"Responsabilidad con pensamiento positivo"

UNIVERSIDAD TECNOLÓGICA ISRAEL

TRABAJO DE TITULACIÓN EN OPCIÓN AL GRADO DE: INGENIERO EN ELECTRÓNICA DIGITAL Y TELECOMUNICACIONES

TEMA:

IMPLEMENTACIÓN DE UN PROTOTIPO DE CONTROL DE REGISTROS DE ACCESO MEDIANTE TECNOLOGÍA NFC

AUTOR:

Samuel Esteban Jácome Berrones

TUTOR:

Ing. Flavio David Morales Arévalo, Mg

QUITO, ECUADOR

2018

DECLARACIÓN

Yo, Samuel Esteban Jácome Berrones, estudiante de la carrera de Electrónica Digital y Telecomunicaciones, perteneciente a la Universidad Tecnológica Israel, declaro que el contenido aquí descrito es de mi autoría, y de mi absoluta responsabilidad legal.

Quito DM, enero de 2018

Samuel Esteban Jácome Berrones

C.I: 1720907664

UNIVERSIDAD TECNOLÓGICA ISRAEL

APROBACIÓN DEL TUTOR

En mi calidad de tutor del trabajo de titulación certifico:

Que el trabajo de titulación "**IMPLEMENTACIÓN DE UN PROTOTIPO DE CONTROL DE REGISTROS DE ACCESO MEDIANTE TECNOLOGÍA NFC**", presentado por el Sr. Samuel Esteban Jácome Berrones, estudiante de la carrera de Electrónica Digital y Telecomunicaciones, reúne los requisitos y méritos suficientes para ser sometido a la evaluación del Tribunal de Grado, que se designe, para su correspondiente estudio y calificación.

Quito, D.M. Febrero 17 de 2018

TUTOR

……

Ing. Flavio Morales, Mg

AGRADECIMIENTO

Al culminar el transcurso de mi carrera quiero agradecer en primer lugar a Dios por permitirme finalizar mis estudios con salud y vida, además por haberme bendecido con tantas personas las cuales siempre han estado apoyándome.

Quiero agradecer a mis padres, Amarilis Berrones y José Jácome, por ser los pilares en mi vida por educarme y apoyarme en cada momento difícil, porque gracias a su confianza he logrado cumplir mis metas como persona y como profesional.

Quiero agradecer a mi hermano Gabriel por darme ese impulso para seguir siempre y porque sé que logrará todas sus metas y siempre estaré contigo para apoyarte como tú lo has hecho conmigo.

Quiero agradecer a mi novia Isabel por estar siempre a mi lado y ser tan única porque día a día me ha apoyado para seguir adelante con sus consejos.

Quiero agradecer a la Universidad Tecnológica Israel por los conocimientos y valores impartidos en el transcurso de mi carrera, los cuales han permitido prepararme como persona y profesionalmente.

SAMUEL ESTEBAN JÁCOME BERRONES

DEDICATORIA

Este proyecto tiene dedicatoria exclusiva a mis padres, por guiarme siempre en los caminos de Dios, Amarilis Berrones y José Jácome, porque sin importar cuán difícil sea el momento, han estado junto a mí, son personas increíbles y todos disfrutaremos de este gran paso en mi vida Profesional, ya que gracias a ellos nunca decaí sino que estoy culminando mis estudios con su bendición y aliento.

Contenido

Resumen 11

INTRODUCCIÓN 12

Antecedentes 12

Planteamiento del Problema 13

Formulación del problema 13

Justificación 13

Objetivo General: 14

Objetivos Específicos: 14

Descripción de capítulos 15

1. CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA 16

1.1. Antecedentes 16

1.2. Metodológico 17

1.3. Marco Teórico 17

1.3.1. Introducción a la tecnología NFC 17

1.3.2. Definición de la Tecnología 18

1.3.3. Historia de la Tecnología NFC 19

1.3.4. Orígenes de la Tecnología NFC 20

1.3.5. Usos de la Tecnología NFC 20

1.3.6. Especificaciones técnicas 21

1.3.7. Estandarización de NFC 23

1.3.8. NDEF, Formato de intercambio de Datos NFC 23

1.3.9. Interfaces y protocolos NFC 30

1.3.10. Modos de Funcionamiento 32

1.3.11. Establecimiento de la comunicación NFC 34

1.3.12. Módulo NFC PN532 34

1.3.13. ASPECTOS DE SEGURIDAD NFC 37

1.3.14. ARDUINO 37
1.3.15. Arduino MEGA 38
1.3.16. Arduino NANO 40
1.3.17. Servomotor SG90 41
1.4. Marco Conceptual 43
2. CAPÍTULO 2 PROPUESTA 44
2.1. Descripción de la comunicación (usuario: Administrador) para el sistema de control de accesos 45
2.2. Diseño de placa de control de los módulos NFC y servomotores 50
2.2.1. Placa Arduino MEGA 2560 51
2.2.2. Módulo NFC Pn532 51
2.2.3. Servomotores 52
2.2.4. Diseño Esquemático 53
2.2.5. Diagrama de flujo del sistema de acceso 56
2.2.6. Diseño del Sistema 57
2.2.7. Descripción de las etapas de funcionamiento 59
2.3. Diseño del Prototipo 60
2.4. Diseño de la aplicación móvil 62
2.4.1. Interfaz Visible 62
2.5. Diseño de aplicación en la PC 63
2.5.1. Interfaz gráfica 63
2.6. Programación Arduino NANO 64
2.7. Programación Arduino MEGA 64
2.8. Croquis del prototipo de las oficinas 66
3. CAPÍTULO III IMPLEMENTACIÓN 70
3.1. Desarrollo 70
3.1.1. Fabricación de la placa 70

3.2. Implementación del Sistema de Acceso 72
3.3. Implementación de aplicación móvil 73
3.3.1. Inicio de aplicaciones 73
3.3.2. Aplicación en App Inventor 2 78
3.4. Pruebas de Funcionamiento 79
3.5. Análisis de Resultados 80
3.6. Implementación Final 81
3.6.1. Módulos NFC instaladas en el Prototipo 81
3.7. Presupuesto 82
4. CONCLUSIONES 83
5. RECOMENDACIONES 84
6. REFERENCIAS BIBLIOGRÁFICAS 85
7. ANEXOS 87

Índice De Figuras

Figura. 1. 1. Empresas que apoyan la Tecnología Nfc 19
Figura. 1. 2. Logo tecnología Nfc 21
Figura. 1. 3. Inducción del campo magnético de Nfc 22
Figura. 1. 4. Formato de un registro Ndef 25
Figura. 1. 5. Mensaje Ndef con varios registros 29
Figura. 1. 6. Detección de radio frecuencia, modo selección de dispositivos Nfcio-2 31
Figura. 1. 7. Modo de comunicación Pasiva 32
Figura. 1. 8. Modo de comunicación Activa 33
Figura. 1. 9. Módulo Pn532 35
Figura. 1. 10. Arduino y sus usos 38
Figura. 1. 11. Arduino Mega 39
Figura. 1. 12. Arduino Nano 40
Figura. 1. 13. Descripción de Pines 41
Figura. 1. 14. Composición de un Servomotor 42
Figura. 1. 15. Servomotor Sg90 42
Figura. 1. 16. Comunicación entre dispositivos 45
Figura. 1. 17. Señal modulada en amplitud, M=2 47
Figura. 1. 18. Codificación manchester 48
Figura. 2. 1. Diagrama esquemático del sistema 50
Figura. 2. 2. Placa Arduino Mega 2560 51
Figura. 2. 3. Módulo Pn532 52
Figura. 2. 4. Conexiones Servomotor 53
Figura. 2. 5. Diagrama conexiones arduino Mega- Nano- Servomotor- Módulo Nfc 54
Figura. 2. 6. Diagrama de Conexiones Simplificado 55
Figura. 2. 7. Diagrama diseño de placa Pcb 56
Figura. 2. 8. Diagrama de flujo del sistema 56
Figura. 2. 9. Diagrama general del prototipo 57
Figura. 2. 10. Diagrama en bloques de las etapas de funcionamiento 59
Figura. 2. 11. Descripción de fases 60
Figura. 2. 12. Parte frontal del prototipo 60
Figura. 2. 13. Parte izquierda del prototipo 61

Figura. 2. 14. Diseño del prototipo 61
Figura. 2. 15. Interfaz visible del administrador 62
Figura. 2. 16. Aplicación en Visual Basic 63
Figura. 2. 17. Registros de entradas y salidas en Microsoft Excel 64
Figura. 2. 18. Descripción inicialización del programa 65
Figura. 2. 19. Almacenamiento de datos 65
Figura. 2. 20. Comunicación con la Pc 66
Figura. 2. 21. Opción de recepción Módulo Nfc 66
Figura. 2. 22. Croquis del prototipo 67
Figura. 3. 1. Calentamiento de la impresión sobre la baquelita 70
Figura. 3. 2. Impresión del diagrama en la baquelita 70
Figura. 3. 3. Perforación de la placa 71
Figura. 3. 4. Elementos colocados en la placa 71
Figura. 3. 5. Placa terminada 72
Figura. 3. 6. Implementación del Arduino Mega con el Arduino Nano 72
Figura. 3. 7. Sistema completo para el funcionamiento del prototipo 73
Figura. 3. 8. Encendido del Nfc en el móvil 74
Figura. 3. 9. Aplicaciones instaladas 74
Figura. 3. 10. Interfaz visible del administrador 75
Figura. 3. 11. Interfaz visible del gerente 76
Figura. 3. 12. Interfaz visible del diseñador 76
Figura. 3. 13. Interfaz visible de la oficina de implementación 77
Figura. 3. 14. Programación y diseño de la aplicación 78
Figura. 3. 15. Programación de la aplicación y uus diferentes opciones 79
Figura. 3. 16. Implementación frontal final 81
Figura. 3. 17. Implementación superior final 81

Índice de Tablas

Tabla. 1. 1. Valores de Tnf 27
Tabla. 1. 2. Códigos de transferencia De Nfc 33
Tabla. 1. 3. Datos de referencia 35
Tabla. 1. 4. Descripción de los pines 36
Tabla. 1. 5. Arduinos más usados 37
Tabla. 1. 6. Arduinos más pequeños 40
Tabla. 1. 7. Datos decimales convertidos a binarios 46
Tabla. 1. 8, Tabla de datos binarios transmitidos 46
Tabla. 2. 1. Códigos ocultos de acceso 58
Tabla. 3. 1. Pruebas de funcionamiento 79
Tabla. 3. 2. Presupuesto 82

Resumen

El proyecto “Implementación de un prototipo de control de registros de acceso mediante tecnología NFC” se realizó con el fin de supervisar los ingresos a las oficinas y controlar la puntualidad de los trabajadores mediante un prototipo. Se pudo visualizar a través de la computadora los horarios, fechas y áreas de los ingresos.

