Mini |
| **Voltaje** | DC 12V |
| **Potencia** | 108 dB |
| **Corriente de consumo máximo de operación** | 110 mA |

#### 3.2.4.1.7 Fuente de Alimentación

Para la alimentación de los componentes que conforman el sistema de control de acceso y alarma se necesitan dos fuentes: Una fuente de 8V que alimente el Arduino y otra de 12V que alimente la cerradura electromagnética y la sirena. A continuación, se presenta una tabla de consumo de corriente los dispositivos:

**Tabla 11. Consumo energético de los componentes que conforman el Sistema de Control de Acceso y Alarma**

| Modelo | Descripción | 8V | 12V |
|---|---|---|---|
| Arduino Mega | Controlador | 130 mA | |
| RioRand | Display LCD | 1.6 mA | |
| Docooler | Cerradura Electromagnética | | 430 mA |
| Foto4easy Mini | Sirena | | 110 mA |
| RFID-RC522 | Módulo RFID | 26 mA | |
| **Total** | | **157.6 mA** | **540 mA** |
| **Consumo Total** | **697.6 mA** | | |

**Nota:** El consumo está previsto en condiciones máximas de demanda.

Para suplir la demanda de corriente de todos los componentes se requirió el uso de la fuente DROK 200188 que se muestra a continuación:

Figura 28. Fuente de alimentación DROK 200188

Tabla 12. Características de la fuente de alimentación

| Características | |
|---|---|
| **Modelo** | DROK 200188 |
| **Voltaje de Entrada** | 110V-240VAC, 50/60Hz |
| **Voltaje de Salida** | 13.5V , 10A |
| **Potencia** | 120W |
| **Función** | UPS |

Se utilizo está fuente para garantizar la distribución de energía de forma segura y eficiente, y poder suplir la demanda de corriente que consumen en conjunto todos los elementos de control de acceso y alarma.

Una ventaja de esta fuente es que dispone de una función de alimentación interrumpida (UPS) cuyo propósito principal es suministrar potencial eléctrico al sistema cuando la tensión de la red comercial falle. Para esto será necesario equipar a la fuente con una batería de plomo y acido, que estén en el rango de los 12V y 12AH. Por otra parte, se podrá cargar la batería automáticamente; cuando la batería se encuentre completamente cargada, deja de cargar y así prolongar la vida útil de la batería.

Esta fuente entrega aproximadamente diez veces más de la potencia requerida, no obstante, se instaló con el propósito de garantizar un amplio margen de servicio para futuras ampliaciones.

#### 3.2.4.1.8 Reguladores de voltaje

Debido a que la fuente tiene un voltaje de salida de 13.5V, para ajustar la tensión de la fuente, y suministrar potencial eléctrico a los componentes que conforman el sistema, se optó usar los reguladores de voltaje de la serie 7808 y 7812 (Ver **Figura 29**).

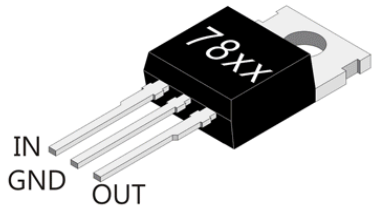
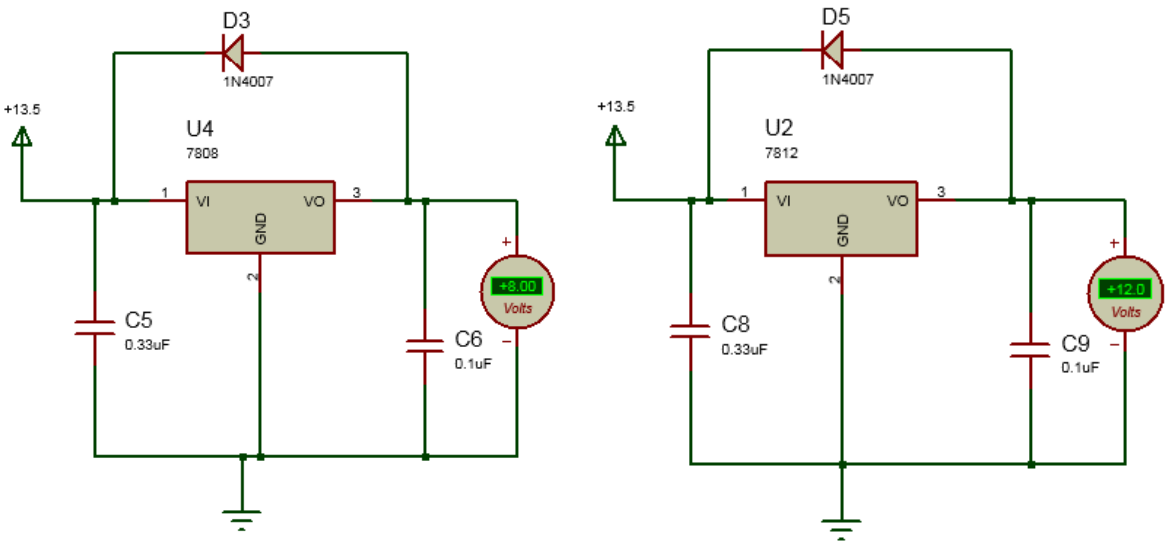



**Figura 29. Regulador de voltaje de la serie 78xx**

Son reguladores lineales de tension, que están diseñados para entregar una tension constante y estable, disponen de tres terminales (Voltaje de entrada, Tierra, Voltaje de salida) y especificaciones similares que sólo difieren en la tensión de salida suministrada. En el caso del regulador 7808, el voltaje de salida es de 8V (Suficiente para suministrar potencial eléctrico al Arduino Mega), y el regulador 7812 es de 12V (Suficiente para suministrar potencial eléctrico a la cerradura electromagnética y la alarma). Ambos proporcionan una corriente máxima de 1A.

La tensión de alimentación (Vin) debe ser un poco más de 2 voltios superior a la tensión de salida (Vout) que entrega el regulador y menor a 35V. Los condensadores tienen los valores recomendados por el fabricante para que proporcionen una función estabilizadora de tensión. Los diodos conectados en la entrada (Con el cátodo del diodo) y salida (Con el ánodo del diodo) de los reguladores es opcional, y sirve para proteger al regulador contra posibles cortos

circuitos en la entrada o alguna regresion de voltaje. En la **Figura 30** se aprecia la conexión de los reguladores de voltaje.




a) Regulador 7808

b) Regulador 7812

**Figura 30. Conexión de Reguladores de voltaje**

### 3.2.4.2 Interface Optoacoplada Entre Dispositivos Digitales y Analógicos

El sistema cuenta con dispositivos de alta potencia que funcionan con un voltaje de alimentación de 12 $V_{DC}$; los cuales deben ser controlado por una etapa digital a través del Arduino Mega (que trabaja por señales TTL que van de 0 a 5 $V_{DC}$).

La manipulación de altas corrientes puede traer daños serios al Arduino Mega. Por lo tanto, es necesario que la interconexión entre ambas etapas (Potencia y Digital) se haga por medio de acoplamiento que permita aislar eléctricamente los dos sistemas. Esto se puede lograr gracias a los optocopladores, mediante el cual se obtiene un acoplamiento óptico y, al mismo tiempo un aislamiento eléctrico. De esta manera se puede manipular los

dispositivos de potencia sin que el Arduino Mega corra algún peligro. En la **Figura 31** se aprecia la conexión de la cerradura electromagnética.

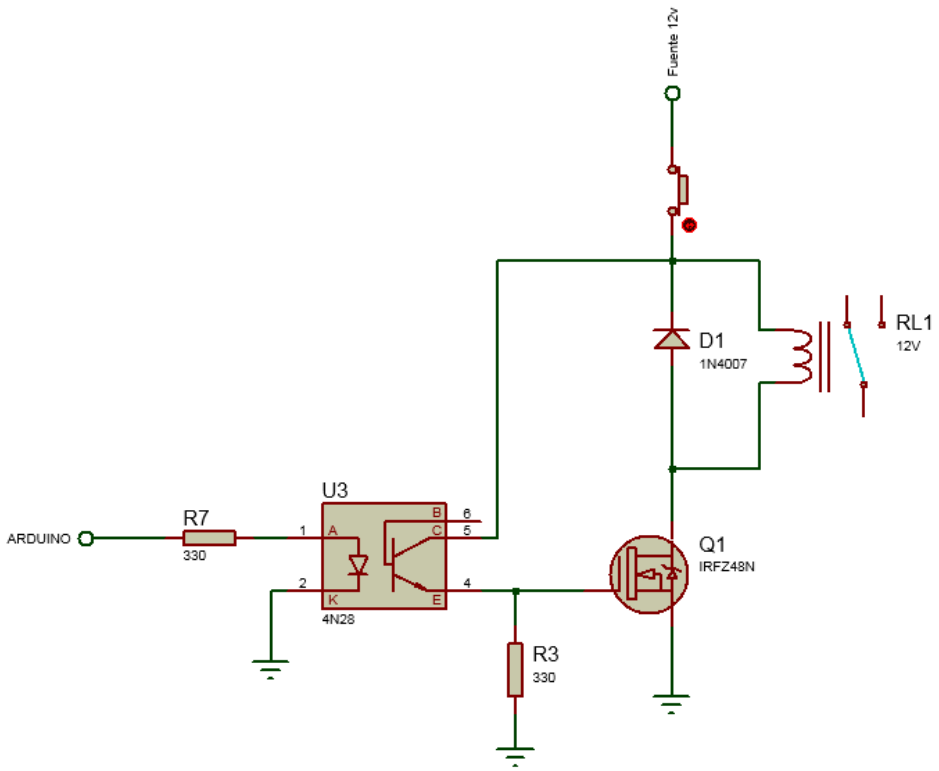



**Figura 31. Conexión de la cerradura electromagnética.**

*Diseño: Ing. José Manuel Arcia Salmerón*

Para el control de la cerradura electromagnética se utilizó el optocoplador 4N28; este optocoplador cuenta con un LED (del inglés, *light-emitting diode*) infrarrojo en su interior que actúa como emisor y un fototransistor BJT que actúa como receptor.

Cuando el Arduino se encuentra en alto (5v) por la resistencia R7 pasa una corriente de:

$$Ley\ de\ Ohm = \frac{5v}{R7} = \frac{5v}{330\Omega} = 15.15\ mA$$

Según la hoja de datos es suficiente para que el LED se active. Al momento que el LED se activa emite una luz infrarroja que satura el fototransistor, esto

significa que el fototransistor va a permitir el paso de la corriente (que actúa como si fuera un circuito cerrado).

Al permitir el paso de la corriente se produce un voltaje en el GATE del MOSFET IRFZ48N de:

$$V_{Gate\ Mosfet} = V_{CE\ Optocoplador} = I * R_3$$

$$V_{Gate\ Mosfet} = (\frac{12v}{330\Omega})(330\Omega)$$

$$V_{Gate\ Mosfet} = (0.0363\ mA)\ (330\Omega)$$

$$V_{Gate\ Mosfet} = 11.979\ V$$

Según la hoja de datos del IRFZ48N para que el MOSFET conmute, el voltaje del GATE y el SOURCE debe ser mayor de 4V ($V_{TH}$). Como el SOURCE va conectado directamente a tierra tenemos 0v.

Entonces, $V_{GS} > V_{TH}$ ; $V_{TH} = 4V$

$$V_{GS} = V_{Gate} - V_{Source} = 11.979V - 0V = 11.979V$$

Por lo tanto, el voltaje del GATE y el SOURCE del MOSFET es de:

$$V_{GS} = 11.979V$$

Así que se cumple la condición $V_{GS} > V_{TH}$ $(11.979V > 4V)$; lo que significa que va a activar la cerradura electromagnética.

Cuando el Arduino Mega se encuentra en bajo (0v) no conduce corriente y, por ende, el LED infrarrojo del 4N28 se encuentra desactivo y el fototransistor se comporta como un circuito abierto, por lo tanto, la cerradura electromagnética se encuentra desactivo.

El diodo que se encuentra en paralelo con la cerradura electromagnética sirve como protección ante los picos de fuerza electromotriz producidos por la carga inductiva de la bobina en el momento de conmutación. (Núñez Zeledón & Zeledón Espinoza, 2013, pág. 59)

Cuando se interrumpe la corriente de excitación externa, la corriente del inductor fluye a través del diodo, la cual es gradualmente disipada por la caída de voltaje del diodo y la resistencia del inductor. Una de las desventajas de usar un diodo de protección simple como protección es que permite que la corriente siga fluyendo por un tiempo, haciendo que el inductor permanezca encendido por un tiempo un poco más largo. (Wikipedia, 2017)

El botón de salida es normalmente cerrado, cuando el usuario presiona el botón interrumpe el flujo de corriente (Actuando como circuito abierto), y como consecuencia la cerradura magnética pierde su propiedad magnética y liberando la puerta. Cuando el botón regresa a su estado permite otra vez el flujo de corriente, activando de nuevo la cerradura electromagnética. En la **Figura 32** se aprecia la conexión de la sirena acústica.

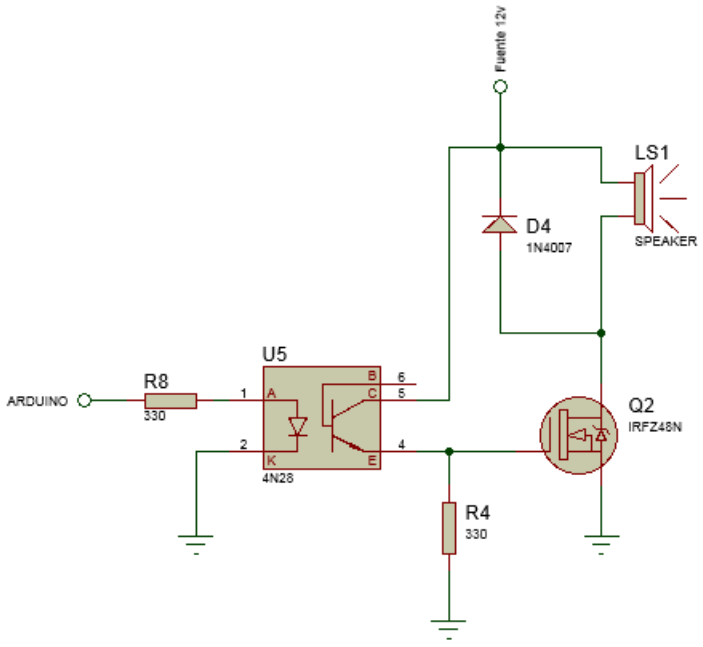



**Figura 32. Conexión sistema de la sirena acústica.**

*Diseño: Ing. José Manuel Arcia Salmerón*

El funcionamiento del control alarma es similar al anterior, con única diferencia que en este caso se activa la sirena.

### 3.2.5 Diseño del Software

En el siguiente apartado se describe el algoritmo de control, el cual contiene las tareas que llevara a cabo el sistema para dar solución al problema.

#### 3.2.5.1 Algoritmo

Basándose en las especificaciones del sistema se realizó el algoritmo (Ver **Figura 33**). El algoritmo funciona de la siguiente manera:

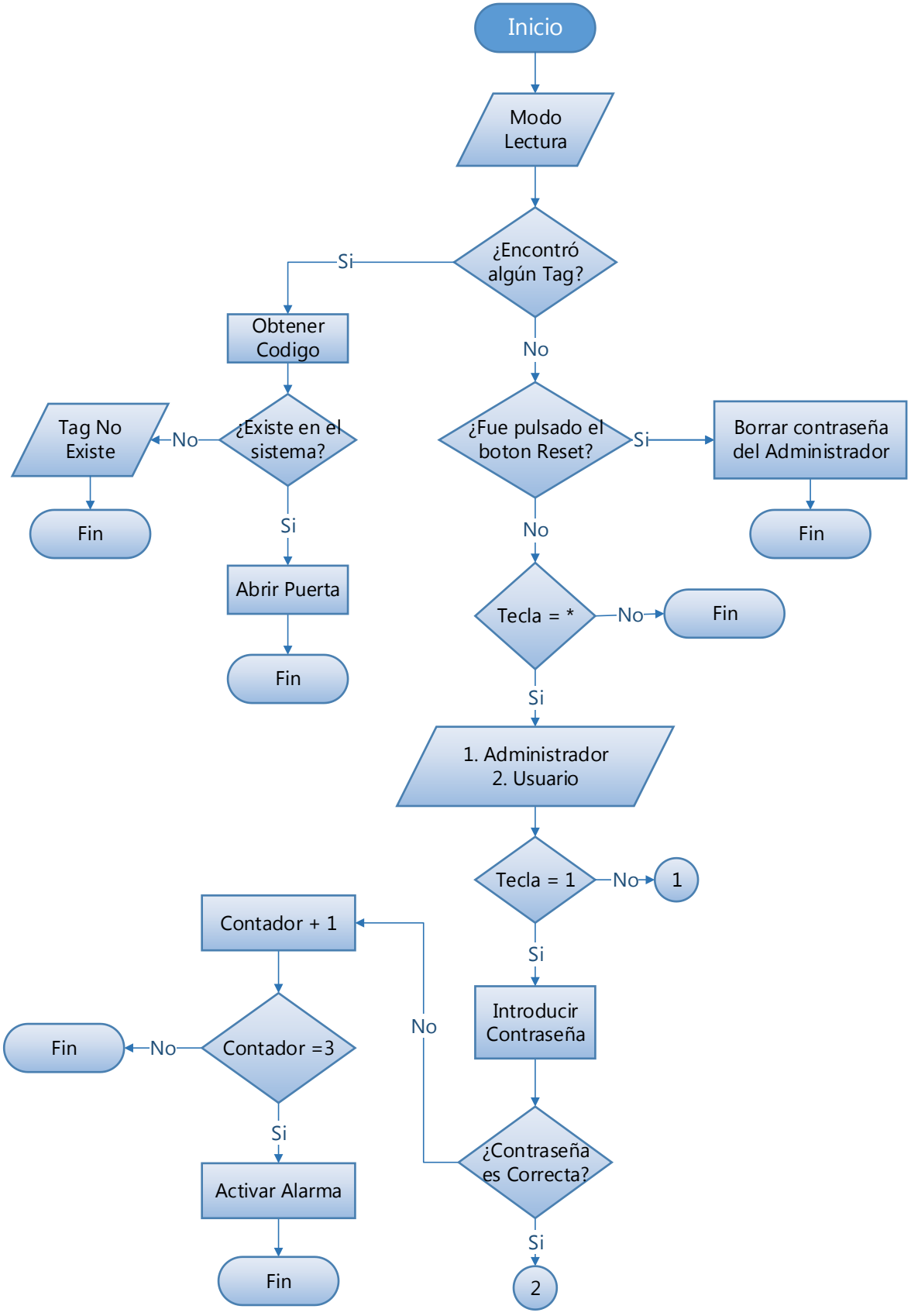
El módulo RFID estará censando constantemente hasta que detecte la presencia de una tarjeta, cuando esta es leída procede a verificar en la base de datos del sistema, si existe, abre la puerta, de lo contrario no.

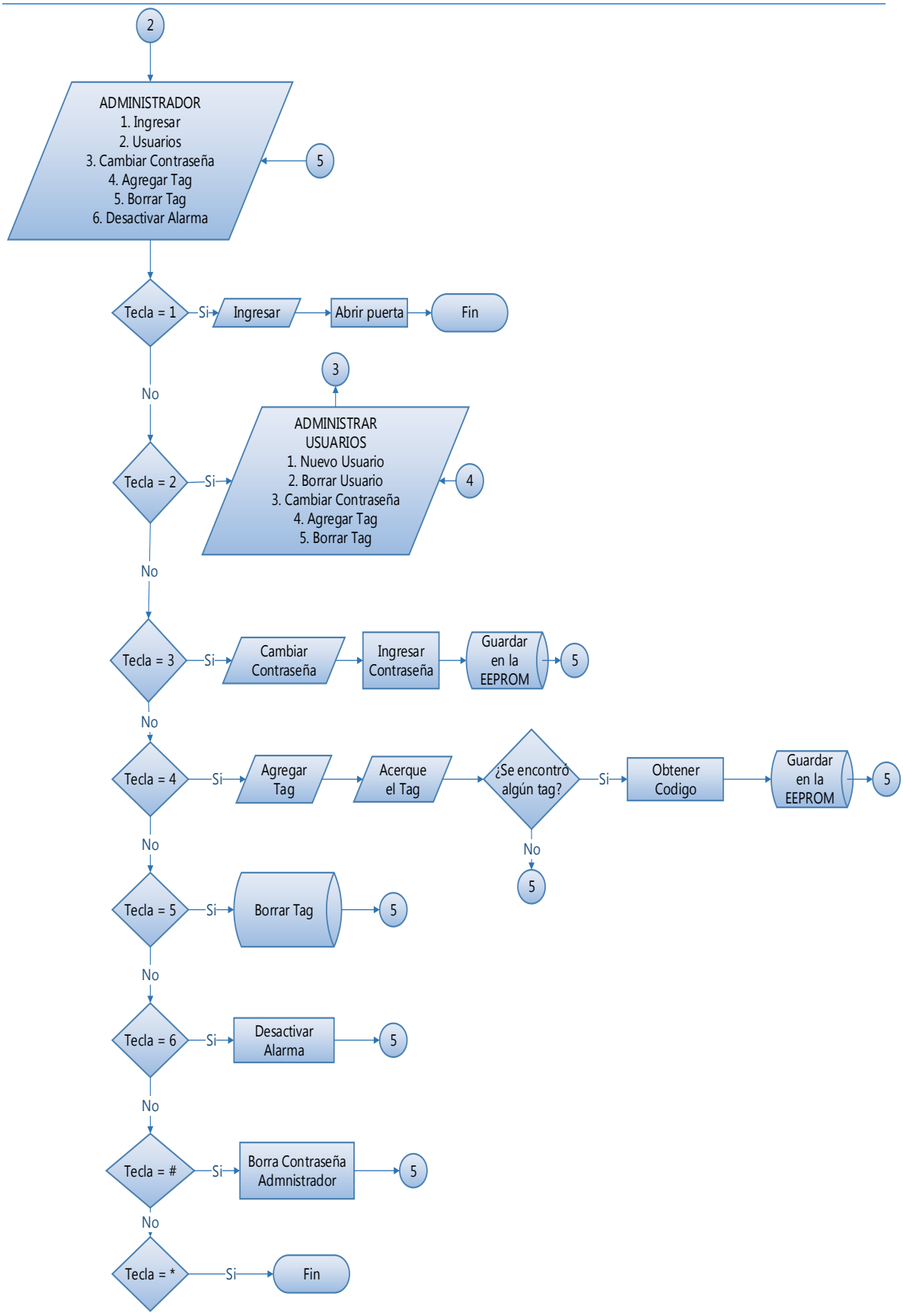
El sistema dispone de dos menús, una para el usuario y otra para el administrador. Para acceder a dichos menús necesita ingresar una contraseña, en caso de ingresar 3 veces mal la contraseña se activa la alarma.

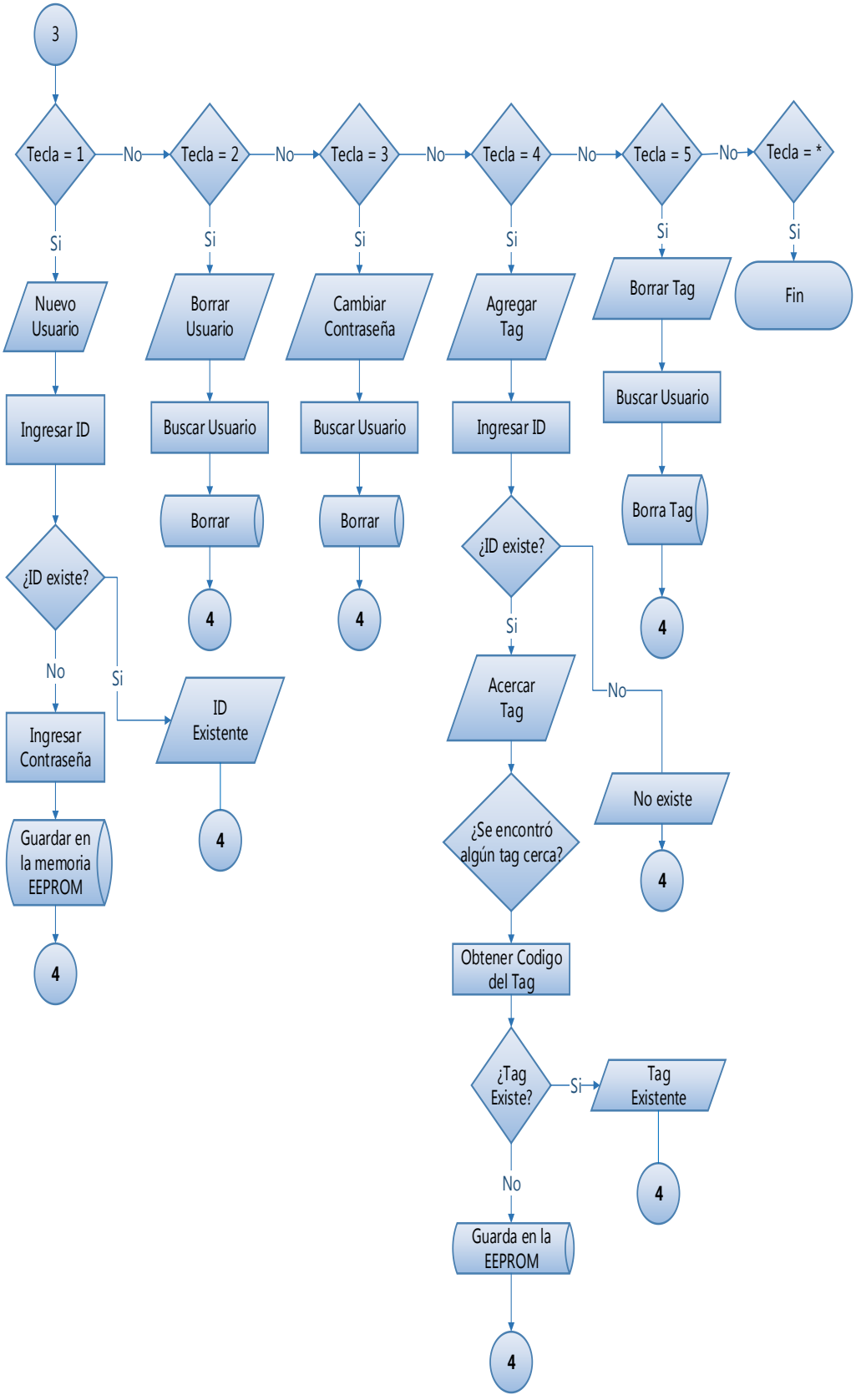
El administrador podrá agregar y borrar usuarios, así como también, añadir y borrar tags. Además, podrá cambiar la contraseña de los usuarios con el propósito de restablecer el acceso del menú, en caso de que se le haya olvidado la contraseña.

Si el administrador o el usuario pierde la tarjeta, podrá acceder al local a través del menú correspondiente, además podrá eliminar el tag del sistema para evitar que un intruso tenga acceso al lugar.

En caso de que el administrador olvide su contraseña podrá cambiarlo en el menú correspondiente por una nueva, pero si no dispone con el acceso al menú, el sistema cuenta con un botón de reinicio de contraseña.


Inicio
Modo Lectura
¿Encontró algún Tag?
Si
No
Obtener Codigo
¿Existe en el sistema?
No
Tag No Existe
Fin
Si
Abrir Puerta
Fin
¿Fue pulsado el boton Reset?
Si
Borrar contraseña del Administrador
Fin
No
Tecla = *
No
Fin
Si
1. Administrador
2. Usuario
Tecla = 1
No
1
Si
Introducir Contraseña
¿Contraseña es Correcta?
No
Si
2
Contador + 1
Contador =3
Fin
No
Si
Activar Alarma
Fin


2
ADMINISTRADOR
1. Ingresar
2. Usuarios
3. Cambiar Contraseña
4. Agregar Tag
5. Borrar Tag
6. Desactivar Alarma
5
Tecla = 1
Si
Ingresar
Abrir puerta
Fin
No
3
Tecla = 2
Si
ADMINISTRAR USUARIOS
1. Nuevo Usuario
2. Borrar Usuario
3. Cambiar Contraseña
4. Agregar Tag
5. Borrar Tag
4
No
Tecla = 3
Si
Cambiar Contraseña
Ingresar Contraseña
Guardar en la EEPROM
5
No
Tecla = 4
Si
Agregar Tag
Acerque el Tag
¿Se encontró algún tag?
Si
Obtener Codigo
Guardar en la EEPROM
5
No
5
No
Tecla = 5
Si
Borrar Tag
5
No
Tecla = 6
Si
Desactivar Alarma
5
No
Tecla = #
Si
Borra Contraseña Admnistrador
5
No
Tecla = *
Si
Fin


3
Tecla = 1
No
Tecla = 2
No
Tecla = 3
No
Tecla = 4
No
Tecla = 5
No
Tecla = *
Si
Nuevo Usuario
Ingresar ID
¿ID existe?
No
Ingresar Contraseña
Guardar en la memoria EEPROM
4
Si
ID Existente
4
Borrar Usuario
Buscar Usuario
Borrar
4
Cambiar Contraseña
Buscar Usuario
Borrar
4
Agregar Tag
Ingresar ID
¿ID existe?
Si
Acercar Tag
¿Se encontró algún tag cerca?
Obtener Codigo del Tag
¿Tag Existe?
Si
Tag Existente
4
No
Guarda en la EEPROM
4
No
No existe
4
Borrar Tag
Buscar Usuario
Borra Tag
4
Fin

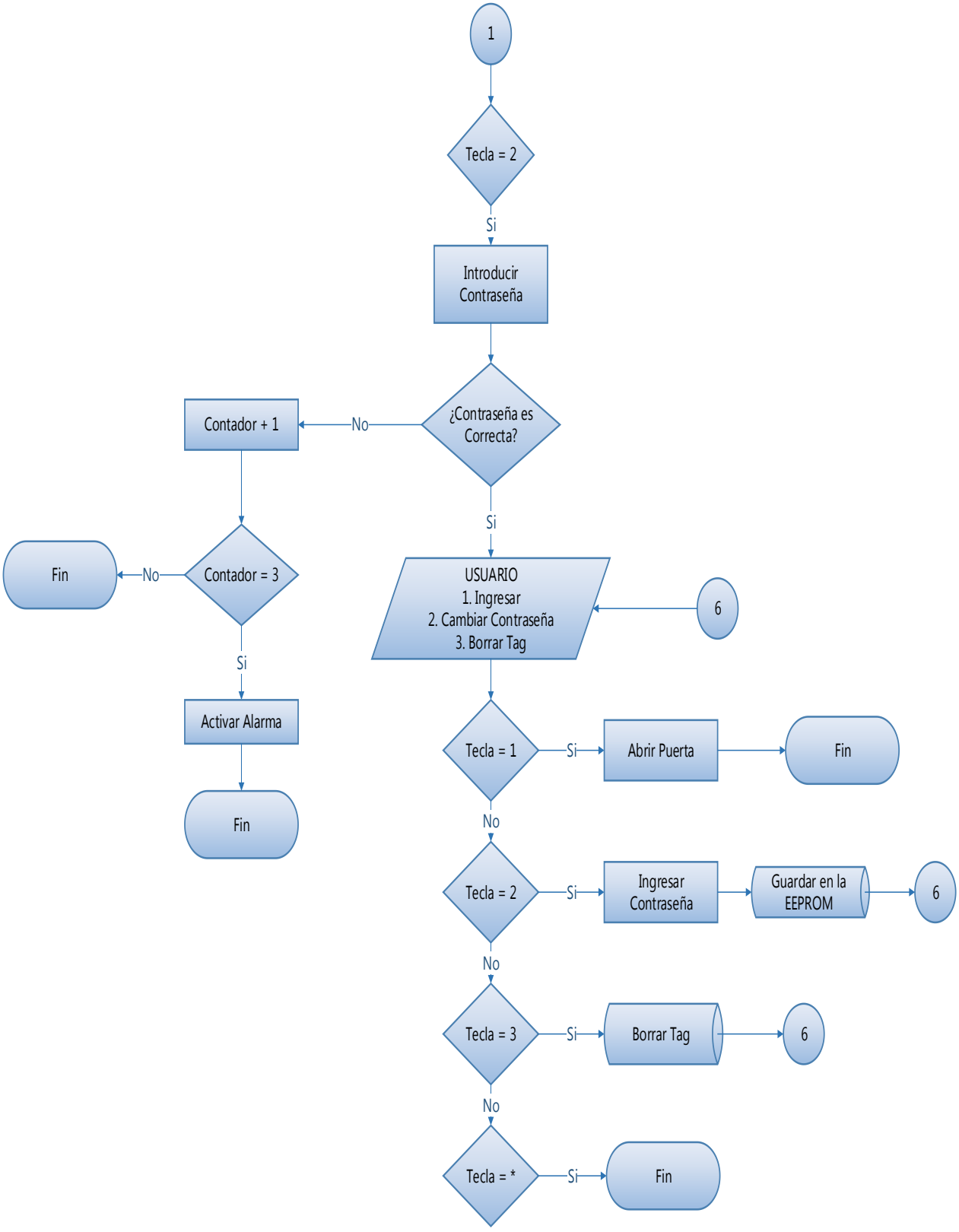



**Figura 33. Algoritmo del Sistema de Control Acceso RFID**

## 3.3 IMPLEMENTACION DEL PROTOTIPO

El sistema de seguridad electrónica se implementó a nivel prototipo utilizando los componentes descritos anteriormente. Durante la elaboración de este prototipo se llevaron a cabo varios procesos, dentro de los cuales se pueden mencionar; la programación de los microcontroladores, fabricación de las tarjetas electrónicas, instalación de los componentes del sistema en el local, pruebas y corrección de errores.

