UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO

CARRERA:
INGENIERÍA ELECTRÓNICA

Trabajo de titulación previa a la obtención del título de:
INGENIEROS ELECTRÓNICOS

TEMA:
DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL DE ACCESO MULTINIVEL EN BASE DE RECEPTORES NEAR FIELD COMMUNICATION (NFC)

AUTORES:
DARWIN SEGUNDO CACUANGO GUACHALÁ
EDUARDO JAVIER ZAPATA NARVÁEZ

DIRECTORA:
LUISA FERNANDA SOTOMAYOR REINOSO

Quito, abril de 2015

DECLARATORIA DE RESPONSABILIDAD Y AUTORIZACIÓN DE USO DEL TRABAJO DE TITULACIÓN

Nosotros, autorizamos a la Universidad Politécnica Salesiana la publicación total o parcial de este trabajo de titulación y su reproducción sin fines de lucro.

Además, declaramos que los conceptos, análisis desarrollados y las conclusiones del presente trabajo son de exclusiva responsabilidad de los autores.

Quito, abril de 2015

_____

Darwin Segundo Cacuango Guachalá

CC: 172135074-0

_____

Eduardo Javier Zapata Narváez

CC: 172345253-6

DEDICATORIA

En primer lugar a Dios que ha guiado mi camino a lo largo de la carrera y permitirme cumplir este objetivo.

A mis padres, por su ejemplo y sabios consejos que me ayudan a mejorar como persona cada día, a todo el esfuerzo realizado para que pueda alcanzar esta meta, gracias a todo el apoyo que me han brindado he llegado hasta este momento tan importante de mi formación profesional.

A Tefa la persona que nunca dejo creer en mí, por su apoyo incondicional, por todo el cariño, gracias por cada momento, en fin son muchas personas que han estado a mi lado durante el desarrollo de este proyecto a quienes quiero agradecer su amistad, consejos y palabras de aliento.

Darwin Segundo Cacuango Guachalá

A Dios, por permite culminar esta etapa de mi vida con éxito, a mis padres Patricio Zapata y Gladys Narváez, por su sacrificio, su apoyo incondicional, por haberme dado los estudios, a mis hermanos a quienes quiero demasiado, los que han sido mi motivación para seguir adelante, esperando ser un ejemplo hacia ellos. De manera espacial a mi hermano Diego, quien hizo posible esto, mil gracias hermano.

Mis abuelitos que Dios me da la dicha de poder gozar aun con su presencia. Mi cariño, respeto y sobre todo admiración. A todos mis tíos y tías los cuales son parte fundamental en mi vida, por haberme hecho sentir como un hijo. A mis primos y primas, hermanos de tantas aventuras, me faltan palabras para expresar el cariño por cada uno de ustedes.

A Micky, quien ha sido mi refugio, es parte de mi vida y mi corazón, un eterno gracias. A mi gran amigo y compañero con quien hemos dedicado nuestro tiempo y esfuerzo, superando obstáculos por cumplir este sueño, Darwin lo logramos.

Eduardo Javier Zapata Narváez

AGRADECIMIENTOS

A la Universidad Politécnica Salesiana y a todos los que conforman esta noble institución, autoridades, profesores, por aportar con sus conocimientos a lo largo de la carrera para formarnos como buenos profesionales y personas de bien.

A nuestra tutora de tesis Ing. Luisa Sotomayor por su tiempo, dedicación y conocimientos compartidos hacia nosotros en la realización del proyecto.

Al Ing. Rafael Jaya por sus recomendaciones para culminar el presente proyecto de titulación.

ÍNDICE

INTRODUCCIÓN **1**

CAPÍTULO 1 ..... **2**

1.1 Justificación 2

1.2 Alcance del proyecto 2

1.3 Objetivos 3

1.3.1 Objetivo general 3

1.3.2 Objetivos específicos 3

1.4 Análisis del problema 3

CAPÍTULO 2 ..... **5**

DESCRIPCIÓN DE LA TECNOLOGÍA NEAR FIELD COMMUNICATION (NFC) **5**

2.1 Definición de la tecnología NFC 5

2.2 Características y funcionamiento 6

2.2.1 Modos de funcionamiento. 7

2.2.1.1 Modo de comunicación pasivo o iniciador. 7

2.2.1.2 Modo de comunicación activo u objetivo. 8

2.3 Evolución y avances de la tecnología NFC 9

2.3.1 Tarjetas sin contacto 10

2.3.1.1 MIFARE 10

2.3.1.2 FELICA 11

2.4 Estándares de comunicación 12

2.4.1 Estándares de regulación 13

2.4.1.1 Norma ISO/IEC 18092/ECMA-340/ETSI TS 102 190 13

2.4.1.2 Norma ISO/IEC 21481/ECMA-352/ETSI TS 102 312 14

2.4.1.3 Norma ISO 15693 (Vicinity Cards) (Gómez, 2013). 16

2.4.1.4 Norma ISO 14443 (Proximity Cards) (Gómez, 2013). 16

2.4.2 Formato de intercambio de datos NFC (NDEF) 17

2.4.2.1 Obtención de información de etiquetas NFC 19
2.4.2.2 RTD, Definición de tipo de registro 20
2.5 Aplicaciones de la tecnología NFC 20
CAPÍTULO 3 22
SISTEMA DE COMUNICACIÓN NFC 22
3.1 Arquitectura de la comunicación NFC 22
3.1.1 Arquitectura del NCI 23
3.2.1 Tipos de comunicación NFC 23
3.2.1 Modo de comunicación Peer-to-Peer 24
3.2.2 Modo lectura / escritura 25
3.5 Modo emulación de tarjeta inteligente NFC 26
3.3 Establecimiento de la comunicación NFC 27
3.3.1 Protocolo (SPI) 29
3.3.2 Comunicación RS-485 31
3.4.1 Aspectos de seguridad NFC 33
3.4.1 EavesdropPing (Interceptación) 33
3.4.2 Data corruption (corrupción de datos) 34
3.4.3 Data modification (modificación de datos) 34
3.4.4 Man-in-the-middle attack (Ataques de intermediario) 35
3.5 Lector NFC (Módulo RFID RC522) 35
CAPÍTULO 4 37
DISEÑO E IMPLEMENTACIÓN DEL PROTOTIPO 37
4.1 Descripción del sistema prototipo 37
4.2 Diseño del prototipo 39
4.2.1 Diseño electrónico del prototipo. 41
4.2.1.1 Comunicación SPI 41
4.2.1.2 Conversor 43

4.2.1.3 Etapa de control de potencia 44
4.3 Requerimiento de hardware y software 45
4.3.1 Hardware 45
4.3.1.1 PIC 16F628A 45
4.3.1.2 Circuito integrado SN75176 46
4.3.1.3 Cerradura electromagnética. 47
4.3.1.4 Fuente de voltaje. 48
4.3.1.5 Conversor RS-485 a RS-232 50
4.3.2 Software 50
4.4 Implementación del prototipo 52
4.4.1 Diseño de la tarjeta de control 53
4.4.2 Diseño de aplicación del prototipo 59
4.5 Costos del sistema 64
CAPÍTULO 5 66
ANÁLISIS DE RESULTADOS 66
5.1 Pruebas de funcionamiento del prototipo 66
5.1.1 Pruebas de hardware 66
5.1.2 Pruebas de envío y recepción de información. 68
5.1.3 Prueba de funcionamiento total del sistema 69
5.2 Análisis y resultados 73
CONCLUSIONES 75
RECOMENDACIONES 76
LISTA DE REFERENCIAS 77

ÍNDICE DE FIGURAS

Figura 1. Dispositivo (Peer to Peer).....5
Figura 2. Tecnología NFC 6
Figura 3. Modo de comunicación pasivo.....7
Figura 4. Modo de comunicación activa.....8
Figura 5. Miembros NFC Forum. 10
Figura 6. Tarjeta MIFARE.....11
Figura 7. Tarjeta FeliCa 12
Figura 8. Acuerdo protocolo NFC. 13
Figura 9. NFCIP-2 14
Figura 10. Modo de selección de dispositivo NFCIP-2.....15
Figura 11. Esquema de la estandarizacion NFC. 17
Figura 12. Formato de un registro NDFE 18
Figura 13. Arquitectura dispositivo NFC 22
Figura 14. Modos de operación dispositivo NFC 24
Figura 15. Elementos protocolo Peer to Peer (NFC). 24
Figura 16. Elementos protocolo lectura/escritura (NFC).....26
Figura 17. Pasos transacción NFC 28
Figura 18. Conexión Maestro-Esclavo. 30
Figura 19. Conexión Maestro-Esclavo (Varios) 30
Figura 20. Señales SPI 31
Figura 21. Balanceo de líneas RS-485.....32
Figura 22. Tipo de comunicación RS-485 32
Figura 23. Ataque hombre en medio.....35
Figura 24. Módulo RFID RC522 35
Figura 25. Diagrama general del prototipo 37
Figura 26. Diagrama simplificado del prototipo.....38
Figura 27. Dimensiones del prototipo.....39
Figura 28. Áreas de seguridad 40
Figura 29. Área de control del prototipo.....41
Figura 30. Diagrama conexión Lector-PIC.....42
Figura 31. Diagrama fuente reguladora de voltaje.....42

Figura 32 Diagrama control RS-485 43
Figura 33. Concentrador RS-485 44
Figura 34. Diagrama bloques del proyecto 44
Figura 35. Diagrama conexión de réles 45
Figura 36. PIC 16F628A. 46
Figura 37. SN75176. 46
Figura 38 Cerradura electromagnética 47
Figura 39. Funcionamiento de la cerradura electromagnética 47
Figura 40. Secuencia de funcionamiento de una fuente 48
Figura 41. Diagrama de bloques de una fuente de voltaje 49
Figura 42. Regulador de voltaje 49
Figura 43. Conversor RS-485 a RS-232 50
Figura 44. Visual Studio Express 50
Figura 45. Diagrama de bloques de la tarjeta de control 53
Figura 46. Diagrama de flujo de proceso del proyecto 54
Figura 47. Diseño esquemático electrónico 55
Figura 48. Diseño PCB placa de control 56
Figura 49. Placa de control terminada 56
Figura 50. Esquema de diseño del prototipo 57
Figura 51. Construcción del prototipo 57
Figura 52. Ubicación de cerraduras electromagnéticas 58
Figura 53. Instalación de lectores 58
Figura 54. Prototipo terminado 59
Figura 55. Creación del proyecto (Visual Studio) 60
Figura 56. Lista de controles y propiedades (Visual Studio) 61
Figura 57. Creación de campos de la base de datos 62
Figura 58. Selección de directorio de la base de datos 62
Figura 59. Elección de variables a importar 63
Figura 60. Variables enlazadas con la aplicación 63
Figura 61. Selección de variables y tipo de control 64
Figura 62. Fuente de alimentación del proyecto 67
Figura 63. Led indicador de funcionamiento 67
Figura 64. Parámetros comunicación Terminal X-CTU 68

Figura 65. Recepción de lectura de tarjeta.....68
Figura 66. Envío de codigo hacia prototipo.....69
Figura 67. Ventana de inicio de la aplicación.....70
Figura 68. Ventana de monitoreo del sistema 70
Figura 69. Selección de puerto.....71
Figura 70. Sección para editar información del personal 71
Figura 71. Adquisición de código de tarjeta.....72
Figura 72. Base de datos de usuarios.....72
Figura 73. Prueba de rango de comunicación…….….....74

ÍNDICE DE TABLAS

Tabla 1. Tipos de etiquetas NFC 19
Tabla 2. Evolucion de estandarización NFC 20
Tabla 3. Costo del sistema 65
Tabla 4. Costo del sistema con tecnología arduino 65
Tabla 5. Variación del costo de comercialización del lector 65

ÍNDICE DE ANEXOS

Anexo 1. Planos del prototipo. 80
Anexo 2. Circuitos eléctricos y electrónicos 81
Anexo 3. Programa PIC 16F628A 83
Anexo 4. Programa de la aplicación 93

RESUMEN

El uso de la tecnología de comunicación de campo cercano (NFC) aplicada al control de acceso en distintas áreas, se plantea en este proyecto como una solución alternativa al problema de seguridad en el control de accesos y flujo de personas, aprovechando las ventajas que brinda esta tecnología con alto nivel de seguridad y velocidad de comunicación.

El sistema de control de acceso basado en la tecnología NFC capaz de direccionar a los usuarios según los requerimientos del sistema, se realizó empezando con el análisis de conceptos importantes de la tecnología NFC, a partir de estos conceptos se crea el diseño de la red de acceso para el intercambio de datos entre los dispositivos usando terminales inalámbricos de corto alcance, posteriormente se implementa el prototipo e interconecta el sistema de control de acceso formado por siete (7) receptores NFC empleando una red de microcontroladores, la cual procesa la información adquirida por los receptores NFC, para brindar o limitar acceso a los diferentes usuarios mediante sus privilegios preestablecidos en las distintas áreas a un dispositivo central, desde donde se administra y se monitorea la información adquirida por el sistema, gracias a un software de control donde se visualiza toda la actividad en tiempo real.

El software de control se enlaza con una base de datos donde se almacena información personal de los usuarios con una etiqueta NFC a cada uno.

Finalmente, se ejecutan pruebas de funcionamiento y se analiza los resultados obtenidos con la implementación del prototipo.

ABSTRACT

Using technology Near Field Communication (NFC) applied to access control in different areas is proposed in this project as an alternative solution to security access control and flow of people, taking advantage the advantages offered by this technology with high security and communication speed.

The access control system based on NFC able to direct users according to the system requirements, technology was performed starting with the analysis of important concepts of NFC technology, these concepts from the design of the network is created Access to exchange data between devices using short-range handsets, then the prototype is implemented and interconnects the access control system consisting of eight (8) NFC receivers using a network of microcontrollers, which processes the information acquired by NFC receivers to provide or restrict access to different users through their preset in different areas to a central device, where it is administered and the information acquired by the system is monitored by a control software which is displayed privileges all activity in real time.

The control software is linked to a database where user's personal information is stored with an NFC tag to each.

Finally run performance tests and the results obtained with the implementation of the prototype are analyzed.

INTRODUCCIÓN

La comunicación de campo cercano es una tecnología que permite que dos dispositivos se comuniquen de forma inalámbrica a corta distancia, a pesar que cumplirá 12 años en el mercado desde su lanzamiento el 8 de diciembre del 2003, y los intentos de grandes empresas a nivel mundial de impulsar la tecnología en américa latina y específicamente en Ecuador su uso es muy reducido.

Globalmente el proyecto desarrollado tiene como uno de sus objetivos evidenciar las ventajas de la tecnología NFC e impulsar el estudio y uso de la misma en el país mediante la implementación de una de sus tantas aplicaciones en este caso un sistema de control de acceso alta seguridad y de bajo costo, aprovechando las ventajas que brinda esta tecnología.

El prototipo desarrollado es un sistema que permite controlar el flujo de personas en lugares como hospitales, eventos masivos y sobre todo en oficinas, donde adicionalmente este sistema funcionará como un registro de cumplimiento de personal de jornadas laborales, ya que cada una de las tarjetas NFC entregadas a los empleados de la institución estará enlazada a su información personal, lo cual además de permitir el acceso a las diferentes dependencias también servirá para llevar un registro la jornada laborar de cada uno.

Este trabajo se encuentra organizado por capítulos, el primer capítulo describe el planteamiento del problema que servirá para justificar la realización del proyecto, el objetivo general, los objetivos específicos. En el capítulo dos se describe la tecnología Near Field Communication (NFC), características, avances y aplicaciones. El capítulo tres muestra la arquitectura de comunicación, tipos de comunicación y aspectos de seguridad de la tecnología. Finalmente en el capítulo cuatro se realizó la implementación del sistema prototipo además de sus respectivas pruebas de funcionamiento para posteriormente exponer los resultados y conclusiones obtenidas a lo largo del desarrollo del proyecto.

CAPÍTULO 1

En este capítulo se detallan la justificación e importancia del proyecto, el alcance los objetivos y el análisis del problema.

1.1 Justificación

La seguridad en controles de acceso de personal a diferentes dependencias, especialmente a las restringidas ha sido un tema de gran importancia en todo tipo de proyectos, puesto que en estas áreas se guarda equipos e información importantes para la empresa, en la actualidad existen algunas alternativas de estos sistemas, que permiten brindar seguridad de una manera inteligente, pero todavía poseen costos elevados para la pequeña y mediana industria de nuestro país.

La tecnología NFC aparece como una solución alternativa, por su comunicación inalámbrica que interactúa con una gran variedad de dispositivos (tarjetas, dispositivos móviles), por este motivo se pretende realizar un sistema de control en base a receptores NFC, aprovechando las ventajas que brinda esta tecnología; como su alto nivel de seguridad, flexibilidad y velocidad de comunicación.

Las tecnologías de corto alcance cada vez tienen mayor aceptación debido a las ventajas que presentan ante otro tipo de tecnología, siendo una de sus características que más sobresalen la escalabilidad y la flexibilidad, esta tecnología están en nuestra sociedad pero pasan desapercibidas y son de gran importancia ya que facilitan la vida de las personas.

1.2 Alcance del proyecto

El presente proyecto tiene como finalidad proponer un nuevo sistema de control de acceso de personal basado en la tecnología NFC capaz de direccionar a los usuarios a las diferentes áreas según los requerimientos del sistema.

Para realizar el sistema de control de acceso se va a crear una red de microcontroladores interconectados, la cual procesa la información adquirida por

los receptores NFC para brindar o limitar acceso a los diferentes usuarios mediante sus privilegios preestablecidos en las distintas áreas.

Esta información se podrá administrar y gestionar gracias a un software de control donde se visualiza toda la actividad en tiempo real.

1.3 Objetivos

1.3.1 Objetivo general

Diseñar e implementar un sistema de control de acceso multinivel mediante receptores NFC (Near Field Communication) para controlar el ingreso mediante privilegios de usuario.

1.3.2 Objetivos específicos

- Definir las ventajas y características de la tecnología NFC para aplicar en nuevos sistemas de seguridad de acceso.
- Diseñar la red de acceso para el intercambio de datos entre los dispositivos usando terminales inalámbricos de corto alcance.
- Implementar un prototipo e interconectar el sistema de control de acceso formado por cuatro (4) receptores NFC a un dispositivo central, para exponer el funcionamiento y los resultados obtenidos que permita la administración y la gestión de la información adquirida por el sistema.
- Ejecutar pruebas de funcionamiento y analizar los resultados obtenidos del prototipo implementado.

1.4 Análisis del problema

La seguridad en controles de acceso ha sido un tema de gran importancia en todo tipo de infraestructuras especialmente a las que tienen material o información de gran valor intelectual y económico.

En la actualidad existen algunas alternativas de sistemas de control de acceso de personal que permiten brindar seguridad de una manera inteligente, pero a un alto

costo y con varias limitaciones como: el tiempo de respuesta de los sistemas actuales, restricción de acceso para personal no autorizado a ciertas áreas, la poca apertura e importancia a la nueva tecnología en el campo de control de acceso, encontrando como gran inconveniente la administración y control de personal en distintas áreas.

Por esta razón se plantea como una solución alternativa a este problema, el uso de la tecnología NFC aplicada al control de acceso en distintas áreas, aprovechando las ventajas que brinda esta tecnología, como su alto nivel de seguridad y velocidad de comunicación.

CAPÍTULO 2

DESCRIPCIÓN DE LA TECNOLOGÍA NEAR FIELD COMMUNICATION (NFC)

En este capítulo se hará referencia a la tecnología inalámbrica NFC, la cual, se aplicará en el desarrollo del presente proyecto detallando su definición, características, modos de comunicación, evolución y avances, estándares de comunicación para el correcto intercambio de datos entre dispositivos NFC y sus aplicaciones para ofrecer soluciones en base a esta tecnología.

2.1 Definición de la tecnología NFC

La tecnología NFC se basa en una interfaz inalámbrica, el modo de funcionamiento es similar al Bluetooth, pero la comunicación NFC es más sencilla, ya que no es necesario un emparejamiento previo de los dispositivos que intervienen en la comunicación, solo que los dispositivos NFC estén en un rango de distancia adecuado para que se produzca el intercambio de información. La comunicación se realiza entre las aplicaciones y los dispositivos electrónicos que permiten este intercambio, ubicados a menos de 20 centímetros de distancia para dispositivos conocidos como (peer-to-peer), se refieren a una conexión limitada a dos extremos (Forum, 2004).

Dispositivos (peer to peer)

Figura 1. Comunicación NFC Peer to Peer.

Elaborado por: Darwin Cacuango y Eduardo Zapata.

Este tipo de comunicación no necesita de ninguna licencia debido a que el campo magnético utilizado por NFC 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 (RapidNFC, 2011).

Gracias a su compatibilidad con las tecnologías existentes como Bluetooth y RFID, hace que incremente el interés en esta tecnología haciendo de ella un nuevo punto de desarrollo para futuros proyectos basados en esta tecnología inalámbrica.

2.2 Características y funcionamiento

La tecnología NFC es una extensión de la norma ISO/IEC 14443, lo cual define la comunicación con tarjetas inteligentes, lectores y dispositivos NFC definidos dentro de esta norma, trabaja dentro de la banda ISM (Industrial, Scientific and Medical) de radio frecuencia de 13,56 MHz, es de plataforma abierta estandarizada en la ISO/IEC 18092 y la ECMA-340, la cual define los modos de comunicación para la interfaz de comunicación de campo cercano y protocolo (NFCIP-1) (Forum, 2004).

Esta norma también define los modos de comunicación de la tecnología NFC y específica, los esquemas de modulación, codificaciones, velocidades de transferencia y formato de la trama de la interfaz de radio frecuencia (RF) de dispositivos NFC que soportan velocidades de transmisión de 106, 212, 424 u 848 Kbits/s. ((ECMA), 2004).

Al contar con dispositivos con tecnología NFC y al aproximarlos en un rango no mayor de 20 centímetros, sus campos magnéticos entran en contacto por lo cual, se produce un acoplamiento por inducción magnética para la trasferencia de datos, como se indica en la Figura 2.

Tecnología NFC

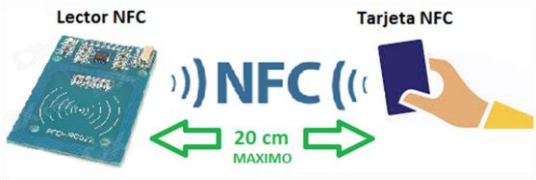


Figura 2. Distancia máxima de comunicación

Elaborado por: Darwin Cacuango y Eduardo Zapata.

Cualquier tipo de dispositivo NFC dentro del estándar ISO/IEC 14443 puede establecer una comunicación para el intercambio de datos e información. NFC fue creada principalmente para el uso en teléfonos móviles, tarjetas inteligentes ya que no se transmite datos masivamente (INTECO, 2013).

2.2.1 Modos de funcionamiento.

Al contar con la capacidad de comunicación bidireccional de NFC se puede establecer comunicación al enviar o recibir datos dependiendo de la función que realice el dispositivo NFC, este puede operar de dos formas, función iniciador (initiator) o función objetivo (target), a excepción de una etiqueta NFC, a continuación se describe los modos que funciona el dispositivo.

- Modo pasivo o iniciador.
- Modo activo u objeto

2.2.1.1 Modo de comunicación pasivo o iniciador.

En el modo de comunicación pasivo o iniciador solo un dispositivo genera el campo electromagnético y es quien comienza la conexión controlando el intercambio de información este campo es aprovechado por el dispositivo de destino.

Los pasos para la comunicación en modo pasivo se indica en la Figura 3 con las funciones descritas a continuación:

Modo de comunicación pasivo

Figura 3. (Rubén Abuín, 2014).

- NFC Iniciador.- Es un dispositivo NFC que consta de su propia fuente de alimentación permitiendo generar su campo electromagnético.
- NFC Objetivo.- El receptor es el dispositivo NFC que responde a las peticiones del iniciador, aprovechando el campo electromagnético que genera el iniciador para concretar la comunicación.

2.2.1.2 Modo de comunicación activo u objetivo.

En el modo de comunicación activo el dispositivo iniciador y el de destino tienen su fuente de energía para generar su propio campo electromagnético y poder transmitir información aleatoriamente desactivándolo mientras esperan la respuesta.

Modo de comunicación activa

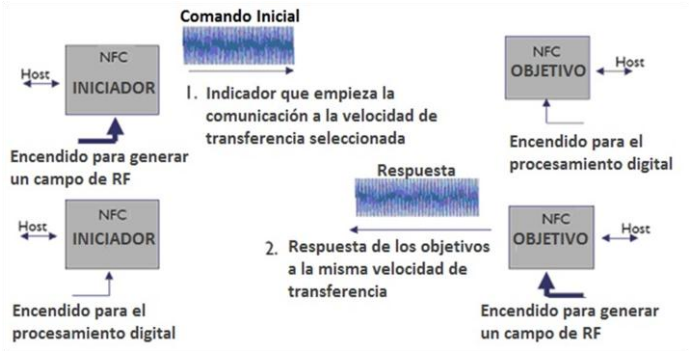


Figura 4. (Rubén Abuín, 2014).