Este sistema esta diferenciado con otros por tener una aplicación móvil (exclusiva de la empresa) que fue la que permitió aceptar o negar el ingreso a personas que se integren a sus puestos de trabajo y así evitar cualquier acto delictivo.

El control de los accesos permitió que las oficinas estén seguras, ya que al integrar esto se contrarresto cualquier ingreso no autorizado evitando así los robos de cualquier artículo cuando las oficinas estén cerradas o abiertas y supervisando si alguien intento ingresar a alguna oficina.

La integración del sistema en el prototipo ayudó a mantener las instalaciones protegidas ante eventos en el área de entregas ya que no permitió que personas ajenas a las oficinas ingresen ya sea por recoger su pedido o intenciones de robar.

PALABRAS CLAVES: Control, registros, accesos, NFC, NDEF.

Abstract

The project "Implementation of a prototype of control of access records using NFC technology" was carried out with the purpose of supervising the incomes to the offices and controlling the punctuality of the workers through a prototype. It was possible to visualize the schedules, dates and areas of the income through the computer.

This system is differentiated with others by having a mobile application (exclusive of the company) that was the one that allowed to accept or deny the entry to people who integrate to their jobs and thus avoid any criminal act.

The control of the accesses allowed the offices to be safe, since by integrating this counterbalanced any unauthorized entry thus preventing the theft of any item when the offices are closed or open and supervising if someone tries to enter an office.

The integration of the system in the prototype helped to keep the facilities protected against events in the delivery area as it did not allow people outside the offices to enter either to pick up their order or intentions to steal.

KEYWORDS: Control, records, access, NFC, NDEF.

INTRODUCCIÓN

Los sistemas de acceso han cambiado mucho desde el tiempo con respecto a los métodos para asegurarse, que han permitido que el usuario sienta más confianza con el paso de los años.

Existiendo diversos métodos de acceso que las empresas utilizan para su seguridad, por ejemplo mediante tarjetas, huellas dactilares, iris de ojos entre otros.

Pero la tecnología avanza cada día y con ello nuevas ideas se van incrementando dando además de solo restricciones, alarmas y otros medios.

Antecedentes

El prototipo está diseñado para simular una empresa de reparación de computadoras e instalación de software, de forma individual como reparaciones para empresas, la cual contiene profesionales especializados para cualquier tipo de problema en un computador, y lo cual requiere un nivel alto de seguridad.

La falta de control de accesos en cada oficina hace que sea necesario el implementar el sistema en el prototipo, que además muestre la administración y control del personal, siendo indispensable integrar los dispositivos tecnológicos y proteger los bienes de la empresa y equipos de los clientes.

Finalmente a través del diseño y elaboración del plan para el mejoramiento del control de la seguridad y normativas de los horarios de ingreso de los trabajadores se consideró muy necesario el sistema de control.

Planteamiento del Problema

El prototipo está basado en un grupo de oficinas en las cuales su primera falencia era la inseguridad de los bienes de la empresa, ya que no contaba con un sistema para controlar los accesos de los trabajadores ni su horario de ingreso.

Muchas empresas hoy en día se encuentran enfocadas en su trabajo y no se informan correctamente sobre el uso de los sistemas de control de acceso. Por lo cual tienen un bajo concepto de estos medios de seguridad, juzgando su valor económico o la calidad. Siendo información completamente errónea ya que día a día la tecnología avanza y economiza más y más sus productos

Formulación del problema

Las oficinas de empresas en el Ecuador tienen gran vulnerabilidad con respecto a la delincuencia, esto ha conllevado a la inseguridad de los clientes y los equipos de la empresa. Se puede comprobar la necesidad de obtener un sistema de control de accesos mediante tecnología NFC (COMUNICACIÓN DE CAMPO CERCANO) para un mayor cuidado y administración de los equipos.

El prototipo da a conocer la necesidad de integrar el control de accesos para obtener mayor seguridad y control de las personas que ingresan a las oficinas, sus horarios e intentos de ingresar a lugares no permitidos.

Justificación

Las diversas formas de robo en estos tiempos preocupan a la población ya que los ciudadanos no se sienten en un ambiente de confianza. Por este motivo el introducir un control de accesos que permita que los trabajadores accedan y también controlen sus horarios según su puesto de trabajo, conlleva a ser una necesidad de la empresa protegiendo cualquier anomalía que se pudiera presentar.

Estos aspectos mencionados se han analizado en el prototipo para ser implementados en el sistema de control de accesos mediante tecnología NFC. Se procede a instalar los módulos NFC en cada puerta los cuales permitirán o no el acceso de las personas dentro del área establecida.

Adicionalmente se colocaron servomotores para simular la apertura de las puertas, y mediante una aplicación en el celular la cual contiene un código cada una, se podrá ingresar a las oficinas.

Esto se complementó con una aplicación realizada en Visual Basic para la computadora, que permite saber los movimientos que se produjeron durante el día. Como la hora, fecha, la puerta que fue abierta o intentada abrir y el usuario que lo realizó.

Objetivo General:

- Implementar un prototipo de identificación de accesos con tecnología NFC y control de registros.

Objetivos Específicos:

- Establecer las técnicas y parámetros para la implementación de la tecnología NFC en nuevos sistemas de seguridad de accesos.
- Investigar los módulos necesarios para la implementación del sistema de control mediante Arduino para la programación y una aplicación en Android para el acceso.
- Diseñar una aplicación en Android para el intercambio de datos con un receptor NFC
- Implementar un prototipo que identifique mediante la observación en un PC, el control de accesos acerca de los días permitidos, horarios.
- Realizar pruebas de implementación y el control de los registros obtenidos

Descripción de capítulos

Dentro del documento escrito se tiene la introducción, en donde se detalla los antecedentes, problemática, objetivo general y específicos además de la hipótesis, siguiendo después con los siguientes capítulos:

En el primer capítulo está la fundamentación teórica, la cual consta de marco teórico y metodológico, detallando ahí las tecnologías usadas en el control de accesos y otros elementos que forman parte en el prototipo para un óptimo funcionamiento.

El capítulo dos contiene el diseño del prototipo, diagramas en bloque, flujo, los cálculos realizados, las placas en las que se programó y otros materiales utilizados.

El capítulo tres, contiene todo acerca de la implementación del control de accesos en el prototipo, observando así el desarrollo, la programación de los Arduinos, el elaborado de la aplicación móvil y los diagramas explicativos. Además, se observará las pruebas que se obtuvieron del funcionamiento y así según los resultados obtenidos se garantice el correcto funcionamiento cumpliendo así con cada objetivo planteado.

Finalmente, se encuentran las conclusiones, recomendaciones, la bibliografía y anexos. Encontrando ahí todo lo referente a los manuales de usuario, técnico y los Datasheets de cada componente teniendo así una correcta manipulación del sistema.

1. CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

Existen diversas formas de conocer acerca de sistemas de control de accesos mediante tecnología NFC (*Near Field Communication*), y sus principios de funcionamiento. Libros, páginas web y tesis describen de formas muy generales los procedimientos de conexión de los dispositivos NFC. Siendo los sistemas de control de accesos implementados en empresas, escuelas, colegios, universidades, departamentos. De varios documentos importantes vale la pena mencionar unos pocos. La compañía NFC Forum, fundada en marzo de 2004 gracias a la colaboración de sus miembros en la determinación de las características y estándares de NFC. La cual en el año 2006 realiza su primer lanzamiento mundial en la especificación técnica de la tecnología NFC.

La compañía tecnológica IDTrónica Sistemas ha lanzado al mercado μAccess, un control de accesos basado en la tecnología NFC y siendo uno de los pocos productos en el mercado basados en esta tecnología. Siendo NFC adaptada a la domótica con el fin de mejorar la accesibilidad de sus usuarios.

NFC Forum ofrece varias capacitaciones de uso de la tecnología, en la que los miembros de la compañía preparan a sus estudiantes para que en un futuro, ellos sean quienes innoven en nuevos diseños de compatibilidad y seguridad.

El diseño de un control de accesos mediante comunicación con la tecnología NFC es un proyecto abierto de ingeniería, el cual puede ser realizado de diversas formas, sin afectar el rendimiento habitual del NFC. Este documento propone un procedimiento eficaz y entendible en la elaboración de sistemas de control de acceso, siendo así una guía para la innovación o mejoramiento.

1.1. Antecedentes

La tecnología con su gigantesco crecimiento diario, siempre ha buscado el bienestar humano, automatizando procesos para que no haya la necesidad de un manejo manual que pueda ocasionar errores, de tal manera que brinda comodidad al simplificar la vida de las personas. Entre estos avances aparece el teléfono celular, que en un principio se creó como un dispositivo netamente de comunicación para dar movilidad a las personas.