### 3.3.1 Programación del Microcontrolador

Para la programación del Arduino Mega se utilizó el IDE[13] oficial de Arduino que es de libre uso, basado en el algoritmo de control propuesto. El lenguaje que se utilizo es el propio de Arduino, siendo este una combinación de los lenguajes C y C++ con algunas características añadidas y otras reducidas.

#### 3.3.1.1 IDE Arduino 1.6.9

Algunas características del IDE de Arduino, es que la estructura básica del lenguaje de programación es bastante simple y se compone de al menos dos partes. Estas dos partes necesarias, o funciones, encierran bloques que contienen declaraciones, estamentos o instrucciones. En la **Figura 34** se puede observar el entorno de Arduino y su estructura básica.

[13] IDE: Entorno de desarrollo integrado, en inglés Integrated Development Environment (IDE), es una aplicación informática que proporciona servicios integrales para facilitarle al desarrollador o programador el desarrollo de software.

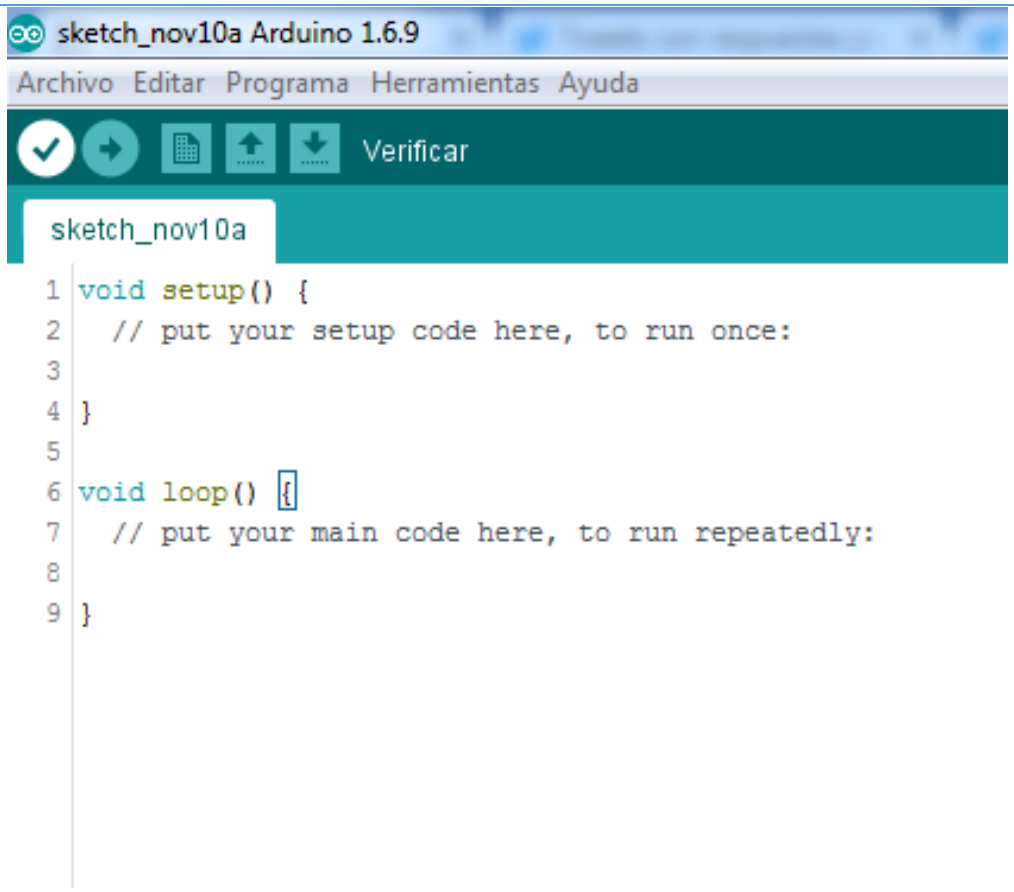



**Figura 34. IDE Arduino 1.6.9**

En donde **setup()** es la parte encargada de obtener las configuraciones y **loop()** es la que contienen el programa que se ejecutará cíclicamente. Ambas funciones son necesarias para que el programa trabaje.

La función de configuración **setup()** debe contener la declaración de las variables. Es la primera función a ejecutar en el programa, se ejecuta sólo una vez, y se utiliza para configurar o inicializar el modo de trabajo de los pines de entrada y salida, configuración de la comunicación, etc.

La función bucle **loop()** contiene el código que se ejecutara continuamente (lectura de entradas, activación de salidas, etc). Esta función es el núcleo de todos los programas de Arduino y la que realiza la mayor parte del trabajo.

#### 3.3.1.1.1 Desarrollo del programa en el IDE de Arduino

Para el desarrollo del programa se requiere hacer una revisión de todos los procesos que se quieren ejecutar en el algoritmo de control, lo que permite al programador verificar los aspectos involucrados (control analógico/digital, comunicaciones, adquisición, comparación y almacenamiento de datos, entre otros) a nivel de software.

Para algunos procesos complejos, el IDE de Arduino contiene más de 30 librerías por defecto (Además, se pueden anexar algunas librerías creadas por la comunidad) capaces de realizar varias rutinas específicas, estas pueden ser para interactuar con dispositivos de hardware o realizar cálculos matemáticos, a continuación, se citan las librerías que contienen las rutinas utilizadas en el desarrollo del proyecto:

1. LiquidCrystal.h
2. Keypad.h
3. EEPROM.h
4. SPI.h
5. MFRC522.h

En total se requirieron 5 librerías para llevar a cabo las tareas de control de acceso y alarma. A continuación, se presenta una breve descripción de las tareas que realiza cada uno de las librerías:

**LiquidCrystal.h:** Está librería contiene instrucciones que permiten visualizar caracteres en el módulo Display LCD.

**Keypad.h:** Está librería contiene instrucciones que permiten controlar el teclado.

**EEPROM.h:** Está librería contiene instrucciones para poder almacenar datos en la memoria EEPROM del Arduino.

**SPI.h:** Está librería contiene una serie de instrucciones para poder comunicar el Arduino con cualquier dispositivo que funcione con el protocolo de comunicación SPI.

**MFRC522.h:** Está librería contiene las instrucciones necesarias para poder controlar el modulo lector RFID-RC522.

Para Almacenar las credenciales de los usuarios se requirió hacer uso de un arreglo bidimensional, los datos contenido en este arreglo se almacenan en la memoria EEPROM del Arduino.

Para hacer el menú se requirió hacer uso de las instrucciones switch case anidado, eso quiere decir que, en algunos switch los casos pueden contener otros switch case.

Para realizar el proceso de lectura de la tarjeta se usó las instrucciones:

```
if ( ! rfid.PICC_IsNewCardPresent())

return;

if ( ! rfid.PICC_ReadCardSerial())

return;
```

Estas instrucciones se encargan de obtener los datos de la tarjeta, una vez obtenidos los datos se guardan en una matriz unidimensional de manera temporal, seguido, se realizan las operaciones correspondientes. Una vez terminada la operación se usan las instrucciones:

```
rfid.PICC_HaltA();
rfid.PCD_StopCrypto1();
```

Que permiten terminar la comunicación con el modulo lector y seguido detiene el canal de encriptación.

Para comunicar el Arduino con la computadora, se usó el protocolo de comunicación EUSART (Descrito en **Anexo E. 2**) y luego se configuro el puerto serial (Tanto del Arduino como Labview) a una velocidad de 9600 baudios[14]. Para enviar un dato se utilizó la instrucción:

Serial.println();

Al finalizar la ardua tarea de programar, se copilo el programa y se obtuvo el siguiente resultado que se aprecia en la siguiente figura:

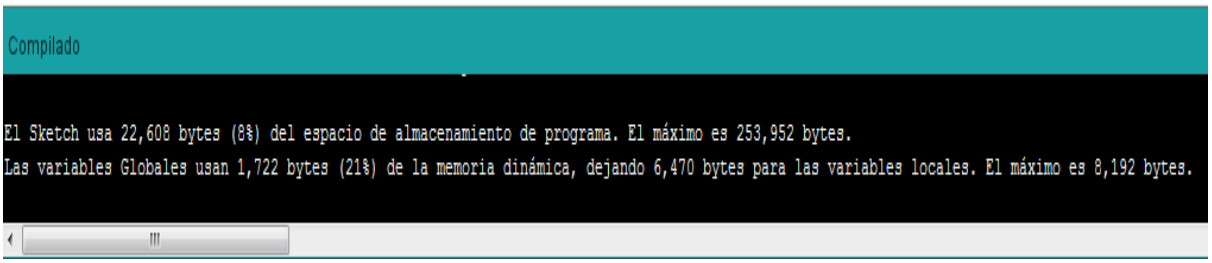



**Figura 35. Cantidad de memoria que utilizo el Firmware en el Arduino Mega**

Se puede apreciar que el firmware ocupo tan solo 22.608 kbytes de espacio en la memoria flash del Arduino Mega, lo que representa apenas el 8% del espacio de memoria en el Arduino. La capacidad máxima es de 253.952 kbytes.

También se puede ver que las variables globales ocuparon tan solo 1.722 kbytes de espacio en la memoria SRAM del Arduino, lo que representa apenas el 21% del espacio de memoria. La capacidad máxima del Arduino es de 8.192 kbytes. En **Anexo I** se adjunta el código de programación.

[14] Baudios: Unidad de medida de la velocidad de transmisión de señales que se expresa en símbolos por segundo.

### 3.3.2 Diseño del Circuito del Sistema

Para realizar el diseño del sistema se ocupó el software Proteus con el objetivo de encontrar posibles errores a la hora del diseño y así obtener rendimiento óptimo del sistema.

#### 3.3.2.1 Proteus 8.4

Proteus es un entorno integrado diseñado para la realización completa de proyectos de construcción de equipos electrónicos en todas sus etapas: diseño, simulación, depuración y construcción.

La aplicación Proteus está compuesta básicamente por tres programas principales (Wikipedia, 2016):

- ISIS (Intelligent Schematic Input System): Es el programa que se utiliza para diseñar el esquema eléctrico del circuito que se desea realizar.
- VSM (Virtual System Modeling): Es un módulo asociado a ISIS que permite simular los circuitos en tiempo real (incluyendo MCU[15] y MPU[16]).
- ARES (Advance Routing Modeling): Es el software que se utiliza para la fabricación de placas de circuito impreso, permitiendo la edición, elaboración de pistas y ubicación de componentes.

Con Proteus se podrá realizar el diseño de un circuito electrónico a través de ISIS, que luego podrán ser simuladas a través del entorno VMS con el propósito de encontrar posibles errores en el diseño, seguido se procede a pasarlos a un circuito impreso a través del entorno de ARES. En la **Figura 36** se aprecia el entorno grafico de Proteus.

[15] MCU: Microcontrolador
[16] MPU: Microprocesador

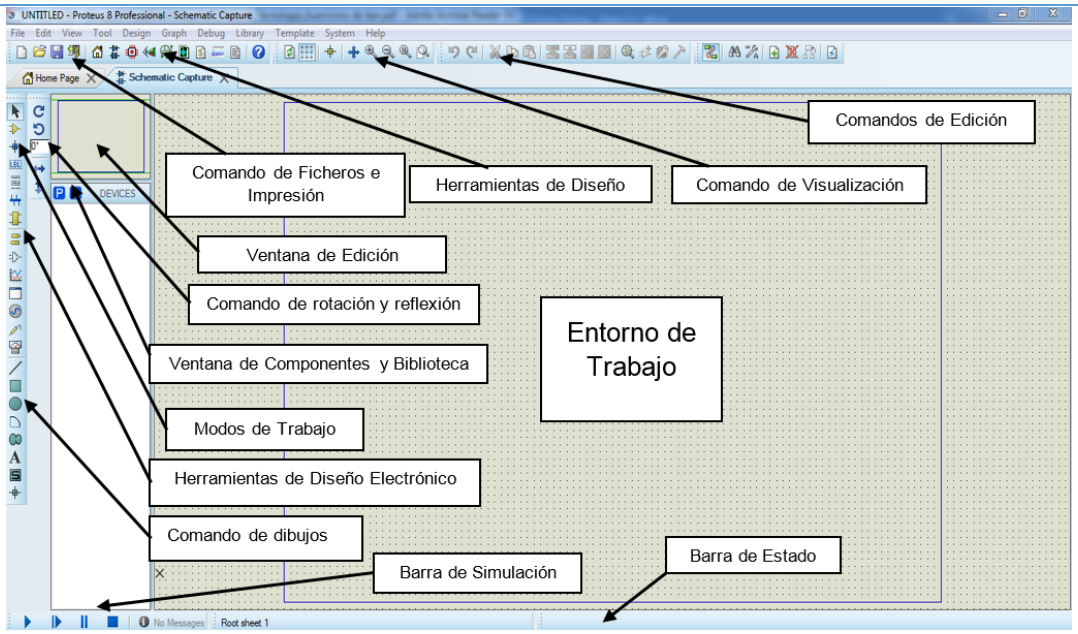



**Figura 36. Entorno Grafico de Proteus**

Sin la utilización de la suite Proteus, el proceso para construir un equipo electrónico basado en un microprocesador se componía de las siguientes etapas (Ver **Figura 37**):

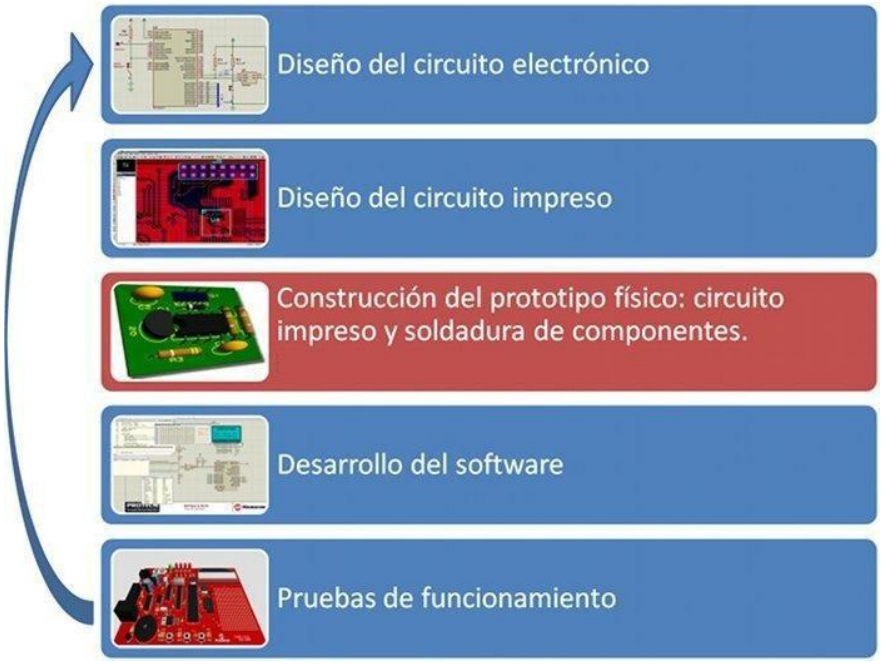



**Figura 37. Procesos de desarrollo de un prototipo utilizando herramientas tradicionales de diseño**

Con las herramientas de diseño tradicionales, el desarrollo del software y la comprobación del prototipo, no puede realizarse hasta que este no se desarrolla. Esto puede suponer meses de atraso. Si se localiza un error en el hardware, la totalidad del proceso se debe repetir, lo que con lleva altos costes económicos. En la **Figura 38** se muestra los procesos de desarrollo de un prototipo usando Proteus.


**Figura 38. Procesos de desarrollo de un prototipo usando Proteus**

Usando Proteus, el desarrollo del software puede comenzar tan pronto como el diseño esquemático este acabado y la combinación del hardware y el software nos permite testear el prototipo y ver si funciona.

#### 3.3.2.1.1 Simulación del Hardware en Proteus

En esta etapa pone a prueba el diseño, la programación y la funcionalidad. A pesar de que Proteus no dispone en la librería del módulo RFID-MFRC522 se logró simular gran parte del sistema, el cual significo un gran avance a la hora de implementarlo. En la **Figura 39** se aprecia la simulación del sistema de control de acceso en Proteus.

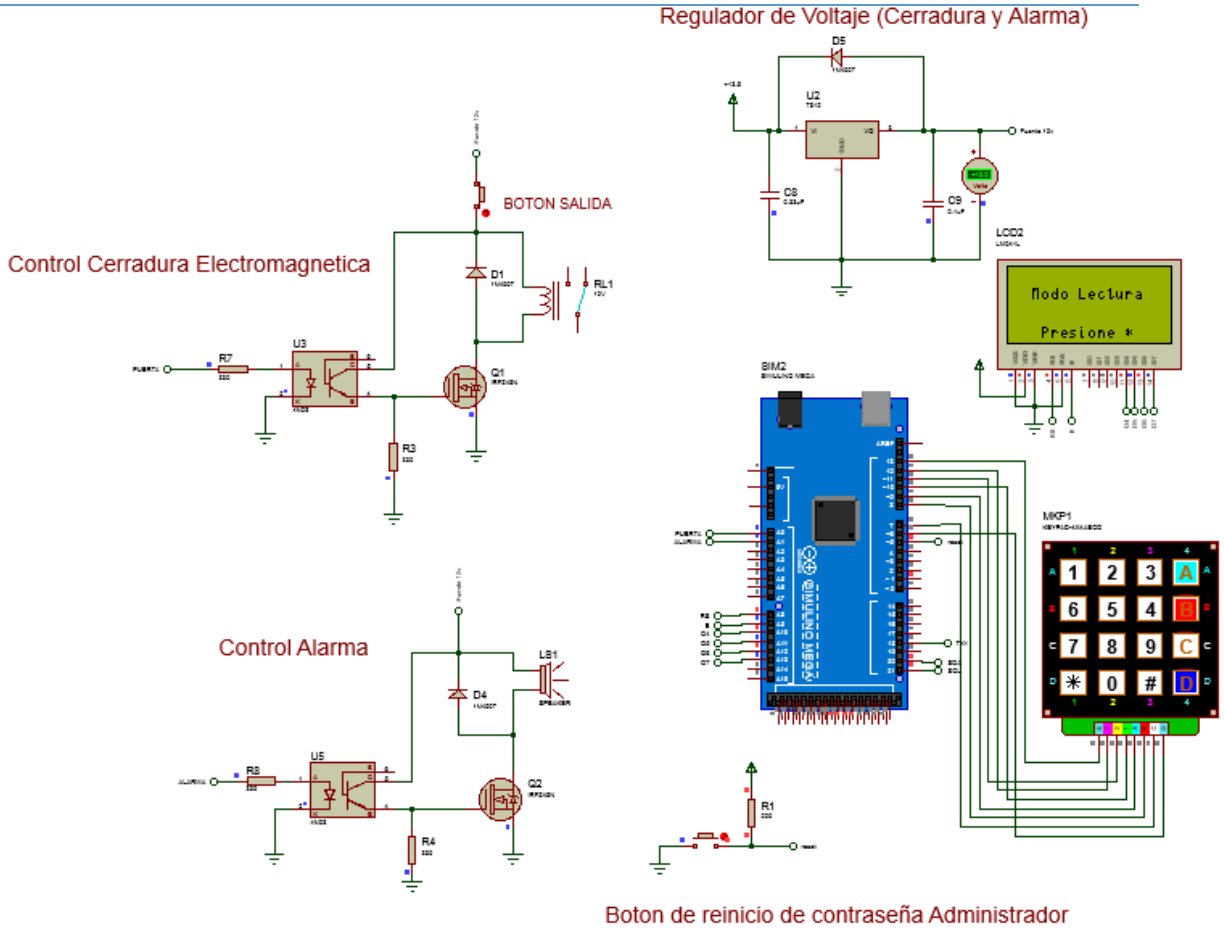



**Figura 39. Simulación del Sistema de Control de Acceso**

### 3.3.3 Registro de Usuarios

Para poder registrar los tiempos de accesos de los usuarios se realizó un programa a través del software LabVIEW. La cual se encarga de adquirir los datos enviados atreves del Arduino Mega para ser procesarlos, y seguido, guardarlos en una hoja de Excel.

Una de las razones por la que se utilizó este software, es porque ya había tenido una experiencia previa en algunas clases de la carrera, por lo que, me fue fácil de aprender y usar.

#### 3.3.3.1 NI LabVIEW 2015

LabVIEW (Laboratory Virtual Instrument Engineering Workbench) es una plataforma y entorno de desarrollo para diseñar sistemas, con un lenguaje de programación gráfica. Utilizado en aplicaciones que involucren adquisición, control, análisis y presentación de datos. Las ventajas que proporciona el uso de LabVIEW se resumen en las siguientes:

- Reducción de tiempo de desarrollo de las aplicaciones al menos unas 10 veces, ya que es muy intuitivo y rápido de aprender.
- El sistema cuenta con una gran flexibilidad, permitiendo cambios y actualizaciones tanto del hardware como del software.
- Da la posibilidad a los usuarios de crear soluciones completas a problemas muy complejos.
- Con un único sistema de desarrollo se integran las funciones de adquisición, análisis y presentación de datos.
- El sistema está equipado con un compilador gráfico para lograr la máxima velocidad de ejecución posible.
- Se puede añadir aplicaciones escritas en otros lenguajes. LabVIEW es un entorno de programación diseñado para el desarrollo de aplicaciones, similar a otros sistemas de desarrollo comerciales que utilizan el lenguaje C o Python. Sin embargo, LabVIEW se diferencia de dichos programas en un importante aspecto: los citados lenguajes de programación se basan en líneas de texto para crear el código fuente del programa, mientras que LabVIEW emplea la programación gráfica o lenguaje G para crear programas basados en diagramas de bloques.

LabVIEW posee extensas librerías de funciones y subrutinas. Además de las funciones básicas de todo lenguaje de programación, LabVIEW incluye librerías específicas para la adquisición de datos, control de instrumentación VXI, GPIB y comunicación serie, análisis presentación y guardado de datos.

LabVIEW también proporciona potentes herramientas que facilitan la depuración de los programas.

Cada VI de LabVIEW cuenta con dos interfaces: Panel Frontal y Diagrama de bloques. Estás cuentan con paletas que contienen los objetos necesarios para implementar y desarrollar tareas. En la **Figura 40** se aprecia el entorno de LabVIEW.

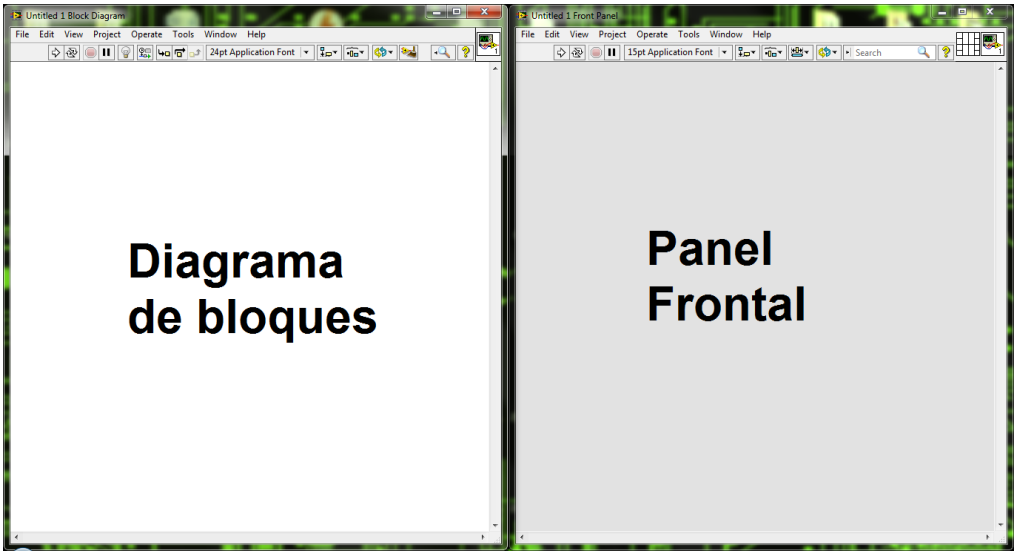



**Figura 40. Entorno de LabVIEW**

El panel frontal permite realizar la presentación que el usuario final verá, en él se colocan todos los diferentes elementos que permiten el intercambio de información entre el programa y la persona que ejecutará la aplicación. Mientras que, el diagrama de bloques contiene el código del programa. Por medio del lenguaje gráfico se “dibuja” el algoritmo que procesa la información que se adquiere por medio del panel frontal.

#### 3.3.3.1.1 Algoritmo

Para el desarrollo del algoritmo se requiere que el programa este monitoreando constantemente para saber si el Arduino envió algún dato, en caso de ser recibido obtiene el dato junto con la hora y la fecha (la cual es obtenida por medio de la computadora) y seguido procede a guardar en una hoja de Excel. En la **Figura 41** se aprecia el algoritmo de control.

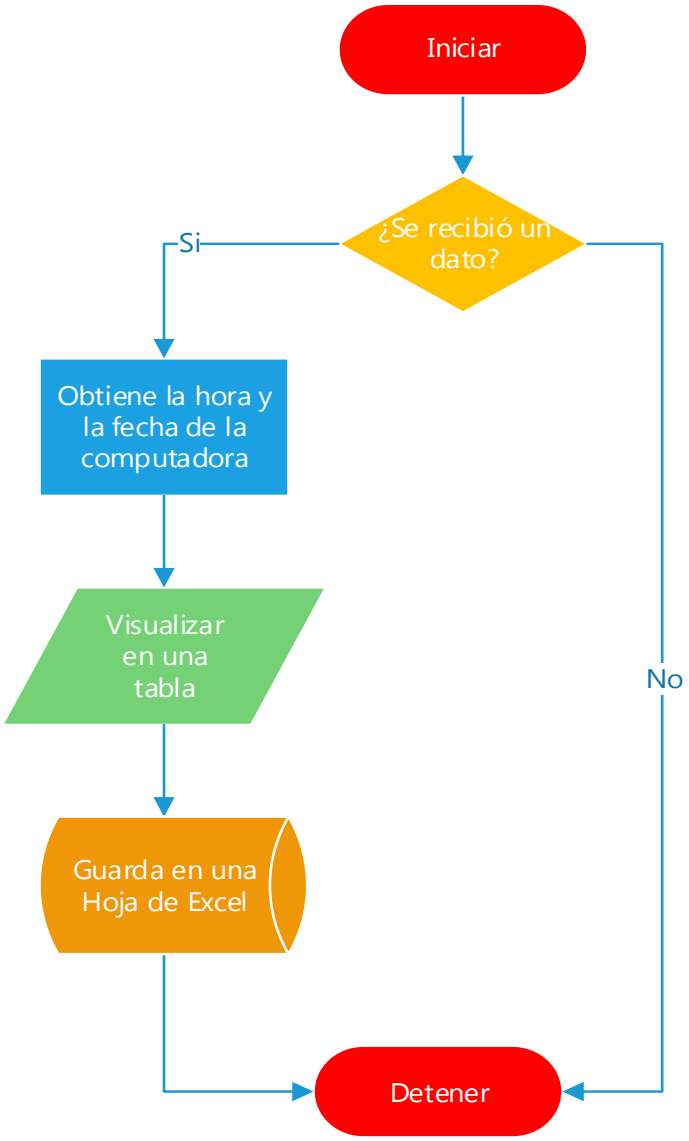



**Figura 41. Algoritmo de Control de Adquisición de datos LabVIEW**

### 3.3.3.1.2 Desarrollo del Software en el IDE de LabVIEW

Por medio de los instrumentos VISA serial se emplea la comunicación con el Arduino, primero se configura por defecto la velocidad a 9600 baudios, seguido se le pone un delay de 100 milisegundos para que los datos sean recibidos ordenadamente. Una vez recibido el dato, se lee el número de bits, si el número de bits es mayor que cero, lee el dato y seguido adquiere la hora y la fecha de la computadora. Estos son almacenados temporalmente en un arreglo, para después ser guardados a una hoja de Excel. En la **Figura 42** se muestra el programa de Adquisición de datos en LabVIEW.