En la figura 4, se puede observar que el dispositivo NFC iniciador y objetivo generan su propio campo electromagnético, un dispositivo desactiva su campo en espera de una respuesta del otro, conjuntamente los dos dispositivos pueden ponerse de acuerdo a qué velocidad trabajar y reajustar el parámetro en cualquier instante de la comunicación.

La tecnología NFC con estas características posee las siguientes ventajas:

- **Velocidad:** al interactuar los dispositivos NFC entre sí en el rango determinado su respuesta es instantánea, reduciendo los tiempos de comunicación.
- **Versatilidad:** los dispositivos NFC pueden comunicarse con otras tecnologías lo cual permite ampliar sus aplicaciones creando una gran opción en el mercado.
- **Comunicación:** la comunicación inalámbrica NFC resulta más rápida y sencilla que otras tecnologías como wifi o bluetooth que requieren de un proceso de emparejamiento previo al inicio de la comunicación.
- **Seguridad:** se considera como segura debido al rango de alcance determinado para el intercambio de datos entre dispositivos es pequeño.
- **Estandarizado:** al encontrarse dentro de las norma ISO/IEC puede trabajar con diversas tecnologías a nivel mundial y trabajar en un frecuencia que no tiene ninguna restricción (Forum, 2004).

2.3 Evolución y avances de la tecnología NFC

La tecnología NFC comenzó a desarrollarse en el año 2002 en una acción conjunta de Philips y Sony, con el fin de conseguir un protocolo compatible con tecnología sin contacto existente, como resultado de esta investigación surgió la tecnología NFC (Gómez, 2013).

Esta fue aprobada como ISO/IEC 18092 en diciembre de 2003 y más tarde como un estándar ECMA, y en marzo de 2004, Philips, Sony y Nokia formaron el NFC Forum para avanzar en el desarrollo de las especificaciones NFC consiguiendo que empresas como Google, Visa, At&t, PayPal, etc. Apoyen el desarrollo de esta tecnología para la búsqueda de soluciones rápidas y sencillas usando tarjetas sin contacto.

Actualmente el NFC Forum cuenta con unos 115 miembros, algunos de estos se muestra en la figura 5.

Miembros NFC Forum

Figura 5. (Forum, 2004).

2.3.1 Tarjetas sin contacto

Las tarjetas sin contacto son el avance tecnológico de las tarjetas inteligentes estas fueron inventadas y patentadas en los setenta. Existe una controversia acerca del inventor original entre los supuestos están, Juergen Dethloff de Alemania, Arimura de Japón y Roland Moreno de Francia, a este último se le adjudica este gran descubrimiento que dio lugar a la creación de las tarjetas de crédito y ha evolucionado hasta las tarjetas SIM utilizadas en los teléfonos móviles.

Las tarjetas sin contacto tienen mayor costo en comparación a las tarjetas de contacto, pero tienen una vida útil más extensa y mayor confiabilidad, un ejemplo de estas son las conocidas Smart Card o tarjetas inteligentes entre las que a nivel mundial destacan las tarjetas MIFARE y FELICA de donde deriva la tecnología NFC.

2.3.1.1 MIFARE

Son tarjetas inteligentes sin contacto desarrollada por Mikron, con aproximadamente 250 millones de tarjetas inteligentes y 1,5 millones de módulos lectores vendidos es una tecnología RFID a 13,56 MHz fabricada por varias empresas, la más famosa de ellas es Philips Electronics, se le considera como tarjeta inteligente debido a que la tecnología MIFARE permite leer y escribir en la tarjeta, con una distancia típica de lectura de 10 cm (unas 4 pulgadas). La

distancia de lectura siempre depende de la potencia del módulo lector, existiendo lectores de mayor y menor alcance. (Austria, 2002-2014)

Las tarjetas inteligentes sin contacto, así como los lectores y escritores de tarjetas MIFARE fueron desarrollados originalmente para transacciones de pago en sistemas de transporte público. Los lectores sin contacto son más eficientes disminuyendo tiempo de respuesta y fáciles de usar en comparación a los de contacto, los dispositivos sin contacto requieren de un mantenimiento mínimo al no sufrir desgastes importantes.

Tarjeta MIFARE

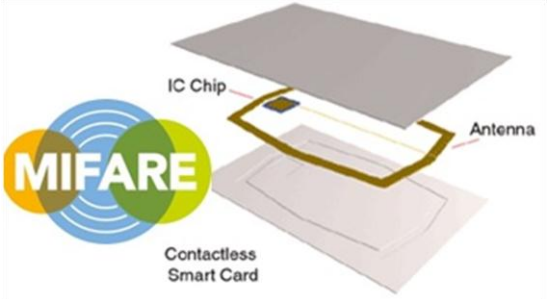


Figura 6. (Austria, 2002-2014).

En la figura 6, se muestra la estructura de una tarjeta MIFARE laminada por ambas caras por PVC y en su interior consta contienen un chip que permite almacenar y recuperar datos remotos.

2.3.1.2 FELICA

La tarjeta FeliCa surgió en el año 1988 fue creada por Sony Corporation en Japón, el nombre se refiere a *la tarjeta de Felicity*, que significa literalmente "felicidad". Los pioneros en usar este tipo de tarjeta fue Octopus en Hong Kong, la tecnología se utiliza en una variedad de tarjetas también en países como Singapur, Japón y el Estados Unidos.

Las tarjetas de FeliCa son dispositivos pasivos que no poseen fuente de alimentación propia. El lector de tarjetas crea una señal para poder interactuar con la tarjeta FeliCa, lee la información de ella completando el proceso de

comunicación, usada en gran parte del mercado a nivel mundial, donde su mayor participación se encuentra en Japón aproximadamente con 212 millones, seguido de Hong Kong y por ultimo Singapur.

La estructura de las tarjetas sin contacto FeliCa como se muestra en la figura 7, consiste en un chip IC y una antena con esta composición no se limitan solo a montarse en tarjetas, sino también en teléfonos móviles y relojes.

Tarjeta FeliCa

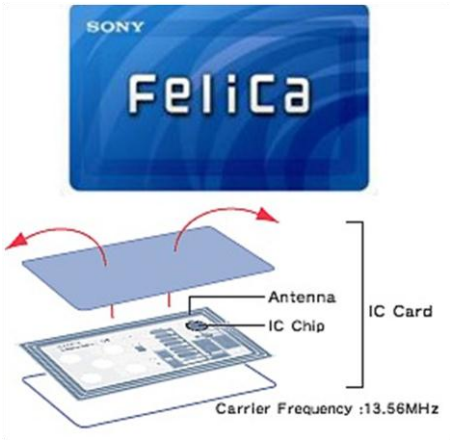


Figura 7.

Elaborado por: Darwin Cacuango y Eduardo Zapata.

2.4 Estándares de comunicación

La tecnología NFC y las Smart Card están reconocidos por organizaciones internacionales como ETSI (European Telecommunications Standards Institute), ISO/IE (International Organization for Standardization/Interntional Electro_technical Comission) y ECMA (European Association for Standardizing Information and Communication Systems) las mismas que proporcionan los esquemas de modulación, codificación, velocidad de trasferencia y la interfaz de radiofrecuencia (RF) de los dispositivos NFC (Forum, 2004).

Para comunicarse entre los dispositivos NFC se creó un formato estandarizado para la trasferencia de datos. Este formato es propiedad de NFC Forum, una

asociación industrial sin fines de lucro encargada de regular la interacción inalámbrica y la interoperabilidad entre dispositivos NFC.

2.4.1 Estándares de regulación

Los estándares de regulación de la tecnología NFC es un proceso que contiene especificaciones técnicas de aplicación que cubren protocolos de comunicación y formatos de intercambio de datos NFC, y se encuentra regulado por los siguientes estándares:

2.4.1.1 Norma ISO/IEC 18092/ECMA-340/ETSI TS 102 190

Esta Norma define los modos de comunicación para la interfaz Near Field Communication Interface and Protocol-1 (NFCIP-1 modo peer-to-peer) utilizando dispositivos de acoplamiento inductivo que operan en la frecuencia central de 13,56 MHz para la interconexión de periféricos. Esta norma define, en particular, los esquemas de codificación, modulación, las velocidades de transferencia, y arquitectura para las tazas de transferencia de datos de 106, 212, 424 Kbps, así como esquemas de inicialización y las condiciones requeridas para el control de colisión de datos durante el inicio de la comunicación peer-to-peer entre dispositivos. Este tipo de comunicación se asemeja a tecnologías inalámbricas de corto alcance, pero la transferencia física de datos es diferente ((ECMA), 2004).

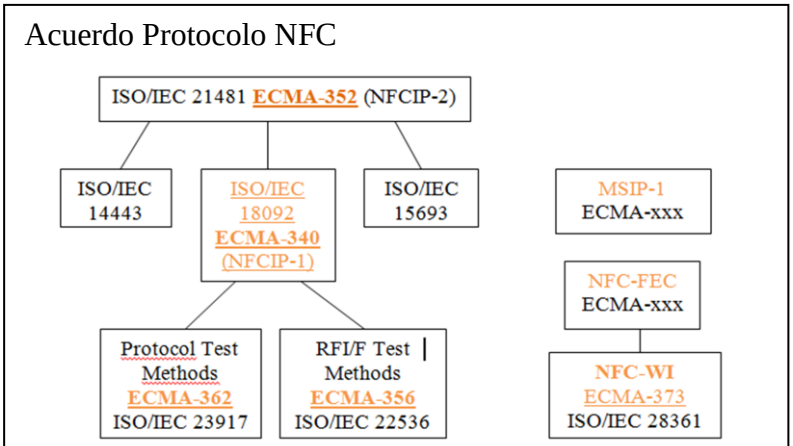


Figura 8. ((ECMA), 2004).

Las normas o estándares que conforman la ECMA-340 se observan en la figura 8, donde está integrada por la ECMA-362 (Protocol Test Methods), que especifica los métodos de prueba de protocolo de ECMA-340, además de los especificados en ECMA-356 (RF Interface Test Methods), específica los métodos de prueba para RF-NFCIP-1 dispositivos con antenas ajustadas dentro del área rectangular de 50 mm por 40 mm. ((ECMA), 2004).

2.4.1.2 Norma ISO/IEC 21481/ECMA-352/ETSI TS 102 312

Near Field Communication Interface and Protocol-2 (NFCIP-2), se especifica al comenzar la comunicación, estableciendo automáticamente el modo adecuado de comunicación entre dispositivos de aplicación de ECMA-340, como se muestra en la figura 9, es el conjunto de la norma ISO/IEC 14443 y la ISO/IEC 15693 donde se detalla acerca de las Tarjetas de proximidad sin contacto.

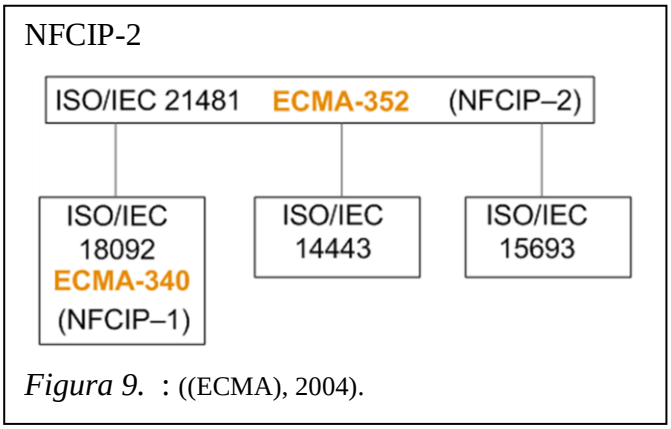


Figura 9. : ((ECMA), 2004).

Este estándar previene la interferencia entre otras comunicaciones que se estén produciendo en la banda de 13.56 MHz, definiendo los distintos modos tales como:

- **Modo NFC.-** Modo en el cual un dispositivo NFCIP-2 opera de acuerdo a lo especificado en ECMA 340.
- **Modo PICC (Proximity Integrated Circuit Card). -** Modo en el cual un dispositivo NFCIP-2 funciona como está especificado en ISO/IEC14443.
- **Modo PCD (Proximity Coupling Device). -** Dispositivo NFCIP-2 que opera tal como está especificado en ISO/IEC 14443.

- **Modo VCD (Vicinity Coupling Device).** - Modo en el cual un dispositivo NFCIP-2 funciona como está especificado en ISO/IEC 15693. ((ECMA), 2004).

En la figura 10, se describe el proceso de selección de los diferentes modos de un dispositivo NFCIP-2:

Modo de selección de dispositivos NFCIP-2

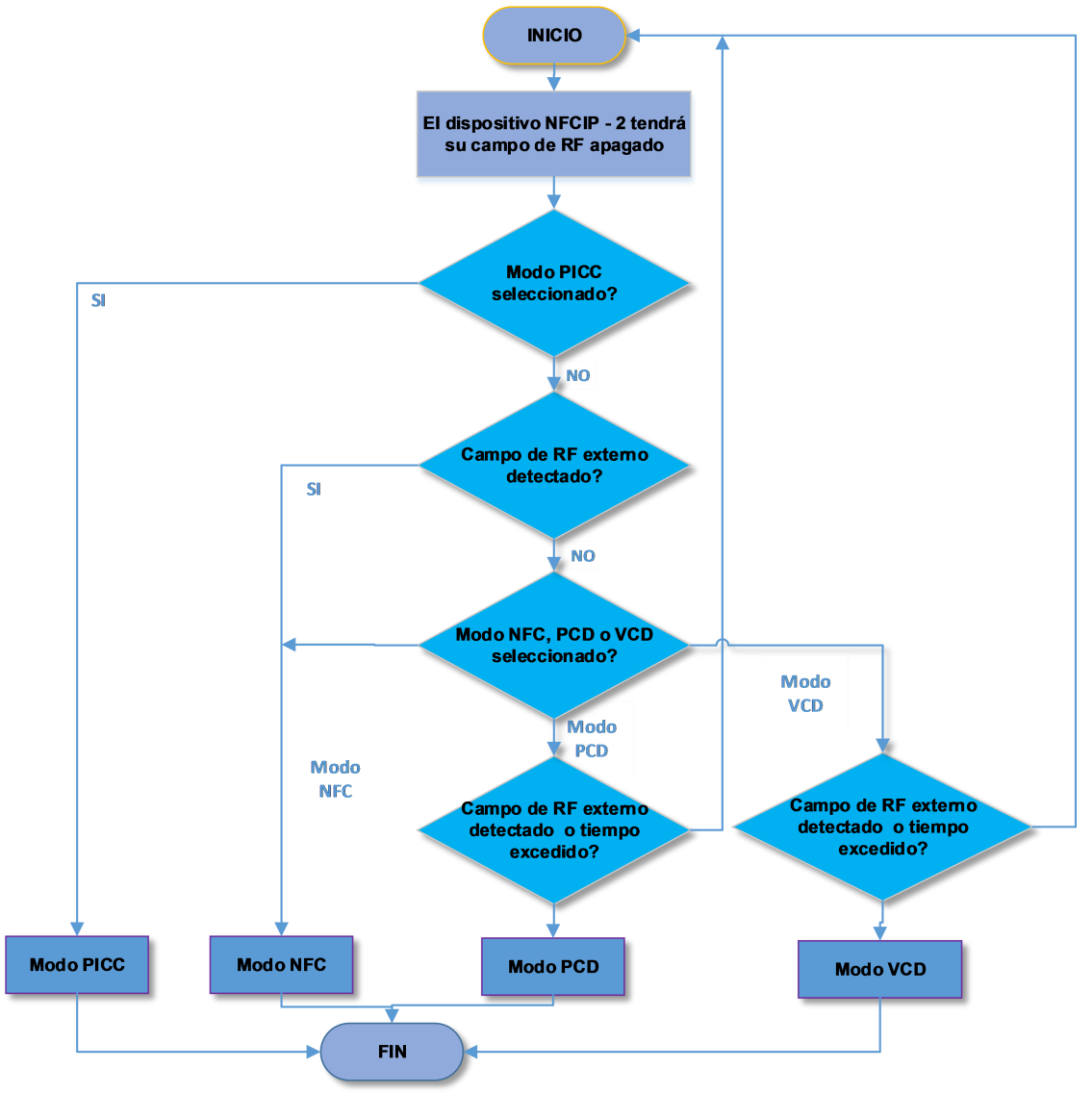


Figura 10. ((ECMA), 2004).

Elaborado por: Darwin Cacuango y Eduardo Zapata.

2.4.1.3 Norma ISO 15693 (Vicinity Cards) (Gómez, 2013).

Este estándar permite especificar el protocolo de transmisión, ayuda a prevenir las colisiones y la interfaz aire con la capa de enlace. Las principales características de ISO 15693 son:

- Define la estructura física de la etiqueta y protocolos de transmisión.
- Trabaja en la frecuencia de 13.56Mhz
- Puede operar el lector y la etiqueta para lectura y escritura hasta un metro de distancia.
- Evita colisiones de la información predeterminando una velocidad de 26 Kbps.
- Habitualmente el control de acceso utiliza este tipo de tarjetas.

2.4.1.4 Norma ISO 14443 (Proximity Cards) (Gómez, 2013).

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 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.

NFC Forum también ha completado 16 normas que permitan a los dispositivos NFC funcionar con todas las tarjetas sin contacto y NFC, a continuación se presenta un esquema detallado de la estandarización de la tecnología NFC.

Esquemas de la estandarización NFC.

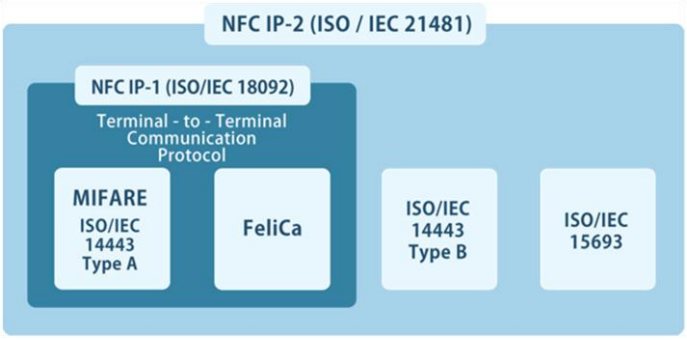


Figura 11. (TOOPAN FORMS CO, 2012).

NFC Forum creó un formato común para la trasmisión de datos entre los dispositivos y etiquetas NFC llamado NDEF.

2.4.2 Formato de intercambio de datos NFC (NDEF)

NFC fórum ha creado un formato estandarizado NDEF (Data Exchange Format) el cual permite almacenar y trasportar diferente información, desde mensajes Multipurpose Internet Mail Extensions *"MIME"* (Extensiones de Correo de Internet de Propósitos Múltiples) hasta documentos Record Type Definition *"RTD"* (Definición del Tipo de Registro) y de menor tamaño como Uniform Resource Locator *"URLs"* (Localizador de Recursos Uniforme) (Forum, 2004).

El formato NDEF permite el encapsulamiento de los mensajes en el intercambio de datos entre dispositivos o a una etiqueta NFC, define la arquitectura y el registro ordenado de los mensajes NDEF. Al poseer una etiqueta NFC esta no permite una interacción con el usuario ya que es un elemento pasivo no podría mostrar ninguna información al usuario no genera su propia energía por lo que necesitaría de un dispositivo activo para que funcione, el cual genera su propia energía y tiene una interacción con el usuario, el campo magnético generado por el dispositivo permite el funcionamiento de los elementos pasivos.

NDEF posee una estructura en donde se puede enviar uno o más payload o información útil para el usuario, de diferente tipo y tamaño encapsulados en un solo mensaje documentos XLM hasta imágenes en formato JPEG, etc. El payload está compuesto por una, longitud e identificador opcional (Rubén Abuín, 2014).

- **Tipo de Payload.-** Específica la clase de información transportada, al saber el tipo de información o payload se la puede despachar para la aplicación apropiada. Los tipos de identificadores podrían ser URIs, MINE o tíPICos específicos NFC (NFC-specific).
- **Longitud de la carga *(payload).-*** Indican el tamaño o longitud del payload encapsulados en un mensaje. El campo PAYLOAD_LENGTH es un octeto para registros pequeños. Los registros pequeños están indicados establecidos por el bit de bandera Short Record (SR) en 1.
- **Identificador payload.-** Entrega información en forma URI para identificar la carga para la aplicación del usuario apropiada.

Los registros NDEF son de longitud variable pero todos tienen un formato común que se representa en la figura 12 que se describe a continuación.

Formato de un registro NDFE

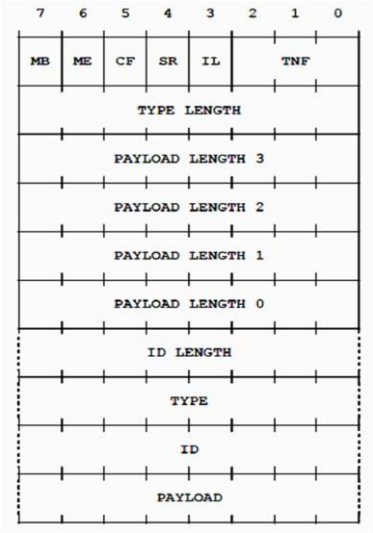


Figura 12. (Chérrez, 2010).

La información de los registros NDFE se presenta en octetos la transmisión es de izquierda a derecha y de arriba hacia abajo por lo que es el bit más significativo es el bit del extremo izquierdo.

2.4.2.1 Obtención de información de etiquetas NFC

Son dispositivos pasivos, pequeños que contienen información, principalmente están compuestos por espirales metálicas a las cuales se adjuntan componentes de memoria y comunicación. Por su diseño son fáciles para presentar como tarjetas de visitas, llaveros, pulseras, stickers.

Su funcionamiento es semejante a los códigos de barras y QR (Quick Response), sin el reconocimiento óptico del código. Las etiquetas o elementos pasivos se activan y transmiten la información almacenada en su interior al acercarlo a un dispositivo activo el cual genera el campo de radiofrecuencia este dispositivo puede ser un teléfono inteligente, un lector NFC iniciando el proceso de comunicación.

La información que se adquiera dependerá del tipo de etiqueta NFC, ya que cumplen varios tipos de funciones y capacidades como memoria, su tasa de transferencia de datos y los modos de interacción por lo que fueron estandarizados por NFC fórum y están detallados a continuación en la figura 13.

Tabla 1

Tipos de etiquetas NFC

Tipo	Estándar	Modos	Memoria	Velocidad
Tipo 1	ISO 1443 Tipo A	Solo lectura Lectura / Escritura	96 bytes	106 kbits/seg
Tipo 2	ISO 1443 Tipo A	Solo lectura Lectura / Escritura	48 bytes	106 kbits/seg
Tipo 3	Sony - Felica	Solo lectura	2 Kbytes	212 kbits/seg
Tipo 4	ISO 1443 Tipo A y B	Solo lectura Lectura / Escritura	32 Kbytes	106 kbits/seg 424 kbits/seg

Nota. (Forum, 2004).

Elaborado por: Darwin Cacuango y Eduardo Zapata.

2.4.2.2 RTD, Definición de tipo de registro

RTD (Record Type Definition) da las pautas para los tipos de registro NFC que se incluyen en los mensajes NDEF que se transmiten entre los dispositivos y etiqueta NFC. Esto permite realizar aplicaciones específicas NFC.

Tabla 2.

Evolución de estandarización NFC

FECHA	ESTANDARIZACIÓN DE LA TECNOLOGÍA NFC
Dic-2002	El estándar NFC fue registrado como ECMA 340
Dic-2003	Internacionalmente se acreditó el estándar ISO/IEC 18092 (NFCIP-1)
Ene-2005	Internacionalmente se acreditó el estándar ISO/IEC 21481 (NFCIP-2)
Dic-2005	Estándar ECMA-356 "Métodos de prueba del protocolo NFC IP-1"
Jun-2006	Estándar ECMA-373 "Interfaces cableadas para NFC (NFC-WI)"
Jun-2010	Estándar ECMA-385 "Protocolos y Servicios de Seguridad NFCIP-1"
Jun-2010	Estándar ECMA-386 "NFC-SEC-01: Criptografía NFC-SEC usando ECDH y AES"

Nota. (Forum, 2004)

2.5 Aplicaciones de la tecnología NFC

La tecnología NFC se implementado inicialmente en los teléfonos móviles, desarrollando aplicaciones capaces de ejecutar acciones previamente configuradas siempre que verifiquen el código de una etiquetas NFC.

El objetivo de todas las tecnologías incluyendo NFC es lograr mejorar aspectos de nuestra vida cotidiana, algunos de ellos relacionados con la automatización de tareas, otros con las transacciones electrónicas y con la identificación segura.

NFC se caracteriza por su interfaz intuitiva facilitando su uso, a pesar de esta ventaja aún no se ha logrado una masificación de la tecnología pero ha despertado un gran interés en las grandes empresas para invertir en el desarrollo de la misma.

Mientras la tecnología NFC siga en desarrollo con el tiempo se estará en capacidad de explotar su potencialidad y ampliar su campo de aplicación como:

- Ventas de entradas electrónicas: entradas de aerolíneas, entradas de eventos/conciertos y otros.
- Dinero electrónico

- Llaves electrónicas: llaves de carros, llaves de casa/oficina, llaves de cuartos de hotel, etc.
- Identificación de documentos
- NFC puede ser usado para configurar e iniciar otras conexiones de red.