Pero su crecimiento acelerado ha hecho que se convierta en una herramienta diaria e indispensable capaz de converger un sinnúmero de aplicaciones y tecnologías dentro del mismo, algunos ejemplos son NFC, Bluetooth, Wifi, etc.

1.2. Metodológico

La investigación aplicada confirmatoria es la metodología utilizada en este proyecto ya que su objetivo es examinar que el funcionamiento de campo y natural tenga la eficacia correcta para la validez del documento.

El usuario mediante la creación de varios escenarios mostrará la calidad y eficiencia del proyecto por lo cual se basa en un método experimental dejando así un resultado satisfactorio o negativo.

1.3. Marco Teórico

Una definición general de control de acceso, hace alusión al mecanismo en función de la identificación ya autenticada que provee el acceso a datos o recursos. Es común encontrar controles de acceso para múltiples aplicaciones y en diversas formas. Por ejemplo, es usual encontrar controles de acceso por software cuando ingresamos una contraseña para acceder a un e-Mail, o usar la huella dactilar para desbloquear un celular, en estos casos se usa un control de acceso para acceder a información. Otro ejemplo común es cuando se usa identificación biométrica para acceder a una oficina, en estos casos se usa la seguridad electrónica para administrar el acceso a recursos físicos.

Para este caso en particular, el control de acceso es la habilidad de conceder o denegar el acceso a un espacio físico (áreas restringidas según el tipo de trabajador).

1.3.1. Introducción a la tecnología NFC

La tecnología inalámbrica NFC, por sus siglas en inglés *Near Field Communication*, aparece como un progreso en la convergencia de aplicaciones dentro del teléfono móvil al

ofrecer los servicios de las tarjetas inteligentes (pagos, identificaciones, accesos a datos o recursos.) y las ventajas de las tecnologías inalámbricas de corto alcance mediante su uso.

Una de las características más significativas de NFC es su compatibilidad con las tecnologías inalámbricas ya existentes como Bluetooth y RFID, lo que hace aún más interesante su uso e incrementa el interés de más y más empresas en su inversión y desarrollo, por eso se han puesto en marcha proyectos pilotos, principalmente en Europa, para probar su desenvolvimiento con los llamados servicios de proximidad, aquellos servicios a los que se puede tener acceso con sólo acercar el teléfono móvil a un lector o terminal que ofrezca este servicio.

A pesar de que ya existen una variedad de tecnologías de corto alcance como Bluetooth, RFID, el interés que NFC está generando tiene que ver con la potencialidad que promete para el desarrollo e implementación de novedosas e interesantes aplicaciones, como el pago a través del celular, control de accesos a entidades públicas y privadas, con un nivel de seguridad mejorado y también permitiendo que la experiencia de los usuarios en servicios ya existentes sea atractiva, es decir que su interacción guste a los usuarios para que a la hora de elección se inclinen hacia esta.

Es por tal motivo que con la realización de este proyecto de titulación, se da a conocer el entorno que conforma la tecnología NFC, brindando, más allá de un estudio, una experiencia real al implementar un prototipo que permita sacar provecho de sus características y ventajas para la satisfacción del usuario y que así se juzgue de manera objetiva si la creación de NFC estará justificada.

1.3.2. Definición de la Tecnología

Near Field Communication o Comunicación de Campo Cercano, por sus siglas NFC, es una tecnología de comunicación inalámbrica de corto alcance que permite el intercambio bidireccional de datos entre dispositivos a una distancia corta aproximadamente de 10 cm.

La idea de desarrollar esta tecnología fue crear un nuevo protocolo que preste compatibilidad con las tecnologías sin contacto de corto alcance ya existentes, razón por la que NFC es una extensión simple del estándar ISO/IEC 144435 de tarjetas de proximidad

(tarjetas RFID sin contacto) que combina la interface de una tarjeta inteligente y de un lector dentro de un mismo dispositivo.

Un dispositivo NFC puede comunicarse con cualquier tarjeta inteligente y lector, existentes dentro del estándar ISO/IEC 14443 (estándar internacional relacionado con las tarjetas de identificación electrónicas, en especial las tarjetas de proximidad), tan bien como con otros dispositivos NFC, y es por lo tanto compatible con la infraestructura sin contacto ya en uso para la transportación pública, el control de accesos y para pagos, por ejemplo en el caso de Málaga en donde hay un proyecto piloto para la utilización de NFC en su transportación pública e identificación de trabajadores en empresas públicas. NFC está destinado principalmente para el uso en teléfonos celulares ya que no está orientada para la transmisión masiva de datos como Wi-Fi por ejemplo.

1.3.3. Historia de la Tecnología NFC

NFC (Comunicación de corto alcance) inició sus primeros avances gracias a Sony y Philips, quienes buscaban protocolos compatibles con todas las tecnologías. NFC fue aprobado en diciembre de 2003 como el estándar ISO 18092 (Define los modos de comunicación de corto alcance), para inmediatamente en marzo de 2004 aliarse a Nokia y crear NFC Forum (Compañía para la innovación de NFC).

Desde que surgió NFC Forum la tecnología fue consiguiendo grandes patrocinios de empresas como Google, Visa, At&t, PayPal.

Figura. 1. 1. Empresas que apoyaban la tecnología NFC

Fuente: (Contreras, 2012)

La tecnología NFC es compatible con tarjetas de proximidad, en las cuales no hace falta el contacto para establecer comunicación y es aceptado por el estándar ISO-14443 (Proximidad sin contacto).

La tecnología de corto alcance (NFC) como su nombre lo indica hace referencia a que su distancia de contacto debe ser máxima de 10cm para una comunicación óptima.

NFC es la unión de radio frecuencia (RFID) y de otras tecnologías interconectadas. Además su frecuencia es libre de uso es decir no necesita de pagos o internet para su funcionamiento.

Está orientada a una comunicación rápida por lo que su tasa de transferencia es de hasta 424kbit por segundo, lo cual no es mucha información, es útil para identificar y validar personas, instrumentos o herramientas.

NFC promete grandes avances para los siguientes años ya que su uso aumentará de manera que grandes aplicaciones podrían ser sustituidas, por ejemplo los códigos QR y el comercio electrónico. Siendo así de gran utilidad hasta para pagos, reemplazando a las tarjetas de crédito y el efectivo.

1.3.4. Orígenes de la Tecnología NFC

NFC tuvo su primer lanzamiento en el año 2006, en donde surgió la primera especificación técnica de esta tecnología. NFC Forum fue la organización encargada de regular las características y determinar los estándares de NFC, siendo apoyada por nuevas empresas como: Dell, Intel, Microsoft y Samsung.

Los creadores de la tecnología NFC tienen diversos objetivos a realizar para el futuro, uno de los cuales es enseñar el uso correcto y óptimo a la población

1.3.5. Usos de la Tecnología NFC

Su uso principal es el intercambio de datos en forma inalámbrica lo que permite que a su futuro tenga usos exclusivos de gran utilidad como son el pago y recogida e intercambio de información.

Figura. 1. 2. Logo Tecnología NFC

Fuente: (Simpson, 2017)

Identificación: La identificación es un medio utilizado en la mayoría de lugares de trabajo para una mayor seguridad, lo cual sería tan simple como acercar el teléfono con la aplicación NFC a un dispositivo receptor.

Sincronización instantánea de dispositivos: En caso de la sincronía de dispositivos la tecnología NFC es la más indicada, ya que no es necesario de emparejamientos como Bluetooth, por ejemplo para utilizar altavoces la sincronización es automática, solo ubicando los dos dispositivos a una distancia adecuada.

Automatización de acciones: Para realizar estas acciones se necesita de una tag o etiquetas NFC, las cuales son pequeñas tarjetas programables, que con solo pasar un móvil se puede cambiar diversas configuraciones como: hora de alarmas, cambio de sonidos, etc.

Pago con el teléfono móvil: Uno de los usos más eficientes de esta tecnología es el pago mediante el móvil, ya que es tan fácil y rápido realizar compras sin necesidad de tarjetas u otros medios.

Mediante estos métodos la tecnología NFC ha ingresado al mercado de una manera agresiva y a sido acogida por la mayoría de personas para facilitar diversas actividades, las cuales serán indispensables en poco tiempo.

1.3.6. Especificaciones técnicas

NFC fue aprobado como un estándar ISO/IEC (ISO (International Organization for Standardization) e IEC (International Electrotechnical Commission) el 08 de diciembre del 2003 y posteriormente como un estándar ECMA (Organización internacional basada en membresías de estándares para la comunicación y la información).

El estándar ISO/IEC 14443 se comunica vía inducción de campo magnético, donde dos lazos de antena son localizados dentro de cada campo cercano del otro, formando efectivamente un transformador núcleo de aire.