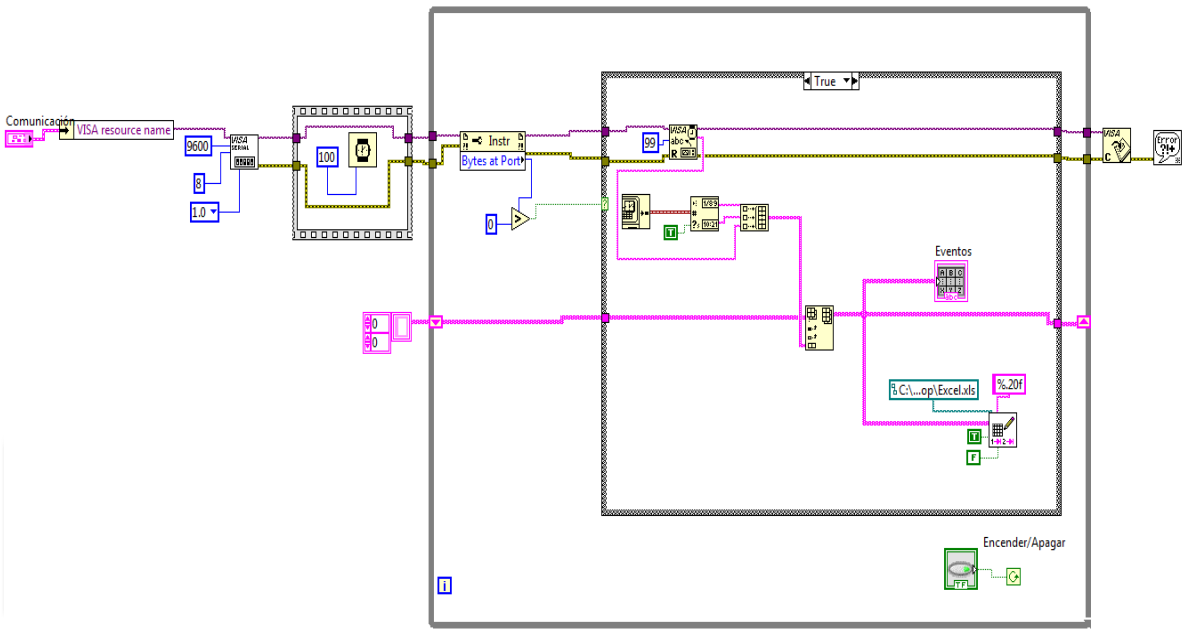
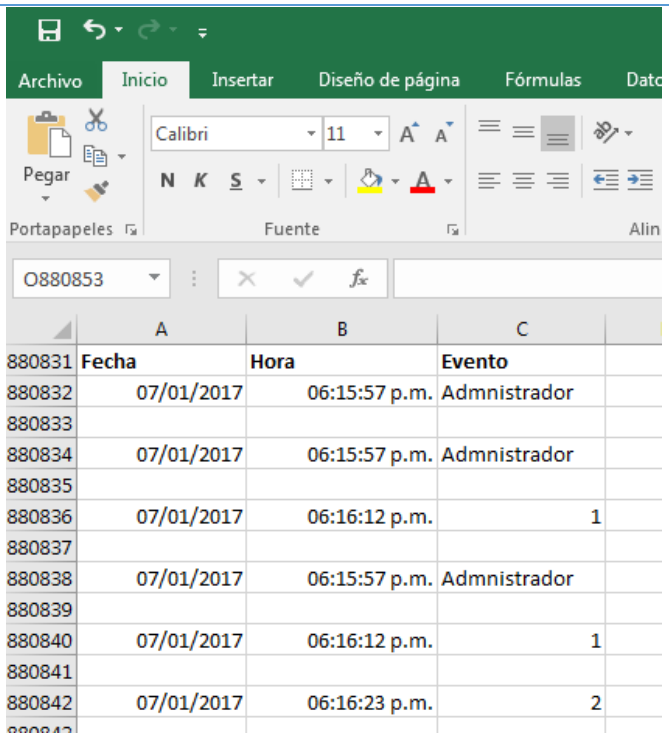



**Figura 42. Programa de Adquisición de datos en LabVIEW**

Después de terminar la ejecución del programa se crea una hoja de Excel automáticamente en el escritorio de la computadora, que contiene los datos como lo son la hora, fecha y evento, tal como se muestra en la siguiente figura:



Archivo | Inicio | Insertar | Diseño de página | Fórmulas | Dato

Pegar | Portapapeles | Calibri | 11 | Fuente | Alin

O880853

| | A | B | C |
|---|---|---|---|
| 880831 | **Fecha** | **Hora** | **Evento** |
| 880832 | 07/01/2017 | 06:15:57 p.m. | Admnistrador |
| 880833 | | | |
| 880834 | 07/01/2017 | 06:15:57 p.m. | Admnistrador |
| 880835 | | | |
| 880836 | 07/01/2017 | 06:16:12 p.m. | 1 |
| 880837 | | | |
| 880838 | 07/01/2017 | 06:15:57 p.m. | Admnistrador |
| 880839 | | | |
| 880840 | 07/01/2017 | 06:16:12 p.m. | 1 |
| 880841 | | | |
| 880842 | 07/01/2017 | 06:16:23 p.m. | 2 |

**Figura 43. Registro de Eventos guardados en Excel**

### 3.3.4 Montaje del Sistema en Tabla de Nodo

Antes de montar los componentes en la tarjeta de circuito impreso (PCB), es necesario poner a prueba el diseño en una tabla de nodos (Ver **Figura 44**) para asegurar el buen funcionamiento del mismo.

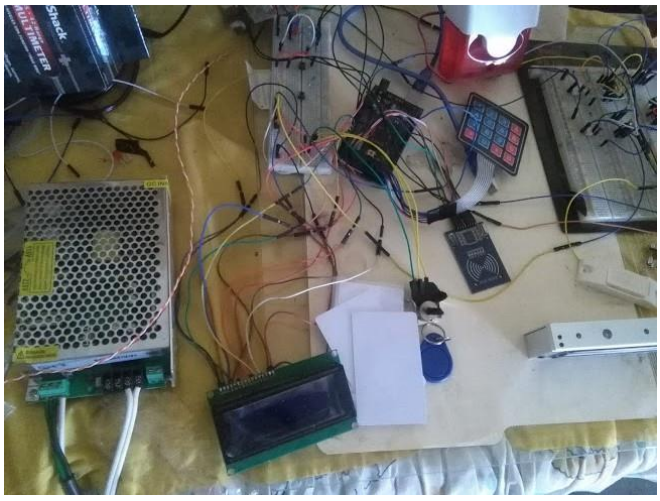


**Figura 44. Montaje de los componentes en tabla de nodo**

### 3.3.5 Diseño de las Tarjetas de circuito impreso (PCB)

Para el diseño de las placas de circuito impreso se utilizó CadSoft EAGLE PCB Design Software, el cual integra varias funciones y herramientas que facilitan esta labor. Una ventaja de este software es que dispone de una biblioteca con gran variedad de componentes, además contiene librerías creadas por empresas o aficionados que la distribuyen alrededor de la red de forma gratuita.

Para realizar el diseño de una PCB, primero debemos de hacer el esquema del circuito en el editor de diagramas electrónicos que dispone EAGLE (Ver **Figura 45**).

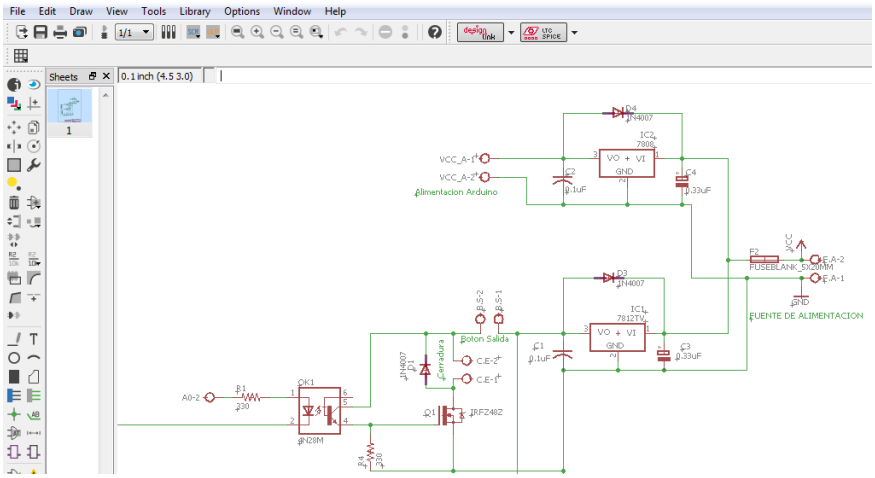



**Figura 45. Esquema del circuito en EAGLE**

EAGLE contiene de un Netlist que se encarga de pasar la lista de componentes del panel de diagramas electrónicos al editor de PCBs, el cual nos encargaremos de configurar las dimensiones de la placa y de realizar la interconexión de los componentes. En la **Figura 46** se aprecia el diseño de la PCB en EAGLE.

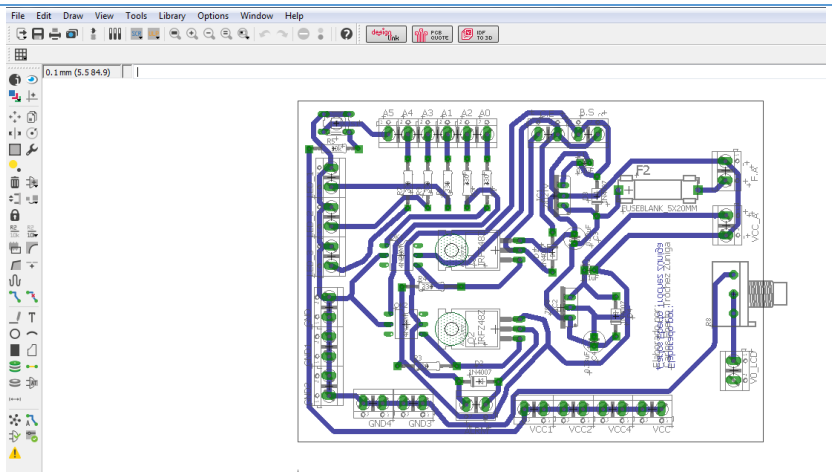



**Figura 46. Diseño del PCB en EAGLE**

### 3.3.6 Fabricación de la PCB

En esta etapa, se procede a realizar la fabricación de la PCB. Este proceso se describe detalladamente en el **Anexo G**. En la **Figura 47** se aprecia la placa PCB ya finaliza.

**Figura 47. Placa PCB del sistema de control de Acceso RFID**

## 3.4 PRUEBAS DE FUNCIONAMIENTO Y CORRECCIÓN DE ERRORES

En esta etapa se realiza pruebas para verificar el correcto funcionamiento del sistema. A continuación, se describen los problemas que fueron corregidos:

- **El sistema no detectaba los IDs de los usuarios correctamente**:

Debido a que se trabajó con arreglos bidimensional para almacenar la identificación de los usuarios, el Arduino no detectaba correctamente si un ID está guardado en el sistema, por lo que, se guardaba otra vez y de esta forma contaminaba el arreglo. Para solucionarlo, se ocupó algunas de las funcionalidades de lenguaje C++, para manipular los caracteres y de esta forma obtener un rendimiento óptimo del sistema.

- **La contraseña del Administrador no guarda**:

Al momento de que se cambia la contraseña del Administrador no guardaba, por lo que, para acceder al menú, se tenía qué hacer con la contraseña que viene por defecto en el sistema. Para solucionarlo, se cambió el estado (De Bajo a Alto) con el que el Arduino lee el botón de reinicio de contraseña del administrador.

- **El módulo RFID se queda congelado al momento de leer una tarjeta**:

Al momento que se pasaba la tarjeta cerca del módulo RFID lo leía correctamente, pero luego de pasarla por segunda vez ya no la detectaba. El problema se presentaba en un error en la función, debido a que era una función de tipo “Void” no devolvía ningún valor entero que continua la función, así que se cambió por una función de tipo “Bool”.

- **Problemas con el botón de salida del usuario**:

La idea era que al momento de que el usuario presiona el botón de salida, se ejecutara una interrupción en el Arduino para que se desactivara la cerradura electromagnética durante un periodo de tiempo de 5 segundo para que el usuario

pueda salir de local y luego activara de nuevo la cerradura para mantener cerrada la puerta del local. No se pudo hacer de esta manera debido a que, en el momento que se ejecuta una interrupción, el sistema debe terminar de ejecutar dicha interrupción lo más rápido posible y volver al programa principal y, por ende, los Timer de los microcontroladores no funcionan en las interrupciones. Para solucionar este problema, se colocó el botón de salida entre la fuente y la cerradura electromagnética.

- **El Mosfet del circuito de la cerradura electromagnética se cortocircuita:**

Al momento de energizar la placa PCB, el Mosfet del circuito de la cerradura electromagnética se cortocircuitaba, por lo que, la cerradura electromagnética se mantenía constantemente encendida, lo cual no permitía desactivarla para permitir el ingreso de un usuario. El problema surgía porque el ánodo del diodo de protección se encontraba conectado después del “Drain” de Mosfet, a simple vista no parecía que fuera un problema debido a que comparten el mismo nodo. Y debido a esto, las corrientes inversas que genera la cerradura electromagnética destruía el Mosfet, por lo que, el diodo no ejercía ninguna función. El problema se resolvió puenteando un cable del ánodo del diodo hacia la terminal negativa de la cerradura electromagnética, de esta manera se hace un solo nodo entre la cerradura, el Mosfet y el diodo.

## 3.5 RESULTADOS

Como resultado de este proyecto fue la construcción de un sistema de control de acceso utilizando tecnología RFID, el cual funciona de manera óptima y eficientemente. A continuación, se listan las características del prototipo elaborado:

- El sistema permite el acceso del personal autorizado por medio de la validación de las tarjetas RFID en el punto de acceso de forma rápida y eficiente.
- Se generan alarmas locales para avisar a sus usuarios cuando se presenta un evento que viole la política de seguridad.
- El sistema dispone de un menú de Usuario y Administrador, cuya función de describe a continuación:

    - El menú del administrador tiene el privilegio de agregar, borrar y cambiar contraseña de los usuarios, así como también agregar o eliminar los tags de cada uno. Además, dispone de una opción para desactivar la alarma, cuando la sirena se encuentre activa.
    - El menú del usuario dispone con las únicas opciones de cambiar contraseña y eliminar tag.
    - Ambos menús tanto usuarios y Administrador disponen con una opción que permite el ingreso al local, en caso de que pierdan u olviden la tarjeta RFID.

- Cuando un usuario entra al local, se libera la puerta y se registran su identificación junto con la hora y la fecha en una hoja de Excel.
- El sistema está diseñado para que funcione cuando haya cortes de energia.

## 3.6 COSTO DEL PROTOTIPO

A continuación, se presentan los costos de los dispositivos utilizados en proceso de diseño e implementación del prototipo.

**Tabla 13. Componentes comprados en el extranjero – Parte 1**

| Modelo | Descripción | Cantidad | Precio Amazon | Total USD (Enviado a Nic) |
|---|---|---|---|---|
| **Elegoo Mega 2560** | Controlador | 1 | $ 12.21 USD | $ 30.53 USD |
| **Teclado** | Teclado | 1 | $ 6.99 USD | $ 24.21 USD |
| **RFID RC522** | Modulo Lector | 1 | $ 7.99 USD | $ 27.79 USD |
| **RioRand LCD Module** | Display LCD | 1 | $ 10.99 USD | $ 27.96 USD |
| **UID Card 13.56 MHz RFID** | Tarjetas RFID | 10 | $ 15.50 USD | $ 43.50 USD |
| **DROCK** | Fuente de alimentación 13.5V, 10A | 1 | $ 26.58 USD | $ 55.25 USD |
| **Subtotal 1** | | | | **$ 209.24 USD** |

**Nota:** Componentes incluyen costo de envió

Tabla 14. Componentes comprados en el extranjero – Parte 2

| Modelo | Descripción | Cantidad | Precio Amazon | Total USD |
|---|---|---|---|---|
| **Docooler** | Cerradura Electromagnética 180 KG | 1 | $32.99 USD | $ 32.99 USD |
| **Foto4easy** | Sirena | 1 | $7.57 USD | $ 7.57 USD |
| **Seco-Larm** | Botón de salida | 1 | $4.66 USD | $ 4.66 USD |
| **Costo de envió** | Comisión por compra; I.V.A, Flete, etc. | | | $ 37.29 USD |
| **Subtotal 2** | | | | **$ 82.51 USD** |

**Nota**: Costos de envió de los componentes se incluyen aparte

Tabla 15. Componentes comprados en Nicaragua

| Modelo | Cantidad | Total C$ |
|---|---|---|
| **Terminal Block** | 15 | C$ 179.91 |
| **Placas de cobre** | 1 | C$ 50.00 |
| **IRFZ48N** | 2 | C$ 70.00 |
| **1N4007** | 4 | C$ 16.00 |
| **L7808** | 1 | C$ 18.00 |
| **L7812** | 1 | C$ 18.00 |
| **Capacitores 0.33 µF** | 2 | C$ 15.00 |
| **4N28** | 2 | C$ 68.00 |
| **Resistencias 330 ohm** | 2 | C$ 5.00 |
| **Porta fusible** | 1 | C$ 15.00 |
| **Fusible 1A** | 1 | C$ 10.00 |

| | | |
|---|---|---|
| **Potenciómetro 2k** | 1 | C$ 35.00 |
| **Capacitores 0.1 µF** | 2 | C$ 20.00 |
| **Resistencias 220 ohm** | 2 | C$ 12.00 |
| **Aislante TO 220** | 4 | C$ 10.00 |
| **Cajas eléctricas** | 2 | C$ 2100.00 |
| **Disipadores** | 3 | C$ 40.00 |
| **Total, en Córdobas** | | **C$ 2,681.91** |
| **Total, en Dólares (Tipo de cambio: C$ 29.90)** | | **$ 89.69 USD** |
| **Subtotal 3** | | **$ 89.69 USD** |

**Tabla 16. Costos Adicionales**

| Descripción | Total USD |
|---|---|
| **Computadora** | $ 200.00 USD |
| **Labview** | $ 440.00 USD |
| **Servicios Profesionales (Diseño, Programación, Construcción)** | $ 100. 00 USD |
| **Subtotal 4** | **$ 740.00 USD** |

Tabla 17. Costos totales de todo el proyecto

| | |
|---|---|
| Subtotal 1 | $ 209.24 USD |
| Subtotal 2 | $ 82.51 USD |
| Subtotal 3 | $ 89.69 USD |
| Subtotal 4 | $ 740.00 USD |
| **Total Final** | **$ 1121.44 USD** |

El sistema de control de acceso RFID tiene un valor de $ 1121.44 USD, sin embargo, se puede reducir los costos del prototipo, si la adquisición de datos se hace por medio de un software libre como Visual Studio o si los eventos de entrada de los usuarios se guardan directamente en una memoria microSD, el sistema tendría un valor final aproximado de $ 681.44 USD.

No obstante, existen factores no cuantificados que impactan directamente en el costo real del prototipo, tales como: herramientas de diseño, construcción e instalación, entre otras.

Por otro lado, la compra de los dispositivos se disminuye cuando se realizan compras por mayores cantidades.

# CAPITULO 4. CONCLUSIONES Y RECOMENDACIONES

## 4.1 Conclusiones

Los objetivos de este proyecto, se han cumplido satisfactoriamente, ya que se ha adquirido una experiencia muy valiosa en sistemas basados en identificación por radiofrecuencia (RFID), además de adquirir nuevas habilidades en diseño y simulación de circuitos electrónicos y, programación de microcontroladores.

En base a los objetivos que fueron propuestos durante la fase inicial de este proyecto y los resultados obtenidos se llegó a las siguientes conclusiones:

- Se realizó la construcción de un sistema de control de acceso utilizando tecnología de identificación de radiofrecuencia (RFID), cuyo sistema es de fácil manejo, rápido y eficiente.
- El Sistema desarrollado responde a los requerimientos de la UNI ONLINE, de manera que integra las características y funcionalidades descritas en el presente documento.
- La lógica de control del sistema principal se logró implementar utilizando la tecnología de las placas de Arduino, lo cual proporciono gran flexibilidad para programar las tareas y acciones de control, así como para interactuar con otros dispositivos de hardware del sistema.

## 4.2 Recomendaciones

Durante el transcurso del proyecto, han surgido nuevas ideas, que podrían ser implementadas en un desarrollo futuro, a continuación, se describen las más significativas.

- Crear una interfaz en Android para que los usuarios que disponen de la funcionalidad de NFC en sus celulares, puedan acceder al local sin necesidad de usar tarjetas RFID.
- Añadir otro sistema RFID que permita el monitoreo de equipos, ambos sistemas deberán estar comunicado inalámbricamente, cuando un equipo sea removido de un lugar, se comunique con el sistema principal y active la alarma.
- Anexar al sistema sensores de movimiento (PIR) que permita la detección de intrusos cuando no haya ningún directivo de la UNI-Online en el local.
- Agregar sensores de humo al sistema, para cuando haya un incendio libere la puerta.
- Adicionar al sistema la capacidad de control sobre las luminarias, aire acondicionado, computadoras, etc. Apagarlos cuando no haya ocupantes.
- Incluir un GSM Shield para cuando se detecte un intruso, el sistema notifique a través de una llamada a los directivos de UNI-Online.
- Hacer pruebas con el Arduino Mega para comprobar si se puede enviar datos a través del puerto del Arduino a la computadora con un cable mayor de 1 metro, esto debido a que el cable que trae el Arduino es de apenas unos 25 cm.
- Agregar una batería de plomo y acido de 12V 12AH para que el sistema siga funcionando cuando haya cortes de energía, y de esta forma brindar seguridad y control en dichas situaciones.
- Guardar los eventos de entrada de los usuarios en una memoria microSD en formato Excel, para que pueda ser visualizados a través de una computadora y, de esta manera reducir el costo del prototipo.

# V. Bibliografía

Alvarado Sanchez, J. A. (2008). *Sistema de Control de Acceso con RFID.* México . Obtenido de http://www.cs.cinvestav.mx/TesisGraduados/2008/tesisJorgeAlvarado.pdf

Arduino. (2016). *Arduino*. Obtenido de https://www.arduino.cc/en/Main/ArduinoBoardMega2560

Cacuango Guachalá, D. S., & Zapata Narváez, E. J. (2015). *IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL DE ACCESO MULTINIVEL EN BASE DE RECEPTORES NEAR FIELD COMMUNICATION (NFC).* Quito. Obtenido de http://dspace.ups.edu.ec/bitstream/123456789/10160/1/UPS%20-%20ST001789.pdf

Chandramouli, R., Grance, T., Kuhn, R., & Landau, S. (2005). *Security Standards for the RFID Market.* Obtenido de http://csrc.nist.gov/staff/Kuhn/phillips-karygiannis-kuhn05.pdf

Ciudad Herrera , J. M., & Casanovas , E. S. (s.f.). *ESTUDIO, DISEÑO Y SIMULACIÓN DE UN SISTEMA DE RFID BASADO EN EPC.* Obtenido de http://upcommons.upc.edu/bitstream/handle/2099.1/3552/40883-2.pdf

Finkenzeller, K. (2010). *RFID Handbook,fundamentals and applications in contactless smart cards, radio frequency identification and near-field communication.* India: Wiley & sons. Obtenido de http://aries.ektf.hu/~dream/e107/e107_files/downloads/rfidhand.pdf

Flores Cortez, O. O. (2009). BATALLA DE MICROCONTROLADORES ¿AVR O PIC? El Salvador. Obtenido de

https://microcontroladores2utec.files.wordpress.com/2009/11/180909_articulo_colaboracion_boletin_fica_omar_otoniel_flores.pdf

Herrera Lozada, J. C., Pérez Romero, P., & Marciano Melchor, M. (2009). *Tecnología RFID Aplicada al Control de Accesos.* Obtenido de http://www.gelbukh.com/polibits/2009_40/40_08.pdf

Herrera, J. M. (s.f.). *ESTUDIO, DISEÑO Y SIMULACIÓN DE UN SISTEMA DE RFID BASADO EN EPC.* doi:

ITU. (2012). *Reglamentos de Radiocomunicaciones - Articulos.* Obtenido de https://www.itu.int/dms_pub/itu-s/oth/02/02/S02020000244501PDFS.pdf

Núñez Zeledón, S. O., & Zeledón Espinoza, M. J. (2013). *DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL DE ACCESO Y ALARMA CONTRA.* Managua, Nicaragua.

Portillo García, J., Bermejo Nieto, A. B., Bernardos Barbolla, A., & Martínez Salles, I. (s.f.). *Tecnología de identificación por radiofrecuencia (RFID): aplicaciones en el ámbito de la salud.* Madrid. Obtenido de http://www.madrimasd.org/informacionIDI/biblioteca/Publicacion/Vigilancia-tecnologica/descargar_documentos/fichero.asp?id=VT13_RFID.pdf

Romero Quezada, B. A. (2016). *ANÁLISIS DE SEGURIDAD DE LA TARJETA BIP! CHILENA COMO MEDIO DE PAGO.* Santiago de Chile. Obtenido de http://repositorio.uchile.cl/bitstream/handle/2250/139911/Analisis-de-seguridad-de-la-Tarjeta-Bip!-chilena-como-medio-de-pago.pdf?sequence=1

Stanford, V. (2003). *Pervasive Computing Goes the Last Hundred Feet with RFID Systems.* Obtenido de http://echo.iat.sfu.ca/library/stanford_03_pervasiveComp_RFID.pdf

Sweeney II, P. j. (2005). *RFID for Dummies.* Canada: Wiley. Obtenido de http://www.mums.ac.ir/shares/hit/hardware/pdf/Wiley,.RFID.for.Dummies.(2005).LinG.LotB.pdf

Vergara, Z. V. (2013). *Sistema de Control de Acceso y Monitoreo con Tecnologia RFID para el Departamento de Sistemas de la Universidad Politecnica Salesiana Sede Guayaquil.* Guayaquil. Obtenido de http://dspace.ups.edu.ec/bitstream/123456789/5380/1/UPS-GT000473.pdf

Verle, M. (2009). *MICROCONTROLADORES PIC – PROGRAMACIÓN EN C CON EJEMPLOS.* Obtenido de https://learn.mikroe.com/ebooks/microcontroladorespicc/

Want, R. (2006). An Introduction to RFID Technology. *IEEE Pervasive Computing, 5*, págs. 25-33. Obtenido de https://www.cs.cmu.edu/~15-821/READINGS/PAPERS/want2006.pdf

Weinstein, R. (2005). *RFID: A Technical Overview and Its Application to the Enterprise.* Obtenido de http://www.laxcen.com/pdf/1355910457RFID.pdf

*Wikipedia*. (2014). Obtenido de https://es.wikipedia.org/wiki/Banda_ISM

*Wikipedia*. (2016). Obtenido de https://es.wikipedia.org/wiki/RFID

Wikipedia. (2016). *Wikipedia*. Obtenido de https://es.wikipedia.org/wiki/Proteus_Design_Suite

*Wikipedia*. (2017). Obtenido de https://en.wikipedia.org/wiki/Snubber

# VI. ANEXOS

# Anexo A: Arduino Mega

## A.1 Características Técnicas del Arduino Mega

Arduino Mega dispone de un microcontrolador modelo ATmega2560; este microcontrolador es un CMOS (del inglés, *complementary metal-oxide-semiconductor*) de 8 bits que pertenece a la familia de microcontroladores ATMEL, cuyo CPU es arquitectura Harvard, opera a una frecuencia de 16 MHz y está fabricado con la tecnología RISC (del inglés *Reduced Instruction Set Computer*) lo que proporciona al ATmega2560 dos grandes ventajas:

- Puede ejecutar hasta 135 instrucciones en un solo ciclo de reloj.
- El CPU cuenta con 32 Instrucciones de propósito general para trabajar con los registros.
- Capacidad de procesamiento de hasta 16 MIPS (Mega Instrucciones Por Segundo)

En la **Figura 48** se aprecia el diagrama de bloques del microcontrolador ATmega2560.

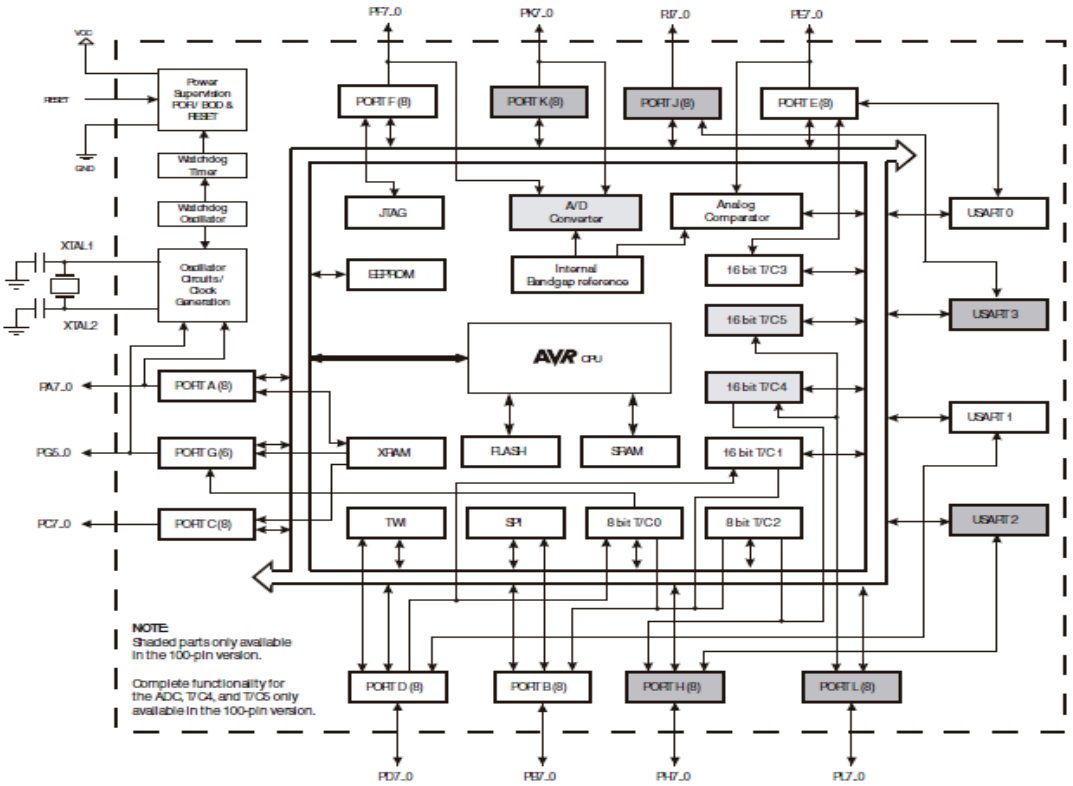



**Figura 48. Diagrama de bloques del microcontrolador ATmega2560**

Una de las ventajas de la tarjeta Arduino Mega 2560 es que dispone de un microcontrolador ATmega16U2 programado para actuar como convertidor USB a serie, la cual puede ser usado para enviar o recibir datos atreves de una computadora.