También NFC estará en capacidad de interactuar con un sin número de dispositivos electrónicos como cámaras, televisores, máquinas expendedoras, computadores, etc.

Uno de los pioneros en el uso de la tecnología es el sector del transporte, a continuación se menciona algunos proyectos NFC desarrollados internacionalmente.

En diciembre del 2008 la aplicación eCLOWN fue publicada, la cual permite leer y copiar el contenido del chip de pasaportes biométricos.

La empresa Motorola está realizando pruebas acerca de las seguridades en los teléfonos con tecnología NFC para que tengan la capacidad de guardar la confidencialidad de datos bancarios con el objetivo de realizar transacciones financieras seguras con su uso.

La empresa de transportes TransSys, Visa Europe y Nokia son algunas de las empresas creadoras de un proyecto piloto en la ciudad de Londres capaz de realizar pagos de billetes de viaje mediante tecnología NFC.

Otro proyecto es desarrollado por la empresa Metropolitana de Transportes de Málaga en unión con la operadora Orange, Mobipay, Indra y Oberthur, que consiste en el pago de pasajes de autobús mediante el móvil con tecnología NFC además ofrece una aplicación que informa el número de viajes restantes.

Un proyecto ya hecho realidad es conocido como Smart Poster, teniendo como finalidad la comercialización de música a través de NFC, esta aplicación fue desarrollada en conjunto por Visa y Universal Music con el fin de facilitar la venta de pistas musicales aprovechando los Smart Poster.

CAPÍTULO 3

SISTEMA DE COMUNICACIÓN NFC

3.1 Arquitectura de la comunicación NFC

La arquitectura de la tecnología NFC posee características similares a la tecnología RFID, la tecnología NFC es sólida y única debido a que trabaja en tres tipos de configuraciones haciendo de esta más eficiente que otras.

Arquitectura dispositivo NFC

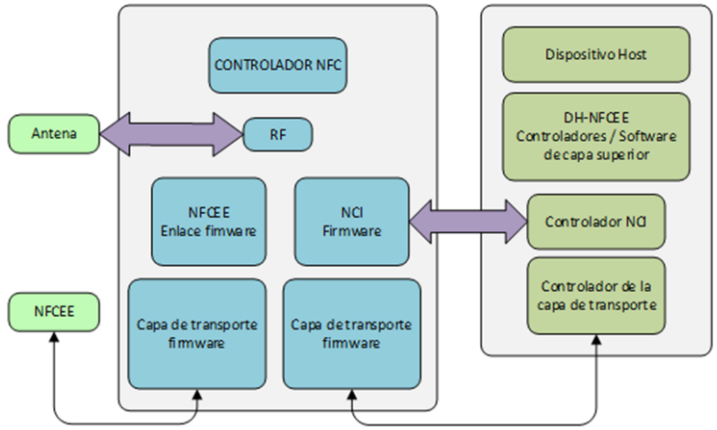


Figura 13. (Forum, 2004).

Elaborado por: Darwin Cacuango y Eduardo Zapata.

Como se muestra en la figura 13 el segmento principal es el Controlador NFC contiene el procesador principal del dispositivo, a este segmento se conecta el Dispositivo Host que contiene uno o más ambientes de ejecución NFC, cuando uno o más entornos de ejecución NFC son integrados o conectados al NFCC se los conocen como NFCEEs (NFC Execution Environment).

La comunicación entre el Controlador NFC y Dispositivo Host está definida por NCI (NFC Controller Interface), el cual está encargado de definir el formato de los datos que se intercambian en la comunicación.

En el segmento Controlador NFC se encuentran todos los bloques de instrucciones que se conectan mediante el NCI al Dispositivo Host que es el segmento en donde se ejecutan las instrucciones NFC.

3.1.1 Arquitectura del NCI

La interfaz controladora NFC (NCI), tiene la función de establecer la comunicación entre los segmentos NFC Controller y Device Host, esta es una interfaz estándar que facilita la integración de dispositivos NFC de diferentes fabricantes y también facilita usar controladores de NFC con diferentes procesadores de aplicaciones y diferentes pilas de software NFC (Forum, 2004).

El NCI proporciona a los usuarios una interfaz lógica que se puede utilizar con diferentes medios de transporte físicos, tales como UART, SPI, y I2C, protocolos de comunicación comúnmente usados.

3.2.1 Tipos de comunicación NFC

Dependiendo de la modalidad de operación, la tecnología NFC presenta el siguiente diseño de arquitectura mostrado en la figura14.

En donde la capa “RF Layer” es común para los modos de comunicación de NFC, en esta capa se establece los estándares, especificaciones analógicas, protocolos digitales en las que trabaja la comunicación NFC debidamente establecidas en los estándares ISO 18092, 14443. (Forum, 2004).

La tecnología NFC puede trabajar en tres configuraciones diferentes como se muestra en la figura:

- Modo de comunicación Peer-to-Peer
- Modo lectura / escritura
- Modo emulación de tarjeta inteligente NFC

Modos de operación dispositivos NFC

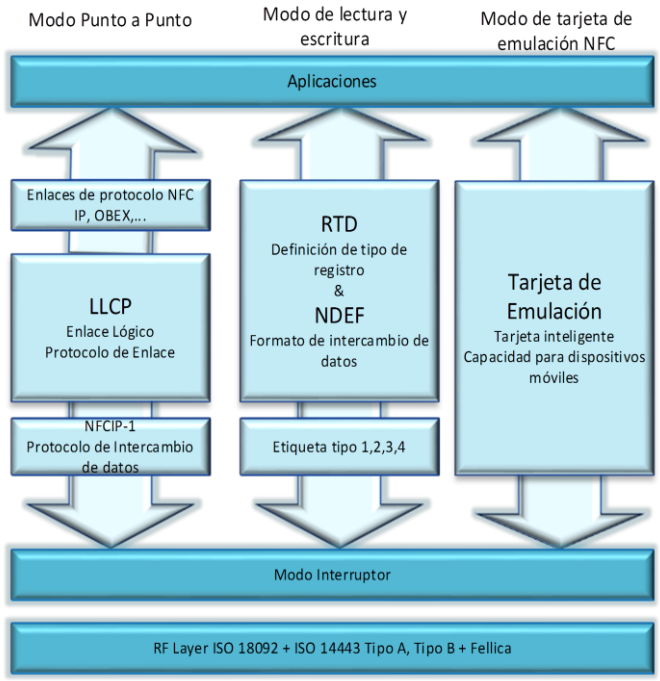


Figura 14. **(Forum, 2004)**.

Elaborado por: Darwin Cacuango y Eduardo Zapata.

3.2.1 Modo de comunicación Peer-to-Peer

El modo Peer-to-Peer (Punto a Punto), es el modo clásico de comunicación NFC en donde se establece una conexión bidireccional de datos entre dispositivos, con una velocidad de 424kBit/seg aproximadamente (RapidNFC, 2011).

Elementos protocolo Peer to Peer (NFC)

<table>
<tr><td colspan="2">Aplicaciones</td></tr>
<tr><td rowspan="2">Protocolo de intercambio NDEF y otros protocolos</td><td>Protocolos registrados NFC</td></tr>
<tr><td>Protocolo de enlace</td></tr>
<tr><td colspan="2">Enlace lógico de protocolo de control (LLCP)</td></tr>
<tr><td colspan="2">Protocolo digital</td></tr>
<tr><td colspan="2">Análogo</td></tr>
</table>

Figura 15. (Vedat, Ok, & Ozdenizci, February 2012).

Elaborado por: Darwin Cacuango y Eduardo Zapata.

En la figura 15, se observa la estructura de elementos de protocolos de un dispositivo NFC que se encuentra operando en modo Peer-to-Peer (Punto-Punto),

en donde se nota que NFC utiliza protocolos análogos y digitales estandarizados por NFCIP-1.

NFC utiliza a nivel de la capa de enlace el protocolo de control de enlace lógico (LLCP), el mismo que es usado para la activación, supervisión y desactivación de la comunicación. El modo de transferencia se lo hace de modo asincrónico balanceado, es decir que cualquier dispositivo puede iniciar la transmisión, supervisar y enviar información en cualquier momento.

El protocolo de intercambio NDEF se utiliza para enviar mensajes con el formato NDEF en el modo Peer-to-Peer, al igual que en las especificaciones de operación de los tipos de etiquetas NFC.

El protocolo de enlace proporciona enlaces estándar para protocolos NFC registrados y permite su uso interoperable.

Los protocolos NFC registrados son aquellos que el Foro NFC define un enlace para el protocolo de control de enlace lógico, por ejemplo IP, OBEX27, entre otras.

Las aplicaciones en modo Peer-to-Peer son de referencia y están establecidas por el NFC Fórum que pueden ser ejecutadas a través del protocolo de intercambio simple NDEF como por ejemplo imprimir desde una cámara, intercambiar imágenes entre dos celulares, etc.

3.2.2 Modo lectura / escritura

En el modo lectura/escritura el dispositivo NFC puede leer los cuatro tipos de Etiquetas definidos en NFC Forum, Cuando se establece esta configuración los dispositivos NFC pueden intercambiar pequeñas cantidades de información como por ejemplo información de texto, una dirección web o un número telefónico. Este modo tiene compatibilidad de RF con la ISO/IEC 14443 y FeliCa.

En la actualidad este modo de comunicación es el más usado, en donde el dispositivo NFC se encuentra en modo activo y lee una etiqueta RFID pasivo, intercambiando la información entre dispositivos.

Elementos protocolo lectura/escritura (NFC)

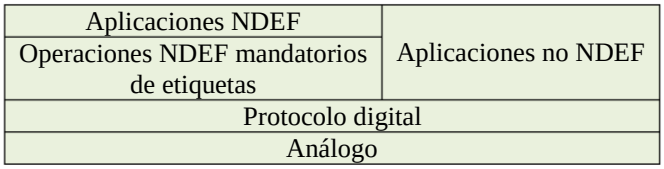


Figura 16. (Vedat, Ok, & Ozdenizci, February 2012).

Elaborado por: Darwin Cacuango y Eduardo Zapata.

La figura 16, muestra la estructura de elementos de protocolos de un dispositivo NFC que se encuentre operando en modo lectura/escritura, se observa que NFC utiliza protocolos análogos y digitales en la capa inferior, donde el protocolo análogo determina el rango de radio frecuencia operable de los dispositivos NFC. Los protocolos digitales refieren a los aspectos digitales de la comunicación NFC debidamente establecidas en los estándares ISO/IEC 18092, 14443 (Forum, 2004).

La operación de etiquetas son comandos e instrucciones que se deben seguir para habilitar a dispositivos activos NFC capaces de realizar operaciones de escritura y lectura sobre etiquetas NFC.

Las aplicaciones NDEF se basan en especificaciones NDEF que es el formato de datos normalizados para el intercambio de información entre dispositivos NFC a diferencia de las aplicaciones no NDEF que se definen por especificaciones propias de los creadores de las mismas.

3.5 Modo emulación de tarjeta inteligente NFC

El modo emulación de tarjeta el dispositivo NFC puede emular el comportamiento y propiedades de una tarjeta inteligente definida con el estándar ISO/IEC 14443. El lector NFC no puede distinguir entre un dispositivo que esté operando en modo emulación o una tarjeta ordinaria lo que implica una ventaja debido a que la

estructura existente para tarjetas inteligentes puede ser aprovechada por la tecnología NFC sin tener que reemplazarla (Forum, 2004).

En este modo el dispositivo NFC en modo emulación no genera su campo de radio frecuencia comportándose como una etiqueta estándar, esperando el campo del lector para iniciar la comunicación, también se puede utilizar las características de seguridad avanzadas, siendo útil para transacciones bancarias, gestión de entradas, en general para las gestiones de pagos rápidos, control de accesos, etc.

La estructura de elementos de protocolo del modo de emulación de tarjeta son:

- Aplicaciones
- Protocolo Digital
- Análogo

Los dispositivos que operan en este modo usan protocolos similares al de tarjetas inteligentes ya que son compatibles con los mismos estándares.

3.3 Establecimiento de la comunicación NFC

La comunicación o transacción NFC sigue una misma secuencia de operación que consta de los siguientes pasos: descubrimiento de dispositivos NFC, autenticación, negociación, transferencia de información y confirmación. El protocolo NFC incluye un procedimiento para la autenticación segura y mecanismos anti-colisión para evitar la intercepción del canal de comunicación.

Pasos Transacción NFC

Figura 17.

Elaborado por: Darwin Cacuango y Eduardo Zapata.

Los pasos de una transacción NFC que se indican en la figura 17, tienen una función específica y siempre están presentes, a continuación se detalla cada uno de ellos:

1. Descubrimiento: en esta fase los dispositivos inician la etapa de exploración entre dispositivos.
2. Autenticación: en esta parte los dispositivos verifican si el otro dispositivo está autorizado o si deben establecer algún tipo de cifrado para la comunicación.
3. Negociación: los dispositivos definen parámetros como la velocidad de transmisión, la identificación del dispositivo, el tipo de aplicación, su tamaño, y la acción a solicitada.
4. Transferencia: una vez negociados los parámetros para la comunicación, se puede realizar el intercambio de datos.
5. Confirmación: el dispositivo receptor confirma el establecimiento de la comunicación y la transferencia de datos (Forum, 2004).

Cabe destacar que la tecnología NFC reduce el tiempo de establecimiento de la comunicación en relación a otras tecnologías inalámbricas para efectuar el enlace.

3.3.1 Protocolo (SPI)

El estándar SPI (Syncronous Peripheral Interface) es utilizado para la comunicación serial entre dispositivos. El SPI fue inicialmente creado por Motorola y adoptado posteriormente por diferentes fabricantes, como Microchip y Atmel. Se trata de un enlace de datos en serie, síncrono, y que opera en modo full dúplex, es decir, las señales de datos viajan en ambas direcciones en forma simultánea.

El canal SPI fue diseñado para aplicaciones de trasmisión de datos a velocidades altas (10 Mbps) y distancias cortas, del orden de 10 a 20 cm, o bien dentro de un mismo PCB (circuito impreso), entre 2 dispositivos, por ejemplo, un microcontrolador y otro dispositivo electrónico cualquiera. Las señales de trasmisión de datos y control del canal SPI, usan niveles de voltaje TTL o bien 3.3 volts, dependiendo de la tecnología de fabricación del dispositivo.

Los dispositivos SPI se comunican entre sí utilizando un bus de 4 señales:

- Master Output Slave Input (MOSI)
- Master Input Slave Output (MISO)
- Signal Clock (SCK)
- Slave Select (SS)

El protocolo SPI también usa un esquema maestro/esclavo, en el cual el maestro inicia el protocolo de trasmisión de los datos. En ocasiones, las interfaces SPI son circuitos que están disponibles como parte del hardware en los microcontroladores o en dispositivos como módulos lectores RFID en este caso el lector usado en el prototipo.

Es posible implementar una comunicación SPI, utilizando 4 bits de entrada/salida de un microcontrolador junto con un firmware adecuado que maneje el protocolo SPI. Las señales denominadas MOSI y MISO son portadoras de los datos en ambas direcciones mientras la señal SCK es la señal de reloj que sincroniza la recepción de los datos.

La señal SS habilita el esclavo correspondiente. Sin embargo, añadiendo varias líneas SS, puede implementar una red de varios circuitos SPI, controlados por el mismo dispositivo Master.

A continuación se muestra los tipos de conexión del protocolo SPI figura 18, figura 19.

Conexión Maestro-Esclavo

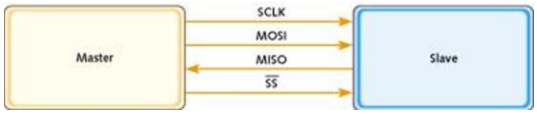


Figura 18. (RapidNFC, 2011).

Conexión Maestro-Esclavo (Varios)

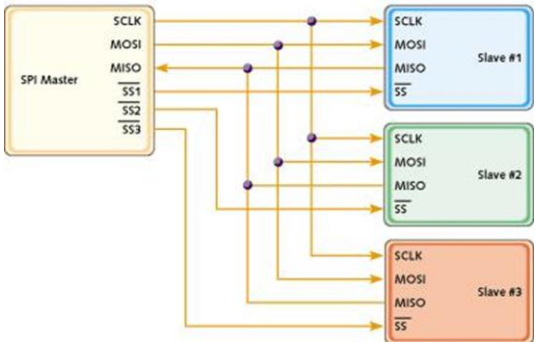


Figura 19. **(**RapidNFC**,** 2011**)**.

Nótese en la figura 19 cada esclavo tiene su propia línea de selección de esclavo (SS) para permitir la comunicación de uno a la vez.

Cuando el maestro selecciona un dispositivo esclavo y genera una señal de reloj (SCK), los datos pueden fluir en ambas direcciones simultáneamente (full dúplex), ya que el mismo reloj funciona para los 2 dispositivos, maestro y esclavo.

SPI no especifica un protocolo de alto nivel para el diálogo maestro-esclavo, y no cuenta con un mecanismo de hardware para la confirmación (acknowledge) o validación de la recepción de los datos.

Tratándose de un canal de trasmisión síncrono, la velocidad de trasmisión de SPI depende de la frecuencia de generación del reloj (señal SCK), por lo que puede tomar cualquier valor. No existen velocidades estándar de trasmisión, como en el caso del RS-232. La velocidad máxima típica de SPI puede llegar a 10 Mbps.

Señales SPI

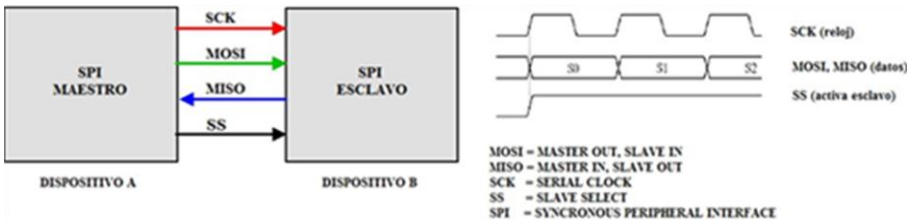


Figura 20. (PIC, 2012)

3.3.2 Comunicación RS-485

Cuando se necesita transmitir a largas distancias y altas velocidades mayores que RS-232 la comunicación RS-485 es la solución, al usar enlaces con RS-485 no hay limitación de conexión de dispositivos es decir, dependiendo de la distancia, velocidad de transmisión se pueden conectar hasta 32 nodos con un simple par de cables.

Las ventajas que brinda este protocolo en comparación al RS-232 se puede mencionar:

a) Bajo costo.- Los Circuitos Integrados para trasmitir y recibir son baratos y solo requieren una fuente de +5V para poder generar una diferencia mínima de 1.5v entre las salidas diferenciales.
b) Capacidad de interconexión.- RS-485 es una interface multi-enlace con la capacidad de tener múltiples transmisores y receptores.
c) Longitud de Enlace.- En un enlace RS-485 puede tener hasta 1200 metros de longitud, comparado con RS-232 que tiene unos límites típicos de 15 metros.
d) Rapidez.- La velocidad de comunicación puede ser como 10 Mega bits/ segundo.

La razón por la que RS-485 puede transmitir a largas distancias, es porque utiliza el balanceo de líneas. Cada señal tiene un par de cables, sobre uno de ellos se encontrará un voltaje y en el otro se estará su complemento, de esta forma, el receptor responde a la diferencia entre voltajes figura 21.

Balanceo de líneas RS-485

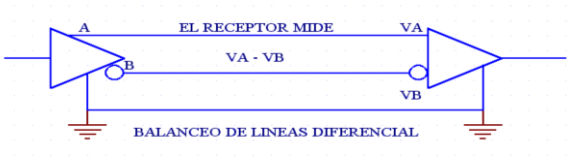


Figura 21. (Pérez, 2014) .

El margen de ruido es menor que el de un enlace RS-232, no hay que olvidar que RS-485 maneja señales diferenciales y que cancela la mayoría del ruido a través de su enlace.

La interfaz RS-485 puede ser cableada de dos formas: con dos cables o con cuatro cables. El modo de conexión mediante dos cables no permite comunicación full duplex, y requiere que los datos sean transferidos en un solo sentido cada vez. Para operaciones half duplex, los dos pines de transmisión deben estar conectados a los dos pines de recepción (TD+ a RD+ y TD- a RD-). El modo de conexión mediante cuatro cables permite la transferencia de datos full dúplex figura 22.

Tipo de comunicación RS-485

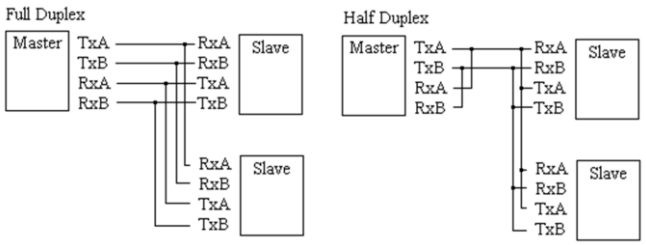


Figura 22. (Educativa, 2014) .

3.4.1 Aspectos de seguridad NFC

La tecnología NFC es considerada segura por su rango de alcance para establecer comunicación que se limita a pocos centímetros, pero NFC por sí sola no garantiza comunicaciones seguras.

NFC no está en capacidad de ofrecer seguridad contra terceros, es vulnerable a la modificación de datos. Las aplicaciones son las encargadas de establecer un canal seguro para la comunicación esto se da mediante el uso de protocolos criptográficos.

La ventaja en seguridad NFC es debido a su corta distancia de operación en la comunicación, y para poder robar información durante la comunicación se debe estar dentro del rango del establecimiento de la comunicación.

Se debe aclarar que actualmente las aplicaciones relacionadas con este tipo de tecnología están en etapas de prueba hasta poder implementar de forma masiva, a partir de esto se presenta problemas de seguridad aprovechando las debilidades de la tecnología como de sus aplicaciones.

El análisis de seguridad NFC solo se basan en estudios realizados por investigadores quienes han determinado que existen varias áreas importantes para la seguridad de las comunicaciones de campo cercano, algunas de las principales áreas son:

- EavesdropPing (Espionaje, Interceptación)
- Data corruption (Corrupción de datos)
- Data modification (Modificación de datos)
- Man-in-middle attack (Ataques de intermediario)

3.4.1 EavesdropPing (Interceptación)

El EavesdropPing o interceptación es un tipo de ataque en donde se trata de interceptar una comunicación con el fin de adquirir un porcentaje de información y poder usarla con fines destructivos.

Este ataque se presenta en toda comunicación inalámbrica debido a que utilizan ondas de radio para el intercambio de información y al propagarse por el aire están expuestas a que usuarios no deseados puedan recolectar información de estas señales.

Para interceptar una comunicación el atacante debe colocar una antena intermedia que escuche y reciba la señal de radio frecuencia generada por la transferencia inalámbrica de información entre dos dispositivos NFC, se cree que un ataque de este tipo involucra mayor dificultad por la distancia de transferencia entre emisor y receptor que oscila entre los 10 centímetros.

3.4.2 Data corruption (corrupción de datos)

Este problema de seguridad en NFC es un ataque de prohibición de servicio, en donde el atacante no se limita a escuchar la transmisión sino que la distorsiona con el objetivo de destruir la información mediante el envío de señales de radio frecuencia que generan ruido e interferencia en la transmisión original

Es posible que los dispositivos NFC puedan evitar este tipo de ataques, ya que son capaces de detectar señales de radio frecuencia antes de enviar datos y si llegase a detectar una señal mayor a la de emisión de datos puede ser cancelado previniendo el ataque.

Para este caso, el atacante necesitaría transmitir datos en la misma frecuencia y un momento determinado, el cual es calculable si se conoce bien la modulación y codificación.

3.4.3 Data modification (modificación de datos)

Este tipo de ataque no se enfoca en impedir la comunicación sino que el atacante trata de modificar los datos enviados, para esto el atacante intercepta y manipula los datos, haciendo una modificacion parcial o total de los valores binarios de los datos y luego proceder a enviarlos al receptor, el éxito que tenga el atacante dependerá de la exactitud de la modulación de la señal que emita.

3.4.4 Man-in-the-middle attack (Ataques de intermediario)

En este tipo de ataque la comunicación es interceptada por un tercero entre los dos dispositivos NFC con el fin de almacenar o manipular los datos sin que los extremos de comunicación tengan conocimiento del ataque figura 23.

Ataque Hombre-en-medio

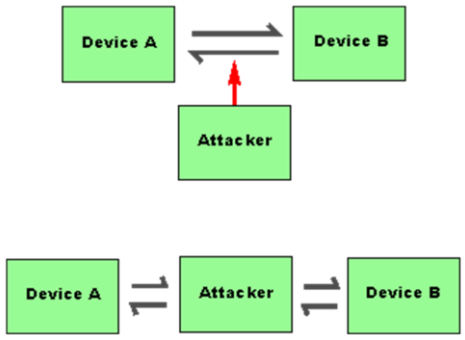


Figura 23. Tipos de ataque de intermediario

Fuente: (Vedat, Ok, & Ozdenizci, February 2012).

Este tipo de ataque se considera difícil de conseguir en una comunicación NFC, una recomendación para minimizar el riesgo por completo, utilizar un modo de comunicación activo-pasivo. De esta manera sería posible escuchar y detectar cualquier usuario no deseado o permitido.