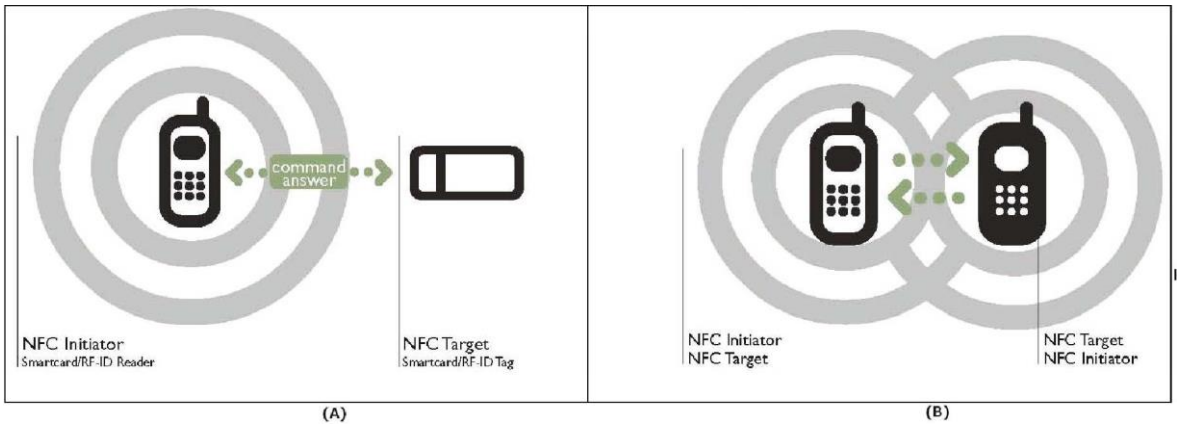


Figura. 1. 3. Inducción del campo magnético de NFC

Fuente: (internacional, Forum, 2017)

Opera dentro de la banda ISM (Industrial, Scientific and Medical) de radio frecuencia de 13,56 MHz disponible globalmente sin restricción y sin necesidad de licencia para su uso, con un ancho de banda de casi 2 MHz.

NFC es una tecnología de plataforma abierta estandarizada en la ISO/IEC 18092 y la ECMA-340. Estos estándares especifican los esquemas de modulación, codificación, velocidades de transferencia y formato de la trama de la interfaz RF de dispositivos NFC, así como los esquemas de inicialización y condiciones requeridas para el control de colisión de datos durante la inicialización para ambos modos de comunicación, activo y pasivo. También definen el protocolo de transporte, incluyendo los métodos de activación de protocolo y de intercambio de datos.

La distancia de trabajo con antenas compactas estándar es aproximadamente 20 cm, aunque generalmente efectivo es cercano a los 10 cm. Las velocidades de transmisión que soporta esta tecnología son de 106, 212, 424 u 848 kbits/s.

La comunicación NFC es bidireccional, por lo tanto los dispositivos NFC son capaces de transmitir y recibir datos al mismo tiempo. De esta manera, ellos pueden verificar el campo de Radio Frecuencia y detectar una colisión si la señal recibida no coincide con la señal transmitida.

1.3.7. Estandarización de NFC

Dentro de los estándares de NFC se ha establecido un formato común de datos para que los dispositivos NFC puedan compartir información entre sí. Estos estándares señalan las especificaciones que permiten la comunicación y son propiedad del NFC Forum, una asociación industrial sin fines de lucro encargada de regular la interacción inalámbrica y la interoperabilidad entre dispositivos NFC.

1.3.8. NDEF, Formato de intercambio de Datos NFC

NFC Forum ha definido un formato de datos común llamado NDEF, por sus siglas en inglés NFC Data Exchange Format, el cual puede ser usado para guardar y transportar diferentes tipos de elementos, que van desde cualquier objeto escrito *MIME* (Extensiones de correo de Internet de Propósitos múltiples) hasta documentos *RTD* (Definición del tipo de registro) ultra pequeños, tales como *URLs* (Localizador de recursos uniforme).

La Especificación NDEF define un formato de encapsulación de mensaje para el intercambio de datos entre dispositivos NFC o de un dispositivo NFC a una etiqueta NFC y las reglas para construcción de un mensaje NDEF válido y también de una cadena ordenada de registros NDEF. La diferencia entre una etiqueta y un dispositivo NFC es que la primera no permite una interacción con el usuario y por sí sola no podría mostrar ninguna información al usuario, además es pasiva es decir que no genera su propia energía de funcionamiento y necesita de un dispositivo activo para que funcione. En cambio un dispositivo NFC permite una interacción del usuario así como es el propio generador de su energía y a través de su campo de inducción puede estimular y generar la energía para el funcionamiento de los elementos pasivos.

NDEF es un formato binario ligero que puede encapsular una o más payloads (Es el conjunto de datos transmitidos, que es en realidad el mensaje enviado) de diferente tipo y tamaño dentro de la estructura de un solo mensaje. El payload está identificado por un tipo, una longitud y un identificador opcional.

- Longitud de la carga (payload): Indica el número de octetos de payload, es decir, indica la longitud de payload encapsulada en un registro. Se encuentra dentro de los primeros 8 octetos de un registro. El Campo PAYLOAD_LENGTH es un octeto para registros pequeños y cuatro octetos para registros normales. Los registros pequeños están indicados estableciendo el bit de la bandera SR (short record) en 1.

- Tipo de Payload: Indica la clase de datos que está siendo transportado en el payload de ese registro. El tipo del primer registro, por convención, debería proveer el contexto de procesamiento no solo para el primer registro sino para todo el mensaje NDEF. Los tipos de identificadores podrían ser *URIs* (Identificador de recurso uniforme), MIME o tipos específicos NFC (*NFC-specific*). Al identificar el tipo de carga útil, es posible despachar la carga para la aplicación del usuario apropiada.

- Identificador de Payload: La payload puede dar un identificador opcional en la forma de una URI absoluta o relativa; esto permite a las cargas que soportan URI vincular tecnologías de referencia con otras cargas.

NDEF es simplemente un formato de mensaje, es decir que solo especifica la estructura del formato por lo que no se debe pensar que declara algún tipo de circuito o algún concepto de conexión o que pueda especificar el intercambio de información. El formato de datos de NDEF es el mismo tanto para un dispositivo NFC como para una etiqueta NFC, por lo que la información de NDEF es independiente del tipo de dispositivos que se estén comunicando.

Dentro del formato de un mensaje NDEF se puede enviar un variado tipo de información como:

- Puede encapsular documentos XML14, fragmentos XML, datos encriptados, e imágenes como archivos JPEG, GIF, etc.
- Encapsular cadenas de información.

- Agregar documentos múltiples y entidades que están asociados lógicamente de alguna manera. Por ejemplo, se puede encapsular un mensaje NFC- specific y un conjunto de archivos adjuntos de tipos estandarizados que tienen referencia desde ese mensaje NFC-specific.
- Encapsulado compacto de pequeños payloads.

Formato del Registro NDEF

Los registros NDEF son de longitud variable pero todos tienen un formato común que se representa a continuación con la siguiente figura:

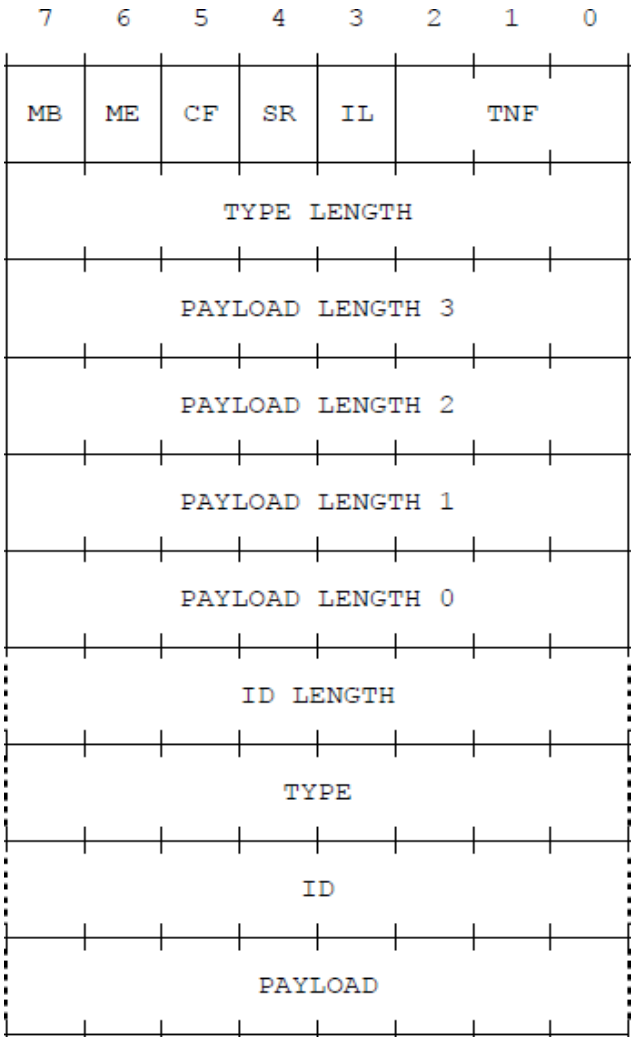


Figura. 1. 4. Formato de un Registro NDEF

Fuente: (internacional, Forum, 2017)

La información de los registros NDEF se presenta en nivel de octetos. El orden de transmisión es de izquierda a derecha y de arriba hacia abajo; de esta manera el bit más significativo del octeto es el bit del extremo izquierdo y para una cadena de octetos es igual, el bit más significativo es el de la extrema izquierda de todo el campo de octetos y es el que se transmite primero.

A continuación se detallan los campos que conforman el formato del registro NDEF:

- MB (Message Begin): Es una bandera de 1 bit que cuando se constituye indica el inicio de un mensaje NDEF.

- ME (Message End): Esta bandera es un campo de 1 bit que si se establece, ya que en el caso de una payload fragmentada esta bandera solo se establece en el segmento de terminación de esta payload fragmentada, indica el final de un mensaje NDEF.

- CF (Chunk Flag): Es una bandera de 1 bit que de establecerse indica que es el primer segmento de registro o que es un segmento de registro del medio de una payload fragmentada.

- SR (Short Record): Se conforma por 1 bit y al establecerse indica que el campo PAYLOAD_LENGTH es un solo octeto y no cuatro octetos como lo es para un registro NDEF normal.

  Este registro pequeño está destinado para una encapsulación compacta la cual permite que pequeños payloads sean parte de campos de payloads con un tamaño de entre 0 a 255 octetos. Un mismo mensaje NDEF podría tener tanto registros NDEF normales como registros cortos.