Para la programación del microcontrolador ATmega2560 en el Arduino Mega viene precargado con un gestor de arranque (bootloader) que permite cargar nuevo código sin necesidad de un programador por hardware externo. Se comunica utilizando el protocolo STK500. También puede evitarse el gestor de arranque y programar directamente el microcontrolador a través del puerto ICSP (In Circuit Serial Programming).

Dispone de 54 pines de entradas y salida (De los cuales 15 pines proveen salida PWM). Además, contiene 16 entradas analógicas. Para la comunicación con otros dispositivos dispone de:

- **4 puertos seriales UART (Transmisor-Receptor Asíncrono Universal)**:
  - Serie: Pin 0 (RX) y Pin 1 (TX)
  - Serie 1: Pin 19 (RX) y Pin 18 (TX)
  - Serie 2: Pin 17 (RX) y Pin 16 (TX)
  - Serie 3: Pin 15 (RX) y Pin 14 (TX).

  Usados para recibir (RX) transmitir (TX) datos a través de puerto serie TTL. Los pines Serie: 0 (RX) y 1 (TX) están conectados a los pines correspondientes ATmega16U2 (Convertidor USB a TTL).

- **1 puerto SPI (Interfaz periférica serial):** Pin 50 (SS), Pin 51 (MOSI), Pin 52 (MISO) y Pin 53 (SCK).
- **1 puerto I2C (Bus serie de datos):** Pin 20 (SDA) y Pin 21 (SCL).

El Arduino Mega dispone de interrupciones externas en los pines:

- Pin: 2 (interrupción 0)
- Pin: 3 (interrupción 1)
- Pin: 18 (interrupción 5)
- Pin: 19 (interrupción 4)
- Pin: 20 (interrupción 3)
- Pin: 21 (interrupción 2).

Estos pines se pueden configurar para lanzar una interrupción en un valor LOW (0V), en flancos de subida o bajada (cambio de LOW a HIGH (5V) o viceversa), o en cambios de valor.

El Arduino Mega puede ser alimentado vía la conexión USB o con una fuente de alimentación externa. El origen de la alimentación se selecciona automáticamente. La placa puede trabajar con una alimentación externa de entre 6 a 20 voltios. Si el voltaje suministrado es inferior a 7V, el pin de 5V puede proporcionar menos de 5 V y la placa puede volverse inestable; si se usan más de 12V los reguladores de voltaje se pueden sobrecalentar y dañar la placa. El rango recomendado es de 7 a 12 voltios.

Los pines de alimentación son los siguientes:

- **VIN**. La entrada de voltaje a la placa Arduino cuando se está usando una fuente externa de alimentación.
- **5V**. Utilizado para suministrar corriente a otros dispositivos de la placa. Esta puede provenir de VIN a través de un regulador integrado en la placa, o proporcionada directamente por el USB u otra fuente estabilizada de 5V.
- **3V3**. Una fuente de voltaje de 3.3V generada por un regulador integrado en la placa. La corriente máxima soportada 50mA.
- **GND**. Pines de toma de tierra.

En la **Tabla 18** se mencionan las principales características del Arduino Mega.

Tabla 18. Principales características de Arduino Mega

| Características | |
|---|---|
| **Modelo** | Arduino Mega |
| **Microcontrolador** | ATmega2560 |
| **Voltaje de Operación** | 5v |
| **Voltaje de Entrada** | 7-12V |
| **Voltaje de Entrada (límites):** | 6-20V |
| **Pines de Entrada/Salida** | 54 (de los cuales 15 proveen salida PWM) |
| **Pines análogos de entrada** | 16 |
| **Corriente DC por cada Pin Entrada/Salida** | 40 mA |
| **Corriente DC entregada en el Pin 3.3V** | 50 mA |
| **Memoria EEPROM** | 4kbytes (100,000 Ciclos de escritura) |
| **Memoria SRAM** | 8kbytes (Con tecnología flash, El chip se puede reprogramar hasta 100.000 veces) |
| **Memoria Flash** | 256 kbytes |
| **Frecuencia de operación** | 16 MHz |
| **Comunicación** | Comunicación Serial: 4 Puertos<br>Comunicación SPI: 1 Puerto<br>Comunicación I2C: 1 Puerto |
| **Interrupciones** | 6 |
| **Timer** | 6 |
| **Convertidores ADC** | 16 |

## A.2 Esquema del Arduino Mega

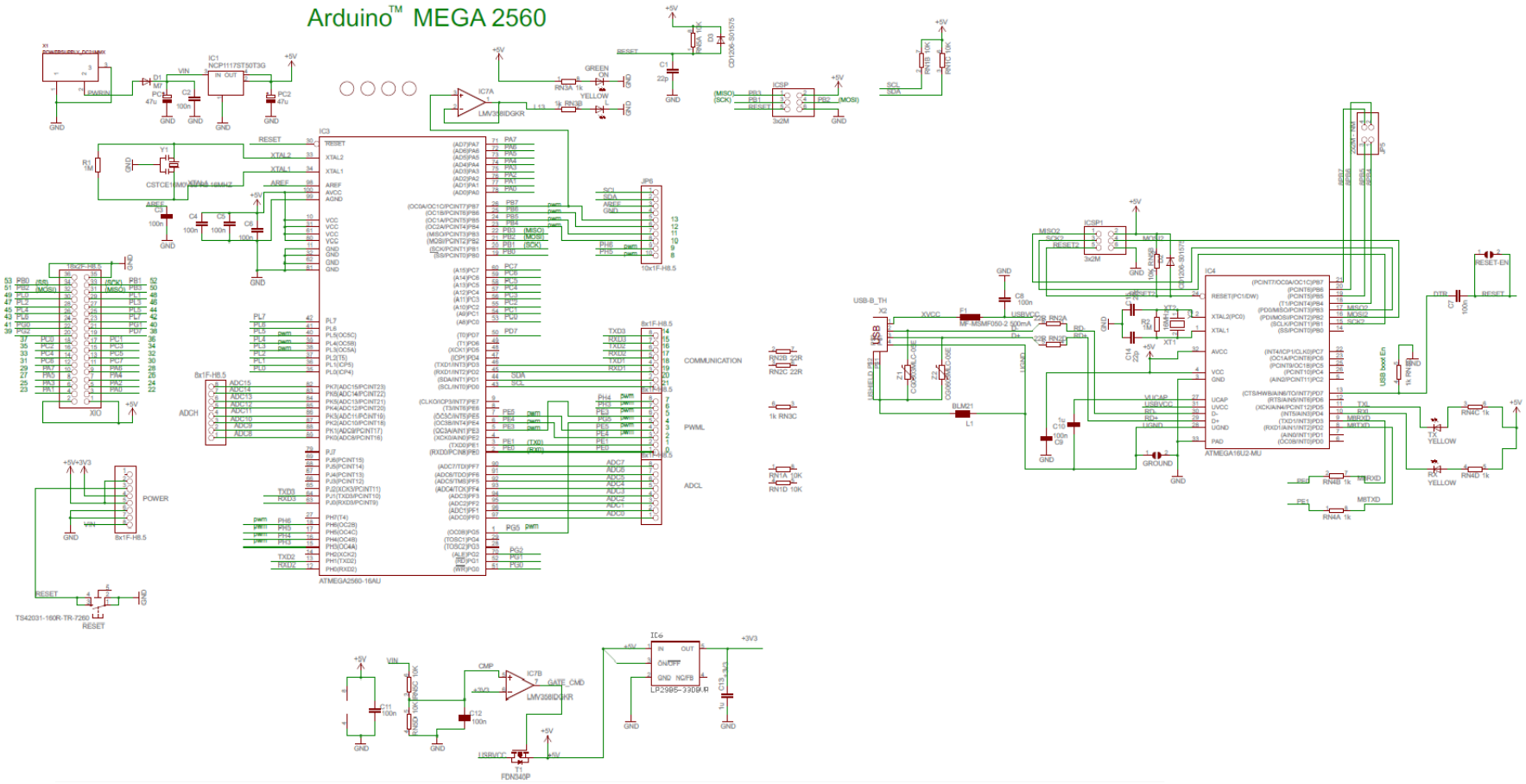



**Figura 49. Esquema del Arduino Mega**

## A.3 Mapeo de Pines del Microcontrolador Atmega2560.

A continuación, se muestra la asignación de pines para el Atmega2560. Microcontrolador utilizado en el Arduino 2560.

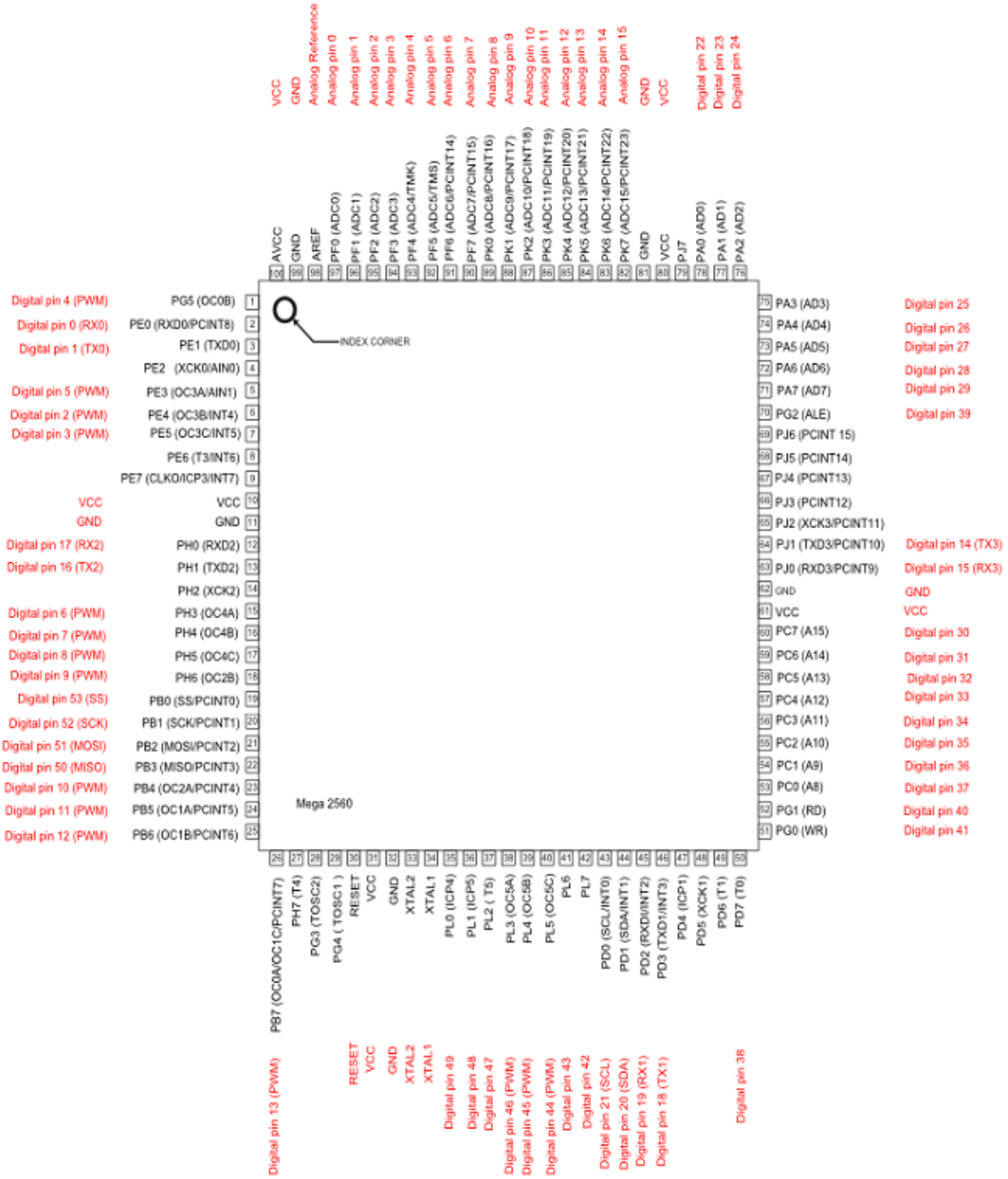



**Figura 50. Mapeo de Pines del Microcontrolador Atmega2560**

# Anexo B: Módulo RFID-RC522

## B.1 Funcionamiento del Módulo Lector RFID-RC522

El MFRC522 soporta el modo lectura y escritura para tarjetas con estándar ISO/IEC 14443 A/MIFARE utilizando diversas velocidades de transferencia y protocolos de modulación. En la **Figura 51** se aprecia el MFRC522 en modo lectura y escritura.

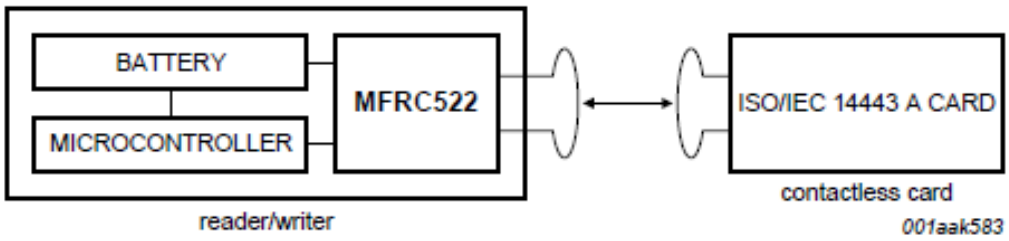



**Figura 51. MFRC522 modo lectura y escritura.**

### B.1.1 Transferencia de Datos del Lector hacia la Tarjeta Pasiva

Para la trasferencia de datos del lector hacia la tarjeta se realiza por medio de la modulación ASK (Modulación por desplazamiento de amplitud) y Codificación Miller Modificado (Ver **Figura 52**), a velocidades de transferencia de 106 kBd hasta 848 kBd.

La modulación ASK consiste en generar una señal de diferente amplitud en la onda portadora en función del dato a enviar. El MFRC522 lo hace en conjunto con el código Miller modificado.

El código Miller modificado representa un ‘1’ por una transición negativa en la mitad del periodo del dato que se quiere trasmitir, mientras que el ‘0’ binario es representado con la continuidad del nivel de la señal hasta el próximo periodo de bit con una transición positiva. En la **Figura 52** se aprecia la Modulación ASK con código Miller modificado.

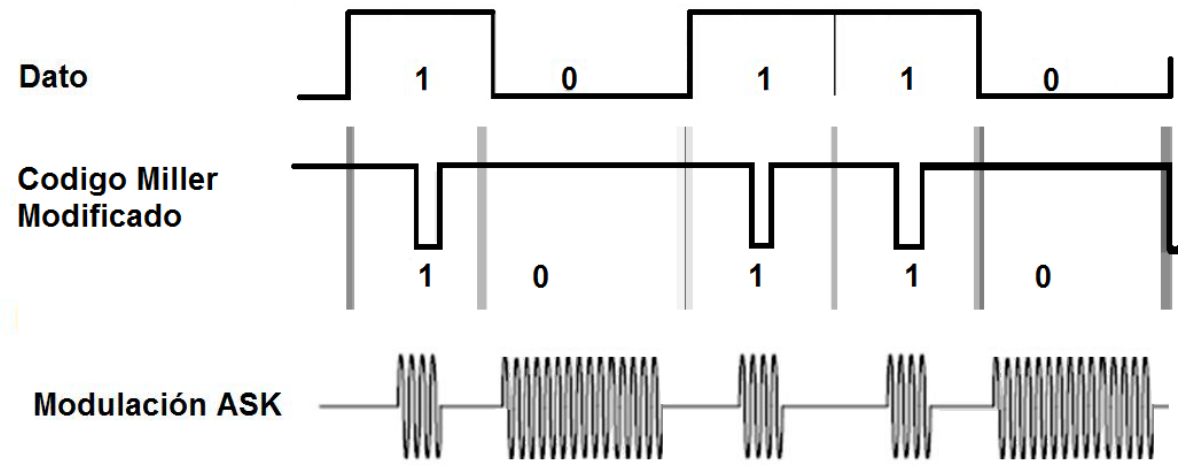



Figura 52. Modulación ASK con código Miller modificado

### B.1.2 Transferencia de Datos de la Tarjeta Pasiva hacia el Lector

Para la transferencia de datos de la tarjeta hacia el lector se realiza por medio de la modulación de subportadora, utilizando la codificación Manchester o BPSK (Modulación por desplazamiento de fase), a velocidades de transferencia de 106 kBd hasta 848 kBd.

La modulación de subportadora consiste en que primeramente el MFRC522 envía una señal modulada de tipo ASK hacia la tarjeta pasiva (Ver **Figura 53**), la tarjeta recibe esta señal y procede a realizar una segunda modulación de la subportadora con la señal portadora (La cual da como resultado la frecuencia final a la que se va a transmitir los datos de la tarjeta hacia el MFRC522).

Más técnicamente, la subportadora es una señal ya modulada, que se modula a continuación, en otra señal de frecuencia y anchura de banda más altas.

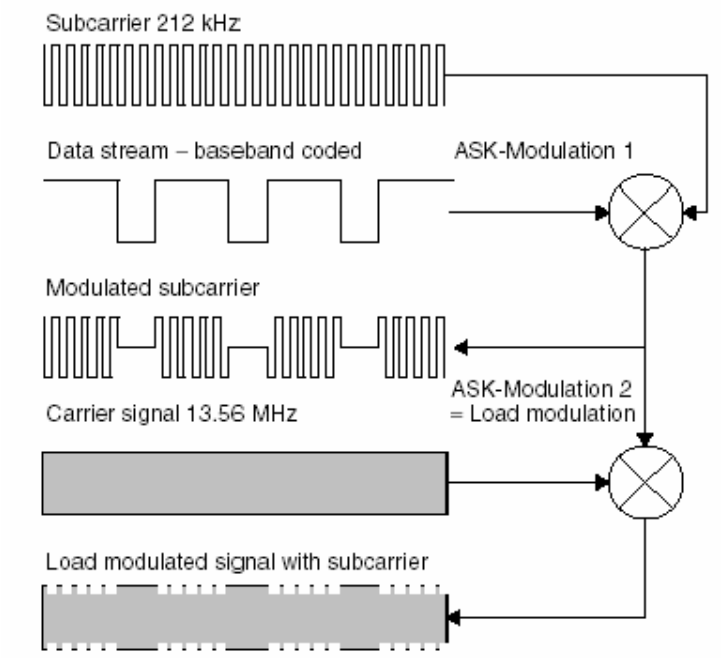



**Figura 53. Proceso detallado de una modulación múltiple, con una subportadora modulada en ASK**

La subportadora es codificada a velocidades de transferencia de 106 kBd con código Manchester, esta codificación consiste en que un ‘1’ binario es representado por una transición negativa en la mitad del periodo de bit y un ‘0’ binario es representado por una transición positiva. En la **Figura 54** se aprecia el código Manchester.

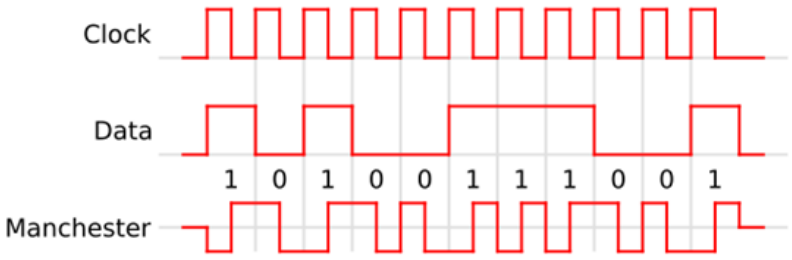



**Figura 54. Código Manchester**

A velocidades superiores a los 212 kBd la subportadora se modulada con BPSK (Modulación por Desplazamiento de Fase binaria). El esquema más simple de BPSK usa dos ondas senoidales para representar 0 y 1, como se muestra en la **Figura 55**.

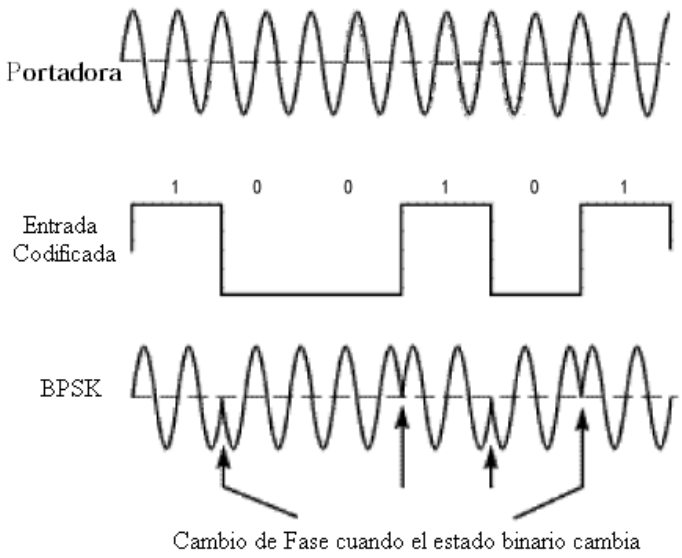



**Figura 55. Modulación BPSK (Modulación por desplazamiento de fase binaria)**

En la **Tabla 19** se aprecia con más detalles los tipos de modulaciones y codificaciones usadas por el lector (MFRC522) y las tarjetas (Estándar ISO/IEC 14443 A/MIFARE) en diferentes velocidades de transmisión.

Tabla 19. Comunicación entre el lector MFRC522 y tarjetas

| Dirección de comunicación | Tipo de Señal | Velocidad de transferencia | | | |
|---|---|---|---|---|---|
| | | 106 kBd | 212 kBd | 424 kBd | 848 kBd |
| **Lector a Tarjeta (Envió de datos del MFRC522 a la tarjeta)** | Modulación (Lector) | 100 % ASK | 100 % ASK | 100 % ASK | 100 % ASK |
| | Codificación | Miller Modificado | Miller Modificado | Miller Modificado | Miller Modificado |
| **Tarjeta a Lector (El MFRC522 recibe datos de la tarjeta)** | Modulación (Tarjeta) | Load modulation con subportadora | Load modulation con subportadora | Load modulation con subportadora | Load modulation con subportadora |
| | Frecuencia de la Subportadora | 13.56 MHz | 13.56 MHz | 13.56 MHz | 13.56 MHz |
| | Codificación | Manchester | BPSK | BPSK | BPSK |

En la **Tabla 20** se aprecia las principales características del módulo lector RFID- RC522.

Tabla 20. Características del módulo lector RFID-RC522

| Características | |
|---|---|
| Modelo | RFID-RC522 |
| Corriente de Operación | 13-26mA a 3.3V |
| Corriente de Stand By | 10-13mA a 3.3V |
| Corriente de Sleep-Mode | 80uA |

| | |
|---|---|
| Distancia de Lectura | 0 a 60mm |
| Estándar | Soporta productos con estándar ISO/IEC 14443 A/MIFARE |
| Compatibilidad | Soporta productos MF1xxS20, MF1xxS70 y MF1xxS50 encriptación en modo lectura y escritura |
| Comunicación | Soporta interfaces:<br>○ Comunicación SPI, cuenta con una velocidad máxima hasta 10 Mbit/s.<br>○ Comunicación I2C, cuenta con una velocidad de 400 kBd hasta 3400 kBd.<br>○ Comunicación serial UART RS232, cuenta con una velocidad hasta 1228.8 kBd |
| Cristal de cuarzo | 27.12 MHz |
| Velocidad de Máxima de Comunicación con la Tarjeta | 848 kBd |
| Dimensiones | 40 x 60 mm |
| Temperatura de Operación | -20 a 80ºC |
| Humedad de operación | 5%-95% |

# Anexo C: Mifare 1k S50

## C.1 Características Técnicas de las tarjetas Mifare 1k S50

### C.1.1 Organización de la Memoria

Las tarjetas Mifare 1k S50 cuentan con una memoria EEPROM de 1kb; y se encuentra dividida en 16 sectores, cada sector dispone de 4 bloques de los cuales 3 bloques se pueden guardar información del usuario (Con excepción del sector 0 que solo dispone de 2 bloques para guardar información). La información es de formato libre, y se puede modificar con comando simples de lectura y escritura. Cada bloque de memoria tiene la capacidad de almacenar hasta 16 Bytes. En la **Figura 56** se aprecia la organización de la memoria de las tarjetas Mifare 1k S50.

| | | Byte Number within a Block | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Sector** | **Block** | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | **Description** |
| 15 | 3 | Key A | | | | | | Access Bits | | | | Key B | | | | | | Sector Trailer 15 |
| | 2 | | | | | | | | | | | | | | | | | Data |
| | 1 | | | | | | | | | | | | | | | | | Data |
| | 0 | | | | | | | | | | | | | | | | | Data |
| 14 | 3 | Key A | | | | | | Access Bits | | | | Key B | | | | | | Sector Trailer 14 |
| | 2 | | | | | | | | | | | | | | | | | Data |
| | 1 | | | | | | | | | | | | | | | | | Data |
| | 0 | | | | | | | | | | | | | | | | | Data |
| : | : | | | | | | | | | | | | | | | | | |
| : | : | | | | | | | | | | | | | | | | | |
| : | : | | | | | | | | | | | | | | | | | |
| 1 | 3 | Key A | | | | | | Access Bits | | | | Key B | | | | | | Sector Trailer 1 |
| | 2 | | | | | | | | | | | | | | | | | Data |
| | 1 | | | | | | | | | | | | | | | | | Data |
| | 0 | | | | | | | | | | | | | | | | | Data |
| 0 | 3 | Key A | | | | | | Access Bits | | | | Key B | | | | | | Sector Trailer 0 |
| | 2 | | | | | | | | | | | | | | | | | Data |
| | 1 | | | | | | | | | | | | | | | | | Data |
| | 0 | | | | | | | | | | | | | | | | | Manufacturer Block |

**Figura 56. Organización de la memoria EEPROM de las tarjetas Mifare 1k S50**

### C.1.1.1 Bloque de Manufactura

Estas tarjetas son los elementos que proporcionarán la identificación por radiofrecuencia del usuario que accede al interior del inmueble. Por su construcción, proporcionan alta seguridad y no son fácilmente duplicables.

Desde su fabricación es grabada (en el bloque 0 y sector 0) con un identificador único (conocido como UID por sus siglas en inglés) que no es posible de modificar (Ver **Figura 57**), por lo cual solo posee exclusivamente permisos de lectura establecidos por hardware.

Los primeros 4 bytes son el identificador único de la tarjeta. El siguiente byte, conocido como BCC, equivale a la aplicación sucesiva de XOR sobre todos los bytes del UID. Por último, los demás 11 bytes contienen datos de la empresa manufacturera de la tarjeta.

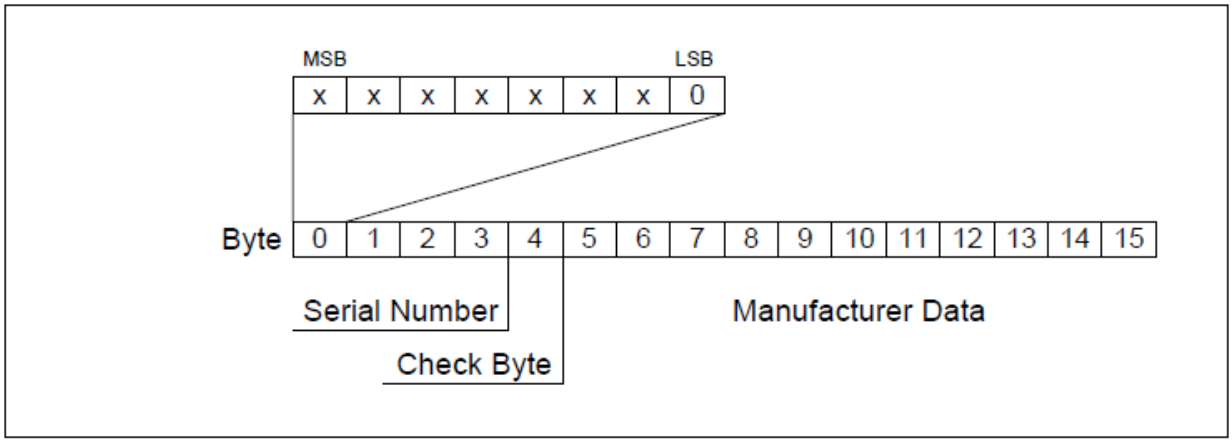



**Figura 57. Bloque de manufactura**

### C.1.1.2 Bloques de Datos

Los sectores utilizan dos claves de acceso llamadas 'Key A' y 'Key B (Opcional)', como se muestra en la **Figura 58**. Estas llaves se almacenan en el cuarto bloque (Bloque N° 3) junto con los permisos de acceso a cada uno de los

tres bloques (por lo que no son hábiles para almacenar datos). Estos permisos pueden ser: lectura y escritura.

Estas llaves y condiciones de acceso ocupan un rol importante en el proceso de cifrado de la comunicación tarjeta - lector. Por ejemplo, para efectuar operaciones sobre algún bloque es necesario primero autentificarse con el sector al cual pertenece dicho bloque.

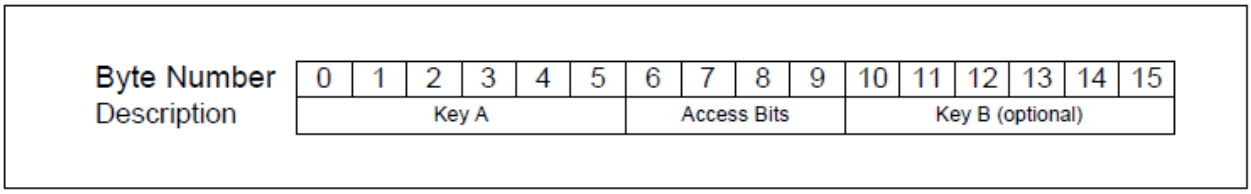



**Figura 58. Bloque N° 3**

Los bloques restantes pueden dividirse en dos tipos dependiendo de la configuración explicitada en los bits con las condiciones de acceso de su sector: bloques de datos y bloques de valor. Un bloque se denomina bloque de datos cuando sus 16 bytes están disponibles para ser utilizados para almacenar datos, sin mayor limitación.

Por el otro lado, cuando se usa un bloque como bloque de valor este sólo tiene 4 bytes disponibles para guardar un valor numérico con signo, sobre el cual es posible realizar operaciones como incremento, decremento, restaurar y transferir, además de las básicas operaciones de lectura y escritura. El espacio restante se utiliza para respaldar este valor una vez más y almacenar también su inverso. Los últimos cuatro bytes, se utilizan para almacenar una dirección del bloque de tamaño 1 byte, valor también duplicado junto a su inverso dos veces (Ver **Figura 59**). Para almacenar los valores negativos, se utiliza el estándar two's complement.

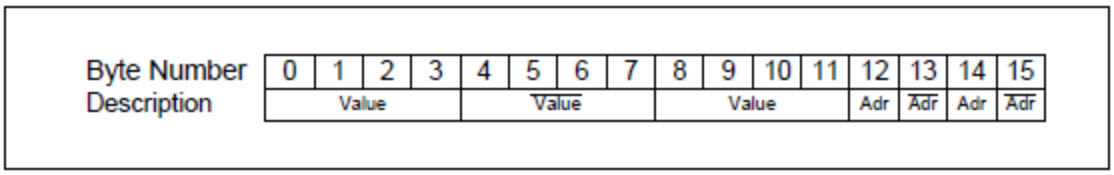



**Figura 59. Representación de los bytes de un bloque de valor**

Cabe destacar, además que toda la información respaldada en bloques se guarda siguiendo la representación de palabras little-endian, es decir, almacenando desde el byte menos significativo hacia el más significativo de los valores, en la dirección más pequeña de la memoria hacia la más grande.