3.5 Lector NFC (Módulo RFID RC522)

El módulo el lector consta de un chip MFRC522 es un CI de lectura/escritura de comunicación sin contacto a 13,56 MHz.

Módulo RFID RC522.

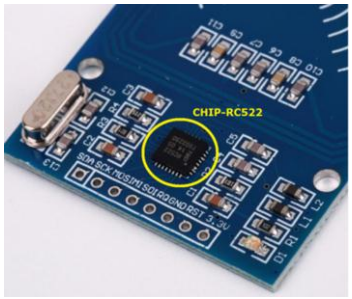


Figura 24. (Arduino, 2015).

En la figura 24, se indica el chip del lector NFC, este controla una antena lectora/escritora diseñada para comunicarse con tarjetas y llaveros bajo la norma ISO/IEC 14443A/MIFARE. El módulo receptor provee una implementación sólida y eficiente para la demodulación y decodificación de señales de tarjetas y llaveros compatibles

El MFRC522 admite la comunicación sin contacto y utiliza velocidades de transferencia de hasta 848 kBd en ambas direcciones (Forum, 2004). A continuación, se describe las características principales del lector:

- Circuitos analógicos integrados para demodular y decodificar respuestas.
- Compatible con ISO/IEC 14443 A/MIFARE
- Distancia de operación en modo de Lectura/Escritura de hasta 50 mm, dependiendo del tamaño de la antena.
- Comunicación a una velocidad de transferencia de hasta 848 kBd.
- SPI de hasta 10 Mbit/s
- Interfaz bus I²C hasta 400 kBd en modo rápido, y hasta 3400 kBd en modo alta velocidad
- UART serie RS232 de hasta 1228,8 kBd, con niveles de voltaje dependientes del suministro de voltaje.
- Una memoria FIFO administra el envío y recepción de 64 bytes
- Modos de interrupción flexibles
- Modo de apagado mediante software
- Temporizador programable
- Oscilador interno para conexión a cristal de cuarzo de 27,12 MHz
- Suministro eléctrico de 2,5 a 3,3 V (Netherlands, 2006-2015).

CAPÍTULO 4

DISEÑO E IMPLEMENTACIÓN DEL PROTOTIPO

En este capítulo se describe el diseño del prototipo de control de acceso multinivel basado en tecnología NFC, que tiene la finalidad de realizar el control automático para el acceso de personas autorizadas, dependiendo de los privilegios del usuario, se podrán limitar a ciertas instalaciones, teniendo así un control de la infraestructura, reduciendo la vulnerabilidad de áreas restringidas.

4.1 Descripción del sistema prototipo

El prototipo diseñado para este proyecto se representa en la figura 25, consta de siete accesos, cada uno de estos está formado por circuitos o placas independientes, donde se tiene un lector NFC conectado a un microcontrolador que se encarga de recoger y transmitir la información adquirida, para luego asociarla a través de un concentrador en donde se interconectan todas las placas para enviar dicha información a un computador desde donde se administra y controla dichos accesos, además de adquirir información y visualizar el registro de usuarios de la red.

Diagrama general del prototipo

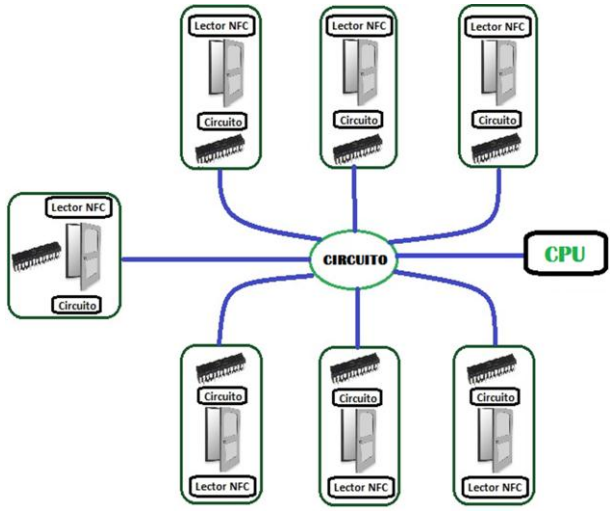


Figura 25.

Elaborado por: Darwin Cacuango y Eduardo Zapata.

Cada uno de los siete accesos están formados por un lector NFC encargado de receptar los datos de la comunicación, funciona conjuntamente con un microcontrolador en donde se procesa toda esta información para posteriormente ser enviada y manipulada según los requerimientos del proyecto, para la comunicación entre el lector y el microcontrolador se usa una comunicación SPI (Syncronous Peripheral Interface) entre dispositivos, se trata de un enlace de datos en serie, síncrono, que opera en modo full dúplex, es decir, las señales de datos viajan en ambas direcciones en forma simultánea.

A partir de que el lector NFC y el microcontrolador poseen la información de la tarjeta o etiqueta NFC, se envía dicha información a un concentrador donde se conectan todos los accesos del prototipo y este pueda re direccionarla a un computador en donde se manejará y administrará esta información logrando visualizar y controlar los registros de acceso del prototipo mediante una aplicación desarrollada para este control.

Diagrama simplificado del prototipo

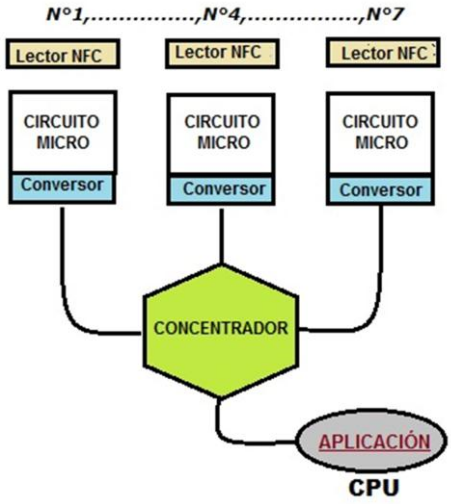


Figura 26.
Elaborado por: Darwin Cacuango y Eduardo Zapata.

En la figura 26, se resume el funcionamiento del prototipo donde existe un conversor encargado de transformar las señales provenientes del microcontrolador (TTL), a niveles lógicos RS-485, todas las placas se interconectan a un concentrador mediante cables RJ-11 para procesar las señales de los mismos

mediante un conversor RS-485 a RS-232 comunicando todos los dispositivos a una aplicación desarrollada para computador mediante cable USB enlazando el hardware y software del prototipo.

4.2 Diseño del prototipo

En este punto se describe el diseño del prototipo donde se tiene varios accesos y niveles de seguridad para verificar la flexibilidad del proyecto.

Dimensiones del prototipo

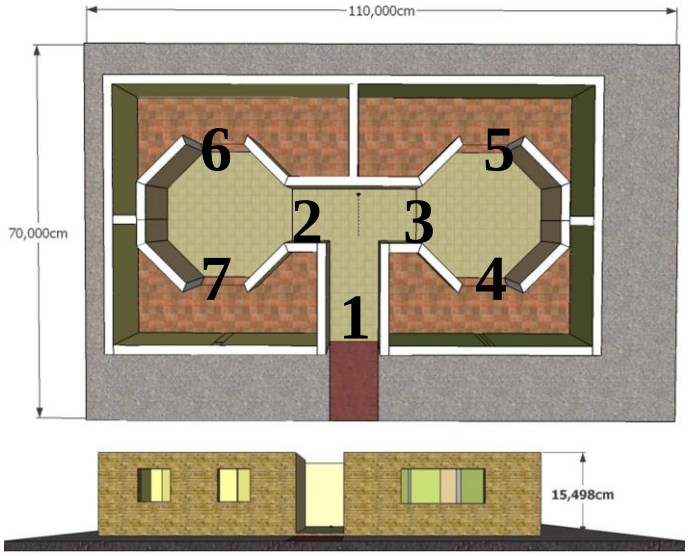


Figura 27.

Elaborado por: Darwin Cacuango y Eduardo Zapata.

En la figura 27, se tiene la vista superior y frontal del prototipo con sus dimensiones y diseño arquitectónico, donde se puede apreciar los accesos de seguridad a implementar, el plano de dimensiones del prototipo se detalla en la sección anexos.

En la construcción del prototipo las paredes que forman la estructura van a tener un ancho de 1,5 [cm] aproximadamente, esta medida tiene como objetivo ubicar los lectores NFC entre las paredes de la estructura y guiar el cableado necesario para el funcionamiento de los lectores y cerraduras magnéticas.

Se procede a relacionar los accesos para obtener áreas de restricción, es decir, existirán áreas en donde no se pueda acceder sin contar con el nivel de seguridad requerido, para la implementación del sistema se creó tres áreas o niveles de seguridad como se muestra en la figura 28.

Áreas de seguridad

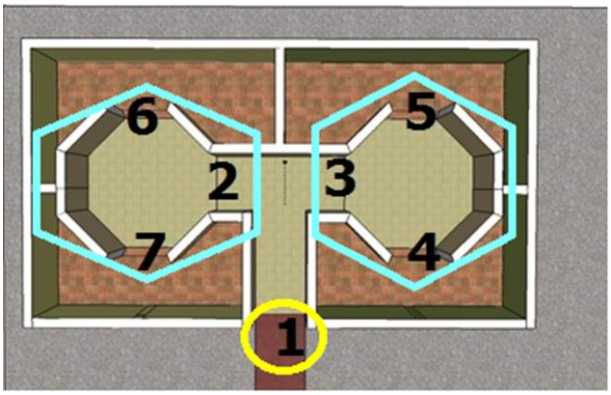


Figura 28.

Elaborado por: Darwin Cacuango y Eduardo Zapata.

Cada acceso enumerado en la figura 28, está formado por un lector NFC y un circuito electrónico controlando así cada uno de los accesos del prototipo. Todos estos circuitos van a agruparse en un cuarto de control donde se tiene toda la parte electrónica del modelo de prueba como se muestra en la figura 29.

La figura 29, se muestra el lugar desde donde está ubicado los controles de los accesos del sistema, aquí se ubican los circuitos o placas elaboradas de cada lector y el concentrador para su posterior comunicación con la aplicación de computador.

En este segmento del modelo de prueba se tiene todas las conexiones eléctricas y electrónicas, cada una referenciada con su respectiva etiqueta para ser interpretada por los usuarios.

Área de control del prototipo

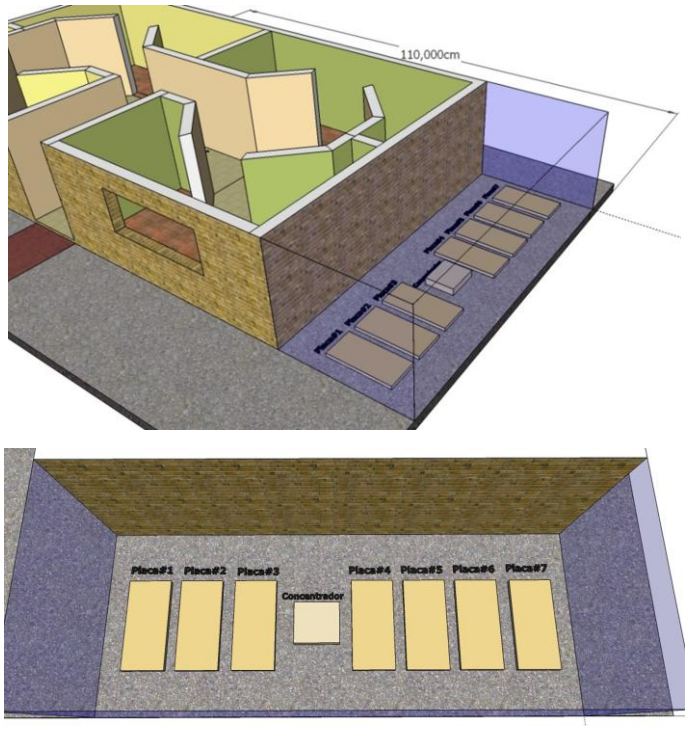


Figura 29.

Elaborado por: Darwin Cacuango y Eduardo Zapata.

4.2.1 Diseño electrónico del prototipo.

El diseño electrónico del prototipo consta de tres partes cada una de ellas de vital importancia debido a que en conjunto logran el correcto funcionamiento del sistema.

- Comunicación SPI
- Conversión
- Etapa de control de potencia

4.2.1.1 Comunicación SPI

La comunicación SPI se da por el microcontrolador PIC16F628a y el lector NFC "RFID MFRC522", en donde van realizan la función de maestro y esclavo respectivamente.

Diagrama conexión Lector-PIC

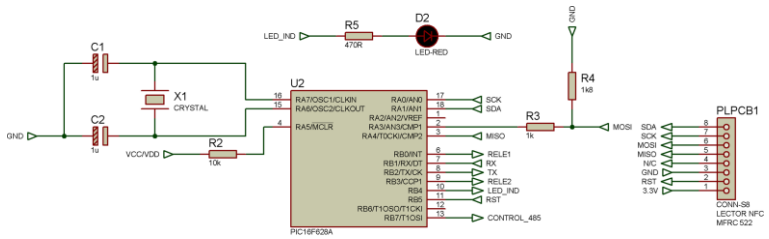


Figura 30.

Elaborado por: Darwin Cacuango y Eduardo Zapata.

En la figura 30, se muestra el diagrama de conexión de la comunicación SPI maestro esclavo, el lector NFC está representado por un conector de 8 Pines PLPCB que a su vez está conectado con el microcontrolador, la trasmisión de datos y control del canal SPI. El detalle de características del programa usado para la comunicación SPI se muestra en la sección anexos.

El microcontrolador funciona con un voltaje de 5 [Vdc], mientras que el lector funciona con 3.3 [Vdc] para obtener estos voltajes exactos se elaboró un circuito adjunto al del control SPI en donde se muestra dichos voltajes.

Diagrama fuente reguladora voltaje

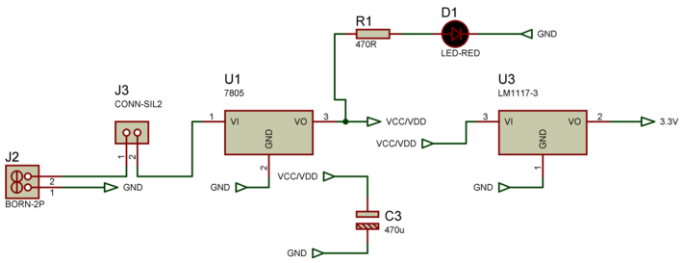


Figura 31.

Elaborado por: Darwin Cacuango y Eduardo Zapata.

En la figura 31, se muestra el circuito realizado para obtener el voltaje exacto que alimenta cada dispositivo para evitar daños.

4.2.1.2 Conversor

Se ha implementado el sistema de conversión RS-232 a Rs-485 para la comunicación entre dispositivos debido a que en una red de múltiples accesos la comunicación entre dispositivos es bastante significativa.

A continuación en la figura 32, se detalla la conexión del circuito integrado SN75176 el cual recibe la señal del microcontrolador Pin (13), estas señales son de tipo TTL y el circuito integrado transforma esta señal o dato a niveles lógicos RS-485.

Diagrama control RS-485

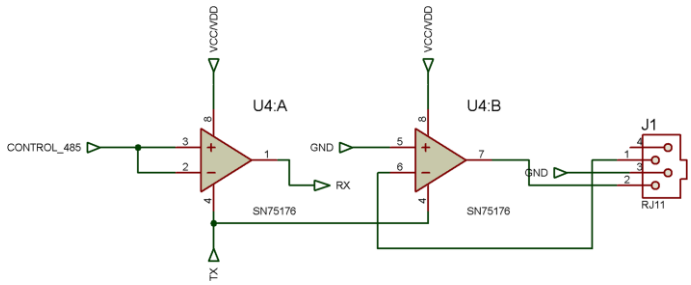


Figura 32.

Elaborado por: Darwin Cacuango y Eduardo Zapata.

Cada lector NFC con su respectiva placa ira conectado a un concentrador de terminales RJ-11, como se muestra en la figura 33, este se usa para enviar datos al computador mediante un conversor RS-485 a RS-232 y posteriormente un conversor RS-232 a USB concluyendo así el proceso de comunicación entre hardware y software.

Concentrador RS-485

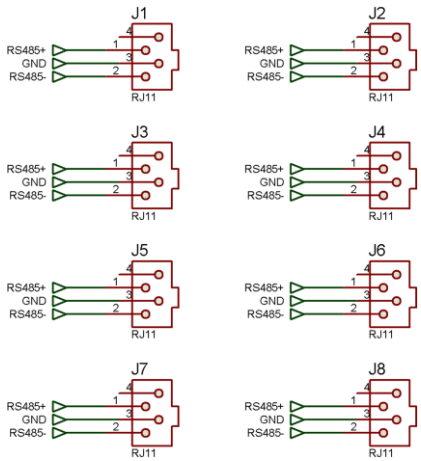


Figura 33.

Elaborado por: Darwin Cacuango y Eduardo Zapata.

En la figura 34, se representa mediante un diagrama de bloques las conexiones de todo el sistema hacia la aplicación.

Diagrama bloques del proyecto

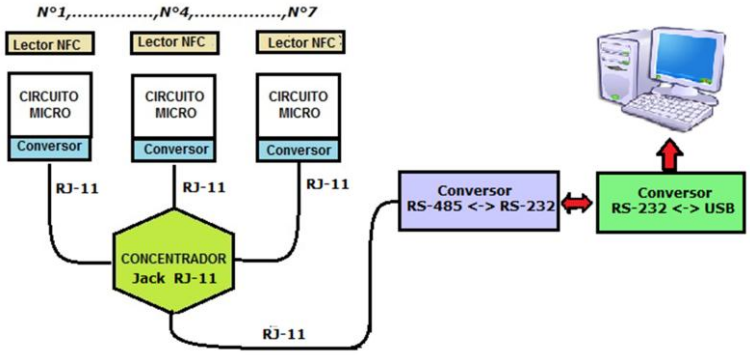


Figura 34.

Elaborado por: Darwin Cacuango y Eduardo Zapata.

4.2.1.3 Etapa de control de potencia

Esta fase del sistema prototipo tiene como objetivo el control de potencia para poder accionar la cerradura electromagnética colocada en cada acceso del prototipo únicamente si ocurre una validación exitosa de información, esta etapa consta de dos relés que se activan con una señal que emite el microcontrolador a

través del Pin 6 para el relé 1 y el Pin 9 para el relé 2. A continuación en la figura 35, se muestra el diagrama de conexión de esta etapa.

Diagrama conexión de relés

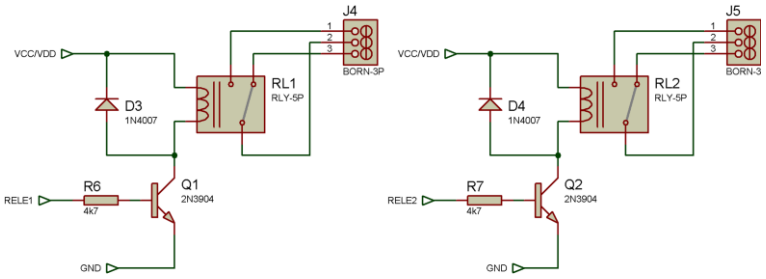


Figura 35.

Elaborado por: Darwin Cacuango y Eduardo Zapata.

4.3 Requerimiento de hardware y software

El prototipo para el control de acceso multinivel consta de dos clases de requerimientos, el hardware se refiere a todos los equipos que se necesitan para la implementación física del sistema y los requerimientos de software son la parte lógica del sistema, corresponde al sistema operativo en donde se ejecutará la aplicación para que funcione el sistema.

4.3.1 Hardware

Para desarrollar el sistema prototipo se desarrollará una comunicación SPI maestro/esclavo entre el microcontrolador PIC 16F628A y el lector NFC "RFID MFRC522" respectivamente.

4.3.1.1 PIC 16F628A

El PIC figura 36 es el encargado del establecimiento de la comunicación SPI con el lector además se encarga de recibir y enviar la información que recibe del lector mediante el uso de tarjetas NFC.

PIC 16F628A.

Figura 36. (Microchip, 2014).

Las características detalladas del PIC 16F628A se muestran en su datasheet (Technology M. , 1998-2015).

4.3.1.2 Circuito integrado SN75176

El integrado SN75176 se encarga de generar los niveles de voltaje necesarios para la transmisión de la red RS485, logrando la correcta recepción y transmisión de datos en el sistema.

Para lograr la comunicación entre hardware y software, y así formar una red donde exista el intercambio de información en el sistema se usa este integrado, el cual elabora una interfaz RS-485.

SN75176.

*Figura 37.*Encapsulado del integrado SN75176.

Fuente: (Instruments, 2015).

Como se muestra en la figura 37, el integrado SN75176, posee una estructura encapsulada con 8 pines, este dispositivo se encarga de hacer la conversión entre los niveles TTL del microcontrolador y las señales del tipo diferencial que utilizan el bus RS-485, las características detalladas del CI. SN75176 se describen en su datasheet (Netherlands, 2006-2015).

4.3.1.3 Cerradura electromagnética.

La Cerradura Electromagnética funciona como medio de apertura a lugares donde la entrada debe ser limitada. Está formada de dos componentes, el electroimán, y el montaje, el primero es un imán este funciona como en la medida que circule corriente por su bobina. Dejan de magnetizar, al momento en que se corta la corriente, en la figura 38 se muestra la cerradura usada en el prototipo.

Cerradura electromagnética.

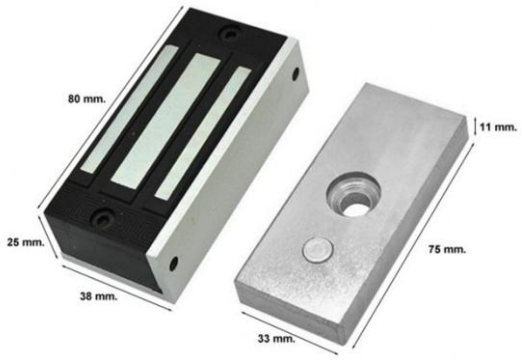


Figura 38. Dimensiones de la cerradura.

Fuente: (Simplifica, 2014).

El electroimán, es el elemento que crea un campo magnético al proporcionarle corriente eléctrica, está compuesto en su interior de un núcleo de hierro, al cual se enrolla un hilo conductor revestido de material aislante como barniz creando una bobina. Al energizar la cerradura se convierte en un imán capaz de atraer objetos metálicos.

Funcionamiento de la cerradura electromagnética.

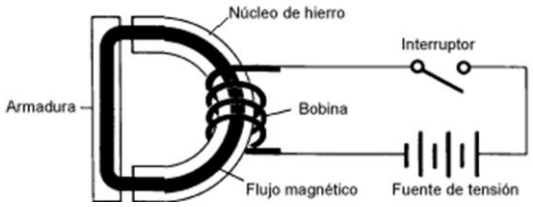


Figura 39. Funcionamiento del electroimán. Fuente: (Educativa, 2014).

En la figura 39, se puede observar las partes de un electroimán, además se puede observar el principio de funcionamiento del mismo, únicamente atraerá objetos metálicos cuando se energice el bobinado del electroimán.

Algunas de las ventajas al usar este tipo de cerraduras son:

- Funcionan con cualquier control de acceso eléctrico o electrónico.
- No hace falta el uso de llaves.
- No produce Arco Voltaico.
- Ideales para lugares donde se maneja alto tránsito de personas.
- Son sumamente aptas para instalaciones en interiores y exteriores.
- No necesitan mantenimiento.
- No sufren desgastes debido a que no cuentan con partes móviles.

Para más detalles acerca de la cerradura dirigirse al datasheet de este dispositivo en los anexos del proyecto.

4.3.1.4 Fuente de voltaje.

Este elemento se ha desarrollado para accionar las cerraduras electromagnéticas debido a que funcionan con 12VDC, además la parte de control de las mismas funcionan con 5VDC por este motivo se ha creado una fuente de voltaje capaz de convertir la tensión alterna en una tensión continua y lo más estable posible para proporcionar a cada circuito del prototipo.

Para construir la fuente se siguen estos pasos en orden de izquierda a derecha, mostrados en la figura 40.

Secuencia de funcionamiento de una fuente.

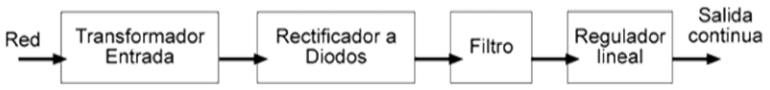


Figura 40. (Molina, 2014).

En la figura 41, se describe mediante un diagrama de bloques la función de cada una de las etapas usadas en la construcción de una fuente de voltaje.

Diagrama de bloques de una fuente de voltaje.

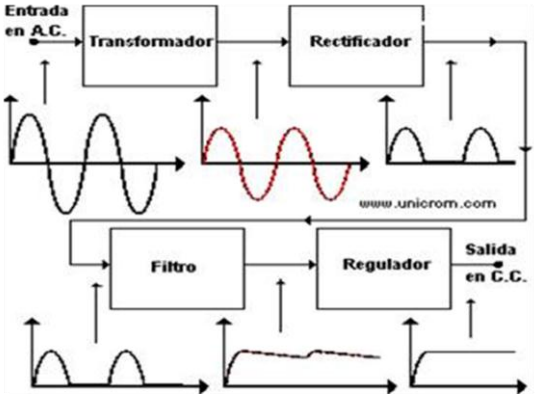


Figura 41. (Molina, 2014).