- IL (ID_LENGTH): La bandera IL es de 1 bit que si se establece indica que el campo ID_LENGTH está presente en la cabecera del registro como un octeto pero si el campo IL es cero entonces éste es omitido de la cabecera y el campo ID también es omitido del registro.

- TNF (TYPE NAME FORMAT): Es un campo de 3 bits que indica la estructura del valor del campo TYPE. Estos valores se detallan a continuación:

Tabla. 1. 1. Valores de TNF

Type Name Format	Valor
Vacío	0x00
Tipo NFC Forum (NFC RTD)	0x01
Tipo de Medios	0x02
URI Absoluto	0x03
Tipo NFC Forum externo	0x04
Tipo Desconocido	0x05
Sin Cambio (Unchanged)	0x06
Reservado	0x07

Fuente: (internacional, Forum, 2017)

EL valor 0x00 (vacío) significa que no hay ningún tipo o payload asociada al registro. De esta manera los campos TYPE_LENGTH, ID_LENGTH y PAYLOAD_LENGTH deben ser cero y por lo tanto los campos TYPE, ID y PAYLOAD respectivamente serían omitidos del registro.

El valor 0x05 (*Unknown*) debería ser usado para indicar que el campo de payload es desconocido. Este valor de TNF ocasiona que el campo TYPE sea omitido del registro NDEF ya que el valor del campo TYPE_LENGTH debe ser cero.

Se recomienda que cuando un analizador NDEF esté recibiendo un registro NDEF de este tipo, provea un mecanismo para guardar pero no para procesar la payload.

EL valor 0x06 (*Unchanged*) no debe usarse en otro registro que no sean los fragmentos de registro del medio y el fragmento del registro terminal que forman payloads segmentadas. Si se establece este valor TNF, el campo TYPE_LENGTH debe ser cero y el campo TYPE será omitido del registro NDEF.

El valor 0x07 (Reservado) es para usos futuros y no debe ser usado.

- TYPE_LENGTH: Este campo es un entero no asignado de 8 bits que representa la longitud en octetos del campo TYPE. Al referirse a un entero no asignado, quiere decir que no es una constante sino que su valor depende de la longitud del campo TYPE.

- ID_LENGTH: Este campo también es un entero no asignado de 8 bits que especifica la longitud del campo ID en octetos y está presente sólo si la bandera IL en la cabecera del registro se establece en 1.

- PAYLOAD_LENGTH: Es un entero no asignado que representa la logitud en octetos del campo PAYLOAD y a su vez el tamaño del campo PAYLOAD_LENGTH depende del valor de la bandera SR. Si la bandera SR está establecida, el campo PAYLOAD_LENGTH representa un solo octeto; pero si esta bandera está vacía, el campo PAYLOAD_LENGTH es de 4 octetos representando un entero no asignado de 32 bits.

- TYPE: Este campo es un identificador que especifica el tipo de payload de la información transmitida. El valor de este campo debe seguir la codificación, la estructura y el formato implícito por el valor del campo TNF. El tamaño máximo de este campo es 255 octetos.

- ID: El valor de este campo es un identificador que tiene la forma de una referencia URI (Identificador de Recursos Uniformes). Para NDEF, una URI es simplemente una cadena de texto que identifica un nombre, una localización o

alguna característica de un determinado recurso. La singularidad requerida del identificador del mensaje es garantizada por el generador. Los fragmentos finales y de la mitad de una payload segmentada no debe tener el campo ID ya que se trata del mismo campo de datos pero en diferentes fragmentos por lo que solamente basta con definir una vez la información completa acerca de todo el payload. Todos los demás tipos de registros podrían tener este campo ID. El tamaño máximo de este campo es 255 octetos.

- PAYLOAD: Dentro de este campo se lleva la carga o información útil para las aplicaciones del usuario y la estructura interna de los datos llevados en este campo es oculta para NDEF. El tamaño máximo del campo PAYLOAD es 2 (elevado a la potencia 32) -1 octetos para un diseño de registro NDEF normal y 255 octetos para un registro pequeño. Pero para tamaños de payload mayores a 2 (elevado a la potencia 32) -1 se segmenta dicha payload para poder ser transmitida en fragmentos (Payloads fragmentadas).

Mensaje NDEF

Un mensaje NDEF está compuesto por uno o varios Registros NDEF. El primer registro de un mensaje está marcado con la bandera MB (Message Begin) y el último registro lleva la bandera ME (Message End). Si un mensaje está compuesto por un solo registro, éste mismo lleva tanto la bandera MB como la bandera ME. El número de registros que un mensaje NDEF puede llevar es ilimitado.

Los mensajes NDEF no deben superponerse, es decir, que las banderas MB y ME no deben ser utilizadas para anidar mensajes NDEF. Los mensajes NDEF pueden ser anidados llevando un mensaje completo como una payload en un registro NDEF.

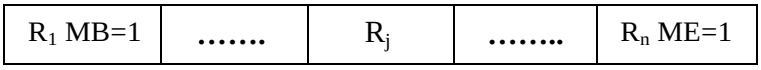


Figura. 1. 5. Mensaje NDEF con varios registros

Fuente: (internacional, Forum, 2017)

1.3.9. Interfaces y protocolos NFC

En la estandarización de la comunicación NFC esencialmente se han definido dos protocolos, NFCIP-1 (*Near Field Communication Interface and Protocol-1*) estandarizado en ISO/IEC 18092 / ECMA – 340 y NFCIP-2 (*Near Field Communication and Protocol-2*) estandarizado en ISO/IEC 21481 / ECMA – 352.

Dentro del protocolo NFCIP-1 se define el enlace de Radio Frecuencia con la que NFC trabaja que es de 13,56 MHz y los modos de operación activo y pasivo con sus rangos de velocidad desde 106 kbits/s hasta 424 kbits/s. También define las características que tienen estos modos de operación, por ejemplo la iniciación y selección del objetivo en el modo pasivo y el evitar colisiones de radio frecuencia en su modo activo.

A su vez, el protocolo NFCIP-2 especifica mecanismos de selección de los modos de comunicación para que no interfiera otras comunicaciones en curso en la frecuencia de 13,56 MHz. Los modos de comunicación que se especifican en este protocolo son:

Modo NFC (identificación de módulos específicos de nfc).

Modo PCD (Proximity Coupling Devices), especificado en la ISO/IEC 14443(identificación de tarjetas de proximidad electrónicas).

Modo VCD (Vicinity Coupling Devices), especificado en la ISO/IEC 15693(tarjetas que son leídas de una distancia más amplia).

La siguiente figura describe el proceso de selección de los diferentes modos de un dispositivo NFCIP-2:

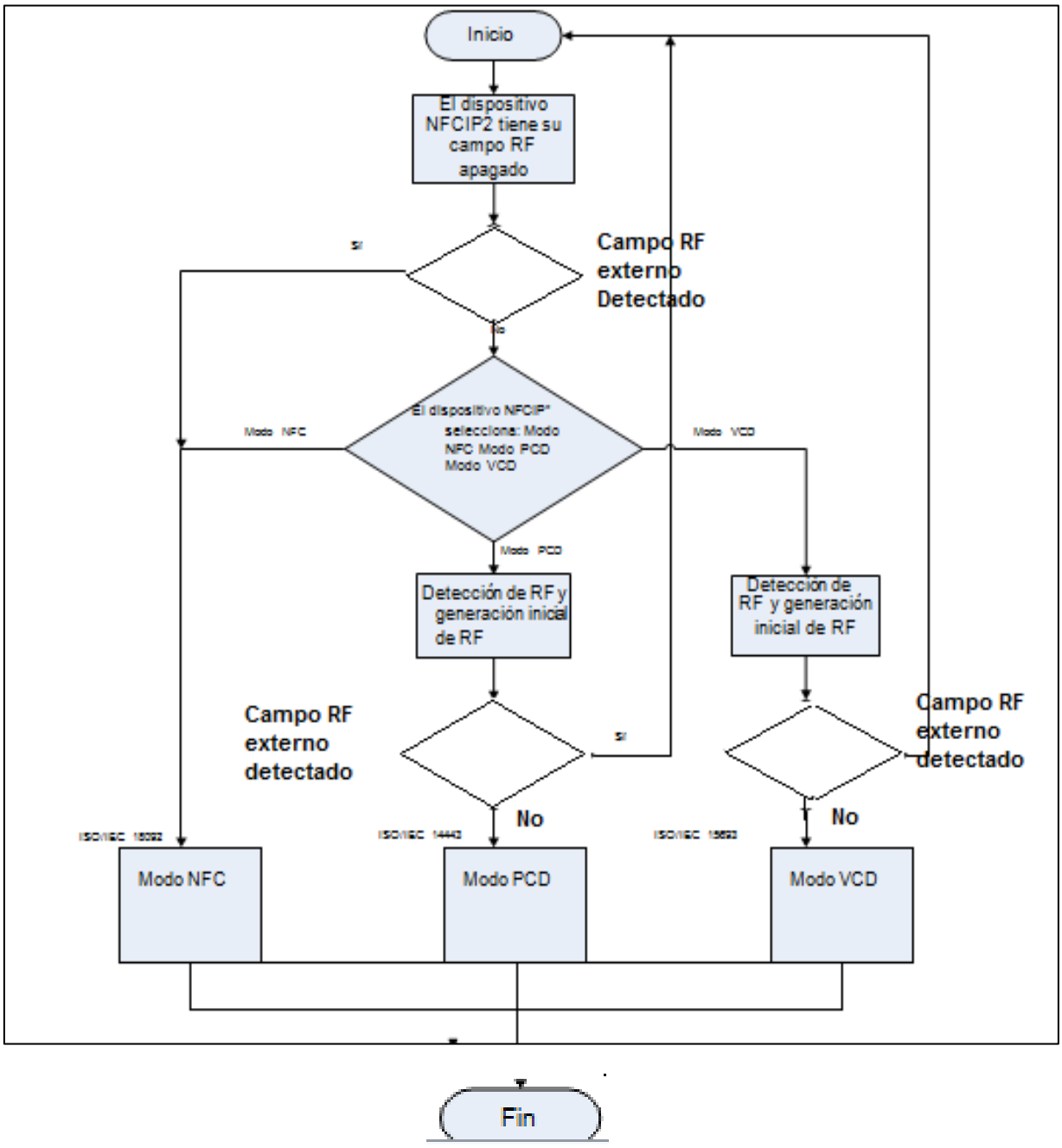


Figura. 1. 6. Detección de Radio Frecuencia y modo de selección de dispositivos NFCIO-2

Fuente: Elaborado por el autor

Cada uno de estos modos de selección trabaja en la banda de los 13,56 MHz y la diferencia de cada uno de estos es la distancia en la detección del campo de RF, los valores mínimos de sus campos para su detección y los procedimientos de inicialización que usan. Esto ayuda para prevenir posibles disturbios en las comunicaciones en curso como ya se lo mencionó.