### C.1.2 Seguridad

Una vez que se acerca la tarjeta a un lector, ésta se activa e inicia un proceso de intercambio con el lector para establecer una comunicación cifrada CRYPTO1. La cual, está diseñado para proveer protección contra escucha del canal, y no para autenticar la tarjeta o el lector.

#### C.1.2.1 CRYPTO1

CRYPTO1 es un cifrador de flujo, esto es, un cifrador de clave simétrica, donde dígitos de texto plano se combinan con una corriente de dígitos pseudoaleatorio, conocida como keystream. En este cifrado, cada dígito de texto plano es operado uno a la vez con el dígito correspondiente del keystream, produciendo un nuevo dígito de la corriente de texto cifrado.

El algoritmo de cifrado de CRYPTO1 es un registro de desplazamiento con retroalimentación lineal (conocidos como LFSR por su sigla en inglés) de 48 bits.

Su polinomio generador del registro es:

P(x) = x 48+x 43+x 39+x 38+x 36+x 34+x 33+x 31+x 29+x 24+x 23+x 21+x 19+x 13+x 9+x 7+x 6+x 5+1

## C. 1.3 Protocolo de comunicación

Para comenzar las interacciones tarjeta-lector primero necesitamos poder alimentar energéticamente nuestra tarjeta entrando al campo magnético del lector. Una vez energizada comienza inmediatamente el proceso conocido como protocolo anticolisión, seguido por el protocolo de autenticación para luego realizar la comunicación encriptada.

El protocolo anticolisión tiene como idea el poder realizar transacciones de modo seguro con una única tarjeta, a pesar de que existieran más de ellas en el campo de lectura. Este comienza con el envío del UID por parte de la tarjeta al lector, quien luego selecciona a la tarjeta respondiendo con el mismo UID, estableciendo así la comunicación.

El protocolo de autenticación tiene cabida al intentar realizarse alguna operación sobre los datos de la tarjeta. Las operaciones modulares, como escritura o lectura, se efectúan sobre bloques completos, por lo cual el lector envía una solicitud a la tarjeta para acceder al sector al cual pertenece el bloque en cuestión. En este momento la tarjeta responde con un desafío aleatorio de 32 bits al lector. Desde este paso, toda la comunicación efectuada es encriptada, esto quiere decir, que la información en cuestión es enviada luego de ser operada, mediante XOR ($\oplus$) con el keystream producido por CRYPTO1. Así, el lector responderá cifrando su propio desafío de 32 bits, más una respuesta al desafío enviado por la tarjeta de 32 bits y también cifrada. Si todo sigue bien, la tarjeta responderá el desafío del lector cifrando esta respuesta y enviándola, completándose así la autenticación.

Hecho esto, la comunicación encriptada puede realizarse sin problemas con el sector autenticado con lo que pueden efectuarse las operaciones pertinentes, respetando siempre los bits con las condiciones de acceso del sector.

### C.1.4 Integridad de Datos

Para trasmitir la información contenida en la tarjeta hacia el lector y evitar un riesgo muy elevado de pérdida de información, ocasionado por la interferencia de ruido externo que produce errores en la trasmisión. Estas tarjetas disponen de mecanismos de protección para la trasmisión de datos muy fiable de la cual se puede destacar:

- 16 bits CRC por bloque
- Bits de paridad para cada byte
- Comprobación del número de bits
- Codificación de bits para distinguir entre "1", "0", y ninguna información
- Monitorización de canales (secuencia de protocolos y análisis de flujo de bits)

#### C.1.4.1 Control de redundancia cíclica (CRC)

El CRC (Control de redundancia cíclica) es código de detección de errores, el lector y la tarjeta cuentan con un polinomio predeterminado para realizar la división, como se muestra en la **Figura 60**, antes que la tarjeta envié los datos primero calcula el CRC y seguido lo anexa al final de los datos, luego cuando estos datos son recibidos por el lector realiza la misma operación hasta llegar a un residuo; el resultado debe ser cero, sino existe error.

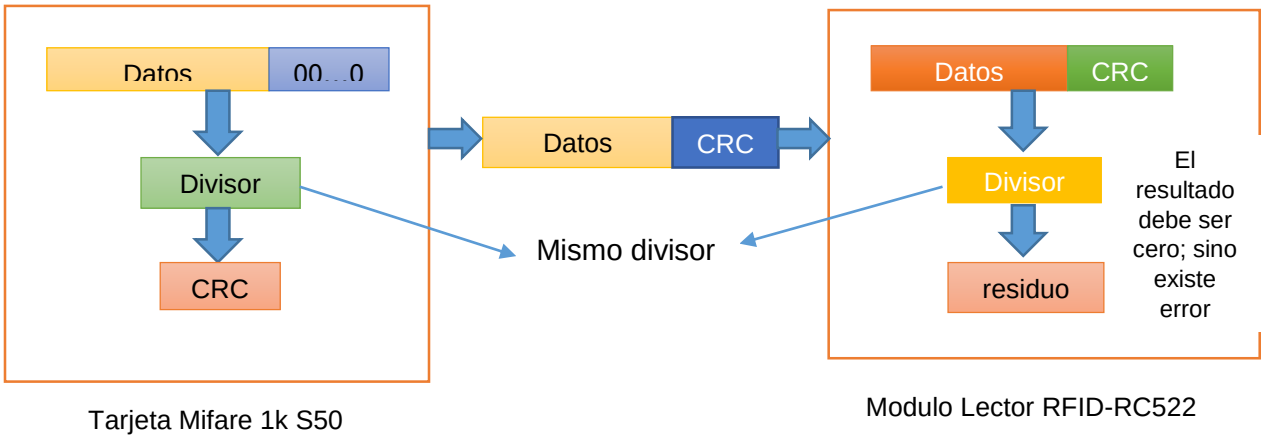



**Figura 60. Código de redundancia cíclica (Entre la tarjeta y el lector RFID)**

### C.1.4.2 Bit de Paridad

El bit de paridad consiste en añadir un bit de más a la cadena datos que se va enviar, y que nos indicará si el número de unos (bits puestos a 1) es par o es impar. Si es par se incluirá ese bit con el valor = 0, y si no es así, lo se incluirá con valor = 1. Cuando esta cadena es enviada al lector realiza la misma operación.

## C.2 Norma ISO 14443 (Proximity Cards)

La norma ISO 14443 define la comunicación con tarjetas inteligentes, lectores y dispositivos NFC definidos dentro de esta norma.

En esta norma se tiene dos tipos de estándares A y B que se emplean en la capa de enlace para la transmisión. Cualquier tipo de sistema de comunicación que cuente con este estándar ISO 14443 puede emplear este tipo de tarjetas para la cual:

- La transmisión entre lector y tarjeta se fija el estándar de comunicación y los protocolos.
- Trabaja dentro de la banda ISM (Industrial, Scientific and Medical) en la frecuencia 13.56 MHz.
- La lectura o escritura entre la tarjeta y el dispositivo tiene que estar en un rango de 10 cm.
- Trabaja a una velocidad de 106 Kbps predeterminada con el fin de evitar colisiones.
- Su nivel de seguridad es muy confiable ya que cuenta con un mecanismo de microprocesadores para la autentificación, posee mensajería de seguridad y tokens criptográficos.

El estándar ISO 14443 consta de cuatro partes y se describen dos tipos de tarjetas: tipo A y tipo B. Las principales diferencias entre estos tipos se encuentran

en los métodos de modulación, codificación de los planes (parte 2) y el protocolo de inicialización de los procedimientos (parte 3).

Las tarjetas de ambos tipos (A y B) utilizan el mismo protocolo de alto nivel (llamado T=CL) que se describe en la parte 4. El protocolo T=CL especifica los bloques de datos y los mecanismos de intercambio:

- Bloque de datos de encadenamiento
- Tiempo de espera de extensión
- Múltiple activación

### C.2.1 Banda ISM

ISM (Industrial Scientific and Medical) es una banda de frecuencia para aplicaciones industriales, médicas y científicas definidas por la ITU (International Telecommunication Union) en el artículo 5 de las regulaciones de radio, concretamente puntos 5.138 y 5.150.

El uso de la banda ISM es abierto a todo usuario sin utilizar licencia, siempre y cuando se respeten las regulaciones que limitan los niveles de potencia. Este hecho fuerza a que las comunicaciones en ISM toleren cierta cantidad de errores, además de requerir mecanismos de protección contra interferencias.

#### C.2.1.1 Articulo 5 de las Regulaciones de Radio de la ITU

En el artículo 5.150 del reglamento de radiocomunicaciones de la ITU, establece las bandas que están designadas para aplicaciones industriales, científicas y médicas, como se aprecia en la siguiente tabla:

**Tabla 21. Frecuencias operan en la Banda ISM**

| Bandas ISM | |
|---|---|
| **13,553 - 13,567 kHz** | Frecuencia central 13,560 kHz |
| **26,957 - 27,283 kHz** | Frecuencia central 27,120 kHz |
| **40,66 - 40,70 MHz** | Frecuencia central 40,68 MHz |
| **902 - 928 MHz** | En la Región 2 (Frecuencia central 915 MHz) |
| **2,400 - 2,500 MHz** | Frecuencia central 2,450 MHz |
| **5,725 - 5,875 MHz** | Frecuencia central 5,800 MHz |
| **24 - 24,25 GHz** | Frecuencia central 24,125 GHz |

# Anexo D: Módulo Display LCD

## D.1 Características Técnicas del Módulo LCD con controlador HD44780

### D.1.1 Conexión

El número de pines del Display es 16 y son compatibles TTL. En la **Tabla 22** se muestra el significado de las señales de cada pin. Hay tres tipos de señales en el LCD: de alimentación, de control y de datos.

**Tabla 22. Descripción de los pines del LCD**

| PIN | SÍMBOLO | DESCRIPCIÓN |
|---|---|---|
| **1** | VSS | Alimentación negativa |
| **2** | VDD | Alimentación positiva |
| **3** | VO | Ajuste del contraste |
| **4** | RS | Selección de Registro:<br>RS = 1 ⇒ Registro de Datos<br>RS = 0 ⇒ Registro de Instrucciones |
| **5** | R/W | Lectura / Escritura:<br>R/W = 1 ⇒ Lectura<br>R/W = 0⇒ Escritura |
| **6** | E | Habilitación del Display<br>E = 1 ⇒ Habilitado<br>E = 0 ⇒ Deshabilitado |
| **7-14** | DB[0..7] | Pines del 0 al 7 del bus de datos bidireccional |
| **15-16** | Led | Iluminación |

La **señal de alimentación** corresponde a los pines 1, 2 y 3. El pin 1 corresponde al negativo, el 2 a la alimentación positiva (normalmente +5Vdc) y el 3 al ajuste del contraste. Habitualmente, al dar alimentación y sin haber mandado

todavía ningún comando, el Display muestra en su pantalla la primera fila con todos sus caracteres en negro. Si no fuera así, se debería proceder al ajuste del contraste. Para ello, se debe instalar un potenciómetro de unos 10KΩ tal y como se observa en la **Figura 61**. Cuanto más cercano a masa esté el voltaje en el pin 3 (VO) mayor será el contraste.

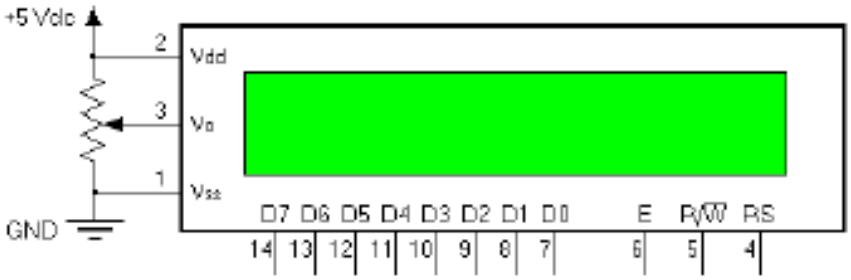



**Figura 61. Conexión del potenciómetro de ajuste de contraste del módulo Display LCD**

Para la **señal de control** corresponde pines 4, 5 y 6. El **pin 4** (RS) sirve para seleccionar el registro de datos o el de instrucciones, poniendo RS=1 o RS=0, respectivamente. El **pin 5** permite leer (R/W=1) o escribir (R/W=0) en el módulo LCD tanto datos como instrucciones. Y, por último, el **pin 6** (E) permite habilitar, con E=1, o deshabilitar el Display (E=0). Sólo cuando está habilitado nos podemos comunicar con él.

Por último, la **señal de datos** corresponde a los **pines del 7 al 14** forman un bus de datos bidireccional de 8 bits (DB7 – DB0) por donde se pueden escribir datos e instrucciones y se puede leer el estado del Display (si está o no ocupado, la posición actual del cursor, etc.). El LCD también puede ser gobernado con un bus de datos de 4 bits.

### D.1.2 DDRAM (Display Data RAM)

El módulo LCD posee una zona de memoria RAM llamada DDRAM (Data Display RAM) donde se almacenan los caracteres que se van a mostrar en la pantalla.

Tiene una capacidad de 80 bytes, 40 por cada línea, de los cuales sólo 32 se pueden visualizar a la vez (16 bytes por línea).

### D.1.3 CGRAM (Character Generator RAM)

Es el área de memoria RAM interna del LCD donde el usuario puede definir sus propios caracteres o gráficos. El tamaño de la CGRAM es de 64 bytes lo que permite crear hasta 8 caracteres de 5×7 puntos o 4 de 5×10. Los caracteres son en realidad de 5×8 puntos, pero las fuentes están definidas en 5×7.

### D.1.4 CGROM (Character Generator ROM)

El LCD dispone de una zona de memoria interna no volátil llamada CGROM donde se almacena una tabla con los 192 caracteres que pueden ser visualizados (Ver **Figura 62**). Cada uno de los caracteres tiene su representación binaria de 8 bits. Para visualizar un carácter debe recibir por el bus de datos el código correspondiente.

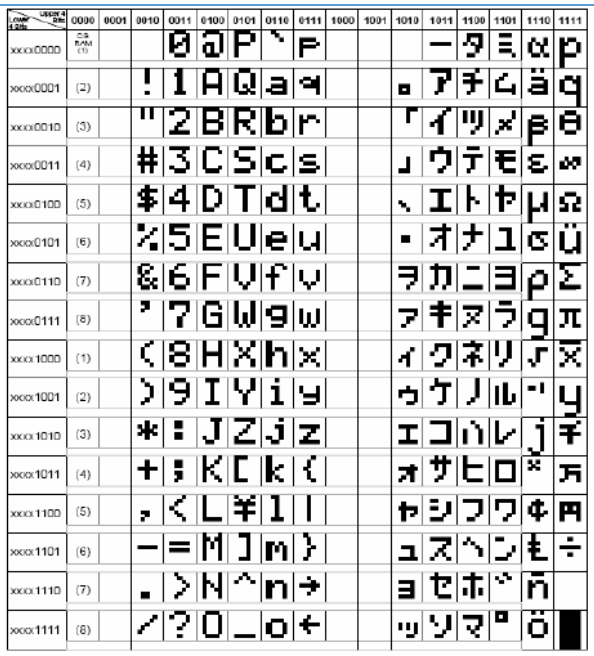


**Figura 62. Juego de caracteres ASCII del HD44780**

### D.1.5 Registros internos

El HD44780 tiene dos registros internos de 8 bits, un registro de datos (DR) y un registro de instrucciones (IR), que se pueden leer y escribir.

El registro de instrucciones almacena el código de la instrucción (clear Display, cursor home, set DDRAM address, etc.) cuando es escrito, mientras que en modo lectura permite leer el bit de ocupado (Busy Flag) y la posición actual del cursor.

El registro de datos almacena de forma temporal el dato que va a ser escrito/leído tanto en/de DDRAM como en/de CGRAM. Los datos escritos en el DR son transferidos automáticamente a la DDRAM o CGRAM mediante una operación interna del propio controlador.

Estos dos registros pueden ser seleccionados mediante la señal RS (pin 4) en modo lectura o escritura según la señal R/W (Pin 5) (Ver **Tabla 23**).

**Tabla 23. Descripción de los pines de un LCD alfanumérico**

| RS | R/W | FUNCIÓN |
|---|---|---|
| **0** | 0 | Escribe en el IR y ejecuta operación interna (clear Display, cursor home,etc.) |
| **0** | 1 | Lee el IR. Lee Busy Flag (DB7) y Address Counter (DB0–DB6) |
| **1** | 0 | Escribe dato en DR y ejecuta operación interna (DR→DDRAM o DR→CGRAM) |
| **1** | 1 | Lee un dato del DR y ejecuta operación interna (DDRAM→DR o CGRAM→DR) |

# Anexo E: Protocolos de Comunicación

## E.1 MÓDULO PUERTO SERIE SÍNCRONO MAESTRO (MSSP)

El MSSP (Puerto serie síncrono maestro – Master Synchronous Serial Port) es un módulo muy útil, y a la vez uno de los circuitos más complejos dentro del microcontrolador. Este módulo permite la comunicación de alta velocidad entre un microcontrolador y otros periféricos u otros microcontroladores al utilizar varias líneas de E/S (como máximo dos o tres líneas). Por eso, se utiliza con frecuencia para conectar el microcontrolador a los visualizadores LCD, los convertidores A/D, las memorias EEPROM seriales, los registros de desplazamiento etc. La característica principal de este tipo de comunicación es que es síncrona y adecuada para ser utilizada en sistemas con un sólo maestro y uno o más esclavos. Un dispositivo maestro contiene un circuito para generación de baudios y además, suministra señales de reloj a todos los dispositivos del sistema. Los dispositivos esclavos no disponen de un circuito interno para generación de señales de reloj. El módulo MSSP puede funcionar en uno de dos modos:

- modo SPI (Interfaz periférica serial – Serial Peripheral Interface); y
- modo I2C (Circuito inter-integrado – Inter-Integrated Circuit).

Como se muestra en la siguiente figura, un módulo MSSP representa sólo una mitad de un hardware necesario para establecer una comunicación serial, mientras que la otra mitad se almacena en el dispositivo con el que intercambia los datos. Aunque los módulos en ambas puntas de línea son los mismos, sus modos de funcionamiento difieren esencialmente dependiendo de si el módulo funciona como Maestro o como Esclavo:

Si el microcontrolador a ser programado controla otro dispositivo o circuito (periféricos), deberá funcionar como un dispositivo maestro. Este módulo generará señal de reloj cuando sea necesario, o sea sólo cuando se requiera recibir y transmitir los datos por software. Por consiguiente, el establecimiento de conexión depende únicamente del dispositivo maestro.

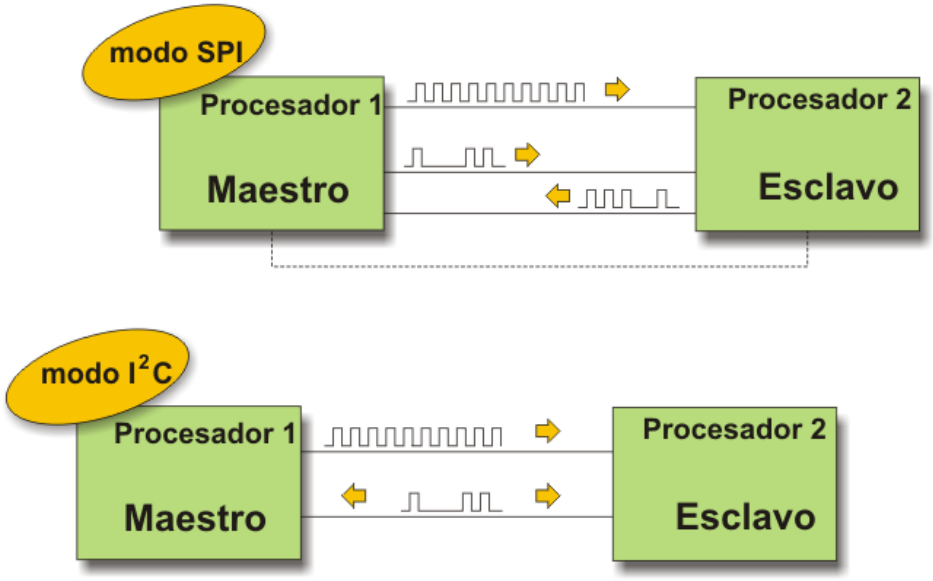



**Figura 63. Modos de funcionamiento del módulo MSSP**

De lo contrario, si el microcontrolador a ser programado está integrado en un dispositivo más complejo (por ejemplo, en una PC), deberá funcionar como un dispositivo esclavo. Como tal, un esclavo siempre tiene que esperar a que un dispositivo maestro envíe la solicitud de transmisión de datos.

### E.1.1 MODO SPI

El modo SPI permite la transmisión y recepción simultánea de datos de 8 bits al utilizar tres líneas de entrada/salida

- SDO – Serial Data Out (salida de datos serie)- línea de transmisión;
- SDI – Serial Data In (entrada de datos serie) – línea de recepción; y
- SCK – Serial Clock (reloj de comunicación) – línea de sincronización.

Adicionalmente, hay una cuarta línea (SS) que se puede utilizar si el microcontrolador intercambia los datos con varios dispositivos periféricos. Refiérase a la siguiente figura.

SS – Slave Select (Selección de esclavo) – Es una línea adicional utilizada para la selección de un dispositivo específico. Esta línea está activa sólo si el microcontrolador funciona como esclavo, o sea cuando el dispositivo externo – maestro requiere intercambiar los datos.

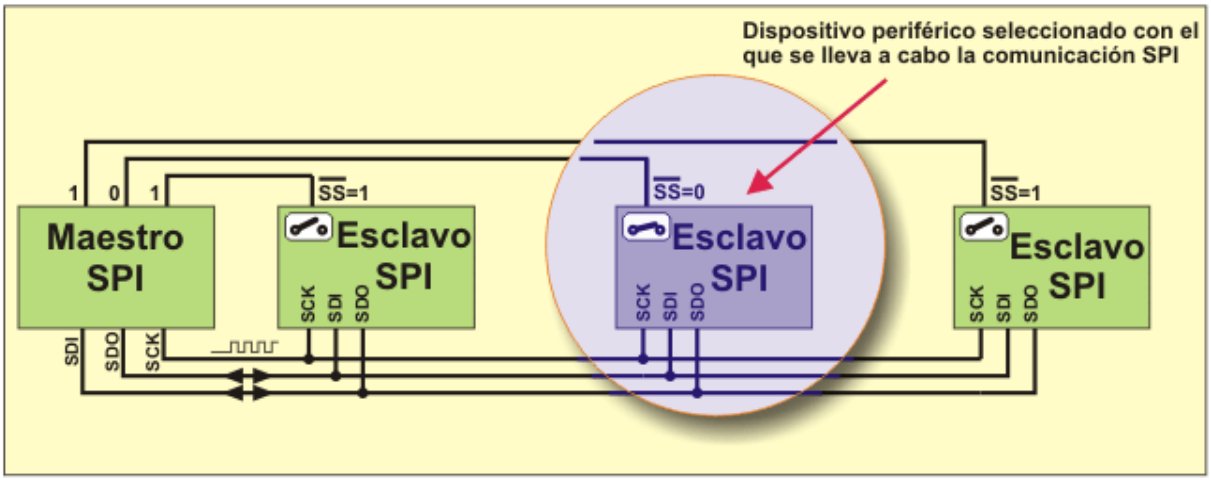



**Figura 64. Protocolo de comunicación SPI**

## E. 2 EUSART

El módulo Transmisor/Receptor Universal Síncrono/Asíncrono mejorado (Enhanced Universal Synchronous Asynchronous Receiver Transmitter EUSART) es un periférico de comunicación serie de entrada/salida. Asimismo, es conocido como Interfaz de comunicación serie (Serial Communications Interface SCI). Contiene todos los generadores de señales de reloj, registros de desplazamiento y búfers de datos necesarios para realizar transmisión de datos serie de entrada/salida, independientemente de la ejecución de programa del dispositivo.

Como indica su nombre, aparte de utilizar el reloj para la sincronización, este módulo puede establecer la conexión asíncrona, lo que lo hace único para algunas aplicaciones. Por ejemplo, en caso de que sea difícil o imposible proporcionar canales especiales para transmisión y recepción de datos y señales

de reloj (por ejemplo, mando a distancia de radio o infrarrojas), el módulo EUSART es definitivamente la mejor opción posible.

### E.2.1 EUSART EN MODO ASÍNCRONO

El EUSART transmite y recibe los datos utilizando la codificación de no retorno a cero NRZ (non-return-to-zero). Como se muestra en la siguiente figura, no se utiliza una señal de reloj y los datos se transmiten de forma muy simple:

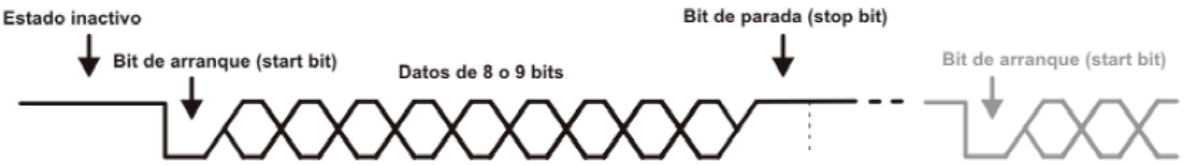



**Figura 65. EUSART EN MODO ASÍNCRONO**

Cada dato se transmite de la siguiente forma:

- En estado inactivo la línea de datos permanece en estado alto (1);
- Cada transmisión de datos comienza con un bit de arranque (START), el cual, siempre es cero (0);
- Cada dato tiene un ancho de 8 o 9 bits (primero se transmite el bit menos significativo LSB); y
- Cada transmisión de datos termina con un bit de parada (STOP), el cual, siempre es uno (1).

# Anexo F: Manual de Usuario

## F.1 Conexiones

El sistema está compuesto por dos cajas, la **Caja 1** contiene el módulo Display LCD, el modulo lector RFID y el teclado. La **Caja 2** contiene el Arduino y la placa PCB.

Para la instalación se requiere que la **Caja 1** se encuentre en la parte de afuera del local y la **Caja 2** junto con la fuente de voltaje, el botón de salida y la alarma se encuentren adentro del local. Para realizar las conexiones de los dispositivos que conforman la **Caja 1** con los dispositivos que contiene la **Caja 2** se requiere de perforar la pared.

Luego se prosigue a realizar las conexiones del módulo Display LCD, el modulo lector RFID y el teclado junto con la cerradura electromagnética y la sirena a los pines correspondientes del Arduino y la placa PCB.