La fuente construida usa en su última etapa los reguladores LM7805 y LM7812 estos dispositivos están diseñados para suministrar una tensión fija. Una característica de este dispositivo es que dispone de protección térmica y limitación de corriente por si se producen cortocircuitos.

Regulador de voltaje.

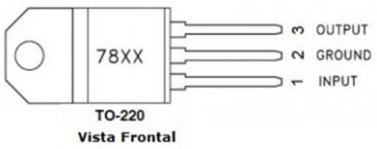


Figura 42. (Electrónica, 2014).

En la figura 42, se muestra la función de los pines del regulador del voltaje esta es la misma para todos los de la familia 78XX, son los más usados debido a su facilidad de uso y bajo costo, la descripción detallada de cada regulador de voltaje se encuentra en el datasheet respectivo (DatasheetCatalog, 2015).

4.3.1.5 Conversor RS-485 a RS-232

La red del prototipo está establecida con la norma RS-485, se debe crear un circuito que convierta estas señales en formato RS-232 para que así pueda conectarse en la red el dispositivo maestro que en este caso es el computador, el cual está encargado de enviar o recibir información.

Conversor RS-485 / RS-232

Figura 43. (Technology S. &., 2015).

En la figura 43, se muestra el Convertidor HXSP-485 es un convertidor bidireccional entre RS-232 y RS-485; se utiliza principalmente para la comunicación entre controladores principales y periféricos, logra la comunicación de solicitud-respuesta entre múltiples máquinas.

4.3.2 Software

El programa para desarrollar la aplicación que se usara en el proyecto será Visual Studio Express para Windows Desktop que posee una variedad de lenguajes de programación incluyendo C #, Visual Basic y C ++, con la tecnología adecuada para desarrollar la aplicación.

Visual Studio Express

Figura 44. (Microsoft, 2015).

En la figura 44, se muestra el nombre y la versión del programa instalado para desarrollar la aplicación, se debe tener en cuenta los requisitos para poder instalar y ejecutar la aplicación en el ordenador.

Los requisitos del sistema para instalar el programa son:

- Windows 7 SP1 (x86 y x64)
- Windows 8 (x86 y x64)
- De Windows 8.1 (x86 y x64)
- Windows Server 2008 R2 SP1 (x64)
- Windows Server 2012 (x64)
- Windows Server 2012 R2 (x64)
- Windows Small Business Server 2011 (x64)

Requisitos de hardware:

- 2,2 GHz o un procesador más rápido
- 1 GB de RAM
- 4 GB de espacio disponible en disco duro

Una vez instalado el programa se procede a desarrollar la aplicación para el sistema siempre teniendo en cuenta los aspectos visuales y técnicos para su correcto funcionamiento.

Para asegurarse de que la aplicación se instalará y se ejecutará correctamente, primero se debe asegurar que todos los componentes de los que depende su aplicación ya estén instalados en el equipo de destino. Por ejemplo, la mayoría de las aplicaciones creadas con Visual Studio tienen una dependencia de .NET Framework; antes de instalar la aplicación, el equipo de destino debe tener la versión correcta de Common Language Runtime.

Visual Studio genera un programa ejecutable de Windows llamado Setup.exe, también conocido como programa previo, este es responsable de la instalación de estos requisitos previos antes de que se ejecute la aplicación.

4.4 Implementación del prototipo

El diseño e implementación del prototipo de control de acceso basado en tecnología NFC tiene varias fases que se pueden resumir en dos Hardware y Software

En la parte del hardware se usa el dispositivo lector NCF y el PIC16F628A con sus sistemas de acondicionamiento electrónico respectivos. Para el software se desarrolló el código y aplicaciones necesarias para el correcto funcionamiento del hardware de acuerdo a las necesidades establecidas.

Para el desarrollo del código, simulación y aplicaciones se utilizó tres paquetes de software, Proteus 8 Professional, mikroC PRO for PIC y el Microsoft *Visual* C# *Express* 2010.

En el primero se desarrolla la simulación y diseño de placas electrónica del prototipo, en el segundo se programan los microcontroladores para la comunicación SPI y poder obtener el ID de las etiquetas, en el tercer software se crea una aplicación compatible con Windows 7, Vista y XP, para conectar el prototipo de forma serial o USB a un PC.

Antes de comenzar la implementación del prototipo es necesario establecer cuáles son sus requisitos de diseño. Los requisitos que debe cumplir o tener el prototipo tanto en hardware como en software son:

- Un elemento central, como un PIC, para manejar el resto de dispositivos.
- Un módulo que permita leer el ID (lector RFID) de las etiquetas RFID.
- Un algoritmo que permita al PIC manejar el lector RFID.
- Un bloque de actuadores e indicadores para controlar las acciones a realizar.
- Un módulo de comunicación para conectar el prototipo a un PC.
- Un algoritmo que permita al PIC enviar y recibir información.
- Una aplicación para PC que reciba los datos enviados por el prototipo, que permita visualiza al ID de las etiquetas RFID y además manejar una

pequeña base de datos en la que se guarde información de las interacciones en el prototipo.

De acuerdo a los requisitos descritos anteriormente se determina que el hardware del prototipo debería tener los módulos y elementos que muestran en la figura 45.

Diagramas de bloques de la tarjeta de control

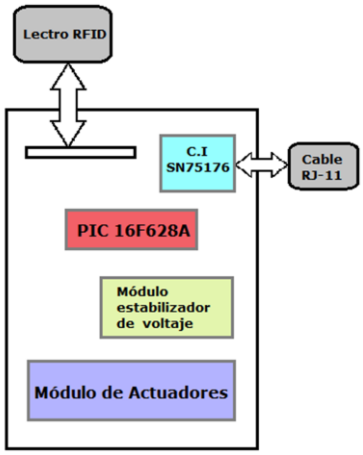


Figura 45.

Elaborado por: Darwin Cacuango y Eduardo Zapata.

Teniendo claro los requisitos de diseño del prototipo se desarrolla la parte lógica, esta se realiza con Proteus VSM es un completo entorno de diseño, que permite realizar todas las tareas de diseño de circuitos electrónicos.

4.4.1 Diseño de la tarjeta de control

Como primer paso en esta etapa se desarrolla el firmware para el microcontrolador, que permite la comunicación entre el lector y el PIC mediante el protocolo SPI, en la figura 46, se describe el funcionamiento de los dispositivos.

Diagrama de flujo de proceso del proyecto

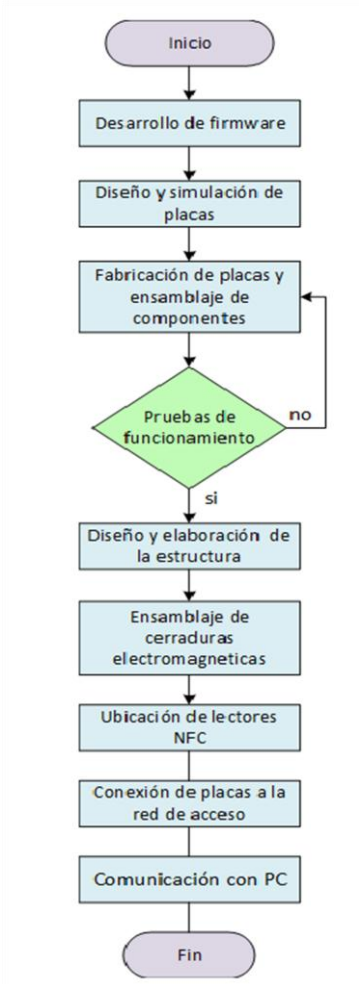


Figura 46.

Elaborado por: Darwin Cacuango y Eduardo Zapata.

Una vez desarrollado el firmware, se realizó la prueba del diseño esquemático del circuito en protoboard, figura47.

Diseño esquemático electrónico.

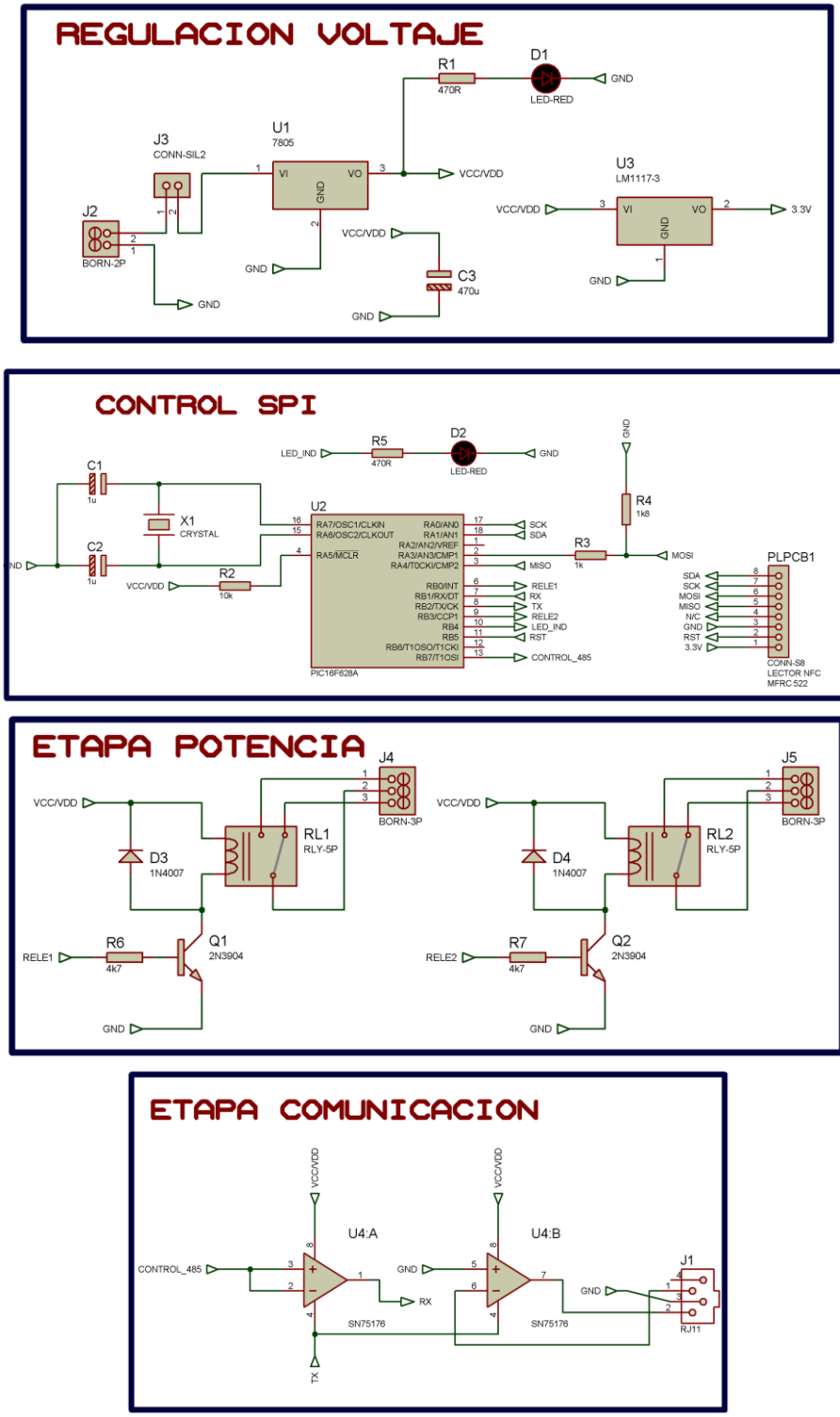


Figura 47.

Elaborado por: Darwin Cacuango y Eduardo Zapata.

Una vez comprobado el funcionamiento del circuito se procede a la fabricación de las siete placas para el prototipo, una de las cuales se muestra en la figura 48, donde se puede observar el diseño de PCB (Printed Circuit Board).

Diseño PCB placa de control.

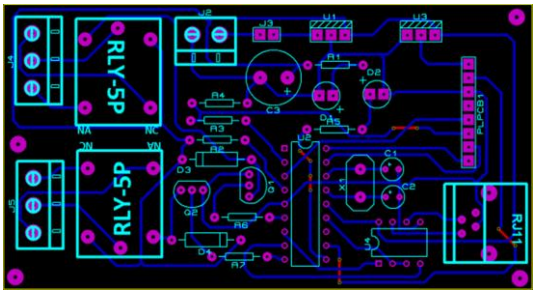


Figura 48.

Elaborado por: Darwin Cacuango y Eduardo Zapata.

Después del proceso de fabricación y ensamblaje de componentes de las placas se realizó pruebas de funcionamiento entre la tarjeta de control y el computador.

Placa de control terminada.

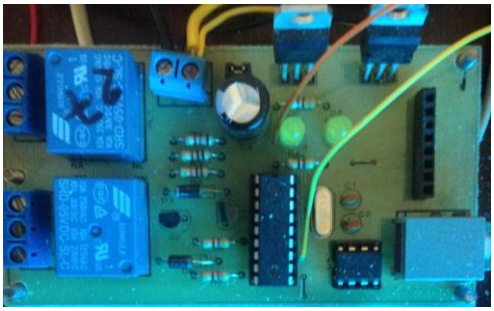


Figura 49.

Elaborado por: Darwin Cacuango y Eduardo Zapata.

En la figura 49, se observa la placa terminada con todos sus componentes y lista para el montaje en el prototipo.

Concluido el proceso de fabricación de las placas se inicia la elaboración del prototipo para el control de acceso multinivel, a continuación se muestra el bosquejo del diseño figura 50.

Esquema de diseño del prototipo

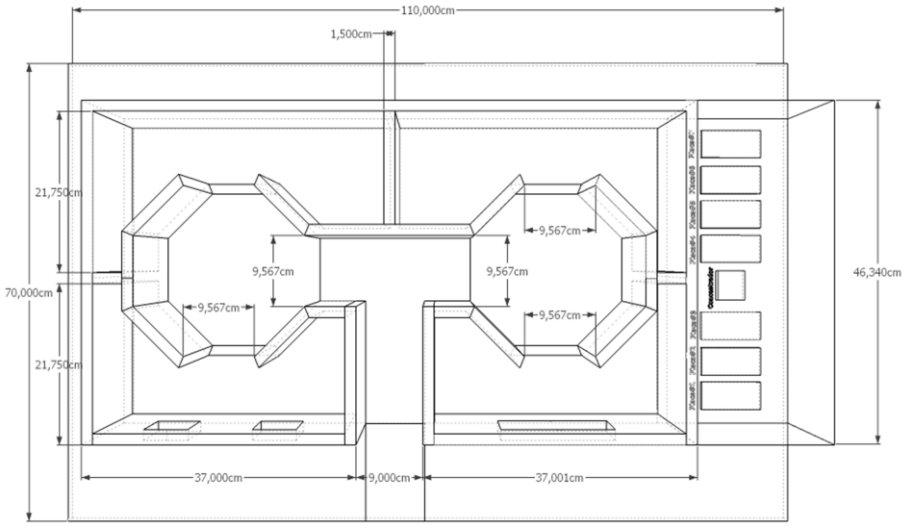


Figura 50.

Elaborado por: Darwin Cacuango y Eduardo Zapata.

Para la construcción de las paredes del modelo se tomó en cuenta las dimensiones que serán ocupadas por el cableado, los lectores y las chapas magnéticas, para lo que se deja un espacio entre la estructura de las paredes como se muestra en la figura 51.

Construcción del prototipo

Figura 51.

Elaborado por: Darwin Cacuango y Eduardo Zapata.

Concluida la fabricación de la estructura se procede a ubicar las chapas electromagnéticas en cada uno de los accesos figura52.

Ubicación de chapas electromagnéticas

Figura 52.

Elaborado por: Darwin Cacuango y Eduardo Zapata.

Ubicadas las chapas electromagnéticas se coloca los lectores NFC en las paredes del prototipo mostrado en la figura 53.

Instalación de lectores.

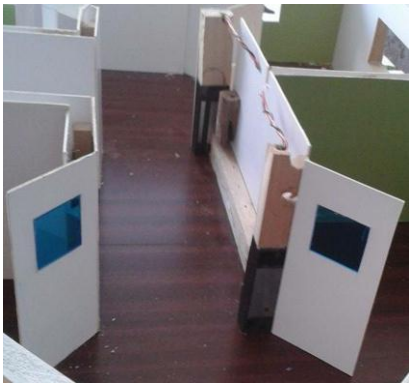


Figura 53.

Elaborado por: Darwin Cacuango y Eduardo Zapata.

Una vez colocados los lectores y chapas se destina un espacio para la ubicación de las placas de cada uno de los accesos, aquí también se conectan los lectores y las cerraduras electromagnéticas a las placas de control, figura 54.

Prototipo terminado

Figura 54.

Elaborado por: Darwin Cacuango y Eduardo Zapata.

Terminado el montaje del prototipo se conecta con el computador para comprobar el correcto funcionamiento de todo el sistema, garantizando que cada dispositivo instalado no haya sufrido ningún desperfecto.

4.4.2 Diseño de aplicación del prototipo

El desarrollo de la aplicación de control del sistema se realizara en Visual Studio Express para Windows aprovechando las ventajas del mismo como se indica en la sección Requerimiento de hardware y software .

Como primer paso para la creación de la aplicación se debe crear un nuevo proyecto en Visual Basic seleccionando la opción Windows Forms Application, como se muestra en la figura 55.

Creación del proyecto (Visual Studio)

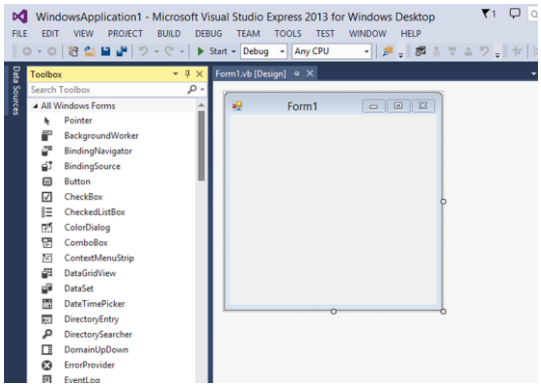


Figura 55.

Elaborado por: Darwin Cacuango y Eduardo Zapata.

Una vez creado el proyecto se crea una ventana llamada Form, que es una plantilla de trabajo para crear la aplicación, en esta ventana se puede agregar diferentes controles según los requerimientos del proyecto.

En la figura 56, se observa el cuadro de herramientas que contiene todos los controles como botones, listas, CheckBox, etc. A medida que se desarrolle la aplicación se irán empleando estos elementos, cada vez que se crea un control en la aplicación aparecerá un cuadro de propiedades y opciones de programación, donde se puede configurar cada uno de los controles empleados.

Lista de controles y propiedades (Visual Studio)

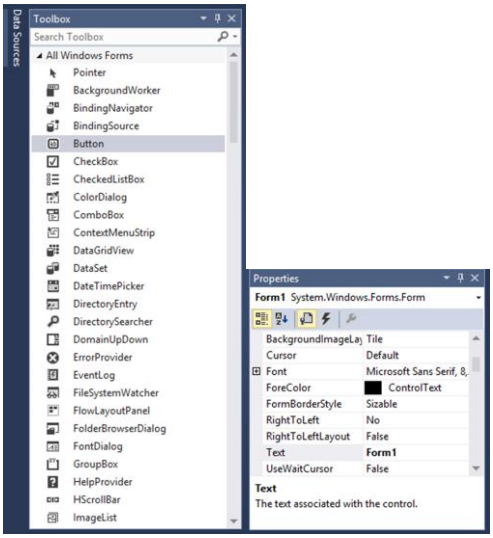


Figura 56.

Elaborado por: Darwin Cacuango y Eduardo Zapata.

Una vez creado el proyecto de la aplicación del sistema se procede a la creación de la base de datos de todo el sistema, en esta base de datos se tendrá información propia de cada usuario, datos informativos como código de acceso, nombres completos, identificación, teléfono, correo electrónico y asignación de accesos.

La creación de la base de datos se realizará en el software Microsoft Access, en la creación de la base de datos se edita una tabla donde se llena ciertos parámetros como asignar un nombre al campo a utilizar, se debe seleccionar el tipo de dato o variable que maneja cada campo, figura 57.

Creación de campos de la base de datos

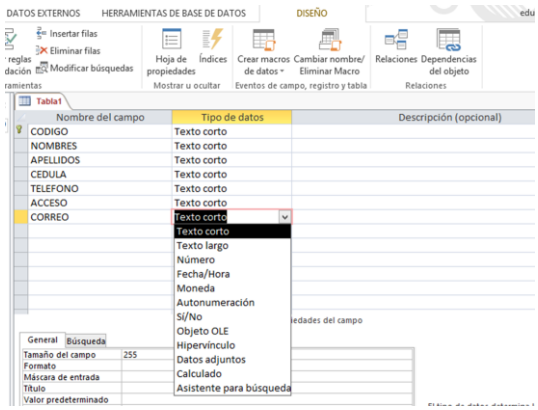


Figura 57.

Elaborado por: Darwin Cacuango y Eduardo Zapata.

Después de crear la tabla con todos los campos necesario, se guarda el proyecto para después enlazar con nuestra aplicación en Visual Basic, esto se realiza en la opción herramientas de la ventana inicial opción Connect to Database, seleccionada el tipo de base de datos a usar se escoge el directorio donde se guardó la misma, figura 58.

Selección de directorio de la base de datos

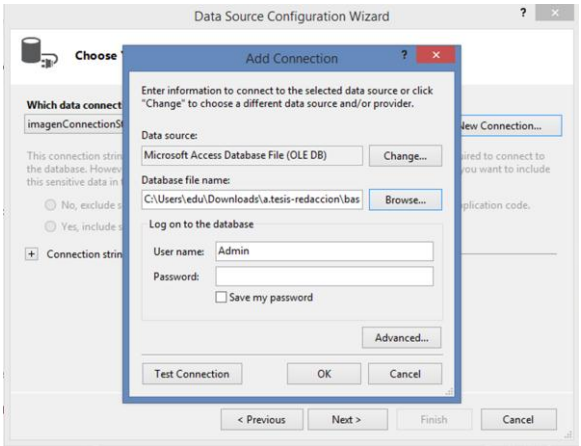


Figura 58.

Elaborado por: Darwin Cacuango y Eduardo Zapata.

Una vez escogida la base de datos a relacionar con la aplicación, se elige todas las variables usadas para que se puedan enlazar y manipular con la aplicación este paso se muestra en la figura 59.

Elección de variables a importar

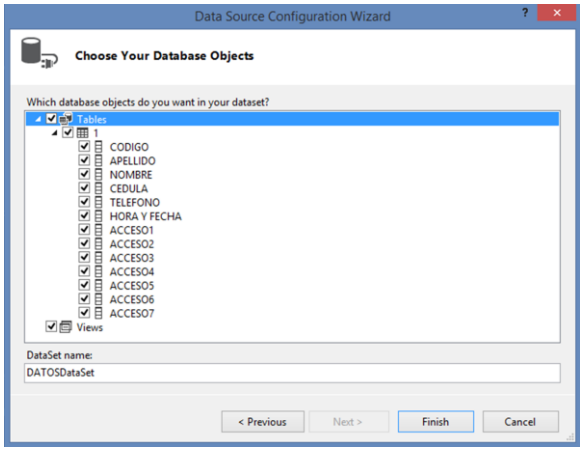


Figura 59.

Elaborado por: Darwin Cacuango y Eduardo Zapata.

Cuando se termina de enlazar la base de datos con la aplicación se puede visualizar una lista de variables creadas en el lado derecho inferior de la ventana de Visual Basic como se observa en la figura 60.

Variables enlazadas con la aplicación

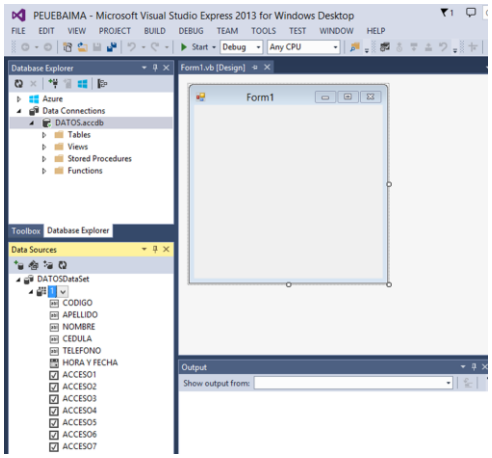


Figura 60.

Elaborado por: Darwin Cacuango y Eduardo Zapata.

Cada variable creada puede tomar distintas formas según el uso requerido como texBox o Button, etc. Al escoger el tipo de herramienta se desplaza a la ventana Form de la aplicación, figura 61.