1.3.10. Modos de Funcionamiento

Existen dos modos de funcionamiento:

- Modo de comunicación Pasiva
- Modo de comunicación Activa

Modo de comunicación pasiva: En este modo solo el dispositivo que inicia la conexión es el encargado de generar el campo electromagnético y el dispositivo de destino aprovecha de la modulación de la carga para poder transferir los datos. El dispositivo de destino podría dibujar su poder de operación desde el campo electromagnético que provee el dispositivo que inicia la comunicación, convirtiendo así al dispositivo de destino en un transponder.

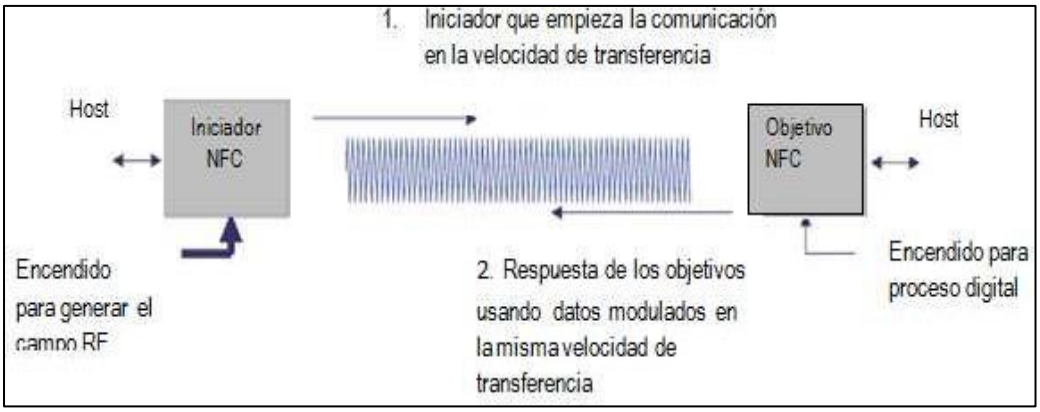


Figura. 1. 7. Modo de comunicación Pasiva

Fuente: (internacional, Forum, 2017)

Modo de comunicación activa: Tanto el dispositivo iniciador de la comunicación como el de destino, se comunican alternadamente generando sus propios campos, es decir, un dispositivo desactiva su campo de RF mientras está esperando por una respuesta. En este modo, ambos dispositivos necesitan tener una fuente de energía para su funcionamiento.

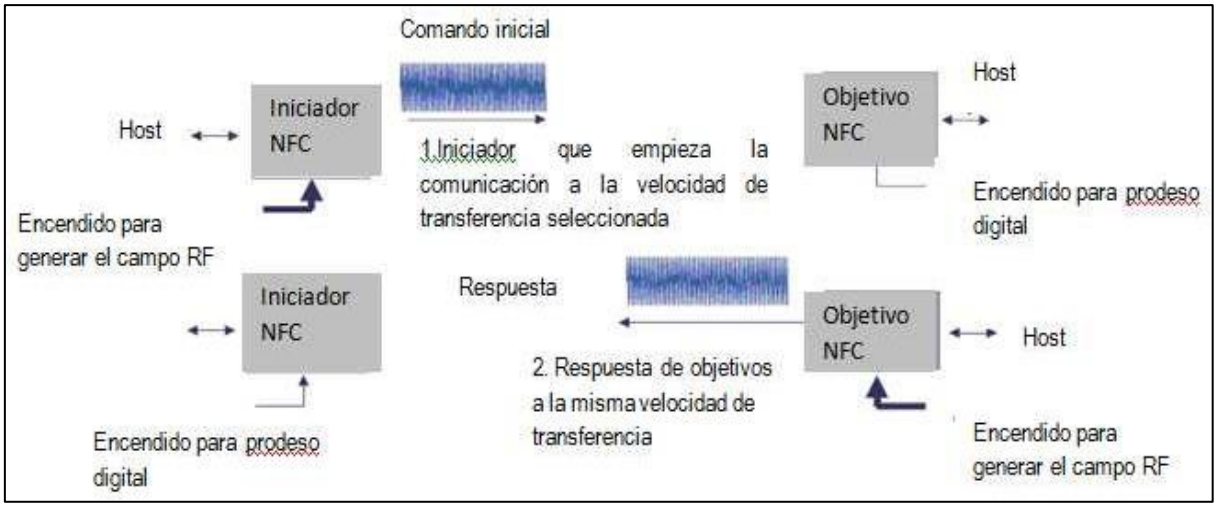


Figura. 1. 8. Modo de comunicación activa

Fuente: (internacional, Forum, 2017)

Tabla. 1. 2. Códigos de transferencia de NFC

Baudios	Dispositivo Activo	Dispositivo pasivo
424 kbaud	Manchester, 10% ASK	Manchester, 10% ASK
212 kbaud	Manchester, 10% ASK	Manchester, 10% ASK
106 kbaud	Modified Miller, 100% ASK	Manchester, 10% ASK

Fuente: Elaborado por el autor

NFC emplea dos diferentes códigos de transferencia de datos. Por ejemplo, si un dispositivo activo transfiere datos a 106 kbit/s, se usa la codificación Miller Modificado con el 100% de modulación. En tanto que para una velocidad de transmisión de 212 y 424 kbit/s se usa el código Manchester con un índice de modulación de 10%.

1.3.11. Establecimiento de la comunicación NFC

La comunicación NFC consta de cinco fases las cuales son importantes ya que tienen una función específica y siempre están presentes en el establecimiento de esta. Estas etapas son:

- Descubrimiento: En esta fase los dispositivos inician la etapa de rastrearse el uno al otro y posteriormente su reconocimiento.
- Autenticación: En esta parte los dispositivos verifican si el otro dispositivo está autorizado o si deben establecer algún tipo de cifrado para la comunicación.
- Negociación: En esta parte del establecimiento, los dispositivos definen parámetros como la velocidad de transmisión, la identificación del dispositivo, el tipo de aplicación, su tamaño, y si es el caso también definen la acción a ser solicitada.
- Transferencia: Una vez negociados los parámetros para la comunicación, se puede decir que ya está realizada exitosamente la comunicación y ya se puede realizar el intercambio de datos.
- Confirmación: El dispositivo receptor confirma el establecimiento de la comunicación y la transferencia de datos.

Cabe destacar que la tecnología NFC no está destinada para la transferencia masiva de datos, pero se puede utilizar para la configuración de otras tecnologías inalámbricas de mayor ancho de banda como Bluetooth o Wi-Fi con la ventaja de que si se utiliza NFC el tiempo de establecimiento de la comunicación es muy inferior que si se utilizaran estas otras tecnologías por sí solas para efectuar el enlace.

1.3.12. Módulo NFC PN532

El Módulo NFC (Near Field Communication) permite integrar transmisiones para comunicaciones inalámbricas, el cual posee un microcontrolador basado en el núcleo 80C52. Para los métodos de comunicación inalámbrica y protocolos, el módulo NFC utiliza el concepto de modulación y demodulación integrada en la banda 13.56 MHZ.

La ventaja principal del módulo NFC es su velocidad en la comunicación, además de no necesitar emparejamiento como sucede con bluetooth o wifi.

El módulo NFC soporta diferentes interfaces de comunicación como son: SPI, I2C y UART dependiendo de la especificación del fabricante los cuales pueden ser Adafruit, Elechouse entre otros. Existen librerías para Arduino, Raspberry y otras plataformas, en las cuales se puede fácilmente realizar la comunicación.

Figura. 1. 9. Módulo PN532

Fuente: Elaborado por el autor

A continuación, se puede observar los voltajes y corrientes de operación del módulo NFC PN532. Además de las condiciones necesarias que deben ser tomadas en cuenta para su uso.