La placa PCB dispone de los siguientes terminales (Ver **Figura 66**):

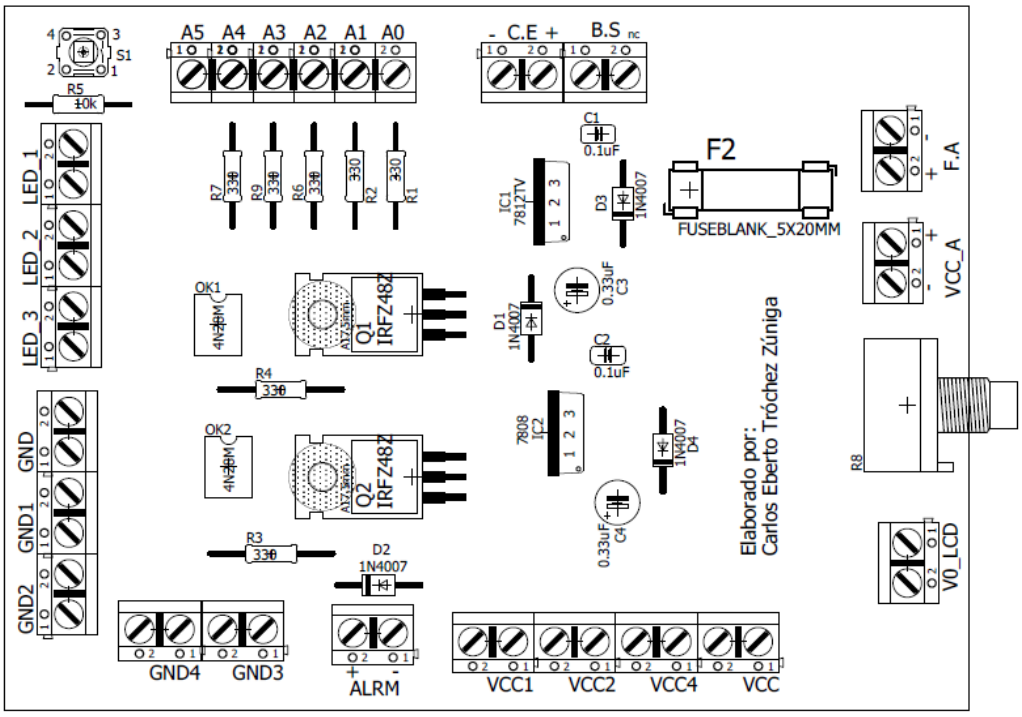



**Figura 66. Terminales del PCB**

A continuación, se describen las terminales del PCB:

- **F.A:** Fuente de alimentación
- **VCC_A:** Alimentación del Arduino
- **B.S**: Botón de salida
- **C**.E: Cerradura electromagnética
- **A0**: Pin A0 del Arduino
- **A1**: Pin A1 del Arduino
- **A2**: Pin A2 del Arduino
- **A3**: Pin A3 del Arduino
- **A4**: Pin A4 del Arduino
- **A5**: Pin A5 del Arduino
- **LED_1**: Led Rojo
- **LED_2**: Led Azul
- **LED_3**: Led Verde
- **GND**: Tierra del Arduino
- **ALARM**: Sirena
- **VCC**: 5V del Arduino
- **V0_LCD**: Pin V0 del módulo Display

Las conexiones del módulo Display LCD se realizan de la siguiente manera (Ver **Tabla 24, 25**):

**Tabla 24. Conexión del módulo Display LCD con las terminales de la tarjeta PCB**

| Pin del Módulo Display LCD | Conexión | Terminales de la Tarjeta PCB |
|---|---|---|
| **VDD** | ⟷ | VCC |
| **VSS** | ⟷ | GND |
| **V0** | ⟷ | V0_LCD (1) |
| **RW** | ⟷ | GND |

| | | |
|---|---|---|
| **A** | ⟷ | VCC |
| **K** | ⟷ | GND |

**Tabla 25. Conexiones del módulo Display LCD con los pines del Arduino Mega**

| Pin del Módulo Display LCD | Conexión | Pin del Arduino Mega |
|---|---|---|
| **RS** | ⟷ | A8 |
| **E** | ⟷ | A9 |
| **D4** | ⟷ | A10 |
| **D5** | ⟷ | A11 |
| **D6** | ⟷ | A12 |
| **D7** | ⟷ | A13 |

Las conexiones del módulo lector RFID se realizan de la siguiente manera (Ver **Tabla 26**):

**Tabla 26. Conexión del módulo lector RFID con los pines del Arduino Mega**

| Pin del Módulo Lector RFID | Conexión | Pin del Arduino Mega |
|---|---|---|
| **3.3V** | ⟷ | 3.3V |
| **RST** | ⟷ | 9 |
| **MISO** | ⟷ | 50 |
| **MOSI** | ⟷ | 51 |
| **SCK** | ⟷ | 52 |
| **SDA** | ⟷ | 53 |
| **GND** | ⟷ | GND |

Para la conexión del teclado 4x4 vamos a etiquetar de izquierda a derecha los pines, el primer pin que comienza en la parte izquierda lo etiquetamos como número 1. La conexión se realiza de la siguiente manera (Ver **Tabla 27**):

**Tabla 27. Pines del Teclado con el Arduino Mega**

| Pines del Teclado | Conexión | Pines del Arduino Mega |
|---|---|---|
| **1** | ⟷ | 37 |
| **2** | ⟷ | 35 |
| **3** | ⟷ | 33 |
| **4** | ⟷ | 31 |
| **5** | ⟷ | 29 |
| **6** | ⟷ | 27 |
| **7** | ⟷ | 25 |
| **8** | ⟷ | 23 |

La conexión de las terminales del Arduino Mega con la placa PCB se realiza de la siguiente manera (Ver **Tabla 28**):

**Tabla 28. Conexión del Arduino Mega con la tarjeta PCB**

| Pines del Arduino Mega | Conexión | Terminales de la tarjeta PCB |
|---|---|---|
| **A0** | ⟷ | A0 |
| **A1** | ⟷ | A1 |
| **A2** | ⟷ | A2 |
| **A3** | ⟷ | A3 |
| **A4** | ⟷ | A4 |
| **5** | ⟷ | A5 |
| **VIN** | ⟷ | VCC_A (+) |
| **GND 1** | ⟷ | VCC_A (-) |
| **GND 2** | ⟷ | GND |

| | | |
|---|---|---|
| **5V** | ⟷ | VCC |

La conexión de la sirena se realiza de la siguiente manera (Ver **Tabla 29**):

**Tabla 29. Conexión de la sirena con la tarjeta PCB**

| Sirena | Conexión | Terminal Tarjeta PCB |
|---|---|---|
| **Rojo (+)** | ⟷ | ALARM (+) |
| **Negro (-)** | ⟷ | ALARM (-) |

La conexión de la cerradura electromagnética se realiza de la siguiente manera (Ver **Tabla 30**):

**Tabla 30. Conexión de la cerradura electromagnética con la tarjeta PCB**

| Cerradura electromagnética | Conexión | Terminal Tarjeta PCB |
|---|---|---|
| **Rojo (+)** | ⟷ | C.E (+) |
| **Negro (-)** | ⟷ | C.E (-) |

Las conexiones de los leds se realizan de la siguiente manera (Ver **Tabla 31**):

**Tabla 31. Conexión de los leds con la tarjeta PCB**

| LED | Conexión | Terminal Tarjeta PCB |
|---|---|---|
| **Rojo** | ⟷ | LED_1 |
| **Azul** | ⟷ | LED_2 |
| **Verde** | ⟷ | LED_3 |

La conexión de la fuente de alimentación se realiza de la siguiente manera (Ver **Tabla 32)**:

Tabla 32. Conexión de la fuente de alimentación

| Conexión | |
|---|---|
| ≟ | Tierra |
| **N** | línea nula |
| **L** | línea viva |
| **BAT+** | Conectar el positivo de la batería |
| **BAT-** | Conectar el negativo de la batería |
| **V+** | Conectar el positivo de la PCB (F.A) |
| **VI-** | Conectar el negativo de la PCB (F.A) |

## F.2 Instalación de la Cerradura Electromagnética

La cerradura electromagnética está compuesta por (Ver **Figura 67**):

- Platina
- Electroimán
- Pieza polar

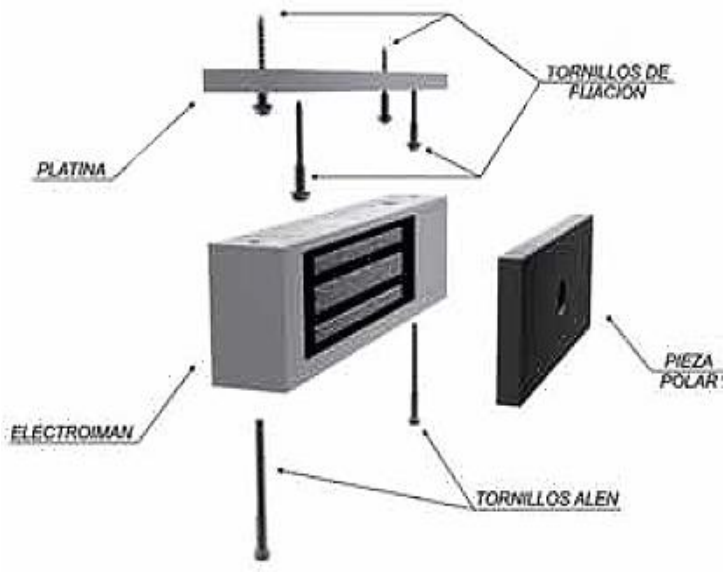



Figura 67. Partes de la cerradura electromagnética

La instalación de una cerradura electromagnética no reviste mayores complejidades y para hacerlo de manera correcta solo hay que seguir estos sencillos pasos:

- Montar rígidamente la platina al marco de la puerta con los tornillos, como se aprecia en la **Figura 68**.
- Montar la cerradura a la platina por medio de los tornillos, como se aprecia en la **Figura 68**.
- Montar sobre la puerta la pieza polar (Ver **Figura 69**) haciéndola coincidir con la cerradura, utilizando los tornillos y gomas. Debe lograrse una oscilación adecuada para facilitar la alineación de las dos piezas.
- Ajustar siempre firmemente los tornillos.

**Nota**: No debe fijarse la pieza polar demasiado ajustada. Hay que dejar oscilar levemente sobre la goma para facilitar la perfecta alineación de la pieza polar y la cerradura.

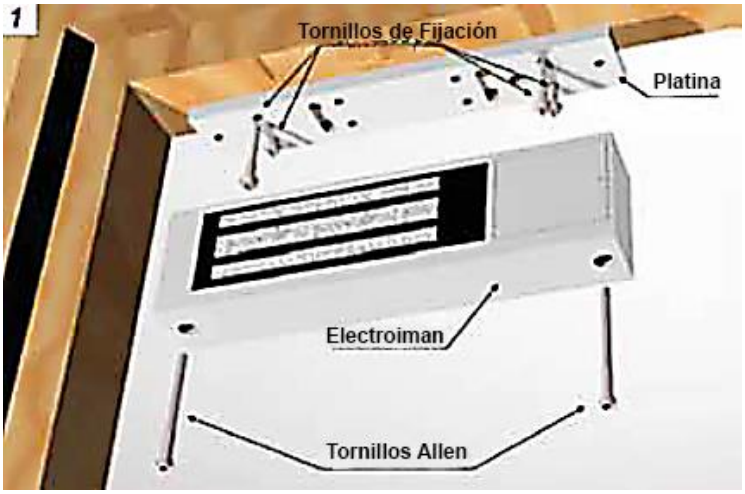



**Figura 68. Montaje de la platina**

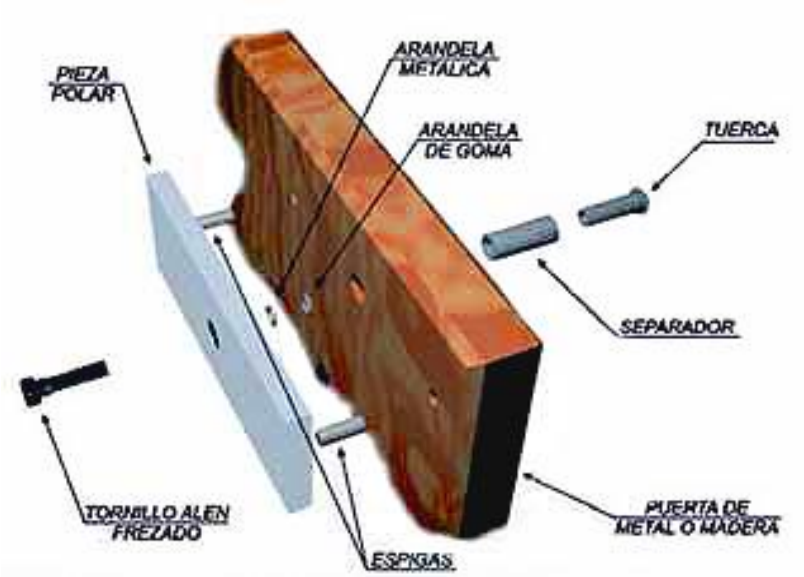



Figura 69. Instalación pieza polar (Puerta)

## F.3 Configuración

El sistema inicializa en modo de lectura (Ver **Figura 70**), esto significa que el sistema va a estar esperando la lectura de un tag.


Figura 70. Modo Lectura

Cuando el sistema detecta un tag, se encarga de tomar la decisión si permitir el paso o no de un usuario, para que el sistema permita el ingreso de un usuario es necesario hacer las configuraciones que se mostrara más adelante.

En caso de querer acceder al local y dispone de un tag autorizado por el sistema, solo acérquelo hacia el modulo lector, luego procede a ingresar al local en el periodo de 5 segundos.

Al **presionar** la tecla **Asterisco**, se accede a la **selección de Menú** como se muestra en la siguiente figura:


**Figura 71. Selección de Menús**

Para acceder al **Menú del Administrador** se presiona la **tecla 1** y para acceder al **Menú de usuario** se presiona la **tecla 2**.

### F.3.1 Menú Administrador

Al acceder al menú del administrador, nos pedirá que ingresemos la contraseña (Ver **Figura 72**).


**Figura 72. Ingresar contraseña para poder acceder al menú del administrador**

**Nota**: Cuando se accede por primera, la contraseña que viene por defecto en el sistema es: 123456

Una vez que se ingresa la contraseña de manera correcta, se presiona la **tecla A** para acceder al menú, en caso de querer borrar la contraseña presione la **tecla B**.

Al ingresar la contraseña correctamente se aprecia el siguiente menú (Ver **Figura 73**):

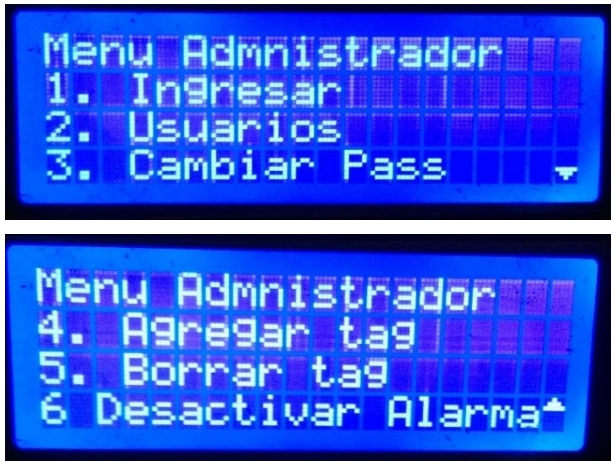



**Figura 73. Menú de Administrador**

**Nota**: Para **subir** de menú se presiona la **tecla C** y para bajar de menú se presiona la **tecla D**. En caso que se quiera salir presione la tecla **Asterisco**.

Dispone de 6 funcionalidades, las cuales se va a describir a continuación:

1. **Ingresar:** Está opción permite al usuario acceder al local, para eso es necesario que presione la **tecla 1**.
2. **Usuarios:** Está opción permite agregar, borrar y cambiar contraseña de los usuarios, así como agregar y borrar tag de los usuarios, para acceder a este menú presione la **tecla 2**.
3. **Cambiar pass:** Está opción permite cambiar la contraseña del administrador, para eso es necesario presionar la **tecla 3**.
4. **Agregar tag:** Está opción permite agregar un tag del administrador al sistema, para eso es necesario acceder presionando la **tecla 4**.

5. **Borrar tag:** Está opción permite borrar el tag del administrador, para es necesario presionar la **tecla 5**.
6. **Desactivar Alarma:** Está opción permite desactivar la alarma cuando se haya activado, para eso es necesario presionar la **tecla 6**.

### F.3.1.1 Configuración de Usuarios

Al acceder al menú **administrar usuarios** se aprecia las siguientes opciones (Ver **Figura 74**):


**Figura 74. Menú administrar usuarios**

Dispone de 5 funcionalidades, las cuales se va a describir a continuación:

1. **Nuevo Usuario:** Permite **agregar** un nuevo usuario al sistema, con el propósito de que pueda acceder al local a través del menú. Para eso es necesario presionar la **tecla 1**.
2. **Borrar Usuario:** Permite **borrar** un usuario del sistema. Para eso es necesario presionar la **tecla 2**.
3. **Cambiar pass:** Permite **cambiar la contraseña** de los usuarios en caso de que pierda el acceso. Para eso es necesario que presione la **tecla 3**.
4. **Agregar tag:** Permite **agregar un tag** a un usuario al sistema. Para eso es necesario que presione la **tecla 4**.

5. **Borrar tag:** Permite borrar un tag de un usuario al sistema. Par eso es necesario que presione la **tecla 5**.

#### F.3.1.1.1 Nuevo Usuario

Cuando se accede a la opción Nuevo Usuario se aprecia la siguiente pantalla (Ver **Figura 75**):

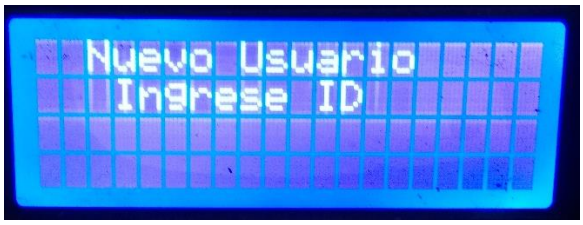



**Figura 75. Ingresar el ID de un nuevo usuario**

Se ingresa el ID del nuevo usuario, el cual debe ser menor de 5 dígitos, una vez ingresado el ID se presiona la **tecla A** (en caso de querer borrar presione la **tecla B**).

Seguido se procede a agregar la contraseña del usuario, como se muestra en la **Figura 76**:

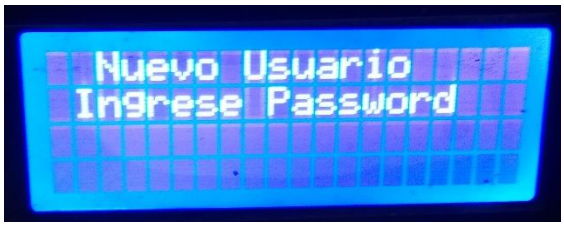



**Figura 76. Ingresar contraseña del nuevo usuario**

**Nota:** La contraseña debe ser mayor de 5 caracteres y menor de 10.

#### F.3.1.1.2 Borrar Usuario

Cuando se selecciona la opción de borrar usuario se aprecia la siguiente pantalla (Ver **Figura 77**):


**Figura 77. Buscar ID de un usuario a borrar del sistema**

**Usuario** es el número de usuario que se encuentra en el sistema y **ID** es la identificación del usuario.

Se busca el ID del usuario que se quiere eliminar presionando las **teclas C** (Para ir a la izquierda) y **D** (Para ir a la Derecha) y seguido se presiona la **tecla A** para proceder a borrar.

#### F.3.1.1.3 Cambiar Password

Al acceder a la opción cambiar password nos aparece la siguiente pantalla (Ver **Figura 78**):


**Figura 78. Búsqueda de un usuario al cual se quiere cambiar la contraseña**

Se busca el ID del usuario que se quiere cambiar la contraseña presionando las **teclas C** (Para ir a la izquierda) y **D** (Para ir a la Derecha) y seguido se presiona la **tecla A** para proceder a cambiar la contraseña.

Luego aparece la siguiente pantalla (Ver **Figura 79**):

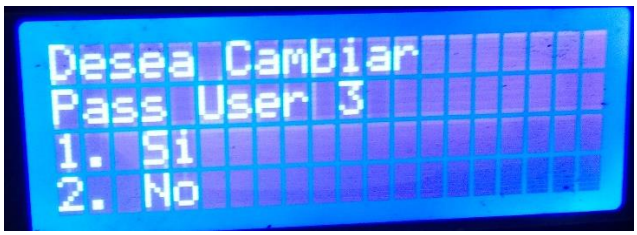



**Figura 79. Desea cambiar la contraseña**

Presione la **tecla 1** si quiere cambiar la contraseña, de lo contrario presione la **tecla 2**.

Seguido procede a cambiar la contraseña del usuario, como se muestra en la **Figura 80**:


**Figura 80. Cambio de contraseña de un usuario**

Una vez ingresado la contraseña presione la **tecla A** para guardar la contraseña, en caso de querer borrar la contraseña presione la **tecla B** y si desea salir del menú presione la tecla **asterisco**.

**Nota:** La contraseña debe ser mayor de 5 caracteres y menor de 10.

### F.3.1.1.4 Agregar Tag

Al acceder a la opción de agregar tag, nos pide que se ingrese el ID de un usuario existente en el sistema (Ver **Figura 81**).


**Figura 81. Ingresar ID de un usuario existente en el sistema para agregar un tag**

Luego de ingresar el ID se presiona la **tecla A** y nos pide que acerquemos el tag al lector RFID (Ver **Figura 82**).


**Figura 82. Guardar un tag en el sistema**

**Nota**: Para guardar un tag en el sistema se debe acercar la tarjeta al módulo lector en un tiempo aproximado de 10 segundos, en caso de no realizarlo en ese periodo se tiene que volver a realizar la operación.

### F.3.1.1.5 Borrar tag

Cuando se selecciona la opción borrar tag se aprecia la siguiente imagen (Ver **Figura 83**):


**Figura 83. Búsqueda de un usuario al que se le quiere borrar el tag**

**Usuario** es el número de usuario que se encuentra en el sistema, **ID** es la identificación del usuario y **Tag** es si un usuario dispone o no de un tag.

Se busca el ID del usuario el cual se quiere eliminar el tag presionando las **teclas C** (Para ir a la izquierda) y **D** (Para ir a la Derecha) y seguido se presiona la **tecla A** para proceder a borrar el tag.

### F.3.2 Menú Usuario

Cuando se selecciona el menú de usuario nos aparece la siguiente pantalla (Ver **Figura 84**):


**Figura 84. Ingresar el ID del usuario autorizado**

Se ingresa el ID del usuario registrado en el sistema y seguido se presiona la **tecla A**. Luego procede a ingresar la contraseña (Ver **Figura 85**)


Figura 85. Ingresar Password del usuario autorizado

Si la contraseña es correcta se muestra el siguiente menú (Ver **Figura 86**):

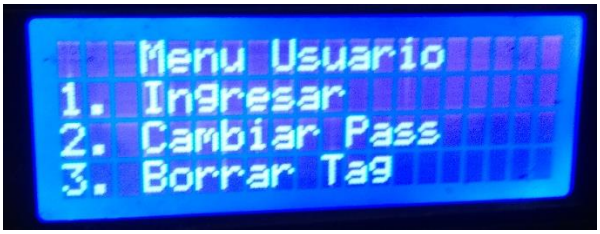



Figura 86. Menú del usuario

Dispone de 3 funcionalidades, las cuales se va a describir a continuación:

1. **Ingresar:** Permite al usuario acceder al local, para eso es necesario que presione la **tecla 1**.
2. **Cambiar pass:** Permite al usuario cambiar la contraseña, en caso que se desee cambiarla, para eso es necesario que presione la **tecla 2**.
3. **Borrar Tag:** Permite al usuario borrar el tag en caso que se haya extraviado, para eso es necesario que presione la **tecla 3**.

# Anexo G. Fabricación de la PCB

## G.1 Fabricación de la PCB

El proceso de fabricación de la PCB se realizó utilizando el método de la plancha, descrito a continuación:

Primero se imprimió el diseño de las pistas proporcionada por el software EAGLE en filmina, como se muestra en la **Figura 87**:

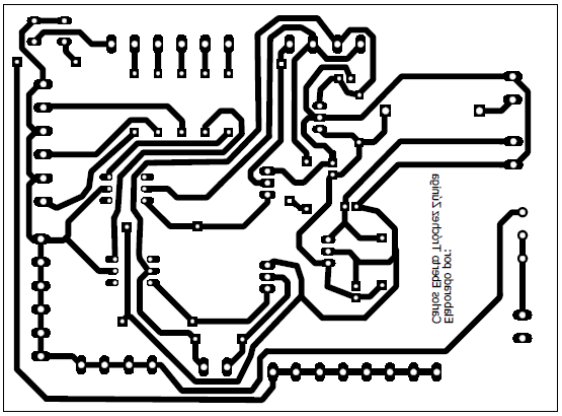


**Figura 87. Impresión en filmina del circuito impreso**

Seguido se coloca la filmina impresa sobre la placa de cobre virgen, se debe colocar cinta adhesiva antes plancharla, como se muestra en la **Figura 88**:

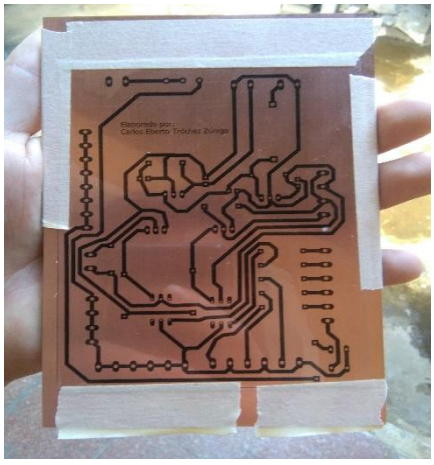


**Figura 88. Papel filmina impresa colocada con cinta adhesiva sobre la placa de cobre virgen**

Luego se procede a planchar suavemente la filmina impresa colocada junto con la placa de cobre, deslizando la plancha por toda la placa en un periodo de tiempo de 15 minutos. Para esto, debemos de colocar un papel encima de la placa, para que la filmina no se dañe con el calor, ni tampoco se adhiera a la plancha. Además, debemos de tener cuidado de no hacer mucha presión con la plancha o de planchar mucho tiempo la placa debido a que el cobre de la placa se puede despegar. Como se aprecia en la **Figura 89**:

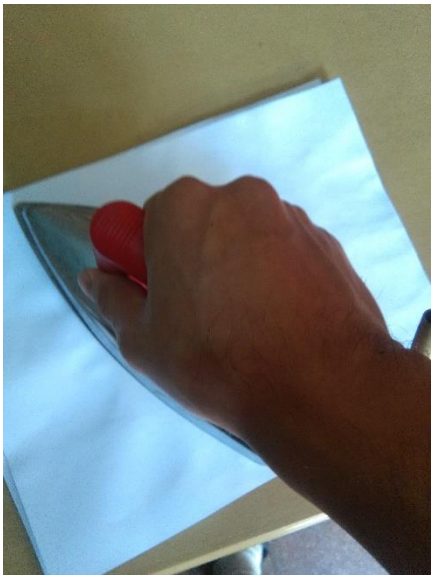


**Figura 89. Planchado de la filmina impresa junto con la placa de cobre**

Después de haberla planchado lo suficiente, se mete la placa en un recipiente con agua, en un periodo de tiempo de 15 minutos. Como se aprecia en la **Figura 90**:

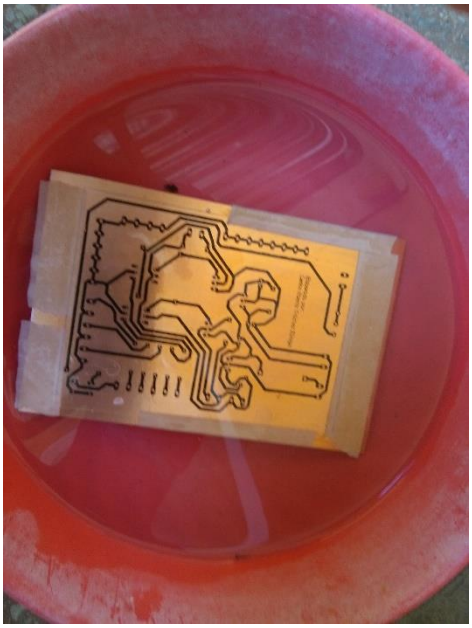


**Figura 90. Placa sumergida en agua después del planchado**

Luego, se quita la filmina suavemente de la placa de cobre y se obtiene el siguiente resultado (Ver **Figura 91**):

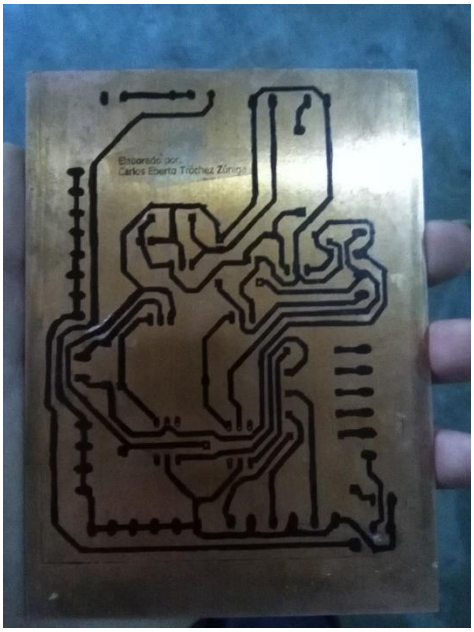


**Figura 91. Pistas impresas en la placa de cobre**

El siguiente paso es sumergir la placa de cobre en percloruro férrico, una solución acida que remueve el cobre donde no está protegido por la tinta de la impresora, en un periodo de tiempo aproximadamente de 45 minutos. El proceso se realizó por medio de baño maría, que consiste en poner dos recipientes, una con agua caliente y otra con el percloruro férrico junto con la placa, con el fin de aligerar proceso (Ver **Figura 92**).

**Figura 92. Tarjeta inmersa en Percloruro Férrico**

Luego, se saca la placa del percloruro férrico y se hace una inspección visual para ver si las pistas quedaron bien, seguido se procede con el multímetro hacer pruebas de continuidad con el propósito de encontrar algún problema en las pistas. (Ver **Figura 93**):

**Figura 93. Placa de cobre después de haber sido sumergido en percloruro férrico**

En esta etapa, se procede a perforar los pads y después montar cada componente en su respectivo lugar, para después soldarlos (Ver **Figura 94**).

**Figura 94. Ubicación de los componentes en la PCB**

Cada punto de soldadura debe garantizar la mejor conductividad entre el pin de un componente y la pista de cobre PCB. Una vez soldados los componentes se produce a realizar una inspección visual con el propósito de verificar el correcto ensamblaje de los componentes, según el diagrama eléctrico de la tarjeta. En la **Figura 95** se aprecia el PCB.

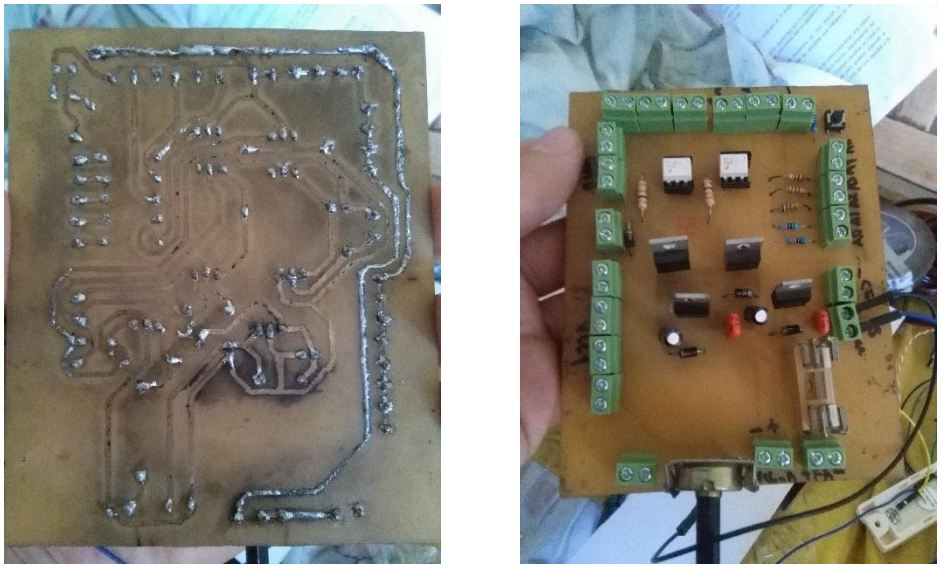


**Figura 95. Placa PCB con sus respectivos componentes soldados**

# Anexo H: Cotización

## H.1 Cotización

### H.1.1 Batería

ExpertPower EXP12120 12 Volt 12 Amp Rechargeable battery


**Figura 96. ExpertPower 12v 12Ah**

**Caracteriticas:**

Bateria: 12V 12Ah

Tipo de bacteria: Plomo y Acido

Precio (Ya enviado a Nicaragua):  $ 125.39 USD

### H.1.2 Modulo GSM Shield

Geeetech SIMCOM SIM900 Quad-band GSM GPRS Shield Development Board for Arduino/Iduino

**Figura 97. Módulo GSM Shield**

**Caracterisiticas:**

El GPRS Shield se basa en el módulo de SIM900 de SimCom y compatible con Arduino y sus clones.

El GPRS Shield le proporciona una manera de comunicarse usando la red del teléfono celular del g/m.

El Shield le permite lograr SMS, MMS, GPRS y audio a través de UART enviando a los comandos (GSM 07,07, 07,05 y SimCom mejorado en los comandos).

El Shield también tiene los 12 GPIOs, 2 PWMs y un ADC del módulo SIM900 (todos son lógica 2V8) presente a bordo.

Quad-Band 850/900/1800/1900 MHz-funcionaría en redes GSM en todos los países de todo el mundo.

Precio (Ya enviado a Nicaragua):  $ 61.87 USD

### H.1.3 Sensor PIR

ENKLOV Wireless PIR Motion Sensor Detector for W1 Home Alarm System Kit


**Figura 98. Sensor PIR**

**Caracteristicas:**

Modo de instalación: Montaje en pared mediante soporte

Tensión de funcionamiento: 4.5V

Transmisión Freq:433MHz

Distancia de transmisión: alrededor de 100m (Área abierta / sin interferencias)

Detección de la cobertura: 8M, campo de visión de 110 grados

Altura de instalación: alrededor de 2,2 m

Batería: batería alcalina de 3 * AAA (incluida)

Precio (Ya enviado a Nicaragua):  $ 33.91 USD

### H.1.4 Sensor de Humo

MQ135 MQ-135 Air Quality Sensor Hazardous Gas Detection Module for Arduino


**Figura 99. Sensor de Humo**

**Caracteristicas:**

El sensor de gas MQ-2 es sensible al GLP, i-butano, propano, metano, alcohol, hidrógeno y humo.

Podría ser utilizado en la fuga de gas que detecta los equipos en familia e industria.