Selección de variables y tipo de control

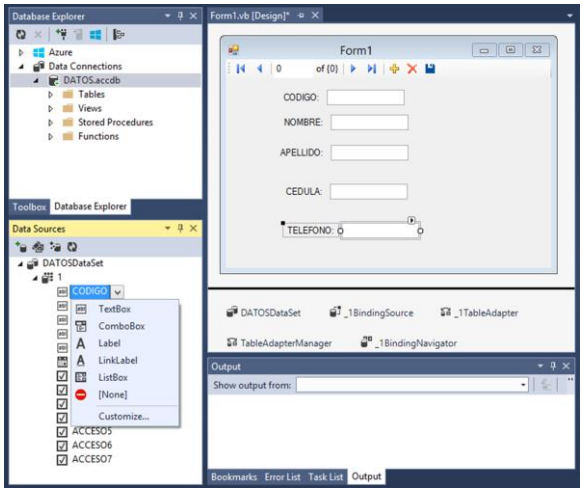


Figura 61.

Elaborado por: Darwin Cacuango y Eduardo Zapata.

Dependiendo de los requerimientos de la aplicación del sistema se irá desarrollando la misma, es decir, se agregará ventanas de trabajo, botones y más herramientas asignado acciones dependiendo del uso, también se debe aclarar que para la comunicación de la aplicación con el prototipo se debe importar la opción SerialPort del menú de herramientas y definir sus características en su respectiva ventana de propiedades.

4.5 Costos del sistema

A continuación, se realiza una descripción de costos del sistema de control de acceso con tecnología NFC, tomando en cuenta el costo de la estructura y las horas de programación para la implementación de la aplicación. A demás se realiza también un detalle de los costos del mismo sistema con el uso de tecnología arduino.

Tabla 3

Costos del sistema

Detalle	Cantidad	Valor unidad ($)	Valor Total ($)
Lector RFID	7	15	105
PIC 16F628A	7	4	28
Cerraduras	7	18	126
Transformador	1	8	8
Circuitería		40	40
Maqueta	1	100	100
Costo de software	400 horas	8	3.200
Total			**$ 3.607**

Nota. Detalle costos del prototipo con PIC

Elaborado por: Darwin Cacuango y Eduardo Zapata.

Tabla 4

Costos del sistema con tecnología arduino

Detalle	Cantidad	Valor unidad ($)	Valor Total ($)
Lector RFID	7	15	105
Arduino UNO	7	40	280
Cerraduras	7	18	126
Transformador	1	8	8
Circuitería		80	80
Maqueta	1	100	100
Costo de software	400 horas	8	3.200
Total			**$ 3.900**

Nota. Detalle costos del prototipo con Arduino

Elaborado por: Darwin Cacuango y Eduardo Zapata.

Además se debe tomar en cuenta la reducción del costo de comercialización del lector usado en este sistema, en la tabla 5 muestra la variación de costos.

Tabla 5

Variación del costo de comercialización del lector

Detalle	Cantidad	Valor unidad ($) Año 2012	Valor unidad ($) Año 2014
Lector RFID	1	20	15
Arduino UNO	1	45	40

Nota. Variación anual de comercialización

Elaborado por: Darwin Cacuango y Eduardo Zapata.

CAPÍTULO 5

ANÁLISIS DE RESULTADOS

En este capítulo se describen las pruebas realizadas de todo el sistema de control de acceso NFC, comprobando el funcionamiento de la parte de hardware y software del proyecto

5.1 Pruebas de funcionamiento del prototipo

Las pruebas de funcionamiento del sistema se ejecutaron una vez implementado todas las partes en el diseño del prototipo además se efectuaron pruebas por separado para detectar y evitar errores en cada una de las fases y posteriormente en todo el sistema.

5.1.1 Pruebas de hardware

En esta etapa se realiza pruebas de los circuitos, tarjetas, chapas magnéticas y todo lo referente a la parte electrónica del sistema.

La primera prueba a realizar es comprobar el funcionamiento de la fuente de poder creada para el sistema la que proporciona dos tipos de alimentación, una parta la parte de las placas electrónicas y la otra para el manejo de las cerraduras electromagnéticas. El voltaje suministrado para las placas de control es de 5 [Vdc], y para las cerraduras electromagnéticas es de 12[Vdc], como se muestra en la figura 62.

Fuente de alimentación proyecto

Figura 62.

Elaborado por: Darwin Cacuango y Eduardo Zapata.

Una vez comprobado que la fuente de alimentación funcione correctamente se conecta al sistema para probar el funcionamiento de las placas y las cerraduras. Cuando se energiza las placas de control un led se enciende indicando que se encuentran en funcionamiento la placa y el lector, figura 63.

Led indicador de funcionamiento

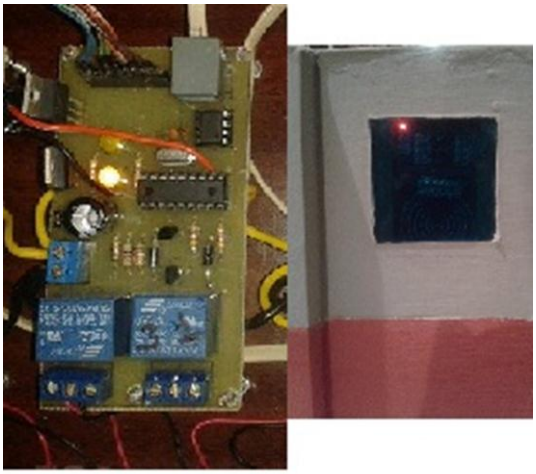


Figura 63.

Elaborado por: Darwin Cacuango y Eduardo Zapata.

Para probar las cerraduras electromagnéticas se usó un pulsador conectado a una fuente para simular los impulsos que genera el PIC cuando tiene éxito la autenticación.

5.1.2 Pruebas de envío y recepción de información.

Como prueba final de las placas se verifico el envío y recepción de datos usando un lector y la aplicación X-CTU usando el puerto serial del computador, además se debe configurar previamente ciertos parámetros para la comunicación como se muestra en la figura 64.

Parámetros comunicación Terminal X-CTU

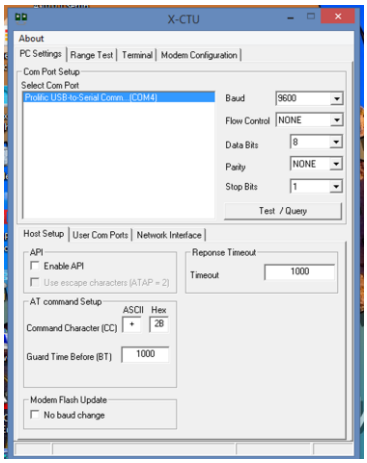


Figura 64.

Elaborado por: Darwin Cacuango y Eduardo Zapata.

Una vez establecida la comunicación entre la aplicación X-CTU y el lector se comprueba la recepción y envió de información entre sí, figura 65.

Recepción de lectura de tarjeta

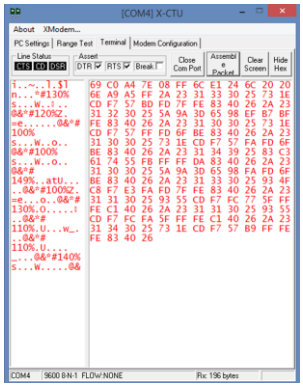


Figura 65.

Elaborado por: Darwin Cacuango y Eduardo Zapata.

Para comprobar el intercambio de información desde el computador hacia el sistema se realiza mediante el envío de paquetes en formato hexadecimal el cual realizara una acción dependiendo el código enviado, mostrado en la figura 66.

Envío de código hacia el prototipo

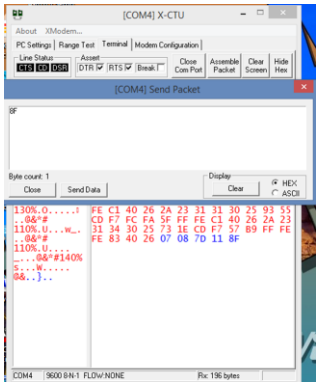


Figura 66.

Elaborado por: Darwin Cacuango y Eduardo Zapata.

5.1.3 Prueba de funcionamiento total del sistema

Comprobado el envío y recepción de información con la parte de hardware del sistema determinando su correcto funcionamiento se procede a la conexión total del sistema.

A continuación se muestra imágenes de la interacción con el prototipo:

Primero se inicia la aplicación creada para el sistema la cual empezará a funcionar a partir del ingreso de un nombre de usuario y contraseña válidos, como se muestra en la figura 67.

Ventana de Inicio de la aplicación

Figura 67.

Elaborado por: Darwin Cacuango y Eduardo Zapata.

Después de autenticarse, el usuario tiene acceso a la ventana donde se ejecuta el monitoreo en tiempo real de las interacciones de usuarios con el prototipo, figura 68.

Ventana de monitoreo del sistema

Figura 68.

Elaborado por: Darwin Cacuango y Eduardo Zapata.

Ya en la ventana de monitoreo se selecciona el puerto de comunicación para interactuar con el prototipo, si el puerto seleccionado es el correcto los indicadores de color amarillo mostrados en la imagen 68 se tornaran azules, como se muestran en la figura 69.

Selección de puerto

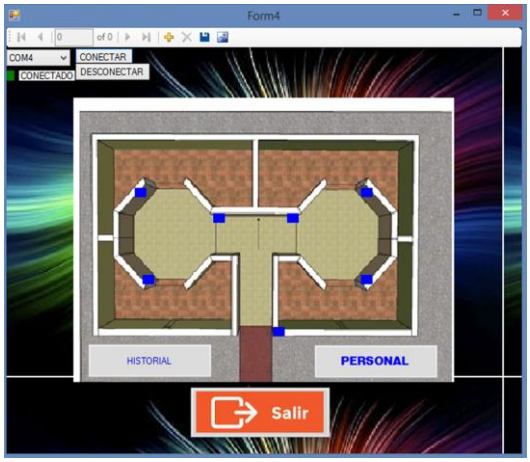


Figura 69.

Elaborado por: Darwin Cacuango y Eduardo Zapata.

Se debe destacar que los indicadores de cada acceso se pondrán de color verde cuando el usuario no tenga ninguna restricción de acceso a cierta instalación, caso contrario se tornaran rojos.

La aplicación también cuenta con una sección donde se puede consultar, editar, ingresar y borrar la información de nuestra base de datos adquiriendo el código de la tarjeta asignada a cada persona, además se puede establecer los privilegios de cada usuario para su acceso a las instalaciones mostrado en la figura 70.

Sección para editar información de personal.

Figura 70.

Elaborado por: Darwin Cacuango y Eduardo Zapata.

Para el registro de un nuevo usuario se selecciona el botón adquirir código, el cual abre una venta en donde se escoge el puerto para obtener el código de una nueva tarjeta y después ingresar la información personal del usuario, como se puede observar en la figura 71.

Adquisición de código de tarjeta

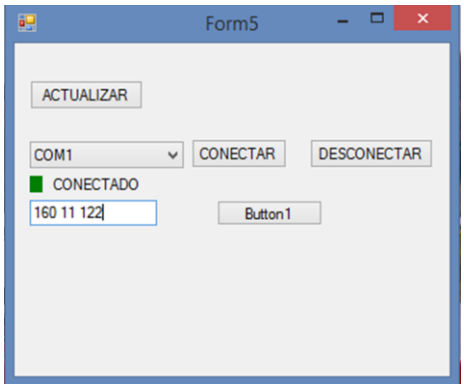


Figura 71.

Elaborado por: Darwin Cacuango y Eduardo Zapata.

Finalmente existe una ventana en donde se presentará la información de las interacciones y el flujo de personas en las instalaciones, detallando la información del usuario enlazada con cada tarjeta, mostrado en la figura 72.

Base de datos de usuarios.

Figura 72.

Elaborado por: Darwin Cacuango y Eduardo Zapata.

5.2 Análisis y resultados

Después de las pruebas realizadas se pudo determinar que el sistema puede operar durante largos periodos de funcionamiento siempre y cuando se implementen disipadores de calor en el sistema para evitar posibles daños.

Se comprobó que la aplicación posee un diseño amigable para la interacción con el usuario del sistema capaz de ser operado por personal sin experiencia en el manejo de aplicaciones de control de acceso.

Se debe considerar que al realizar la implementación del hardware y software se tuvo ciertos inconvenientes al funcionar simultáneamente las dos partes, los cuales se describen a continuación:

Al realizar la primera prueba de funcionamiento entre las partes del prototipo se produjo un error al efectuar la lectura del identificativo de la tarjeta, en donde se adquiría un indeterminado número de datos provenientes desde el lector, lo cual no permitía identificar el código correspondiente a cada tarjeta, impidiendo que la aplicación ejecute las acciones correspondientes, lo que obligo a limitar el rango de adquisición de datos en la aplicación para obtener únicamente el identificativo de la tarjeta logrando la correcta comparación de datos y obtener la información única de cada persona.

En la segunda prueba del sistema se notó el retardo en la visualización de datos en pantalla de la aplicación, lo que también provocaba retraso al ejecutar las diferentes acciones establecidas en la aplicación. La acción tomada ante este inconveniente fue sincronizar la velocidad de comunicación entre la parte de hardware y software además de transferir previamente a la aplicación los datos almacenados en Access para evitar retardos innecesarios al consultar constantemente a otro programa.

Al momento de realizar una prueba total del sistema se manejó diferentes accesos indistintamente lo cual produjo una falla en la ejecución de las acciones tomadas por la aplicación, esta se debía a que cuando se usaba un lector por primera vez la

aplicación no presentaba errores pero al usar otro lector las acciones se realizaban desde la primera interacción, es decir repetía lo ejecutado con la información adquirida por el primer lector y en ocasiones las acciones se intercambiaban, esto se debió a la ejecución de una acción mientras se estaba realizando una operación previa, lo cual alteraba los datos tomados y causaba errores, para corregir este problema se implementó el uso de hilos en el programa de la aplicación, para que se ejecuten estos procesos de verificación de datos y los mismo se habilitaban nuevamente terminada su operación.

Se debe destacar el rango de comunicación de esta tecnología en donde se estipula una distancia máxima de 20 centímetros, a continuación se realiza una prueba del rango de comunicación entre terminales del sistema implementado.

Prueba de rango de comunicación

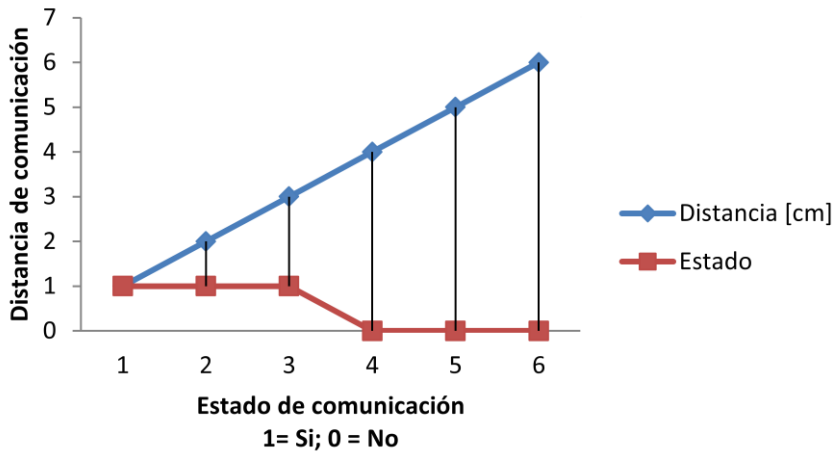


Figura 73.

Elaborado por: Darwin Cacuango y Eduardo Zapata.

En la figura 73, se muestra el rango de comunicación del sistema donde se nota que el rango máximo de comunicación es de 4 centímetros.

CONCLUSIONES

- La implementación del prototipo de control de acceso con tecnología NFC, demuestra su versatilidad y bajo costo en este tipo de aplicaciones, ya que comparando este prototipo con uno de las mismas características realizado con tecnología arduino, este es 10% menos costoso, logrando ser una solución alternativa frente a los sistemas de control de acceso existentes en el mercado.
- El control de acceso con tecnología NFC implementado es una alternativa que ofrece gran seguridad debido a que el intercambio de información entre los dispositivos que intervienen en la comunicación se realiza a una distancia máxima de 4 centímetros, lo que dificulta que agentes externos puedan interferir entre terminales y realicen una clonación de información durante el intercambio de datos.
- Los sistemas de control de acceso ya no están únicamente enfocados hacia las empresas con gran infraestructura, se ha logrado expandir su uso al campo de la domótica e inmótica debido a que su costo ha disminuido considerablemente por su desarrollo en los últimos años, permitiendo expandir la comercialización de dispositivos con tecnología NFC, incluso en Latinoamérica especialmente en Ecuador donde existen limitaciones a la comercialización de insumos de tecnología, en este caso del lector usado en el sistema ha reducido un 35% su valor de comercialización en el transcurso de dos años.
- La tecnología NFC puede ser usada en diferentes campos de aplicación ya que las herramientas de desarrollo y librerías son completas, además estas pertenecen a la biblioteca de código abierto NFC lo que evita costos de licenciamiento y permite la interacción entre dispositivos y etiquetas.
- En el desarrollo de ésta aplicación se usa Thread o hilos, que sirven para realizar varias operaciones al mismo tiempo, esto permite liberar al programa principal de procesos secundarios, logrando que pueda ejecutar acciones en tiempo real, de esta forma los procesos pueden ejecutarse sin causar confusión en los datos y sin pérdida de los mismos.

RECOMENDACIONES

- Para este tipo de accesos se recomiendan el uso de cerraduras electromagnéticas las cuales son muy compatibles con el sistema ya que receptan de una manera muy eficaz las señales emitidas por el sistema.
- Se recomienda realizar pruebas de funcionamiento a cerraduras electromagnéticas, lectores NFC y micro-controladores antes de la instalación en el prototipo, además cuando el sistema esté instalado se recomienda realizar mantenimientos periódicos para asegurar el correcto funcionamiento del mismo.
- Es importante que el sistema tenga una fuente de energía alterna por posibles cortes de energía asegurando el funcionamiento ininterrumpido del sistema brindando seguridad y control en dichas situaciones.
- Para incrementar el nivel de seguridad del sistema se recomienda el uso de móviles que cuenten con tecnología NFC, ya que el dispositivo es de uso personal, además se podría implementar el uso de cámaras fotográficas en cada acceso para además de tener la información personal del usuario tener un registro visual de cada uno.
- El puerto serial debe ser debidamente desconectado (liberado) al momento de cerrar la aplicación para evitar errores y bloqueos del mismo, ya que al no cerrar debidamente el puerto el mismo queda inhabilitado para recibir datos, lo cual obligaría cambiar de puerto.
- Se debe tener muy en cuenta el momento crear y de enlazar la base de datos con nuestra aplicación y escoger correctamente el tipo de variables con las que se va a trabajar y adjuntar a la carpeta de la aplicación creada.

LISTA DE REFERENCIAS

(ECMA), E. C. (2004). *ECMA International*. Recuperado el 6 de noviembre de 2014, de http://www.ecma-international.org/

(2011). Recuperado el 20 de diciembre de 2014, de RapidNFC: http://rapidnfc.com/what_is_nfc

Arduino. (2015). *Arduino Forum*. Recuperado el 21 de enero de 2015, de http://arduino.cc/

Austria, N. S. (2002-2014). *MIFARE*. Recuperado el 4 de octubre de 2014, de http://www.mifare.net/es/technology/nfc/

Chérrez, D. F. (Junio de 2010). *Repositorio digital Escuela Politecnica Nacional.* Recuperado el 28 de noviembre de 2014, de http://bibdigital.epn.edu.ec/bitstream/15000/2227/1/CD-2970.pdf

Chip, N. (2014). *Chips NFC*. Recuperado el 10 de febrero de 2015, de http://www.chipsnfc.com/

DatasheetCatalog. (2015). *DatasheetCatalog.* Recuperado el 10 de diciembre de 2014, de http://pdf.datasheetcatalog.com/datasheet/nationalsemiconductor/LM78XX.pdf

Educativa, I. L. (2014). *Biblioteca Digital*. Recuperado el 16 de enero de 2014, de http://bibliotecadigital.ilce.edu.mx/sites/ciencia/volumen3/ciencia3/112/htm/sec_9.htm

Electrónica. (2014). Recuperado el 5 de enero de 2015, de http://electronica-teoriaypractica.com/reguladores-de-tension-7805-7812-7815-y-7824/

Forum, N. (Junio de 2004). *NFC Forum*. Recuperado el 15 de octubre de 2014, de http://nfc-forum.org

Gómez, E. L. (Julio de 2013). *Repositorio Digital Universidad Politecnica Nacional*. Recuperado el 18 de diciembre de 2014, de http://bibdigital.epn.edu.ec/handle/15000/6440

Instruments, T. (2015). *Texas Instruments*. Recuperado el 24 de febrero de 2015, de http://www.ti.com/

INTECO. (2013). *Cuaderno de notas del Observatorio.* Recuperado el 20 de noviembre de 2014, de Instituto Nacional de Tecnologías de la Comunicación: http://www.inteco.es/

Microchip. (12 de 2014). Recuperado el 10 de octubre de 2014, de http://www.microchip.com/

Microsoft. (2015). *Visual Studio*. Recuperado el 8 de enero de 2015, de https://www.visualstudio.com/es-es/downloads/download-visual-studio-vs

Molina. (2014). *Tutoriales*. Recuperado el 15 de enero de 2015, de http://www.profesormolina.com.ar/tutoriales/tutor1_fuentes.htm

Netherlands, N. S. (2006-2015). *NXP Semiconductors.* Recuperado el 28 de diciembre de 2014, de http://www.nxp.com/documents/data_sheet/MFRC522.pdf

Pérez, I. E. (Diciembre de 2014). *Ingenieria en mircrocontroladores.* Recuperado el 20 de enero de 2015, de http://www.i-micro.com/pdf/articulos/rs-485.pdf

PIC, T. (Enero de 2012). *Tutoriales PIC*. Recuperado el 16 de noviembre de 2014, de http://picfernalia.blogspot.com/

Rubén Abuín, R. d. (2014). *Universidad de Deusto.* Recuperado el 2 de diciembre de 2014, de www.morelab.deusto.es/images/talks/NFC.ppt

Sciences, U. o. (2007). *NFC Research LAB*. Recuperado el 15 de diciembre de 2014, de http://www.nfc-research.at/

Simplifica, T. (18 de 12 de 2014). *T Simplifica*. Recuperado el 18 de diciembre de 2014, de http://www.tsimplifica.com/es/product/cerradura-electromagnetica-mini-de-60kg

Technology, M. (Enero de 1998-2015). *MicroChip*. Recuperado el 24 de noviembre de 2014, de http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en010210

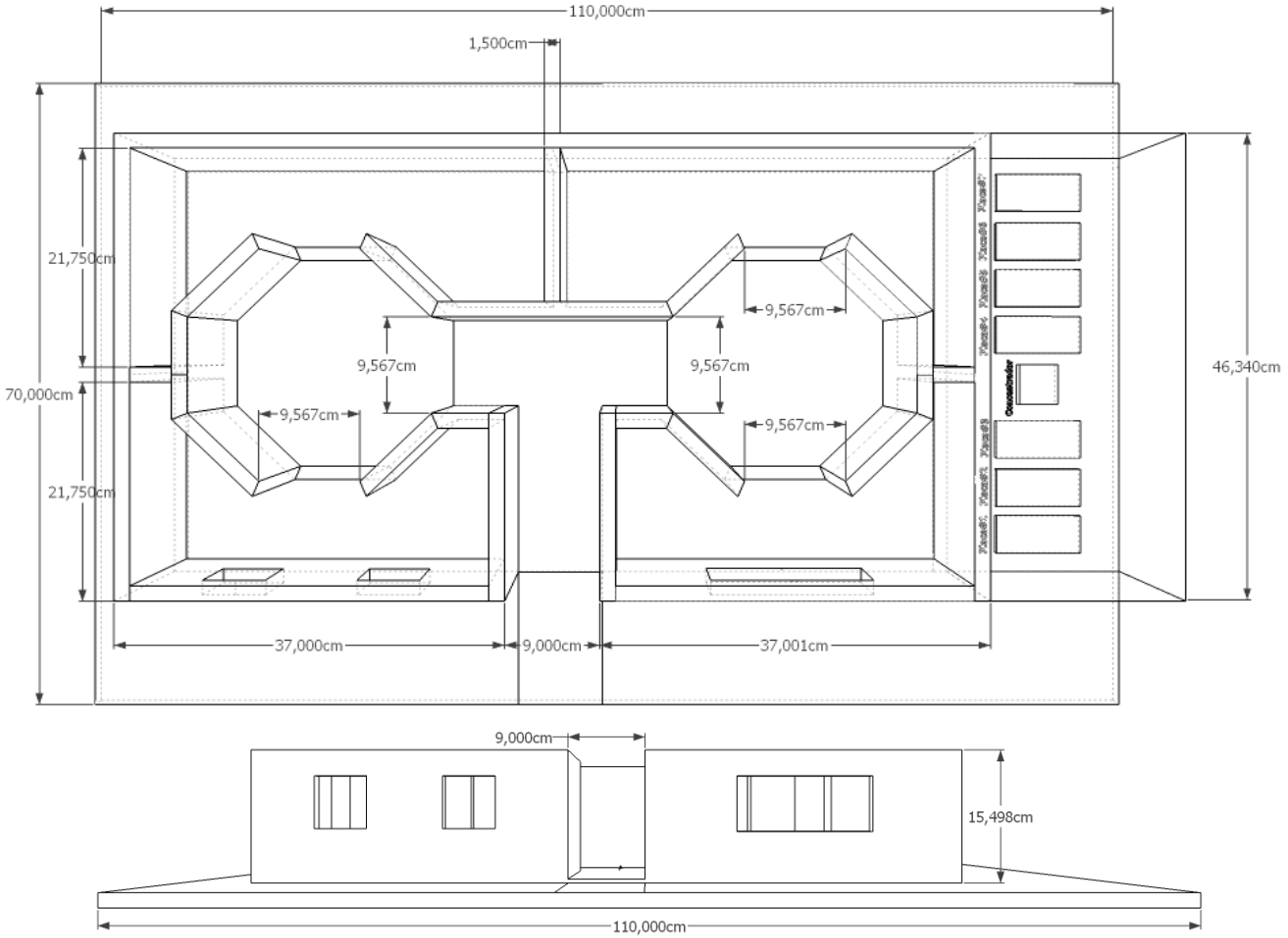
Technology, S. &. (2015). *Soarland & Hexin*. Recuperado el 3 de febrero de 2015, de http://www.hexin-technology.com/RS-232_to_RS-485_Converter-Product-493.html