Tabla. 1. 3. Datos de referencia

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{BAT}	Battery Supply Voltage		2.7		5.4	V
ICVDD	LDO output voltage	VSS = 0V VBAT > 3.3V	2.7	3.0	3.3	V
PVDD	Supply Voltage for host interface	VSS = 0V PVDD < VBAT	1.6		3.6	V
SVDD	Supply Voltage for SAM interface	VSS = 0V VBAT > 3.3V (SVDD Switch Enabled)	2.7	3.0	3.3	V
I_{HPD}	Hard Power Down Current	VBAT=5V, RF level detector off			2	⌠A
I_{SPD}	Soft Power down Current	VBAT=5V, RF level detector on			10	⌠A
I_{ICVDD}	Digital Supply Current	VBAT=5V, RF level detector on, SVDD switch off		25		mA

Fuente: (Arroyo Briones , Contreras Bernal , & Espíritu de la Paz, 2016)

Tabla. 1. 4. Descripción de los pines

Symbol	Pin	Type	Pad Ref Voltage	Description
DVSS	1	PWR		Digital Ground
LOADMOD	2	O	DVDD	Load Modulation output provides digital signal for FeliCa™ and MIFARE® card operating mode
TVSS1	3	PWR		Transmitter Ground: supplies the output stage of TX1 and TX2
TX1	4	O	TVDD	Transmitter 1: delivers the modulated 13.56 MHZ energy carrier
TVDD	5	PWR		Internal Transmitter power supply: supplies the output stage of TX1 and TX2
TX2	6	O	TVDD	Transmitter 2: delivers the modulated 13.56 MHZ energy carrier
TVSS2	7	PWR		Transmitter Ground: supplies the output stage of TX1 and TX2
AVDD	8	PWR		Internal Analog Power Supply
VMID	9	O	AVDD	Internal Reference Voltage: This pin delivers the internal reference voltage.
RX	10	I	AVDD	Receiver Input: Input pin for the reception signal, which is the load modulated 13.56 MHZ energy carrier from the antenna circuit.
AVSS	11	PWR		Analog Ground
AUX1	12	O	AVDD	Auxiliary Output: This pin delivers analog and digital test signals.
AUX2	13	O	AVDD	Auxiliary Output: This pin delivers analog and digital test signals.
OSCIN	14	I	AVDD	Crystal Oscillator Input: input to the inverting amplifier of the oscillator. This pin is also the input for an externally generated clock (fosc = 27.12 MHZ).
OSCOUT	15	O	AVDD	Crystal Oscillator output: Output of the inverting amplifier of the oscillator.
I0	16	I	DVDD	General purpose IO signal Can be used by the embedded firware to select the used host interface.
I1	17	I	DVDD	General purpose IO signal Can be used by the embedded firware to select the used host interface.
TESTEN	18	I	DVDD	Test enable pin: When set to 1 enable the test mode. When set to 0 reset the TCB and disable the access to the test mode.
P35	19	IO	DVDD	General purpose IO signal
NC	20			
NC	21			
NC	22			
PVDD	23	PWR		Pad power supply
P30	24	IO	PVDD	General purpose IO signal. Can be configured to act either as RX line of the second serial interface or general purpose IO. In test mode this signal is used as input and output test signal.
IRQ	25	O	PVDD	Interrupt request: Output to signal an interrupt event to the host (Port 7 bit 0)
RSTOUTN	26	IO	PVDD	Output reset signal. When Low it indicates that the circuit is in reset state.
NSS	27	IO	PVDD	Not Slave Select.
MOSI	28	IO	PVDD	Master Out Slave In.
MISO	29	IO	PVDD	Master In Slave Out.

Fuente: (Arroyo Briones , Contreras Bernal , & Espíritu de la Paz, 2016)

1.3.13. ASPECTOS DE SEGURIDAD NFC

En cuanto a los aspectos de seguridad, se puede decir que la tecnología NFC es inherentemente segura por la característica de su rango de alcance que es limitado a unos pocos centímetros, pero NFC por sí sola no asegura comunicaciones seguras.

NFC no ofrece protección contra los que se dedican a escuchar comunicaciones y es también vulnerable a modificación de datos. Las aplicaciones deben usar protocolos criptográficos de una capa superior para establecer un canal seguro.

Pero esto se contrarresta con la distancia de operación del NFC ya que al ser de tan sólo unos pocos centímetros, el espía debería estar dentro de ese rango y el usuario podría darse cuenta fácilmente.

Un dispositivo pasivo, que no genera su propio campo de radio frecuencia, es mucho más difícil intervenir que un dispositivo activo.

1.3.14. ARDUINO

Arduino permite libremente usar su plataforma, el cual puede ser usado y reformado dependiendo del requerimiento. Está conformado por un entorno de desarrollo y un microcontrolador, su objetivo principal es facilitar el uso de la programación en la electrónica para cualquier proyecto.

Tabla. 1. 5. Arduinos más usados

Característica de Arduino	UNO	Mega 2560	Leonardo	DUE
Tipo de microcontrolador	Atmega 328	Atmega 2560	Atmega 32U4	AT91SAM3X8E
Pines digitales de E/S	14	54	20	54
Entradas analógicas	6	16	12	12
Velocidad de reloj	16MHZ	16MHZ	16MHZ	64MHZ
Memoria de datos (SRAM)	2 Kb	8 Kb	2,5 Kb	96 Kb
Memoria auxiliar (EEPROM)	1 Kb	4 Kb	1 Kb	0 Kb

Fuente: Elaborado por el autor

Mediante Arduino se puede realizar diferentes interacciones controlando diferentes elementos siendo tanto hardware como software.

Por ejemplo:

Un motor se puede mover de izquierda a derecha controlando tiempos y los grados de giro.

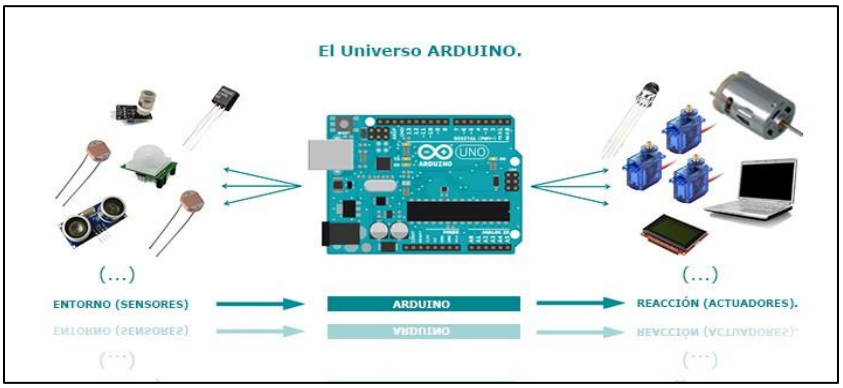


Figura. 1. 10. Arduino y sus usos

Fuente: (Crespo, 2014)

Toda la comunidad apoya al desarrollo de Arduino para que día a día sea replicado y mejorado para nuevas actividades tecnológicas.

Arduino es específicamente una placa y un microcontrolador permitiendo así programar y ejecutar órdenes que antes fueron grabadas en su memoria.

1.3.15. Arduino MEGA

El Arduino MEGA es uno de los microcontroladores más potentes y completos dentro de la familia Arduino, diseñado para resolver cualquier tipo de inconveniente. Posee 70 pines, los cuales 54 son pines digitales de entrada y salida, además de 16 entradas analógicas de entrada y salida. Tiene una conexión USB, una entrada para la alimentación de la placa y un botón de reset.

Arduino se comunica fácilmente a cualquier computadora gracias a su convertidor usb-serie con el cual basta con conectar el dispositivo mediante el cable USB para que se establezca la comunicación.

El Arduino Mega posee requiere de 5v para su funcionamiento el cual es provisto por la computadora, posee 54 pines digitales de entrada y salida de los cuales 15 tienen salida PWM, la corriente entregada por cada pin es de 40mA, posee una memoria flash de 256kb.

Arduino Mega puede ser alimentado con fuentes externas sin que esto afecte el funcionamiento del mismo

Es necesario de un convertidor AC/DC regulado en el rango operativo de la placa, cuando se trabaja con una fuente externa de poder. Su rango aproximado es de 7v a 12v.

El método de programación del Arduino es muy fácil ya que contiene su propio lenguaje y muchos tutoriales de uso.

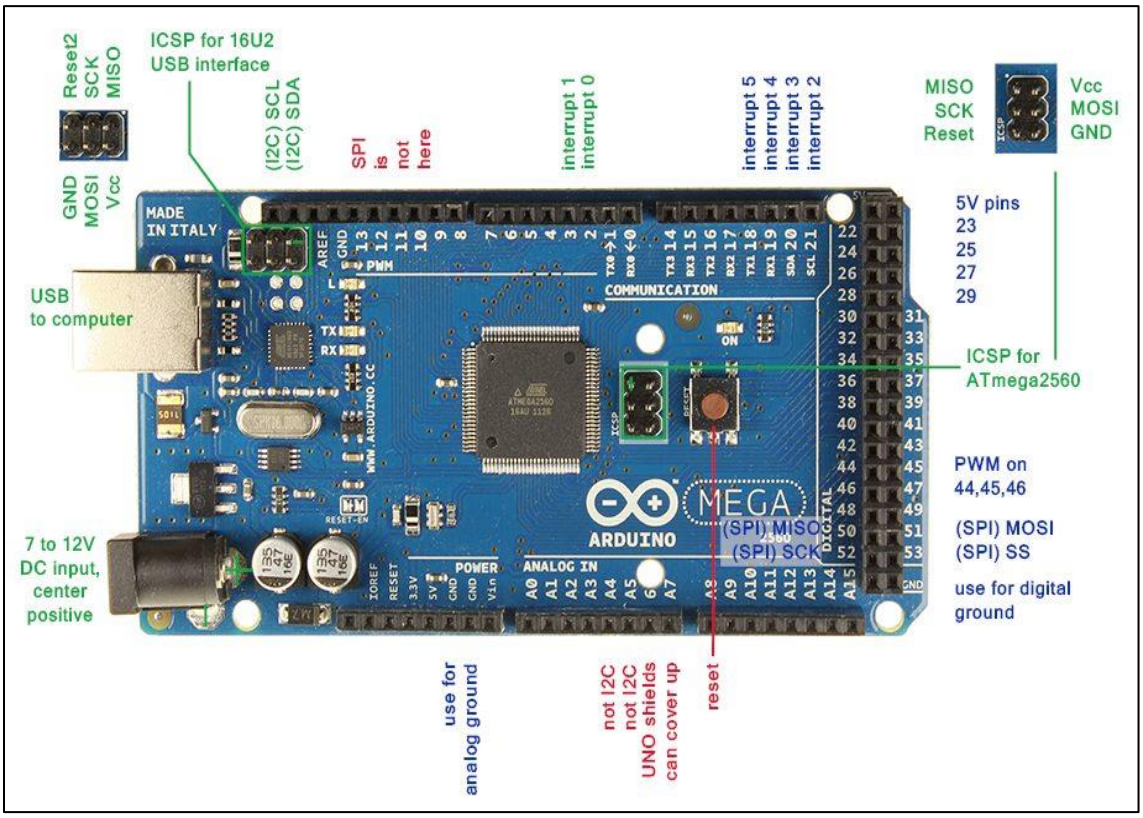


Figura. 1. 11. Arduino MEGA

Fuente: (García Gonzáles, 2013)