Precio (Ya enviado a Nicaragua):  $ 24.21 USD

### H.1.5 Adaptador de microSD

Adafruit MicroSD card breakout board+ [ADA254]

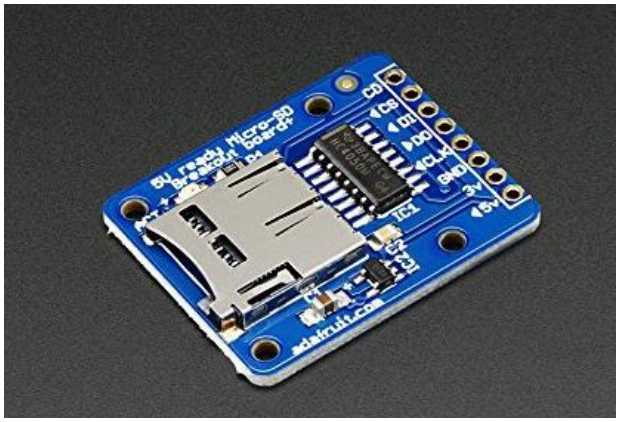



**Figura 100. Adaptador MicroSD**

**Caracteristicas:**

Compatible con Arduino

Costo (Ya enviado a Nicaragua):  $ 28.16 USD

# Anexo I: Código de programación

Autor: Carlos Eberto Tróchez Zúniga. Código en Arduino

```
//--------------------------------------------------------------------------------
#include <LiquidCrystal.h>
#include <Keypad.h>
#include <EEPROM.h>
#include <SPI.h>
#include <MFRC522.h>
//----------Configuración del Mrfc522----------------------
#define SS_PIN 53
#define RST_PIN 9

MFRC522 rfid(SS_PIN, RST_PIN);
MFRC522::MIFARE_Key key;

//----------------------Configuración del LCD---------------
LiquidCrystal lcd(A8, A9, A10, A11, A12, A13);

//-------------------Configuración del Teclado-------------
const byte Filas = 4;
const byte Cols = 4;
byte Pins_Filas[] = {37, 35, 33, 31};
byte Pins_Cols[] =  {29, 27, 25, 23};
char Teclas [ Filas ][ Cols ] =
{
{'1', '2', '3', 'A'},
{'4', '5', '6', 'B'},
{'7', '8', '9', 'C'},
{'*', '0', '#', 'D'}
};
```

```
//------------Tipos de datos-------------------------------------------
char Ingresar_ID[6];
char Ingresar_Contrasena[10];
char Ingresar_Contrasena_Adm[10];
char Ingresar_Contrasena_Confirmacion[10];
char Contrasena1[] = {"123456"};
static int cnt = 0;
byte TagAdm[4];
static int cnt1 = 0;
int i = 0;
int short Puerta = A0;
int short Peligro = A1;
byte Acceso_Negado = A2;
byte Modo_Lectura_LED = A3;
byte Abierto_LED = A4;
int user;
int ID;
char kp;
int Estado;
char Menu;
int Existe;
long Dato;
char Tecla;
boolean estado_reset = HIGH;
int reset = 5;
//---------------------------------------
#define DESPLAZAMIENTO_CONTRASENA 1000
#define TAG_DESPLAZAMIENTO_ADM  820
#define CREDENCIALDESPLAZAMIENTO 20
```

```
//-------Credenciales de los usuarios
struct USUARIOS_CREDENCIALES
{
  char id[6];
  char pwd[10];
  byte rfid[4];
} userioscredenciales;

USUARIOS_CREDENCIALES credenciales[26];
//-----------------Configuraciones---------------------
void setup() {
  lcd.begin(20, 4);
  lcd.display();
  Serial.begin(9600);
  SPI.begin();
  rfid.PCD_Init();
  pinMode(Puerta, OUTPUT);
  pinMode(Peligro, OUTPUT);
  pinMode(Acceso_Negado, OUTPUT);
  pinMode(Modo_Lectura_LED, OUTPUT);
  pinMode(Abierto_LED, OUTPUT);
  pinMode(reset, INPUT);
  digitalWrite(Puerta, HIGH);
  lcd.createChar(1, Flecha_Arriba);
  lcd.createChar(2, Flecha_Abajo);
  lcd.createChar(3, Flecha_Arriba_Abajo);
  lcd.createChar(4, Flecha_Derecha);
  lcd.createChar(5, Flecha_Izquierda);
}
```

```
void loop() {
  Estado = EEPROM.read(DESPLAZAMIENTO_CONTRASENA);
//---Obtiene la contraseña del Adm de la EEPROM
  for (i = 0; i < sizeof(Ingresar_Contrasena_Adm) / sizeof(Ingresar_Contrasena_Adm[0]); i++) {
    EEPROM.get(i * sizeof(Ingresar_Contrasena_Adm[0]), Ingresar_Contrasena_Adm[i]);
  }
//--Obtiene el tag del Adm de la EEPROM
  for (int a = 0; a < sizeof(TagAdm); a++) {
    EEPROM.get(a * sizeof(TagAdm) + TAG_DESPLAZAMIENTO_ADM, TagAdm[a]);
  }
//--Obtiene las credenciales de los usuarios
  for (i = 0; i < sizeof(credenciales ) / sizeof(USUARIOS_CREDENCIALES); i++) {
    EEPROM.get(i * sizeof(USUARIOS_CREDENCIALES) + CREDENCIALDESPLAZAMIENTO, credenciales[i]);
  }
//--Lee el estado de reinicio de contraseña
estado_reset = digitalRead(reset);
//-Borra contraseña
  if (estado_reset == LOW) {
    EEPROM.write(DESPLAZAMIENTO_CONTRASENA, 0);
    for (i = 0; i < sizeof(Ingresar_Contrasena); i++) {
      EEPROM.update(i, 0);
    }
  }
  inicio();   return;
}
```

```
char Teclado () //Función del teclado
{
  do {
    char Tecla = kpd.getKey();
    if (Tecla != NO_KEY)
    {
      return Tecla;
    }
  } while (1);
}

void inicio() {
  digitalWrite(Modo_Lectura_LED, HIGH);
  for (byte i = 0; i < 6; i++) { //Establece el canal de encriptación
    key.keyByte[i] = 0xFF;
  }

  leer_usuario(); //función del lector rfid para leer etiquetas
  lcd.setCursor(2, 1); lcd.print("Modo Lectura");
  lcd.setCursor(3, 3); lcd.print("Presione *");
  Tecla = kpd.getKey();

  if (Tecla == '*') { //Accede al menu
    digitalWrite(Modo_Lectura_LED, LOW);
    Usuario_Admnistrador();
  }
  return;
}
```

```
bool leer_usuario() { //Lee los datos de la etiqueta
  if ( ! rfid.PICC_IsNewCardPresent())
    return false;
  if ( ! rfid.PICC_ReadCardSerial())
    return false;


  for (int user = 0; user < sizeof(credenciales) /
sizeof(USUARIOS_CREDENCIALES); user ++) {


    if ((rfid.uid.uidByte[0] == credenciales[user].rfid[0]) &&
(rfid.uid.uidByte[1] == credenciales[user].rfid[1]) &&
(rfid.uid.uidByte[2] == credenciales[user].rfid[2]) &&
(rfid.uid.uidByte[3] == credenciales[user].rfid[3]) ||
((rfid.uid.uidByte[0] == TagAdm[0]) && (rfid.uid.uidByte[1] ==
TagAdm[1]) && (rfid.uid.uidByte[2] == TagAdm[2]) &&
(rfid.uid.uidByte[3] == TagAdm[3]))) { //Verifica si existe en el
sistema
      if ((rfid.uid.uidByte[0] == TagAdm[0]) && (rfid.uid.uidByte[1] ==
TagAdm[1]) && (rfid.uid.uidByte[2] == TagAdm[2]) &&
(rfid.uid.uidByte[3] == TagAdm[3])) {
        Serial.println("Admnistrador");
      }


      else {
        Serial.println(credenciales[user].id);
      }
      lcd.clear ();
      lcd.setCursor(0, 0); lcd.print("Abrir puerta"); //Permite el acceso
      digitalWrite(Puerta, LOW);
      digitalWrite(Modo_Lectura_LED, LOW);
      digitalWrite(Abierto_LED, HIGH);
```

```
delay(8000);
      digitalWrite(Puerta, HIGH);
      digitalWrite(Modo_Lectura_LED, HIGH);
      digitalWrite(Abierto_LED, LOW);
      lcd.clear ();
      break;
    }
    else { //De lo contrario, niega el acceso
      if (user == (sizeof(credenciales) / sizeof(USUARIOS_CREDENCIALES) - 1)) {
        lcd.clear ();
        lcd.setCursor(0, 0); lcd.print("Tag inexistente");
        digitalWrite(Acceso_Negado, HIGH);
        digitalWrite(Modo_Lectura_LED, LOW);
        delay(3000);
        digitalWrite(Modo_Lectura_LED, HIGH);
        digitalWrite(Acceso_Negado, LOW);
        lcd.clear ();
      }
    }
  }
  rfid.PICC_HaltA(); //desactiva el lector rfid
  rfid.PCD_StopCrypto1(); //finaliza la encriptación
  return true;
}
```

```
void Usuario_Admnistrador() {
  lcd.clear();
  lcd.setCursor(0, 1); lcd.print("1. Administrador");
  lcd.setCursor(0, 2); lcd.print("2. Usuario");
  kp = Teclado ();
  switch (kp) {
    case '1':
      Admnistrador ();
      lcd.clear();
      return;
      break;

    case '2':
      Usuarios();
      lcd.clear();
      return;
      break;

    case 'B':
      lcd.clear();
      inicio();
      return;
      break;

    case '*':
      lcd.clear();
      inicio();
      return;
      break;
```

```
default:
      Usuario_Admnistrador();
      return;
      break;
  }
  return;
}
void Admnistrador () { //funcion agregar contraseña admn
  lcd.clear();
  lcd.setCursor(1, 0); lcd.print("Administrador");
  lcd.setCursor(0, 1); lcd.print("Ingresar Password");
  for ( i = 0; i < sizeof(Ingresar_Contrasena); i++) {
    kp = Teclado ();
    Ingresar_Contrasena[i] = kp;

    if (i < 9) {
      lcd.setCursor(3 + i, 3); lcd.print('*');
    }

    if (Ingresar_Contrasena[i] == 'A') { //si preciona A = Enter
      Ingresar_Contrasena[i] = '\0';
      break;
    }
     else if (Ingresar_Contrasena[i] == 'B') { //si preciona B borra pantalla
      lcd.clear();
      lcd.setCursor(1, 0); lcd.print("Administrador");
      lcd.setCursor(0, 1); lcd.print("Ingresar Password");
      i = -1;
    }
```

```
else if (Ingresar_Contrasena[i] == '*') { //Si preciona *, Exit
    lcd.clear();
    inicio();
    exit;
    return;
  }
  if ((i == (sizeof(Ingresar_Contrasena) - 1))) {
    i--;
  }
 }


  if ((Ingresar_Contrasena[0] == '\0') || (Ingresar_Contrasena[1] == '\0') ||
(Ingresar_Contrasena[2] == '\0') || (Ingresar_Contrasena[3] == '\0') ||
(Ingresar_Contrasena[4] == '\0') || (Ingresar_Contrasena[5] == '\0') ||
(Ingresar_Contrasena[6] == '\0') || (Ingresar_Contrasena[7] == '\0') ||
(Ingresar_Contrasena[8] == '\0') || (Ingresar_Contrasena[9] == '\0')) {


   if ((cnt < 3) && (strcmp (Contrasena1, Ingresar_Contrasena) == 0) &&
(Estado == 0)) {
    Menu_Administrador(); //Si ingreso la contraseña del sist. entra al menu
    return;
   }
   else if ((cnt < 3) && (strcmp (Ingresar_Contrasena_Adm,
Ingresar_Contrasena) == 0) && (Estado == 1)) {
    Menu_Administrador(); //Ingreso la nueva contraseña, entra al menu
    return;
   }
   else { //Si todas las contraseñas son incorrectas
    cnt++;
    lcd.clear();
    lcd.setCursor(2, 1); lcd.print("INCORRECTO");
```

```
delay(1000);
      lcd.clear();
    }
    if (cnt == 3) { //Si el Contador llega a 3 se active la alarma
      cnt = 0;
      digitalWrite(Peligro, HIGH);
      Serial.println("ALARMA");
    }
  }
  return;
}

void Menu_Administrador() { //Menu administrador
  cnt = 0;
  Menu = 0;
  lcd.clear();
  lcd.setCursor(0, 0); lcd.print("Menu Admnistrador");
  lcd.setCursor(0, 1); lcd.print("1. Ingresar");
  lcd.setCursor(0, 2); lcd.print("2. Usuarios");
  lcd.setCursor(0, 3); lcd.print("3. Cambiar Pass");
  lcd.setCursor(19, 3); lcd.write(2);

  kp = Teclado ();
switch (kp) {
    case '1':
      Ingresar();
      Serial.println("Admnistrador");
      lcd.clear();
      return;
```

```
case '2':
   Administracion_de_Usuarios();
   lcd.clear();
   return;
   break;

 case '3':
   Cambiar_Contrasena_Admnistrador();
   Menu_Administrador();
   lcd.clear();
   return;
   break;

 case 'D':
   Menu_Admnistrador_Segunda_Parte();
   lcd.clear();
   return;
   break;

 case 'B':
   Usuario_Admnistrador();
   lcd.clear();
   return;
   break;

 case '*':
   lcd.clear();
   inicio();
```

```
      return;
      break;
}
  return;
}

void Ingresar() {
  lcd.clear();
  lcd.setCursor(3, 1); lcd.print("Abierto");
  digitalWrite(Puerta, LOW);
  digitalWrite(Abierto_LED, HIGH);
  delay(8000);
  digitalWrite(Puerta, HIGH);
  digitalWrite(Abierto_LED, LOW);
  return;
}

void Administracion_de_Usuarios() {
  lcd.clear();
  lcd.setCursor(0, 0); lcd.print("Administrar Usuarios");
  lcd.setCursor(0, 1); lcd.print("1. Nuevo Usuario");
  lcd.setCursor(0, 2); lcd.print("2. Borrar Usuario");
  lcd.setCursor(0, 3); lcd.print("3. Cambiar Pass");
  lcd.setCursor(19, 3); lcd.write(2);
  kp = Teclado ();
switch (kp) {
    case '1':
      Nuevo_Usuario();
```

```
lcd.clear();
      return;
      break;

case '2':
      Borrar_Usuario();
      lcd.clear();
      return;
      break;

    case '3':
      Cambiar_Contrasena_Usuario();
      lcd.clear();
      return;
      break;

    case 'D':
      lcd.clear();
      lcd.setCursor(0, 0); lcd.print("Administrar Usuarios");
      lcd.setCursor(0, 1); lcd.print("4. Agregar Tag");
      lcd.setCursor(0, 2); lcd.print("5. Borrar Tag");
      lcd.setCursor(19, 3); lcd.write(1);
      kp = Teclado ();
      switch (kp) {
        case '4':
          guardar_tag();
          digitalWrite(Modo_Lectura_LED, LOW); //Apaga el led modo lectura
          Administracion_de_Usuarios();
```

```
    return;
    break;

    case '5':
    Borrar_Tag();
    return;
    break;

  case 'C':
    Administracion_de_Usuarios();
    return;
    break;

  case 'B':
    lcd.clear();
    Menu_Administrador();
    return;
    break;

  case '*':
    lcd.clear();
    inicio();
    return;
    break;

  default:
    Administracion_de_Usuarios();
    return;
```

```
        break;
      }
break;


case 'B':
      lcd.clear();
      Menu_Administrador();
      return;
      break;


    case '#':
      for (i = 0; i < sizeof(credenciales) / sizeof(USUARIOS_CREDENCIALES);
i++) {
        EEPROM.put(i * sizeof(USUARIOS_CREDENCIALES) +
CREDENCIALDESPLAZAMIENTO, 0);
      }
      break;


    case '*':
      lcd.clear();
      inicio();
      return;
      exit;
      break;


    default:
      Administracion_de_Usuarios();
      return;
      break;
  }
```

```
void Nuevo_Usuario() { //Funcion agregar nuevo usuario
  lcd.clear();
  lcd.setCursor(2, 0); lcd.print("Nuevo Usuario");
  lcd.setCursor(3, 1); lcd.print("Ingrese ID");

  for (i = 0; i < sizeof(Ingresar_ID); i++) {
    kp = Teclado ();
    Ingresar_ID[i] = kp;
    if (i < 5) {
      lcd.setCursor(5 + i, 3); lcd.print(Ingresar_ID[i]);
    }
    if (Ingresar_ID[i] == 'A') {
      Ingresar_ID[i] = '\0';
      if (Ingresar_ID[0] == '\0') {
        lcd.clear();
        lcd.setCursor(2, 0); lcd.print("Nuevo Usuario");
        lcd.setCursor(3, 1); lcd.print("Ingrese ID");
        i = -1;
      }
      else {
        break;
      }
    }
    else if (Ingresar_ID[i] == 'B') {
      lcd.clear();//Limpia Pantalla
      lcd.setCursor(2, 0); lcd.print("Nuevo Usuario");
      lcd.setCursor(3, 1); lcd.print("Ingrese ID");
      i = -1;
    }
```

```
    else if (Ingresar_ID[i] == '*') {
      lcd.clear();
      inicio();
      exit;
      return;
    }
    if (i == (sizeof(Ingresar_ID) - 1)) {
      i--;
    }
  }
if ((Ingresar_ID[1] == '\0') || (Ingresar_ID[2] == '\0') || (Ingresar_ID[3] == '\0') ||
(Ingresar_ID[4] == '\0') || (Ingresar_ID[5] == '\0')) {

    for (user = 0; user < sizeof(credenciales) /
sizeof(USUARIOS_CREDENCIALES); user ++) {

      if (strcmp (credenciales[user].id, Ingresar_ID) == 0) {
        Existe = 1;
        break;
      }

      else {

        if (user == (sizeof(credenciales) / sizeof(USUARIOS_CREDENCIALES) -
1)) {
          Existe = 2;
        }
      }
    }
```

```
  switch (Existe) {
    case 1:
      Existe = 0;
      lcd.clear ();
      lcd.setCursor(3, 0); lcd.print("USUARIO");
      lcd.setCursor(2, 1); lcd.print("EXISTENTE");
      delay(1000);
      lcd.clear();
      Administracion_de_Usuarios();
      return;
      break;


 case 2:
      Existe = 0;
      for (user = 0; user < sizeof(credenciales) /
sizeof(USUARIOS_CREDENCIALES);) {
        if (credenciales[user].id[6] == '\0') {
          strcpy(credenciales [user].id, Ingresar_ID);
          lcd.clear ();
          lcd.setCursor(3, 1); lcd.print("Guardando");
          delay(1000);
          lcd.clear();
          lcd.setCursor(2, 0); lcd.print("Nuevo Usuario");
          lcd.setCursor(1, 1); lcd.print("Ingrese Password");

        for ( i = 0; i < sizeof(Ingresar_Contrasena); i++) {
            kp = Teclado ();
            Ingresar_Contrasena[i] = kp;
```

```
if (i < 9) {
  lcd.setCursor(3 + i, 3); lcd.print('*');
 }
 if (Ingresar_Contrasena[i] == 'A') {
  Ingresar_Contrasena[i] = '\0';

  if (strlen(Ingresar_Contrasena) < 4) {
   lcd.clear ();
   lcd.setCursor(3, 1); lcd.print("Contrasena");
   lcd.setCursor(5, 2); lcd.print("Corta");
   delay(1000);

   lcd.clear();
   lcd.setCursor(2, 0); lcd.print("Nuevo Usuario");
   lcd.setCursor(1, 1); lcd.print("Ingrese Password");
   i = -1;
  }

  else {
   break;
  }
}
else if (Ingresar_Contrasena[i] == 'B') {
  lcd.clear();
  lcd.setCursor(2, 0); lcd.print("Nuevo Usuario");
  lcd.setCursor(1, 1); lcd.print("Ingrese Password");
  i = -1;
}
```

```
   else if (Ingresar_Contrasena[i] == '*') {
              lcd.clear();
              inicio();
              exit;
              return;
             }

             if (i == (sizeof(Ingresar_Contrasena) - 1)) {
              i--;
             }
            }

        if ((Ingresar_Contrasena[5] == '\0') || (Ingresar_Contrasena[6] == '\0') ||
(Ingresar_Contrasena[7] == '\0') || (Ingresar_Contrasena[8] == '\0') ||
(Ingresar_Contrasena[9] == '\0')) {

             lcd.clear();
             lcd.setCursor(1, 0); lcd.print("Confirme Password");
             lcd.setCursor(1, 1); lcd.print("Ingrese Password");

             for (int i = 0; i < sizeof(Ingresar_Contrasena_Confirmacion); i++) {
              kp = Teclado ();
              Ingresar_Contrasena_Confirmacion[i] = kp;

              if (i < 9) {
               lcd.setCursor(3 + i, 3); lcd.print('*');
              }
             if (Ingresar_Contrasena_Confirmacion[i] == 'A') {
               Ingresar_Contrasena_Confirmacion[i] = '\0';
```

```
  if (strlen(Ingresar_Contrasena_Confirmacion) < 4) {
   lcd.clear ();
   lcd.setCursor(3, 1); lcd.print("Contrasena");
   lcd.setCursor(5, 2); lcd.print("Corta");
   delay(1000);

   lcd.clear();
   lcd.setCursor(1, 0); lcd.print("Confirme Password");
   lcd.setCursor(1, 1); lcd.print("Ingrese Password");
   i = -1;
  }

  else {
   break;
  }
}
else if (Ingresar_Contrasena_Confirmacion[i] == 'B') {
  lcd.clear();
  lcd.setCursor(1, 0); lcd.print("Confirme Password");
  lcd.setCursor(1, 1); lcd.print("Ingrese Password");
  i = -1;
}

else if (Ingresar_Contrasena_Confirmacion[i] == '*') {
  lcd.clear();
  inicio();
  return;
}
```

```
if (i == (sizeof(Ingresar_Contrasena_Confirmacion) - 1)) {
                i--;
              }
            }

          if ((Ingresar_Contrasena_Confirmacion[5] == '\0') ||
(Ingresar_Contrasena_Confirmacion[6] == '\0') ||
(Ingresar_Contrasena_Confirmacion[7] == '\0') ||
(Ingresar_Contrasena_Confirmacion[8] == '\0') ||
(Ingresar_Contrasena_Confirmacion[9] == '\0')) {

            if (strcmp (Ingresar_Contrasena,
Ingresar_Contrasena_Confirmacion) == 0) {

              strcpy(credenciales [user].pwd, Ingresar_Contrasena);
              lcd.clear ();
              lcd.setCursor(3, 1); lcd.print("Guardando");
              delay(1000);

            for (i = 0; i < sizeof(credenciales) /
sizeof(USUARIOS_CREDENCIALES); i++) {
                EEPROM.put(i * sizeof(USUARIOS_CREDENCIALES) +
CREDENCIALDESPLAZAMIENTO, credenciales[i]);
              }

              lcd.clear();
              Administracion_de_Usuarios();
              return;
            }
```

```
                else {
                lcd.clear ();
                lcd.setCursor(2, 1); lcd.print("INCORRECTO");
                delay(1000);
                lcd.clear();
                Administracion_de_Usuarios();
                exit;
                return;
               }
             }
            }
            break;
          }


          else {
            user++;
            if (user == (sizeof(credenciales) /
sizeof(USUARIOS_CREDENCIALES))) {
              lcd.clear ();
              lcd.setCursor(3, 0); lcd.print("Memoria");
              lcd.setCursor(3, 1); lcd.print("Llena");
              delay(1000);
              lcd.clear();
              Administracion_de_Usuarios();
              return;
              break;
            }
          }
```

```
}
  return;
}
void Borrar_Usuario() {
  for (i = 0; i < sizeof(credenciales ) / sizeof(USUARIOS_CREDENCIALES);) {
    EEPROM.get(i * sizeof(USUARIOS_CREDENCIALES) +
CREDENCIALDESPLAZAMIENTO, credenciales[i]);

    if (credenciales [i].id[6] != '\0') {
      lcd.clear();
      lcd.setCursor(1, 0); lcd.print("Borrar Usuarios");
      lcd.setCursor(3, 1); lcd.print("Usuario"); lcd.setCursor(12, 1); lcd.print(i + 1);
      lcd.setCursor(4, 2); lcd.print("ID: "); lcd.setCursor(9, 2);
lcd.print(credenciales[i].id);
      lcd.setCursor(15, 1); lcd.write(4);
      lcd.setCursor(1, 1); lcd.write(5);
      kp = Teclado ();
      switch (kp) {

        case 'C':
         i++;
         if (i == (sizeof(credenciales ) / sizeof(USUARIOS_CREDENCIALES))) {
           i = 0;
         }
         break;

       case 'D':
         i--;
         if (i < 0) {
```

```
 i = (sizeof(credenciales ) / sizeof(USUARIOS_CREDENCIALES)) - 1;
  }
  break;

case 'A':
  lcd.clear();
  Desea_Borrar_Usuario();
  return;
  exit;
  break;

  case 'B':
  lcd.clear();
  Administracion_de_Usuarios();
  return;
  exit;
  break;

case '*':
  lcd.clear();
  inicio();
  return;
  break;

 default:
  Borrar_Usuario();
  return;
  exit;
  break;
```

```
      }
    }

    else {
      lcd.clear();
      lcd.setCursor(1, 0); lcd.print("Borrar Usuarios");
      lcd.setCursor(3, 1); lcd.print("Usuario"); lcd.setCursor(12, 1); lcd.print(i +
1);
      lcd.setCursor(4, 2); lcd.print("No data");
      lcd.setCursor(15, 1); lcd.write(4);
      lcd.setCursor(1, 1); lcd.write(5);
      kp = Teclado ();
     switch (kp) {
        case 'C':
          i++;

          if (i == (sizeof(credenciales ) / sizeof(USUARIOS_CREDENCIALES))) {
            i = 0;
          }
          break;

        case 'D':
          i--;
          if (i < 0) {
            i = (sizeof(credenciales ) / sizeof(USUARIOS_CREDENCIALES)) - 1;
          }
          break;
```

```
case 'B':
          lcd.clear();
          Administracion_de_Usuarios();
          return;
          exit;
          break;

        case '*':
          lcd.clear();
          inicio();
          return;
          exit;
          break;

        default:
          Borrar_Usuario();
          return;
          exit;
          break;
      }
    }
  }
  return;
}
```

```
void Desea_Borrar_Usuario() {
  lcd.clear();
  lcd.setCursor(0, 0); lcd.print("Desea borrar al");
  lcd.setCursor(0, 1); lcd.print("Usuario"); lcd.setCursor(8, 1);
lcd.print(credenciales[i].id);
  lcd.setCursor(0, 2); lcd.print("1. Si");
  lcd.setCursor(0, 3); lcd.print("2. No");


kp = Teclado ();
  switch (kp) {
    case '1':
      lcd.clear();
      lcd.setCursor(3, 1); lcd.print("Borrado");
      strcpy(credenciales [i].id, "");
      strcpy(credenciales [i].pwd, "");
      for (int j = 0; j < sizeof(rfid.uid.size); j++) {
        credenciales [i].rfid[j] = 0;
      }


      EEPROM.put(i * sizeof(USUARIOS_CREDENCIALES) +
CREDENCIALDESPLAZAMIENTO, credenciales[i]);
      delay(1000);
      lcd.clear();
      Administracion_de_Usuarios();
      return;
      break;
```

```
      case '2':
        lcd.clear();
        Borrar_Usuario();
        return;
        break;

      case 'B':
        lcd.clear();
        Borrar_Usuario();
        return;
        break;

      case '*':
        lcd.clear();
        inicio();
        return;
        break;

      default:
        Desea_Borrar_Usuario();
        return;
        break;
    }
   return;
  }
```

```
void Cambiar_Contrasena_Usuario() {

  for (user = 0; user < sizeof(credenciales ) /
sizeof(USUARIOS_CREDENCIALES);) {

    EEPROM.get(user * sizeof(USUARIOS_CREDENCIALES) +
CREDENCIALDESPLAZAMIENTO, credenciales[user]);


if (credenciales [user].id[6] != '\0') {

      lcd.clear();

      lcd.setCursor(1, 0); lcd.print("Cambiar Password");

      lcd.setCursor(3, 1); lcd.print("Usuario"); lcd.setCursor(12, 1);
lcd.print(user + 1);

      lcd.setCursor(4, 2); lcd.print("ID: "); lcd.setCursor(9, 2);
lcd.print(credenciales[user].id);

      lcd.setCursor(15, 1); lcd.write(4);

      lcd.setCursor(1, 1); lcd.write(5);

      kp = Teclado ();

      switch (kp) {

        case 'C':

          user++;

          if (user == (sizeof(credenciales ) /
sizeof(USUARIOS_CREDENCIALES))) {

            user = 0;

          }

          break;


case 'D':

          user--;

          if (user < 0) {

            user = (sizeof(credenciales ) / sizeof(USUARIOS_CREDENCIALES))
- 1;

          }
```

```
      break;

     case 'A':
       lcd.clear();
       Desea_Cambiar_la_contrasena();
       return;
       break;

     case 'B':
       lcd.clear();
       Administracion_de_Usuarios();
       return;
       break;

     case '*':
       lcd.clear();
       inicio();
       return;
       break;

     default:
       Cambiar_Contrasena_Usuario();
       return;
       break;
   }
}
```

```
else {
      lcd.clear();
      lcd.setCursor(1, 0); lcd.print("Cambiar Password");
      lcd.setCursor(3, 1); lcd.print("Usuario"); lcd.setCursor(12, 1); lcd.print(user
+ 1);
      lcd.setCursor(4, 2); lcd.print("No data");
      lcd.setCursor(15, 1); lcd.write(4);
      lcd.setCursor(1, 1); lcd.write(5);
      kp = Teclado ();

      switch (kp) {
        case 'C':
          user++;
          if (user == (sizeof(credenciales ) /
sizeof(USUARIOS_CREDENCIALES))) {
            user = 0;
          }
          break;

    case 'D':
          user--;
          if (user < 0) {
            user = (sizeof(credenciales ) / sizeof(USUARIOS_CREDENCIALES)) -
1;
          }
          break;

        case 'B':
          lcd.clear();
          Administracion_de_Usuarios();
```

```
          return;
          break;

          case '*':
          lcd.clear();
          inicio();
          return;
          break;

        default:
          Cambiar_Contrasena_Usuario();
          return;
          break;
      }
    }
  }
  return;
}

 void Desea_Cambiar_la_contrasena() {
  lcd.clear();
  lcd.setCursor(0, 0); lcd.print("Desea Cambiar");
  lcd.setCursor(0, 1); lcd.print("Pass User"); lcd.setCursor(10, 1);
lcd.print(credenciales[user].id);
  lcd.setCursor(0, 2); lcd.print("1. Si");
  lcd.setCursor(0, 3); lcd.print("2. No");