TOOPAN FORMS CO, L. (2012). *NFC Portal Site*. Recuperado el 13 de octubre de 2014, de http://www.nfc-world.com/en/index.html

Vedat, C., Ok, K., & Ozdenizci, B. (February 2012). *Near Field Communication: From Theory to Practice.*

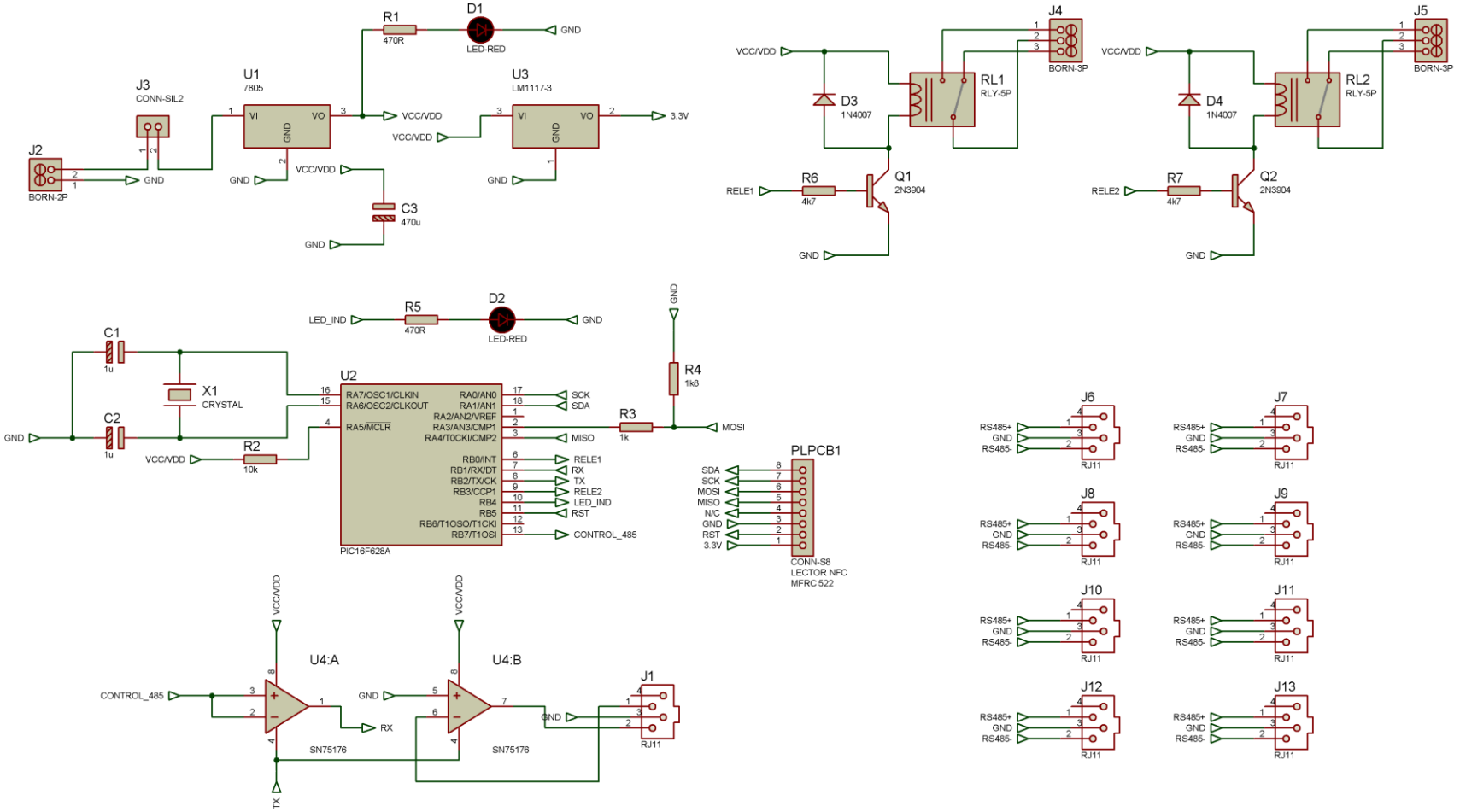
ANEXOS

Anexo 1. Planos del prototipo.



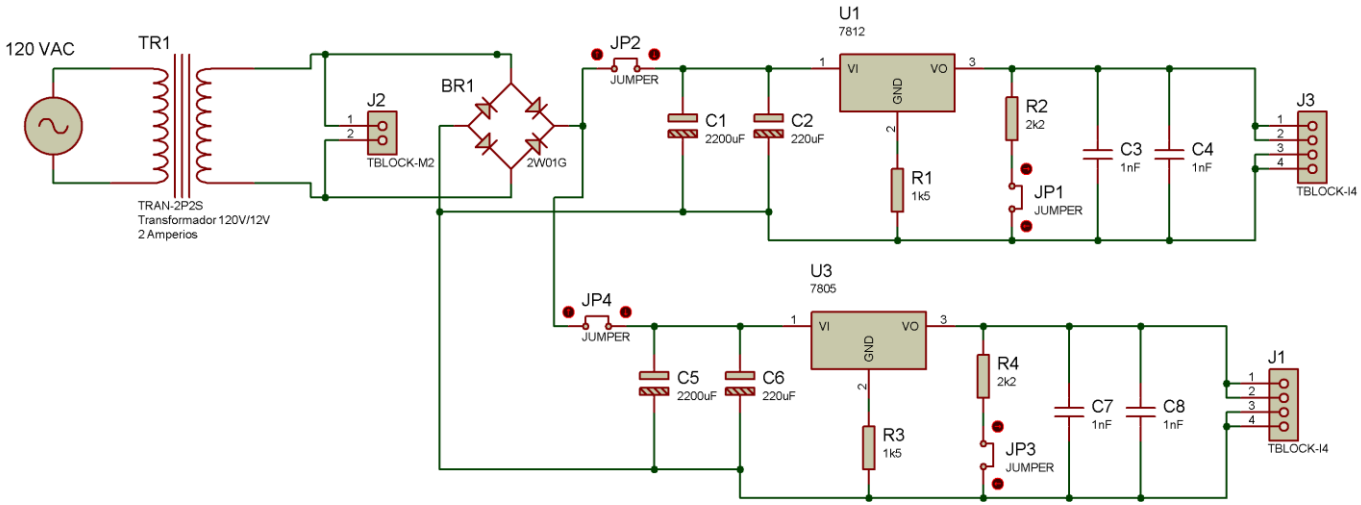
Anexo 2. Circuitos eléctricos y electrónicos

- Diagrama electrónico del circuito del prototipo



-

- Fuente de alimentación del prototipo



Anexo 3. Programa PIC 16F628A

```
// conexiones del módulo SPI

sbit Chip_Select at RA1_bit;
sbit SoftSpi_SDI at RA4_bit;
sbit SoftSpi_SDO at RA3_bit;
sbit SoftSpi_CLK at RA0_bit;

sbit Chip_Select_Direction at TRISA1_bit;
sbit SoftSpi_SDI_Direction at TRISA4_bit;
sbit SoftSpi_SDO_Direction at TRISA3_bit;
sbit SoftSpi_CLK_Direction at TRISA0_bit;
// Fin conexiones del módulo

#define  chipSelectPin RA1_bit       // Seleccion de Chip (CS) =
Selecccion Esclavo (SS)
#define  led RB4_bit
#define rele1 RB0_bit
#define rele2 RB3_bit
#define  beep RB6_bit
#define  NRSTPD RB5_bit   // pin rst del lector

//tamaño maximo del vector
#define MAX_LEN 16

//////////////////////////////////////////////////////////////////////
//definir pin
//////////////////////////////////////////////////////////////////////
//MF522 definir constantes
#define PCD_IDLE              0x00          //Ninguna acción
#define PCD_AUTHENT           0x0E          //clave de autenticación
#define PCD_RECEIVE           0x08          //Recepción de datos
#define PCD_TRANSMIT          0x04          //Transmicion de datos
#define PCD_TRANSCEIVE        0x0C          //transmite y recide datos,
#define PCD_RESETPHASE        0x0F          //Reset
#define PCD_CALCCRC           0x03

// definir constantes tag Mifare
#define PICC_REQIDL          0x26          // encontrar area de la antena
#define PICC_REQALL          0x52          // encuentra tarjeta en el
area de la antena
#define PICC_ANTICOLL        0x93          // anti-colision
#define PICC_SElECTTAG       0x93          // eleccion tarejta
#define PICC_AUTHENT1A       0x60          // clave de autenticación A
#define PICC_AUTHENT1B       0x61          // clave de autenticación B
#define PICC_READ            0x30          // lee bloque
#define PICC_WRITE           0xA0          // escribe bloque
#define PICC_DECREMENT       0xC0          // decremento
#define PICC_INCREMENT       0xC1          // recarga
#define PICC_RESTORE         0xC2          // transferir datos al buffer
#define PICC_TRANSFER        0xB0          // guarda datos en el buffer
#define PICC_HALT            0x50          // pausa

//estado del lector al comunicarse
#define MI_OK                 0
#define MI_NOTAGERR           1
#define MI_ERR                2

//------------------MFRC522 registros---------------
//Commando y estado
#define     Reserved00            0x00
#define     CommandReg            0x01
```

```
#define     CommIEnReg            0x02
#define     DivlEnReg             0x03
#define     CommIrqReg            0x04
#define     DivIrqReg             0x05
#define     ErrorReg              0x06
#define     Status1Reg            0x07
#define     Status2Reg            0x08
#define     FIFODataReg           0x09
#define     FIFOLevelReg          0x0A
#define     WaterLevelReg         0x0B
#define     ControlReg            0x0C
#define     BitFramingReg         0x0D
#define     CollReg               0x0E
#define     Reserved01            0x0F

//Comandos
#define     Reserved10            0x10
#define     ModeReg               0x11
#define     TxModeReg             0x12
#define     RxModeReg             0x13
#define     TxControlReg          0x14
#define     TxAutoReg             0x15
#define     TxSelReg              0x16
#define     RxSelReg              0x17
#define     RxThresholdReg        0x18
#define     DemodReg              0x19
#define     Reserved11            0x1A
#define     Reserved12            0x1B
#define     MifareReg             0x1C
#define     Reserved13            0x1D
#define     Reserved14            0x1E
#define     SerialSpeedReg        0x1F


#define     Reserved20            0x20
#define     CRCResultRegM         0x21
#define     CRCResultRegL         0x22
#define     Reserved21            0x23
#define     ModWidthReg           0x24
#define     Reserved22            0x25
#define     RFCfgReg              0x26
#define     GsNReg                0x27
#define     CWGsPReg              0x28
#define     ModGsPReg             0x29
#define     TModeReg              0x2A
#define     TPrescalerReg         0x2B
#define     TReloadRegH           0x2C
#define     TReloadRegL           0x2D
#define     TCounterValueRegH     0x2E
#define     TCounterValueRegL     0x2F

//prueba de registro
#define     Reserved30            0x30
#define     TestSel1Reg           0x31
#define     TestSel2Reg           0x32
#define     TestPinEnReg          0x33
#define     TestPinValueReg       0x34
#define     TestBusReg            0x35
#define     AutoTestReg           0x36
#define     VersionReg            0x37
#define     AnalogTestReg         0x38
#define     TestDAC1Reg           0x39
#define     TestDAC2Reg           0x3A
#define     TestADCReg            0x3B
#define     Reserved31            0x3C
```

```
#define     Reserved32            0x3D
#define     Reserved33            0x3E
#define     Reserved34            0x3F
//-----------------------------------------------

#define     uchar                 unsigned char
#define     uint                  unsigned int
#define     boolean               bit
#define     ushort                unsigned short

//4 bytes nuemro de tag
uchar  serNum[5];


boolean stringComplete;  // si el string está completo
ushort receive[8];
uchar bytevar1, bytevar2 = 0;
uchar uchar_send[15];

void delay1ms(uint delayTime);
void pulseLED(void);
void beep350(void);
void beep900(void);
void beep1136(void);
void SetFormatRDM630(void);
void MFRC522_Init(void);
void TimeOut1(void);
uchar Separate_hexP10(uchar val);
uchar Separate_hexP1(uchar val);
void Write_MFRC522(uchar addr, uchar val);
uchar Read_MFRC522(uchar addr);
void SetBitMask(uchar reg, uchar mask);
void ClearBitMask(uchar reg, uchar mask);
void AntennaOn(void);
void AntennaOff(void);
void MFRC522_Reset(void);
void MFRC522_Init(void);
uchar MFRC522_Request(uchar reqMode, uchar *TagType);
uchar MFRC522_ToCard(uchar command, uchar *sendData, uchar sendLen, uchar
*backData, unsigned int *backLen);
uchar MFRC522_Anticoll(uchar *serNum);
void CalulateCRC(uchar *pIndata, uchar len, uchar *pOutData);
uchar MFRC522_SelectTag(uchar *serNum);
uchar MFRC522_Auth(uchar authMode, uchar BlockAddr, uchar *Sectorkey,
uchar *serNum);
uchar MFRC522_Read(uchar blockAddr, uchar *recvData);
uchar MFRC522_Write(uchar blockAddr, uchar *writeData);
void MFRC522_Halt(void);

void setup() {
  TRISB = 0B11111111;   // proteccion
  delay1ms(500);        //  proteccion
  TRISA = 0B00000000;
  TRISB = 0B00000010;
  PORTA = 0;
  PORTB = 0;

  OPTION_REG = 0B00001111;
  INTCON = 0B11000000;
  PIE1 = 0B00100000;

  UART1_Init(9600);                       // inicia puerto a 9600bps
  RCSTA = 0B10010000;
  // libreria SPI:
  Soft_SPI_Init();
```

```
  Chip_Select = 1; // SlaveSelect (SS)

    MFRC522_Init();
  stringComplete = 0;
  portb.F7=1;
  delay1ms(1000);
   UART1_Write_Text("Start rfid");
   portb.F7=0;
   delay_ms(100);
}

void main(void)
{
  setup();
  while (1)  // bucle infinito
  {

    uchar i,tmp = 0;
    uchar status;
    uchar str[MAX_LEN];
    uchar RC_size;
    uchar blockAddr;
    status = MFRC522_Request(PICC_REQIDL, str);


    status = MFRC522_Anticoll(str);
    memcpy(serNum, str, 5);
    if (status == MI_OK)
    {
    portb.F7=1;
    delay_ms(100);
     UART1_Write_Text("*#140%");   //nuemro de esclavo

     UART1_Write_Text(serNum);
     UART1_Write_Text("@&&");      //Final trama

     portb.F7=0;
    delay_ms(100);

     for (bytevar1=0;bytevar1<250;bytevar1++)
      {
          beep = !beep;
          delay_us(350);
      }
    }

    MFRC522_Halt();

  }
}

void delay1ms(uint delayTime)
{
  uint loop1;
  for (loop1=0;loop1<delayTime;loop1++)
  {
    delay_ms(1);
  }
}

void pulseLED(void)
{
  led = 0;   // led off
  delay1ms(250);
  led = 1;   // led on
```

```
}

void beep350(void)
{
  for (bytevar1=0;bytevar1<250;bytevar1++)
  {
    beep = !beep;
    delay_us(350);
  }
  beep = 0;
}

void beep900(void)
{
  for (bytevar1=0;bytevar1<250;bytevar1++)
  {
    beep = !beep;
    delay_us(900);
  }
  beep = 0;
}

void beep1136(void)
{
  for (bytevar1=0;bytevar1<250;bytevar1++)
  {
    beep = !beep;
    delay_us(1136);
  }
  beep = 0;
}

void interrupt(void)
{
  ushort RCREG_temp = 0;

  if (PIR1.RCIF) // bandera de dato recibido UART
  {

    PIR1.RCIF = 0;
    RCREG_temp = RCREG;
    if (RCREG_temp == 140)  // rele1 on
    {
      rele1=1;
    }
    if (RCREG_temp == 142)  // rele1 off
    {
      rele1=0;
    }
    if (RCREG_temp == 143)  // rele2 on
    {
      rele2=1;
    }
    if (RCREG_temp == 144)  // rele2 off
    {
      rele2=0;
    }
    if (RCREG_temp == 145)  // led on
    {
      led=1;
    }
    if (RCREG_temp == 146)  // led off
    {
      led=0;
    }
```

```
  }
}
/*
 * Funcion : Write_MFRC5200
 */
void Write_MFRC522(uchar addr, uchar val)
{
  chipSelectPin = 0;

  //formato de dirreccion: 0XXXXXX0
  Soft_SPI_Write((addr<<1)&0x7E);
  Soft_SPI_Write(val);

  chipSelectPin = 1;
}
/*
 * Funcion : Read_MFRC522
 */
uchar Read_MFRC522(uchar addr)
{
  uchar val;

  chipSelectPin = 0;

  //dirrecion formato: 1XXXXXX0
  Soft_SPI_Write(((addr<<1)&0x7E) | 0x80);
  val = Soft_SPI_Read(0x00);

  chipSelectPin = 1;

  return val;
}

void SetBitMask(uchar reg, uchar mask)
{
  uchar tmp;
  tmp = Read_MFRC522(reg);
  Write_MFRC522(reg, tmp | mask);  // set bit mask
}

void ClearBitMask(uchar reg, uchar mask)
{
  uchar tmp;
  tmp = Read_MFRC522(reg);
  Write_MFRC522(reg, tmp & (~mask));  // clear bit mask
}
/*
 * Funcion : Antena On
 */
void AntennaOn(void)
  {
  uchar temp;

  temp = Read_MFRC522(TxControlReg);
  if (!(temp & 0x03))
  {
    SetBitMask(TxControlReg, 0x03);
}
}

/*
  * Funcion : Antena Off

 */
void AntennaOff(void)
```

```
{
  ClearBitMask(TxControlReg, 0x03);
}
/*
 * Funcion : Reset MFRC522
 */
void MFRC522_Reset(void)
{
  Write_MFRC522(CommandReg, PCD_RESETPHASE);
}
/*
 * Funcion : Inicio MFRC522
*/
void MFRC522_Init(void)
{
  NRSTPD = 1;

  MFRC522_Reset();

  Write_MFRC522(TModeReg, 0x8D);
  Write_MFRC522(TPrescalerReg, 0x3E);
  Write_MFRC522(TReloadRegL, 30);
  Write_MFRC522(TReloadRegH, 0);

  Write_MFRC522(TxAutoReg, 0x40);
  Write_MFRC522(ModeReg, 0x3D);

  AntennaOn();                //antena on
}

/*
 * Funcion : MFRC522_Respuesta
 */
uchar MFRC522_Request(uchar reqMode, uchar *TagType)
{
  uchar status;
  uint backBits;                        //recepcion de datos

  Write_MFRC522(BitFramingReg, 0x07);

  TagType[0] = reqMode;
  status = MFRC522_ToCard(PCD_TRANSCEIVE, TagType, 1, TagType,
&backBits);

  if ((status != MI_OK) || (backBits != 0x10))
  {
    status = MI_ERR;
  }

  return status;
}

uchar MFRC522_ToCard(uchar command, uchar *sendData, uchar sendLen, uchar
*backData, uint *backLen)
{
  uchar status = MI_ERR;
  uchar irqEn = 0x00;
  uchar waitIRq = 0x00;
  uchar lastBits;
  uchar n;
  uint i;

  switch (command)
  {
    case PCD_AUTHENT:
```

```
  {
    irqEn = 0x12;
    waitIRq = 0x10;
    break;
  }
  case PCD_TRANSCEIVE:
  {
    irqEn = 0x77;
    waitIRq = 0x30;
    break;
  }
  default:
    break;
}
Write_MFRC522(CommIEnReg, irqEn|0x80);
ClearBitMask(CommIrqReg, 0x80);
SetBitMask(FIFOLevelReg, 0x80);

Write_MFRC522(CommandReg, PCD_IDLE);

//Writing data to the FIFO
for (i=0; i<sendLen; i++)
{
  Write_MFRC522(FIFODataReg, sendData[i]);
}
//Ejecuta comando
Write_MFRC522(CommandReg, command);
if (command == PCD_TRANSCEIVE)
{
  SetBitMask(BitFramingReg, 0x80);
}

i = 2000;
do
{

  n = Read_MFRC522(CommIrqReg);
  i--;
}
while ((i!=0) && !(n&0x01) && !(n&waitIRq));

ClearBitMask(BitFramingReg, 0x80);

if (i != 0)
{
  if(!(Read_MFRC522(ErrorReg) & 0x1B))
  {
    status = MI_OK;
    if (n & irqEn & 0x01)
    {
      status = MI_NOTAGERR;
    }

    if (command == PCD_TRANSCEIVE)
    {
      n = Read_MFRC522(FIFOLevelReg);
      lastBits = Read_MFRC522(ControlReg) & 0x07;
      if (lastBits)
      {
        *backLen = (n-1)*8 + lastBits;
      }
      else
      {
        *backLen = n*8;
      }
```

```
        if (n == 0)
        {
          n = 1;
        }
        if (n > MAX_LEN)
        {
          n = MAX_LEN;
        }

        for (i=0; i<n; i++)
        {
          backData[i] = Read_MFRC522(FIFODataReg);
        }
      }
    }
    else
    {
      status = MI_ERR;
    }

  }
  return status;
}

uchar MFRC522_Anticoll(uchar *serNum)
{
  uchar status;
  uchar i;
  uchar serNumCheck=0;
  uint unLen;

  serNum[0] = PICC_ANTICOLL;
  serNum[1] = 0x20;
  status = MFRC522_ToCard(PCD_TRANSCEIVE, serNum, 2, serNum, &unLen);

  if (status == MI_OK)
  {

    for (i=0; i<4; i++)
    {
      serNumCheck ^= serNum[i];
    }
    if (serNumCheck != serNum[i])
    {
      status = MI_ERR;
    }
  }

  return status;
}

void CalulateCRC(uchar *pIndata, uchar len, uchar *pOutData)
{
  uchar i, n;

  ClearBitMask(DivIrqReg, 0x04);
  SetBitMask(FIFOLevelReg, 0x80);

  for (i=0; i<len; i++)
  {
    Write_MFRC522(FIFODataReg, *(pIndata+i));
  }
  Write_MFRC522(CommandReg, PCD_CALCCRC);
```

```
  i = 0xFF;
  do
  {
    n = Read_MFRC522(DivIrqReg);
    i--;
  }
  while ((i!=0) && !(n&0x04));

  pOutData[0] = Read_MFRC522(CRCResultRegL);
  pOutData[1] = Read_MFRC522(CRCResultRegM);
}

void MFRC522_Halt(void)
{
  uchar status;
  uint unLen;
  uchar buff[4];

  buff[0] = PICC_HALT;
  buff[1] = 0;
  CalulateCRC(buff, 2, &buff[2]);

  status = MFRC522_ToCard(PCD_TRANSCEIVE, buff, 4, buff,&unLen);
}
```

Anexo 4. Programa de la aplicación

FORM 1 - Ventana de seguridad adquisición de usuario y clave

```
Public Class Form1
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
        Dim usuario As String, contraseña As String
        usuario = "administrador"
        contraseña = "123456"
        If (TextBox1.Text = usuario And TextBox2.Text = contraseña) Then
            Form4.Show()
            Me.Hide()
        Else
            Button1.BackColor = Color.Red
        End If

        TextBox1.Text = ""
        TextBox2.Text = ""

    End Sub

    Private Sub TextBox2_TextChanged(sender As Object, e As EventArgs)
Handles TextBox2.TextChanged

    End Sub

    Private Sub TextBox1_TextChanged(sender As Object, e As EventArgs)
Handles TextBox1.TextChanged
        Dim texto1 As String
        texto1 = Text

    End Sub

    Private Sub Button1_MouseEnter(sender As Object, e As EventArgs)
Handles Button1.MouseEnter
        Button1.BackColor = Color.LightGray
    End Sub

    Private Sub Button1_MouseLeave(sender As Object, e As EventArgs)
Handles Button1.MouseLeave
        Button1.BackColor = Color.RoyalBlue

    End Sub
End Class
```