  kp = Teclado ();
```

```
switch (kp) {
  case '1':
    lcd.clear();
    Opcion_Cambiar_Contrasena_Usuario();
    lcd.clear();
    Administracion_de_Usuarios();
    return;
    break;

  case '2':
    lcd.clear();
    Cambiar_Contrasena_Usuario();
    return;
    break;

  case 'B':
    lcd.clear();
    Cambiar_Contrasena_Usuario();
    return;
    break;

  case '*':
    lcd.clear();
    inicio();
    return;
    break;
```

```
void Opcion_Cambiar_Contrasena_Usuario() {
  lcd.clear();
  lcd.setCursor(0, 0); lcd.print("Nueva Contrasena");
  lcd.setCursor(1, 1); lcd.print("Ingrese Password");

  for ( i = 0; i < sizeof(Ingresar_Contrasena); i++) {
    kp = Teclado ();
    Ingresar_Contrasena[i] = kp;
    if (i < 9) {
      lcd.setCursor(3 + i, 3); lcd.print('*');
    }

    if (Ingresar_Contrasena[i] == 'A') {
      Ingresar_Contrasena[i] = '\0';

      if (strlen(Ingresar_Contrasena) < 4) {
        lcd.clear ();
        lcd.setCursor(3, 1); lcd.print("Contrasena");
        lcd.setCursor(5, 2); lcd.print("Corta");
        delay(1000);
        lcd.clear();
        lcd.setCursor(0, 0); lcd.print("Nueva Contrasena");
        lcd.setCursor(1, 1); lcd.print("Ingrese Password");
        i = -1;
      }
      else {
        break;
      }
```

```
}
    else if (Ingresar_Contrasena[i] == 'B') {
      lcd.clear();
      lcd.setCursor(0, 0); lcd.print("Nueva Contrasena");
      lcd.setCursor(1, 1); lcd.print("Ingrese Password");
      i = -1;
    }

    else if (Ingresar_Contrasena[i] == '*') {
      lcd.clear();
      return;
    }

if (i == (sizeof(Ingresar_Contrasena) - 1)) {
      i--;
    }
  }

  if ((Ingresar_Contrasena[5] == '\0') || (Ingresar_Contrasena[6] == '\0') ||
(Ingresar_Contrasena[7] == '\0') || (Ingresar_Contrasena[8] == '\0') ||
(Ingresar_Contrasena[9] == '\0')) {

    lcd.clear();
    lcd.setCursor(1, 0); lcd.print("Confirme Password");
    lcd.setCursor(1, 1); lcd.print("Ingrese Password");
for (int i = 0; i < sizeof(Ingresar_Contrasena_Confirmacion); i++) {
      kp = Teclado ();
      Ingresar_Contrasena_Confirmacion[i] = kp;
```

```
if (i < 9) {
  lcd.setCursor(3 + i, 3); lcd.print('*');
 }
 if (Ingresar_Contrasena_Confirmacion[i] == 'A') {
  Ingresar_Contrasena_Confirmacion[i] = '\0';

  if (strlen(Ingresar_Contrasena_Confirmacion) < 4) {
   lcd.clear ();
   lcd.setCursor(3, 1); lcd.print("Contrasena");
   lcd.setCursor(5, 2); lcd.print("Corta");
   delay(1000);
   lcd.clear();
   lcd.setCursor(1, 0); lcd.print("Confirme Password");
   lcd.setCursor(1, 1); lcd.print("Ingrese Password");
   i = -1;
  }

  else {
   break;
  }
 }

 else if (Ingresar_Contrasena_Confirmacion[i] == 'B') {
  lcd.clear();
  lcd.setCursor(1, 0); lcd.print("Confirme Password");
  lcd.setCursor(1, 1); lcd.print("Ingrese Password");
  i = -1;
 }
```

```
else if (Ingresar_Contrasena_Confirmacion [i] == '*') {
        lcd.clear();
        exit;
        return;
      }

      if (i == (sizeof(Ingresar_Contrasena_Confirmacion) - 1)) {
        i--;
      }
    }

if ((Ingresar_Contrasena_Confirmacion[5] == '\0') ||
(Ingresar_Contrasena_Confirmacion[6] == '\0') ||
(Ingresar_Contrasena_Confirmacion[7] == '\0') ||
(Ingresar_Contrasena_Confirmacion[8] == '\0') ||
(Ingresar_Contrasena_Confirmacion[9] == '\0')) {

      if (strcmp (Ingresar_Contrasena, Ingresar_Contrasena_Confirmacion) ==
0) {
        strcpy(credenciales [user].pwd, Ingresar_Contrasena);
        lcd.clear ();
        lcd.setCursor(2, 1); lcd.print("Guardando");
        delay(1000);

        for (i = 0; i < sizeof(credenciales) / sizeof(USUARIOS_CREDENCIALES);
i++) {
          EEPROM.put(i * sizeof(USUARIOS_CREDENCIALES) +
CREDENCIALDESPLAZAMIENTO, credenciales[i]);
        }
      }
```

```
else {
        lcd.clear ();
        lcd.setCursor(2, 1); lcd.print("INCORRECTO");
        delay(1000);
      }
    }
  }
  return;
}

int guardar_tag() {
  lcd.clear();
  lcd.setCursor(3, 0); lcd.print("Ingrese ID");
  for (i = 0; i < sizeof(Ingresar_ID); i++) {
    kp = Teclado ();
    Ingresar_ID[i] = kp;
    if (i < 5) {
      lcd.setCursor(5 + i, 3); lcd.print(Ingresar_ID[i]);
    }

    if (Ingresar_ID[i] == 'A') {
      Ingresar_ID[i] = '\0';
      if (Ingresar_ID[0] == '\0') {
        lcd.clear();
        lcd.setCursor(3, 0); lcd.print("Ingrese ID");
        i = -1;
      }
```

```
    }
     else {
       break;
     }
   }

    else if (Ingresar_ID[i] == 'B') {
     lcd.clear();
     lcd.setCursor(3, 0); lcd.print("Ingrese ID");
     i = -1;
   }

   else if (Ingresar_ID[i] == '*') {
     lcd.clear();
     return false;
     exit;
   }

   if (i == (sizeof(Ingresar_ID) - 1)) {
     i--;
   }
 }

if ((Ingresar_ID[1] == '\0') || (Ingresar_ID[2] == '\0') || (Ingresar_ID[3] == '\0') ||
(Ingresar_ID[4] == '\0') || (Ingresar_ID[5] == '\0')) {


for (user = 0; user < sizeof(credenciales) /
sizeof(USUARIOS_CREDENCIALES); user ++) {
     if (strcmp (credenciales[user].id, Ingresar_ID) == 0) {
```

```
        lcd.clear();
        lcd.setCursor(2, 0); lcd.print("Acerque el tag");
        digitalWrite(Modo_Lectura_LED, HIGH);
        delay(5000);

        if ( ! rfid.PICC_IsNewCardPresent()) {
          return 0;
        }

        if ( ! rfid.PICC_ReadCardSerial()) {
          return 0;
        }

for (i = 0; i < sizeof(credenciales) / sizeof(USUARIOS_CREDENCIALES); i
++) {

          if ((rfid.uid.uidByte[0] == credenciales[i].rfid[0]) && (rfid.uid.uidByte[1]
== credenciales[i].rfid[1]) && (rfid.uid.uidByte[2] == credenciales[i].rfid[2])
&& (rfid.uid.uidByte[3] == credenciales[i].rfid[3]) || ((rfid.uid.uidByte[0] ==
TagAdm[0]) && (rfid.uid.uidByte[1] == TagAdm[1]) && (rfid.uid.uidByte[2] ==
TagAdm[2]) && (rfid.uid.uidByte[3] == TagAdm[3]))) {

            lcd.clear ();
            lcd.setCursor(0, 0); lcd.print("Tag Existente");
            digitalWrite(Modo_Lectura_LED, LOW);
            digitalWrite(Acceso_Negado, HIGH);
            delay(3000);
            digitalWrite(Acceso_Negado, LOW);
            lcd.clear ();
```

```
          break;
        }


         else {
          if (i == (sizeof(credenciales) / sizeof(USUARIOS_CREDENCIALES) -
1)) {
            for (byte j = 0; j < 4; j++) {
              credenciales[user].rfid[j] = rfid.uid.uidByte[j];
            }


            for (int a = 0; a < sizeof(credenciales) /
sizeof(USUARIOS_CREDENCIALES); a++) {
              EEPROM.put(a * sizeof(USUARIOS_CREDENCIALES) +
CREDENCIALDESPLAZAMIENTO, credenciales[a]);
            }


            lcd.clear ();
            digitalWrite(Modo_Lectura_LED, LOW);
            digitalWrite(Abierto_LED, HIGH);
            lcd.setCursor(5, 1); lcd.print("guardado");
            delay(3000);
            digitalWrite(Abierto_LED, LOW);
            lcd.clear();
          }
        }
      }


    digitalWrite(Modo_Lectura_LED, LOW);
```

```
      rfid.PICC_HaltA();
      delay(1000);
      rfid.PCD_StopCrypto1();
      return 1;
      break;
    }

    else {

      if (user == (sizeof(credenciales) / sizeof(USUARIOS_CREDENCIALES)
- 1)) {
        lcd.clear();
        lcd.setCursor(2, 0); lcd.print("No existe");
        delay(1000);
        lcd.clear();
      }
    }
  }
 }
}

void Borrar_Tag() {
 for (i = 0; i < sizeof(credenciales ) / sizeof(USUARIOS_CREDENCIALES);)
{
  EEPROM.get(i * sizeof(USUARIOS_CREDENCIALES) +
CREDENCIALDESPLAZAMIENTO, credenciales[i]);

  if (credenciales [i].id[6] != '\0') {
    if (credenciales[i].rfid[0] != 0) {
      lcd.clear();
```

```
      lcd.setCursor(1, 0); lcd.print("Borrar Tag");
       lcd.setCursor(3, 1); lcd.print("Usuario"); lcd.setCursor(12, 1); lcd.print(i + 1);
       lcd.setCursor(4, 2); lcd.print("ID: "); lcd.setCursor(9, 2);
lcd.print(credenciales[i].id);
       lcd.setCursor(4, 3); lcd.print("Tag: Si");
       lcd.setCursor(15, 1); lcd.write(4);
       lcd.setCursor(1, 1); lcd.write(5);

       kp = Teclado ();
       switch (kp) {

         case 'C':
          i++;
          if (i == (sizeof(credenciales ) / sizeof(USUARIOS_CREDENCIALES))) {
            i = 0;
          }
          break;

         case 'D':
          i--;

          if (i < 0) {
            i = (sizeof(credenciales ) / sizeof(USUARIOS_CREDENCIALES)) - 1;
          }
          break;

          case 'A':
          lcd.clear();
          Desea_Borrar_Tag();
```

```
          return;
          break;

        case 'B':
          lcd.clear();
          Administracion_de_Usuarios();
          return;
          break;

        case '*':
          lcd.clear();
          inicio();
          return;
          break;

        default:
          Borrar_Tag();
          return;
          break;
      }
    }

    else {
      lcd.clear();
      lcd.setCursor(1, 0); lcd.print("Borrar Tag");
      lcd.setCursor(3, 1); lcd.print("Usuario"); lcd.setCursor(12, 1); lcd.print(i +
1);
      lcd.setCursor(4, 2); lcd.print("ID: "); lcd.setCursor(9, 2);
lcd.print(credenciales[i].id);
```

```
        lcd.setCursor(4, 3); lcd.print("Tag: No");
        lcd.setCursor(15, 1); lcd.write(4);
        lcd.setCursor(1, 1); lcd.write(5);

        kp = Teclado ();
        switch (kp) {
          case 'C':
            i++;

            if (i == (sizeof(credenciales ) / sizeof(USUARIOS_CREDENCIALES)))
{
              i = 0;
            }
            break;

          case 'D':
            i--;
            if (i < 0) {
              i = (sizeof(credenciales ) / sizeof(USUARIOS_CREDENCIALES)) -
1;
            }
            break;

          case 'B':
            lcd.clear();
            Administracion_de_Usuarios();
            return;
            break;
```

```
            case '*':
            lcd.clear();
            inicio();
            return;
            break;

          default:
            Borrar_Tag();
            return;
            break;
        }
      }
    }

    else {
      lcd.clear();
      lcd.setCursor(1, 0); lcd.print("Borrar Tag");
      lcd.setCursor(3, 1); lcd.print("Usuario"); lcd.setCursor(12, 1); lcd.print(i +
1);
      lcd.setCursor(4, 2); lcd.print("No data");
      lcd.setCursor(15, 1); lcd.write(4);
      lcd.setCursor(1, 1); lcd.write(5);
      kp = Teclado ();
      switch (kp) {
        case 'C':
          i++;
if (i == (sizeof(credenciales ) / sizeof(USUARIOS_CREDENCIALES))) {
            i = 0;
          }
```

```
      break;

    case 'D':
      i--;
      if (i < 0) {
        i = (sizeof(credenciales ) / sizeof(USUARIOS_CREDENCIALES)) - 1;
      }
      break;

    case 'B':
      lcd.clear();
      Administracion_de_Usuarios();
      return;
      break;

    case '*':
      lcd.clear();
      inicio();
      return;
      break;

    default:
      Borrar_Tag();
      return;
      break;
   }
  }
}
```

```
return;
}

void Desea_Borrar_Tag() {
  lcd.clear();
  lcd.setCursor(0, 0); lcd.print("Desea Borrar Tag");
  lcd.setCursor(0, 1); lcd.print("Usuario"); lcd.setCursor(8, 1);
lcd.print(credenciales[i].id);
  lcd.setCursor(0, 2); lcd.print("1. Si");
  lcd.setCursor(0, 3); lcd.print("2. No");
  kp = Teclado ();
  switch (kp) {
    case '1':
      lcd.clear();
      lcd.setCursor(3, 1); lcd.print("Borrado");

      for (int j = 0; j < sizeof(rfid.uid.size); j++) {
        credenciales [i].rfid[j] = 0;
      }

      EEPROM.put(i * sizeof(USUARIOS_CREDENCIALES) +
CREDENCIALDESPLAZAMIENTO, credenciales[i]);
      delay(1000);
      lcd.clear();
      Administracion_de_Usuarios();
      return;
      break;
```

```
    case '2':
      lcd.clear();
      Borrar_Tag();
      return;
      break;

    case 'B':
      lcd.clear();
      Borrar_Tag();
      return;
      break;

    case '*':
      lcd.clear();
      inicio();
      return;
      break;

    default:
      Desea_Borrar_Tag();
      return;
      break;
  }
  return;
}
```

```
void Cambiar_Contrasena_Admnistrador() {
  lcd.clear();
  lcd.setCursor(0, 0); lcd.print("Nueva Contrasena");
  lcd.setCursor(1, 1); lcd.print("Ingrese Password");

  for ( i = 0; i < sizeof(Ingresar_Contrasena); i++) {
    kp = Teclado ();
    Ingresar_Contrasena[i] = kp;

    if (i < 9) {
      lcd.setCursor(3 + i, 3); lcd.print('*');
    }

    if (Ingresar_Contrasena[i] == 'A') {
      Ingresar_Contrasena[i] = '\0';

      if (strlen(Ingresar_Contrasena) < 4) {
        lcd.clear ();
        lcd.setCursor(3, 1); lcd.print("Contrasena");
        lcd.setCursor(5, 2); lcd.print("Corta");
        delay(1000);

        lcd.clear();
        lcd.setCursor(0, 0); lcd.print("Nueva Contrasena");
        lcd.setCursor(1, 1); lcd.print("Ingrese Password");
        i = -1;
      }
```

```
      else {
      break;
    }
  }

  else if (Ingresar_Contrasena[i] == 'B') {
    lcd.clear();
    lcd.setCursor(0, 0); lcd.print("Nueva Contrasena");
    lcd.setCursor(1, 1); lcd.print("Ingrese Password");
    i = -1;
  }

  else if (Ingresar_Contrasena[i] == '*') {
    lcd.clear();
    exit;
    return;
  }

  if (i == (sizeof(Ingresar_Contrasena) - 1)) {
    i--;
  }
 }


if ((Ingresar_Contrasena[5] == '\0') || (Ingresar_Contrasena[6] == '\0') ||
(Ingresar_Contrasena[7] == '\0') || (Ingresar_Contrasena[8] == '\0') ||
(Ingresar_Contrasena[9] == '\0')) {
```

```
lcd.clear();
lcd.setCursor(1, 0); lcd.print("Confirme Password");
lcd.setCursor(1, 1); lcd.print("Ingrese Password");

for (int i = 0; i < sizeof(Ingresar_Contrasena_Confirmacion); i++) {
 kp = Teclado ();
 Ingresar_Contrasena_Confirmacion[i] = kp;

 if (i < 9) {
  lcd.setCursor(3 + i, 3); lcd.print('*');
 }

 if (Ingresar_Contrasena_Confirmacion[i] == 'A') {
  Ingresar_Contrasena_Confirmacion[i] = '\0';


  if (strlen(Ingresar_Contrasena_Confirmacion) < 4) {
   lcd.clear ();
   lcd.setCursor(3, 1); lcd.print("Contrasena");
   lcd.setCursor(5, 2); lcd.print("Corta");
   delay(1000);

   lcd.clear();
   lcd.setCursor(1, 0); lcd.print("Confirme Password");
   lcd.setCursor(1, 1); lcd.print("Ingrese Password");
   i = -1;
  }
```

```
else {
          break;
        }
      }

      else if (Ingresar_Contrasena_Confirmacion[i] == 'B') {
        lcd.clear();
        lcd.setCursor(1, 0); lcd.print("Confirme Password");
        lcd.setCursor(1, 1); lcd.print("Ingrese Password");
        i = -1;
      }

      else if (Ingresar_Contrasena_Confirmacion[i] == '*') {
        lcd.clear();
        return;
      }

      if (i == (sizeof(Ingresar_Contrasena_Confirmacion) - 1)) {
        i--;
      }
    }
  }

if ((Ingresar_Contrasena_Confirmacion[5] == '\0') ||
(Ingresar_Contrasena_Confirmacion[6] == '\0') ||
(Ingresar_Contrasena_Confirmacion[7] == '\0') ||
(Ingresar_Contrasena_Confirmacion[8] == '\0') ||
(Ingresar_Contrasena_Confirmacion[9] == '\0')) {
```

```
if (strcmp (Ingresar_Contrasena, Ingresar_Contrasena_Confirmacion) ==
0) {
      lcd.clear();
      lcd.setCursor(3, 1); lcd.print("CORRECTO");
      delay(1000);
      Estado = 1;
      EEPROM.write(DESPLAZAMIENTO_CONTRASENA, Estado);
      delay(500);


      for (int i = 0; i < sizeof(Ingresar_Contrasena) /
sizeof(Ingresar_Contrasena[0]); i++) {
        EEPROM.put(i * sizeof(Ingresar_Contrasena[0]),
Ingresar_Contrasena[i]);
      }
      delay(500);
      lcd.clear();
      lcd.setCursor(3, 1); lcd.print("Guardando");
      delay(1000);
    }


    else {
      lcd.clear();
      lcd.setCursor(2, 1); lcd.print("INCORRECTO");
      delay(2000);
    }
  }
  return;
}
```

```
void Menu_Admnistrador_Segunda_Parte() {
  lcd.clear();
  lcd.setCursor(0, 0); lcd.print("Menu Admnistrador");
  lcd.setCursor(0, 1); lcd.print("4. Agregar tag");
  lcd.setCursor(0, 2); lcd.print("5. Borrar tag");
  lcd.setCursor(0, 3); lcd.print("6 Desactivar Alarma");
  lcd.setCursor(19, 3); lcd.write(1);
  kp = Teclado ();
  switch (kp) {

    case '4':
      Tag_Admnistrador();
      digitalWrite(Modo_Lectura_LED, LOW);
      Menu_Admnistrador_Segunda_Parte();
      lcd.clear();
      return;
      break;

    case '5':
      borrar_tag_admnistrador();
      lcd.clear();
      return;
      break;

    case '6':
      Desactivar_Alarma();
      lcd.clear();
      return;
      break;
```

```
      case 'B':
       Usuario_Admnistrador();
       lcd.clear();
       return;
       break;

     case 'C':
       lcd.clear();
       Menu_Administrador();
       return;
       break;

     case '*':
       lcd.clear();
       inicio();
       return;
       break;

     case '#':
       EEPROM.write(DESPLAZAMIENTO_CONTRASENA, 0);
      for (i = 0; i < sizeof(Ingresar_Contrasena_Adm) /
sizeof(Ingresar_Contrasena_Adm[0]); i++) {
        EEPROM.write(i * sizeof(Ingresar_Contrasena_Adm[0]), 0);
       }
       break;

     default:
       Menu_Admnistrador_Segunda_Parte();
       return;
```

```
    break;
  }
}



int Tag_Admnistrador() {
  lcd.clear();
  lcd.setCursor(2, 0); lcd.print("Acerque el tag");
  digitalWrite(Modo_Lectura_LED, HIGH);
  delay(5000);

  if ( ! rfid.PICC_IsNewCardPresent()) {
    return 0;
  }

  if ( ! rfid.PICC_ReadCardSerial()) {
    return 0;
  }

for (i = 0; i < sizeof(credenciales) / sizeof(USUARIOS_CREDENCIALES); i
++) {

   if (((rfid.uid.uidByte[0] == credenciales[i].rfid[0]) && (rfid.uid.uidByte[1] ==
credenciales[i].rfid[1]) && (rfid.uid.uidByte[2] == credenciales[i].rfid[2]) &&
(rfid.uid.uidByte[3] == credenciales[i].rfid[3])) || ((rfid.uid.uidByte[0] ==
TagAdm[0]) && (rfid.uid.uidByte[1] == TagAdm[1]) && (rfid.uid.uidByte[2] ==
TagAdm[2]) && (rfid.uid.uidByte[3] == TagAdm[3])) ) {
```

```
 lcd.clear ();
  digitalWrite(Modo_Lectura_LED, LOW);
  digitalWrite(Acceso_Negado, HIGH);
  lcd.setCursor(0, 0); lcd.print("Tag Existente");
  delay(3000);
  digitalWrite(Acceso_Negado, LOW);
  lcd.clear ();
  break;
 }

 else {
  if (i == (sizeof(credenciales) / sizeof(USUARIOS_CREDENCIALES) -
1)) {

   for (byte j = 0; j < 4; j++) {
    TagAdm[j] = rfid.uid.uidByte[j];
   }
for (int a = 0; a < sizeof(TagAdm); a++) {
    EEPROM.put(a * sizeof(TagAdm) +
TAG_DESPLAZAMIENTO_ADM, TagAdm[a]);
   }

   lcd.clear ();
   digitalWrite(Modo_Lectura_LED, LOW);
   digitalWrite(Abierto_LED, HIGH);
   lcd.setCursor(5, 1); lcd.print("Guardando");
   delay(3000);
   digitalWrite(Abierto_LED, LOW);
   lcd.clear();
```

```
      }
     }
   }

  rfid.PICC_HaltA();
  delay(1000);
  rfid.PCD_StopCrypto1();
  return 1;
}

void borrar_tag_admnistrador() {
  lcd.clear();  //Limpia pantalla
  lcd.setCursor(0, 0); lcd.print("Desea Borrar Tag");
  lcd.setCursor(0, 2); lcd.print("1. Si");
  lcd.setCursor(0, 3); lcd.print("2. No");
  kp = Teclado ();
  switch (kp) {

    case '1':
      lcd.clear();
      lcd.setCursor(3, 1); lcd.print("Borrado");
      for (int j = 0; j < sizeof(TagAdm); j++) {
        TagAdm[j] = 0;
      }
      for (int a = 0; a < sizeof(TagAdm); a++) {
        EEPROM.put(a * sizeof(TagAdm) + TAG_DESPLAZAMIENTO_ADM,
TagAdm[a]);
      }
      delay(1000);
```

```
 lcd.clear();
  Menu_Admnistrador_Segunda_Parte();
  return;
  break;

 case '2':
  lcd.clear();
  Menu_Admnistrador_Segunda_Parte();
  return;
  break;

 case 'B':
  lcd.clear();
  Menu_Admnistrador_Segunda_Parte();
  return;
  break;

 case '*':
  lcd.clear();
  inicio();
  return;
  break;

 default:
  borrar_tag_admnistrador();
  return;
  break;
}
```

```
 }
  return;
}

void Desactivar_Alarma() {
  digitalWrite(Peligro, LOW);
  lcd.clear();
  lcd.setCursor(0, 1); lcd.print("Alarma Desactivada");
  delay(1000);
}

void Usuarios() {
  lcd.clear();
  lcd.setCursor(4, 0); lcd.print("Usuario");
  lcd.setCursor(3, 1); lcd.print("Ingrese ID");

  for (i = 0; i < sizeof(Ingresar_ID); i++) {
    kp = Teclado ();
    Ingresar_ID[i] = kp;

    if (i < 5) {
      lcd.setCursor(5 + i, 3); lcd.print(Ingresar_ID[i]);
    }

    if (Ingresar_ID[i] == 'A') {
      Ingresar_ID[i] = '\0';
      if (Ingresar_ID[0] == '\0') {
        lcd.clear();
```

```
      lcd.clear();
      lcd.setCursor(4, 0); lcd.print("Usuario");
      lcd.setCursor(3, 1); lcd.print("Ingrese ID");
      i = -1;
    }
    else {
      break;
    }
  }

  else if (Ingresar_ID[i] == 'B') {
    lcd.clear();
    lcd.setCursor(4, 0); lcd.print("Usuario");
    lcd.setCursor(3, 1); lcd.print("Ingrese ID");
    i = -1;
  }

  else if (Ingresar_ID[i] == '*') {
    lcd.clear();
    inicio();
    return;
  }

  if (i == (sizeof(Ingresar_ID) - 1)) {
    i--;
  }
}
```

```
if ((Ingresar_ID[1] == '\0') || (Ingresar_ID[2] == '\0') || (Ingresar_ID[3] == '\0')
|| (Ingresar_ID[4] == '\0') || (Ingresar_ID[5] == '\0')) {


   for (user = 0; user < sizeof(credenciales) /
sizeof(USUARIOS_CREDENCIALES); user ++) {
     if (strcmp (credenciales[user].id, Ingresar_ID) == 0) {
       Existe = 1;
       break;
     }


     else {
       if (user == (sizeof(credenciales) /
sizeof(USUARIOS_CREDENCIALES) - 1)) {
         Existe = 2;
       }
     }
   }


   switch (Existe) {
     case 1:
       lcd.clear();
       lcd.setCursor(4, 0); lcd.print("Usuarios");
       lcd.setCursor(1, 1); lcd.print("Ingrese Password");
       for ( i = 0; i < sizeof(Ingresar_Contrasena); i++) {
         kp = Teclado ();
         Ingresar_Contrasena[i] = kp;
         if (i < 9) {
           lcd.setCursor(3 + i, 3); lcd.print('*');
         }
```

```
      if (Ingresar_Contrasena[i] == 'A') {
        Ingresar_Contrasena[i] = '\0';
        break;
       }

       else if (Ingresar_Contrasena[i] == 'B') {
        lcd.clear();//Limpia Pantalla
        lcd.setCursor(4, 0); lcd.print("Usuarios");
        lcd.setCursor(1, 1); lcd.print("Ingrese Password");
        i = -1;
       }

       else if (Ingresar_Contrasena[i] == '*') {
        lcd.clear();
        inicio();
        return;
       }

       if (i == (sizeof(Ingresar_Contrasena) - 1)) {
        i--;
       }
      }

if ((Ingresar_Contrasena[0] == '\0') || (Ingresar_Contrasena[1] ==
'\0') || (Ingresar_Contrasena[2] == '\0') || (Ingresar_Contrasena[3]
== '\0') || (Ingresar_Contrasena[4] == '\0') ||
(Ingresar_Contrasena[5] == '\0') || (Ingresar_Contrasena[6] ==
'\0') || (Ingresar_Contrasena[7] == '\0') || (Ingresar_Contrasena[8]
== '\0') || (Ingresar_Contrasena[9] == '\0')) {
```

```
        if ((strcmp (credenciales[user].pwd, Ingresar_Contrasena) == 0) &&
(cnt1 < 3)) {
          Menu_Usuario(user);
        }

        else {
          cnt1++;
          lcd.clear();
          lcd.setCursor(2, 1); lcd.print("INCORRECTO");
          delay(1000);
        }

        if (cnt1 == 3) {
          cnt1 = 0;
          digitalWrite(Peligro, HIGH);
          Serial.println("ALARMA");
        }
      }
      break;

    case 2:
      Existe = 0;
      lcd.clear();
      lcd.setCursor(4, 0); lcd.print("USUARIO");
      lcd.setCursor(3, 1); lcd.print("NO EXISTE");
      delay(1000);
      break;
  }
}
```

```
return;
}

void Menu_Usuario(int usuario_numero) {
  cnt1 = 0;
  lcd.clear();
  lcd.setCursor(3, 0); lcd.print("Menu Usuario");
  lcd.setCursor(0, 1); lcd.print("1. Ingresar");
  lcd.setCursor(0, 2); lcd.print("2. Cambiar Pass");
  lcd.setCursor(0, 3); lcd.print("3. Borrar Tag");
  kp = Teclado ();

  switch (kp) {
    case '1':
      Ingresar();
      Serial.println(credenciales[usuario_numero].id);
      return;
      break;

    case '2':
      Opcion_Cambiar_Contrasena_Usuario();
      lcd.clear();
      Menu_Usuario(usuario_numero);
      return;
      break;

     case '3':
      borrar_tag_del_usuario (usuario_numero);
      Menu_Usuario(usuario_numero);
```

```
      return;
       break;

    case 'B':
       Usuario_Admnistrador();
       return;
       break;

    case '*':
       inicio();
       return;
       break;

    default:
       Menu_Usuario(usuario_numero);
       return;
       break;
  }
}

void borrar_tag_del_usuario (int usuario_numero1) {
  lcd.clear();
  lcd.setCursor(0, 0); lcd.print("Desea Borrar Tag");
  lcd.setCursor(0, 2); lcd.print("1. Si");
  lcd.setCursor(0, 3); lcd.print("2. No");
  kp = Teclado ();
  switch (kp) {
    case '1':
       lcd.clear();
```

```
    lcd.setCursor(3, 1); lcd.print("Borrado");
      for (int j = 0; j < sizeof(rfid.uid.size); j++) {
        credenciales [usuario_numero1].rfid[j] = 0;
      }

for (i = 0; i < sizeof(credenciales ) /
sizeof(USUARIOS_CREDENCIALES); i++) {
        EEPROM.put(i * sizeof(USUARIOS_CREDENCIALES) +
CREDENCIALDESPLAZAMIENTO, credenciales[i]);
      }
      delay(1000);
      break;

    case '2':
      lcd.clear();
      Menu_Usuario(usuario_numero1);
      return;
      break;

    case 'B':
      lcd.clear();
      Menu_Usuario(usuario_numero1);
      return;
      break;

    case '*':
      lcd.clear();
      inicio();
      return;
```

```
break;


    default:
      borrar_tag_del_usuario(usuario_numero1);
      return;
      break;
  }
  return;
}
```