FORM 2- Ingresar, borrar, editar y buscar los usuarios mediante los códigos en la base de datos.

```
Public Class Form2

    Private Sub PersonalBindingNavigatorSaveItem_Click(sender As Object, e
As EventArgs) Handles PersonalBindingNavigatorSaveItem.Click
        Me.Validate()
        Me.PersonalBindingSource.EndEdit()
```

```
        Me.TableAdapterManager.UpdateAll(Me.PERSONALDataSet)

    End Sub

    Private Sub Form2_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
        'TODO: This line of code loads data into the
'PERSONALDataSet.personal' table. You can move, or remove it, as needed.
        Me.PersonalTableAdapter.Fill(Me.PERSONALDataSet.personal)
        CODIGOTextBox.Text = ""
        APELLIDOTextBox.Text = ""
        NOMBRETextBox.Text = ""
        CEDULATextBox.Text = ""
        TELEFONOTextBox.Text = ""
        CORREOTextBox.Text = ""
        ACCESO1TextBox.Text = ""
        ACCESO2TextBox.Text = ""
        ACCESO3TextBox.Text = ""
        ACCESO4TextBox.Text = ""
        ACCESO5TextBox.Text = ""
        ACCESO6TextBox.Text = ""
        ACCESO7TextBox.Text = ""

    End Sub

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
        Me.PersonalTableAdapter.INSERTAR(CODIGOTextBox.Text,
APELLIDOTextBox.Text, NOMBRETextBox.Text, CEDULATextBox.Text,
TELEFONOTextBox.Text, CORREOTextBox.Text, ACCESO1TextBox.Text,
ACCESO2TextBox.Text, ACCESO3TextBox.Text, ACCESO4TextBox.Text,
ACCESO5TextBox.Text, ACCESO6TextBox.Text, ACCESO7TextBox.Text,
FECHA_HORA:=System.DateTime.Now.ToString())
        Me.PersonalTableAdapter.Fill(Me.PERSONALDataSet.personal)
        Form3.Show()
        Me.Hide()
        CODIGOTextBox.Text = ""
        APELLIDOTextBox.Text = ""
        NOMBRETextBox.Text = ""
        CEDULATextBox.Text = ""
        TELEFONOTextBox.Text = ""
        CORREOTextBox.Text = ""
        ACCESO1TextBox.Text = ""
        ACCESO2TextBox.Text = ""
        ACCESO3TextBox.Text = ""
        ACCESO4TextBox.Text = ""
        ACCESO5TextBox.Text = ""
        ACCESO6TextBox.Text = ""
        ACCESO7TextBox.Text = ""

    End Sub

    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles
Button2.Click
        Me.PersonalTableAdapter.EDITAR(CODIGOTextBox.Text,
APELLIDOTextBox.Text, NOMBRETextBox.Text, CEDULATextBox.Text,
TELEFONOTextBox.Text, CORREOTextBox.Text, ACCESO1TextBox.Text,
ACCESO2TextBox.Text, ACCESO3TextBox.Text, ACCESO4TextBox.Text,
ACCESO5TextBox.Text, ACCESO6TextBox.Text, ACCESO7TextBox.Text,
```

```
FECHA_HORA:=System.DateTime.Now.ToString(),
Original_CODIGO:=CODIGOTextBox.Text)
        Me.PersonalTableAdapter.Fill(Me.PERSONALDataSet.personal)
        Form3.Show()
        Me.Hide()
        CODIGOTextBox.Text = ""
        APELLIDOTextBox.Text = ""
        NOMBRETextBox.Text = ""
        CEDULATextBox.Text = ""
        TELEFONOTextBox.Text = ""
        CORREOTextBox.Text = ""
        ACCESO1TextBox.Text = ""
        ACCESO2TextBox.Text = ""
        ACCESO3TextBox.Text = ""
        ACCESO4TextBox.Text = ""
        ACCESO5TextBox.Text = ""
        ACCESO6TextBox.Text = ""
        ACCESO7TextBox.Text = ""
    End Sub

    Private Sub Button3_Click(sender As Object, e As EventArgs) Handles
Button3.Click
        Me.PersonalTableAdapter.BORRAR(CODIGOTextBox.Text)
        Me.PersonalTableAdapter.Fill(Me.PERSONALDataSet.personal)
        Form3.Show()
        Me.Hide()
        CODIGOTextBox.Text = ""
        APELLIDOTextBox.Text = ""
        NOMBRETextBox.Text = ""
        CEDULATextBox.Text = ""
        TELEFONOTextBox.Text = ""
        CORREOTextBox.Text = ""
        ACCESO1TextBox.Text = ""
        ACCESO2TextBox.Text = ""
        ACCESO3TextBox.Text = ""
        ACCESO4TextBox.Text = ""
        ACCESO5TextBox.Text = ""
        ACCESO6TextBox.Text = ""
        ACCESO7TextBox.Text = ""
    End Sub

    Private Sub Button4_Click(sender As Object, e As EventArgs) Handles
Button4.Click
        Me.PersonalTableAdapter.FillBy(Me.PERSONALDataSet.personal,
CODIGOTextBox.Text)

    End Sub

    Private Sub Button5_Click(sender As Object, e As EventArgs)
        Me.PersonalTableAdapter.Fill(Me.PERSONALDataSet.personal)



    End Sub

    Private Sub Label1_Click(sender As Object, e As EventArgs) Handles
Label1.Click

    End Sub
```

```
    Private Sub FECHA_HORALabel_Click(sender As Object, e As EventArgs)

    End Sub

    Private Sub Button6_Click(sender As Object, e As EventArgs) Handles
Button6.Click
        Form4.Show()
        Me.Hide()

    End Sub

    Private Sub Label2_Click(sender As Object, e As EventArgs) Handles
Label2.Click

    End Sub

    Private Sub Button6_MouseEnter(sender As Object, e As EventArgs)
Handles Button6.MouseEnter
        Button6.BackColor = Color.RoyalBlue
    End Sub

    Private Sub Button6_MouseLeave(sender As Object, e As EventArgs)
Handles Button6.MouseLeave
        Button6.BackColor = Color.LightGray

    End Sub

    Private Sub Button5_Click_1(sender As Object, e As EventArgs) Handles
Button5.Click
        Me.PersonalTableAdapter.Fill(Me.PERSONALDataSet.personal)
        CODIGOTextBox.Text = ""
        APELLIDOTextBox.Text = ""
        NOMBRETextBox.Text = ""
        CEDULATextBox.Text = ""
        TELEFONOTextBox.Text = ""
        CORREOTextBox.Text = ""
        ACCESO1TextBox.Text = ""
        ACCESO2TextBox.Text = ""
        ACCESO3TextBox.Text = ""
        ACCESO4TextBox.Text = ""
        ACCESO5TextBox.Text = ""
        ACCESO6TextBox.Text = ""
        ACCESO7TextBox.Text = ""

    End Sub

    Private Sub Button5_MouseEnter(sender As Object, e As EventArgs)
Handles Button5.MouseEnter
        Button5.BackColor = Color.LawnGreen

    End Sub

    Private Sub Button5_MouseLeave(sender As Object, e As EventArgs)
Handles Button5.MouseLeave
        Button5.BackColor = Color.LightGray

    End Sub

    Private Sub CODIGOTextBox_KeyPress(sender As Object, e As
KeyPressEventArgs) Handles CODIGOTextBox.KeyPress
```

```
    End Sub

    Private Sub CODIGOTextBox_TextChanged(sender As Object, e As EventArgs)
Handles CODIGOTextBox.TextChanged

    End Sub

    Private Sub Button7_Click(sender As Object, e As EventArgs) Handles
Button7.Click
        Form5.Show()
        Me.Hide()

    End Sub

    Private Sub CODIGOLabel_Click(sender As Object, e As EventArgs)

    End Sub

    Private Sub FECHA_HORADateTimePicker_ValueChanged(sender As Object, e
As EventArgs) Handles FECHA_HORADateTimePicker.ValueChanged

    End Sub
End Class
```

FORM 3-Visualización de la base de datos de cada persona

```
Public Class Form3

    Private Sub PersonalBindingNavigatorSaveItem_Click(sender As Object, e
As EventArgs) Handles PersonalBindingNavigatorSaveItem.Click
        Me.Validate()
        Me.PersonalBindingSource.EndEdit()
        Me.TableAdapterManager.UpdateAll(Me.PERSONALDataSet)

    End Sub

    Private Sub Form3_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
        'TODO: This line of code loads data into the
'PERSONALDataSet.personal' table. You can move, or remove it, as needed.
        Me.PersonalTableAdapter.Fill(Me.PERSONALDataSet.personal)

    End Sub

    Private Sub PersonalDataGridView_CellContentClick(sender As Object, e
As DataGridViewCellEventArgs) Handles PersonalDataGridView.CellContentClick

    End Sub

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
        Me.PersonalTableAdapter.Fill(Me.PERSONALDataSet.personal)

    End Sub

    Private Sub Button1_MouseEnter(sender As Object, e As EventArgs)
Handles Button1.MouseEnter
        Button1.BackColor = Color.Cyan
```

```
    End Sub

    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles
Button2.Click
        Form2.Show()
        Me.Hide()

    End Sub

    Private Sub Button2_MouseEnter(sender As Object, e As EventArgs)
Handles Button2.MouseEnter
        Button2.BackColor = Color.RoyalBlue
    End Sub

    Private Sub Button2_MouseLeave(sender As Object, e As EventArgs)
Handles Button2.MouseLeave
        Button2.BackColor = Color.LightGray

    End Sub

    Private Sub Button1_MouseLeave(sender As Object, e As EventArgs)
Handles Button1.MouseLeave
        Button1.BackColor = Color.LightGray

    End Sub
End Class
```

FORM 4-Visualización y monitoreo en tiempo real de los diferentes accesos.

```
Public Class Form4
    Dim acti As Threading.Thread
    Dim cont As Integer = 0
    Dim hilo As Integer = 0
    Dim tam As Integer
    Dim acce As String
    Dim avi1 As Integer = 5
    Dim ACCESO As String = ""
    Dim posicion As Integer
    Dim confirmacion As String
    Private Sub Form4_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
        'TODO: This line of code loads data into the
'HISTORIALDataSet.Tabla1' table. You can move, or remove it, as needed.
        Me.Tabla1TableAdapter.Fill(Me.HISTORIALDataSet.Tabla1)
        'TODO: This line of code loads data into the
'PERSONALDataSet.personal' table. You can move, or remove it, as needed.
        Label3.Text = ""
        Label4.Text = ""
        tam = PersonalTableAdapter.GetData.Count() - 1
        CheckForIllegalCrossThreadCalls = False
        buscarport()
    End Sub
    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles
Button2.Click
        Form2.Show()
        Me.Close()
    End Sub
```

```
    Private Sub Button3_Click(sender As Object, e As EventArgs) Handles
Button3.Click
        Form1.Show()
        Me.Hide()
    End Sub
    Private Sub Button3_MouseEnter(sender As Object, e As EventArgs)
Handles Button3.MouseEnter
        Button3.BackColor = Color.Coral
    End Sub

    Private Sub Button3_MouseLeave(sender As Object, e As EventArgs)
Handles Button3.MouseLeave
        Button3.BackColor = Color.LightGray
    End Sub

    Private Sub Button2_MouseEnter(sender As Object, e As EventArgs)
Handles Button2.MouseEnter

        Button2.BackColor = Color.DodgerBlue
    End Sub

    Private Sub Button2_MouseLeave(sender As Object, e As EventArgs)
Handles Button2.MouseLeave
        Button2.BackColor = Color.Gainsboro
    End Sub

    Private Sub Button1_MouseEnter(sender As Object, e As EventArgs)
Handles Button1.MouseEnter
        Button1.BackColor = Color.DodgerBlue
    End Sub

    Private Sub Button1_MouseLeave(sender As Object, e As EventArgs)
Handles Button1.MouseLeave
        Button1.BackColor = Color.Gainsboro
    End Sub

    Private Sub SerialPort2_DataReceived(sender As Object, e As
IO.Ports.SerialDataReceivedEventArgs) Handles SerialPort2.DataReceived
        Dim val As String
        val = CStr(SerialPort2.ReadByte)
        If val = 35 Then
            avi1 = 2 'inicia dato puerta
            Label3.Text = ""
        ElseIf val = 37 Then
            avi1 = 3 'termina dato puerta inicia dato tarjeta
        ElseIf val = 64 Then
            avi1 = 4
        End If

        If avi1 = 2 And val <> 35 Then
            Label3.Text += val
            Label4.Text = ""
            cont = 0
        ElseIf avi1 = 3 And cont < 5 And val <> 37 Then
            Label4.Text += CStr(Hex(val))
            cont = cont + 1
        End If
        If avi1 = 4 Then
            If hilo = 0 Then
```

```
                hilo = 1
                If acti.ThreadState <> ThreadState.Initialized Then
                    acti.Start()
                End If
            End If
            val = ""
            avi1 = 5
        End If

    End Sub

    Private Sub PersonalBindingNavigatorSaveItem_Click(sender As Object, e
As EventArgs) Handles PersonalBindingNavigatorSaveItem.Click
        Me.Validate()
        Me.PersonalBindingSource.EndEdit()
        Me.TableAdapterManager.UpdateAll(Me.PERSONALDataSet)

    End Sub
    Private Sub enviar(val As String)
        Dim valor As Byte() = New Byte(0) {}
        valor(0) = val
        If SerialPort2.IsOpen Then
            SerialPort2.Write(valor, 0, valor.Length)
        Else
            MsgBox("no conectado")
        End If
    End Sub
    Private Sub Button4_Click(sender As Object, e As EventArgs) Handles
Button4.Click
        SerialPort2.Close()
        Label2.Text = "DESCONECTADO"
        Panel2.BackColor = Color.Red
        Panel10.BackColor = Color.Yellow
        Panel11.BackColor = Color.Yellow
        Panel12.BackColor = Color.Yellow
        Panel3.BackColor = Color.Yellow
        Panel5.BackColor = Color.Yellow
        Panel7.BackColor = Color.Yellow
        Panel9.BackColor = Color.Yellow
    End Sub
    Private Sub buscarport()
        Try
            ComboBox1.Items.Clear()
            For Each puerto As String In My.Computer.Ports.SerialPortNames
                ComboBox1.Items.Add(puerto)
            Next
            If ComboBox1.Items.Count > 0 Then
                ComboBox1.SelectedItem = 0
            Else
                MsgBox("no encontrado puerto serial")
            End If

        Catch ex As Exception

        End Try
    End Sub

    Private Sub Button5_Click(sender As Object, e As EventArgs) Handles
Button5.Click
        acti = New Threading.Thread(AddressOf Me.activacion)
```

```
        Me.PersonalTableAdapter.FillBy(Me.PERSONALDataSet.personal, "")
        Try
            With SerialPort2
                .BaudRate = 9600
                .Parity = IO.Ports.Parity.None
                .DataBits = 8
                .StopBits = 1
                .PortName = ComboBox1.Text
                .Open()
                If SerialPort2.IsOpen Then
                    Label2.Text = "CONECTADO"
                    Panel2.BackColor = Color.Green
                    Panel10.BackColor = Color.Blue
                    Panel11.BackColor = Color.Blue
                    Panel12.BackColor = Color.Blue
                    Panel3.BackColor = Color.Blue
                    Panel5.BackColor = Color.Blue
                    Panel7.BackColor = Color.Blue
                    Panel9.BackColor = Color.Blue

                Else
                    MsgBox("NO SE PUDO CONECTAR")
                End If

            End With
        Catch ex As Exception

        End Try
    End Sub
    Private Sub activacion()
        While True
            For i As Integer = 1 To tam Step 1
                Label1.Text += CStr(i)
                If Label4.Text = PersonalTableAdapter.GetData(i).CODIGO()
Then
                    posicion = i
                    Label1.Text += CStr(posicion)
                    If Label3.Text = "495048" Then
                        ACCESO = "ACCESO1"
                        If "si" =
PersonalTableAdapter.GetData(posicion).ACCESO1() Then
                            enviar(&H79)
                            Panel12.BackColor = Color.Green
                            Threading.Thread.Sleep(500)
                            enviar(&H7A)
                            Panel12.BackColor = Color.Blue
                            confirmacion = "si"
                            Exit For
                        ElseIf "no" =
PersonalTableAdapter.GetData(posicion).ACCESO1() Then
                            Panel12.BackColor = Color.Red
                            Threading.Thread.Sleep(500)
                            Panel12.BackColor = Color.Blue
                            confirmacion = "no"
                            Exit For
                        End If
                    ElseIf Label3.Text = "494948" Then
                        ACCESO = "ACCESO2"
                        If "si" =
PersonalTableAdapter.GetData(posicion).ACCESO2() Then
```

```
                            enviar(&HC)
                            Panel10.BackColor = Color.Green
                            Threading.Thread.Sleep(500)
                            enviar(&HE)
                            Panel10.BackColor = Color.Blue
                            confirmacion = "si"
                            Exit For
                        ElseIf "no" =
PersonalTableAdapter.GetData(posicion).ACCESO2() Then
                            Panel10.BackColor = Color.Red
                            Threading.Thread.Sleep(500)
                            Panel10.BackColor = Color.Blue
                            confirmacion = "no"
                            Exit For
                        End If
                    ElseIf Label3.Text = "495248" Then
                        ACCESO = "ACCESO3"
                        If "si" =
PersonalTableAdapter.GetData(posicion).ACCESO3() Then
                            enviar(&H8C)
                            Panel11.BackColor = Color.Green
                            Threading.Thread.Sleep(500)
                            enviar(&H8E)
                            Panel11.BackColor = Color.Blue
                            confirmacion = "si"
                            Exit For
                        ElseIf "no" =
PersonalTableAdapter.GetData(posicion).ACCESO3() Then
                            Panel11.BackColor = Color.Red
                            Threading.Thread.Sleep(500)
                            Panel11.BackColor = Color.Blue
                            confirmacion = "no"
                            Exit For
                        End If
                    ElseIf Label3.Text = "494848" Then
                        ACCESO = "ACCESO4"
                        If "si" =
PersonalTableAdapter.GetData(posicion).ACCESO4() Then
                            enviar(&H7)
                            Panel7.BackColor = Color.Green
                            Threading.Thread.Sleep(500)
                            enviar(&H8)
                            Panel7.BackColor = Color.Blue
                            confirmacion = "si"
                            Exit For
                        ElseIf "no" =
PersonalTableAdapter.GetData(posicion).ACCESO4() Then
                            Panel7.BackColor = Color.Red
                            Threading.Thread.Sleep(500)
                            Panel7.BackColor = Color.Blue
                            confirmacion = "no"
                            Exit For
                        End If
                    ElseIf Label3.Text = "495448" Then
                        ACCESO = "ACCESO5"
                        If "si" =
PersonalTableAdapter.GetData(posicion).ACCESO5() Then
                            enviar(&H47)
                            Panel5.BackColor = Color.Green
                            Threading.Thread.Sleep(500)
```

```
                            enviar(&H48)
                            Panel5.BackColor = Color.Blue
                            confirmacion = "si"
                            Exit For
                        ElseIf "no" =
PersonalTableAdapter.GetData(posicion).ACCESO5() Then
                            Panel5.BackColor = Color.Red
                            Threading.Thread.Sleep(500)
                            Panel5.BackColor = Color.Blue
                            confirmacion = "no"
                            Exit For
                        End If
                    ElseIf Label3.Text = "495148" Then
                        ACCESO = "ACCESO6"
                        If "si" =
PersonalTableAdapter.GetData(posicion).ACCESO6() Then
                            enviar(&H13)
                            Panel3.BackColor = Color.Green
                            Threading.Thread.Sleep(500)
                            enviar(&H14)
                            Panel3.BackColor = Color.Blue
                            confirmacion = "si"
                            Exit For
                        ElseIf "no" =
PersonalTableAdapter.GetData(posicion).ACCESO6() Then
                            Panel3.BackColor = Color.Red
                            Threading.Thread.Sleep(500)
                            Panel3.BackColor = Color.Blue
                            confirmacion = "no"
                            Exit For
                        End If
                    ElseIf Label3.Text = "495257" Then
                        ACCESO = "ACCESO7"
                        If "si" =
PersonalTableAdapter.GetData(posicion).ACCESO7() Then
                            enviar(&H3F)
                            Panel9.BackColor = Color.Green
                            Threading.Thread.Sleep(500)
                            enviar(&H42)
                            Panel9.BackColor = Color.Blue
                            confirmacion = "si"
                            Exit For
                        ElseIf "no" =
PersonalTableAdapter.GetData(posicion).ACCESO7() Then
                            Panel9.BackColor = Color.Red
                            Threading.Thread.Sleep(500)
                            Panel9.BackColor = Color.Blue
                            confirmacion = "no"
                            Exit For
                        End If
                    End If
                End If
            Next
            If posicion <> 0 And
PersonalTableAdapter.GetData(posicion).CODIGO() <> "" Then

Me.Tabla1TableAdapter.INSERTAR(FECHA:=System.DateTime.Now.ToString(),
CODIGO:=PersonalTableAdapter.GetData(posicion).CODIGO(),
NOMBRES:=PersonalTableAdapter.GetData(posicion).NOMBRE(),
```

```
APELLIDOS:=PersonalTableAdapter.GetData(posicion).APELLIDO(),
ACCESO:=ACCESO, CONFIRMACION:=confirmacion)
                Me.Tabla1TableAdapter.Fill(Me.HISTORIALDataSet.Tabla1)
            End If
            SerialPort2.RtsEnable = True
            SerialPort2.DiscardInBuffer()
            SerialPort2.DiscardOutBuffer()
            hilo = 0
            posicion = 0
            While hilo = 0

            End While
        End While
    End Sub

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
        Form6.Show()
    End Sub
End Class
```

FORM 5- Obtención de código de tarjetas.

```
Public Class Form5

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
        My.Forms.Form2.CODIGOTextBox.Text = TextBox1.Text
        My.Forms.Form2.Show()
        SerialPort1.Close()
        Label1.Text = "DESCONECTADO"
        Panel1.BackColor = Color.Red
        TextBox1.Clear()
        Me.Hide()

    End Sub

    Private Sub TextBox1_TextChanged(sender As Object, e As EventArgs)
Handles TextBox1.TextChanged

    End Sub
    Private Sub Button3_Click(sender As Object, e As EventArgs) Handles
Button3.Click
        Try
            With SerialPort1
                .BaudRate = 9600
                .Parity = IO.Ports.Parity.None
                .DataBits = 8
                .StopBits = 1
                .PortName = ComboBox1.Text
                .Open()
                If SerialPort1.IsOpen Then
                    Label1.Text = "CONECTADO"
                    Panel1.BackColor = Color.Green
                Else
                    MsgBox("NO SE PUDO CONECTAR")
                End If
```

```
            End With
        Catch ex As Exception
            MsgBox("NO SE PUDO CONECTAR")
        End Try
    End Sub

    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles
Button2.Click
        SerialPort1.Close()
        Label1.Text = "DESCONECTADO"


        Panel1.BackColor = Color.Red
    End Sub

    Private Sub SerialPort1_DataReceived(sender As Object, e As
IO.Ports.SerialDataReceivedEventArgs) Handles SerialPort1.DataReceived
        Dim buffer As String = ""
        Dim cont As Integer = 0
        buffer = SerialPort1.ReadByte
        If cont < 5 Then
            TextBox1.Text += CStr(Hex(buffer))
        End If
        cont = cont + 1
        If cont = 10 Then
            cont = 0
            TextBox1.Clear()
        End If
    End Sub
    Private Sub Form5_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
        buscarport()
        CheckForIllegalCrossThreadCalls = False
    End Sub

    Private Sub ComboBox1_SelectedIndexChanged(sender As Object, e As
EventArgs) Handles ComboBox1.SelectedIndexChanged

    End Sub

    Private Sub Button4_Click(sender As Object, e As EventArgs) Handles
Button4.Click

    End Sub
    Private Sub buscarport()
        Try
            ComboBox1.Items.Clear()
            For Each puerto As String In My.Computer.Ports.SerialPortNames
                ComboBox1.Items.Add(puerto)
            Next
            If ComboBox1.Items.Count > 0 Then
                ComboBox1.SelectedItem = 0
            Else
                MsgBox("no encontrado puerto serial")
            End If

        Catch ex As Exception

        End Try
    End Sub
```

```
End Class
```

FORM 6- Visualización del historial del uso de accesos.

```
Public Class Form6

    Private Sub Tabla1BindingNavigatorSaveItem_Click(sender As Object, e As
EventArgs) Handles Tabla1BindingNavigatorSaveItem.Click
        Me.Validate()
        Me.Tabla1BindingSource.EndEdit()
        Me.TableAdapterManager.UpdateAll(Me.HISTORIALDataSet)

    End Sub

    Private Sub Form6_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
        'TODO: This line of code loads data into the
'HISTORIALDataSet.Tabla1' table. You can move, or remove it, as needed.
        Me.Tabla1TableAdapter.Fill(Me.HISTORIALDataSet.Tabla1)

    End Sub

    Private Sub Tabla1DataGridView_CellContentClick(sender As Object, e As
DataGridViewCellEventArgs) Handles Tabla1DataGridView.CellContentClick

    End Sub
End Class
```