1.3.16. Arduino NANO

Figura. 1. 12. Arduino NANO

Fuente. (Patagoniatec, 2013)

Tabla. 1. 6. Arduinos más pequeños

Arduino Mini	Arduino Micro	Arduino Nano
Esta tarjeta está diseñada para ser montada y usada en una protoboard debido a su tamaño compacto y al arreglo de headers macho. Adecuada para prototipos pequeños y de volúmenes pequeños. Puede considerarse que es una versión miniatura de la tarjeta Arduino UNO. Es la tarjeta más barata dentro de este grupo, aunque se debe erogar más por la adquisición de la tarjeta o el cable FTDI para programarla.	Diseñada para ser usada en tabletas protoboard. Es considerada como la versión pequeña de la tarjeta Arduino Leonardo ya que tiene incorporada la forma de comunicación USB permitiéndole prescindir de procesadores secundarios Por los atributos anteriores –además de tener más pines digitales, analógicos y de PWM-, es la tarjeta más cara de las tres.	También diseñada para usarse en espacios pequeños. Es considerada como otra versión miniatura del Arduino UNO, sólo que se diferencia del Mini por poseer un puerto mini USB para programarla y energizarla. Su precio es intermedio al ser más cara que el Mini pero más barata que la Micro.

Fuente: Elaborado por el autor

El Arduino NANO es la versión más pequeña del Arduino Uno, sus pines facilitan las conexiones de componentes por lo tanto no es necesario utilizar muchos cables. Su tamaño es de gran ventaja con respecto a los Arduinos más utilizados como el UNO, MEGA o Leonardo, que después fue superada por el Arduino Micro y el Arduino Mini. Siendo diseñados exclusivamente para protoboards.

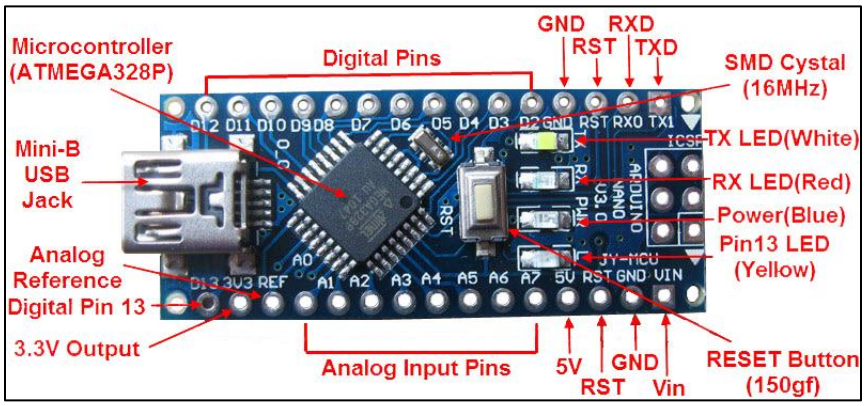


Figura. 1. 13. Descripción de pines

Fuente: (Patagoniatec, 2013)

El Arduino además de ser alimentado por el cable USB puede ser alimentado por un cable USB mini B y una fuente externa de 6 a 20v en el pin30 o una fuente externa de 5v en el pin27.

El Arduino NANO AT 328 posee 32KB, posee además 2 KB de SRAM y 1KB de EEPROM.

Posee 8 entradas analógicas y cada una provee 10 bits de resolución. Por defecto miden entre 5 voltios, pero siendo posible cambiar el rango.

1.3.17. Servomotor SG90

La función de los servomotores es moverse a un ángulo fijo en respuesta a una señal de control, y mantenerse fijos en dicha posición. Los servos son motores de corriente continua (CC), su uso frecuente es en Aero modelismo y en robótica ya que su funcionamiento es específico.

Un servomotor está conformado por un motor de CC, un circuito de control y un conjunto de engranajes (para mayor control).

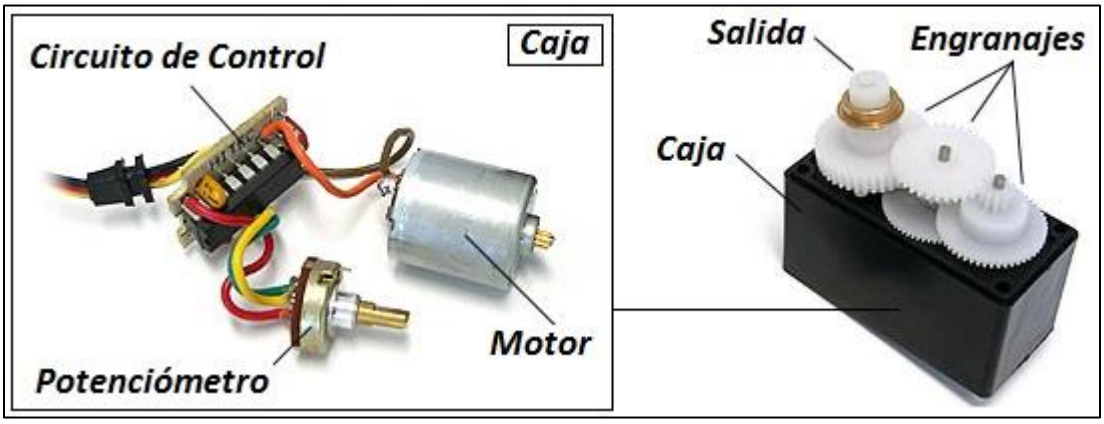


Figura. 1. 14. Composición de un servomotor

Fuente: (Del Campo García, 2016)

El movimiento de los servomotores por lo general puede ser entre 0 º y 180 º, funcionan con un voltaje de 5V y su control se lo realiza mediante PWM (modulación por ancho de pulsos) para transmitir la información a través del canal y así elegir el ángulo deseado.

Figura. 1. 15. Servomotor SG90

Fuente: (Del Campo García, 2016)

1.4. Marco Conceptual

Códigos. – Término utilizado para referirse a una combinación de símbolos, utilizado para protección y seguridad.

Interfaz Gráfica. – Consiste en mostrar un entorno visual mediante un dispositivo o equipo electrónico, de una manera simple para el entendimiento del usuario.

RFID. – Es un sistema que almacena datos mediante la comunicación o identificación por radiofrecuencia así poder recuperar datos que por motivos externos se perdieron o no pudieron ser visualizados.

NFC Forum. – Es la compañía creadora de la tecnología NFC la cual se caracteriza en la comunicación de corto alcance y permite que cualquier persona investigue y aporte a nuevas ideas para su mejora.

Memoria EEPROM. – Es una memoria no volátil en la que se almacena datos sin riesgo de perderlos al retirar la alimentación del Arduino

Memoria SRAM. - Es la zona de memoria donde el sketch crea y manipula las variables cuando se ejecuta. Es un recurso limitado y se debe supervisar su uso para evitar agotarlo.

2. CAPÍTULO 2 PROPUESTA

Los sistemas de control de acceso son la tecnología con más demanda en el mercado actual, se ha migrado de sistemas mecánicos y con personal especializado, a tener procesos de control de entrada y salida completamente automatizados con diferentes tipos de tecnologías y dispositivos. Es importante realizar un estudio adecuado, segmentando las zonas, los grupos de acceso, el nivel de acceso de cada usuario, medir la cantidad de personas que transitan por cada zona y establecer claramente los objetivos de cada control de acceso.

Es importante el estudio y diseño previo a cualquier instalación y puesta en marcha de un proyecto de seguridad y control de acceso. Una adecuada integración de los dispositivos electrónicos con los dispositivos electromecánicos permitirá incluso reducir drásticamente los costos de personal y totales del proyecto, haciendo incluso que un sistema de control de accesos se pueda pagar literalmente solo en un tiempo muy corto.

A continuación una breve explicación del funcionamiento del sistema como opción para el control de accesos:

La implementación del sistema de control de registros de acceso mediante tecnología NFC propuesto, consiste en registrar y controlar los ingresos y salidas de los trabajadores. Para ello, es necesario de un identificador (aplicación móvil), para así poder ingresar a la empresa genérica.

El sistema está diseñado para reconocer, mediante la aplicación móvil, que tipo de usuario desea ingresar (Administrador, Gerente, Diseñador, Implementador). Mediante esta función cada usuario tiene diferentes permisos de acceso.

Como lo son: Administrador: acceso a todas las oficinas, Gerencia: acceso a la puerta principal de entrada y a la oficina de gerencia, Diseñador: acceso a la puerta principal y a la oficina de diseño, Implementador: acceso a la puerta principal y a la oficina de implementación.

Finalmente, cada acceso es leído mediante una aplicación en Visual Basic, la cual con la ayuda de Microsoft Excel registra la hora, fecha y el número de puerta que fue abierto.

En caso de problemas con los trabajadores, el sistema permite bloquear o desbloquear usuarios para mayor seguridad de la empresa genérica.

2.1. Descripción de la comunicación (usuario: Administrador) para el sistema de control de accesos

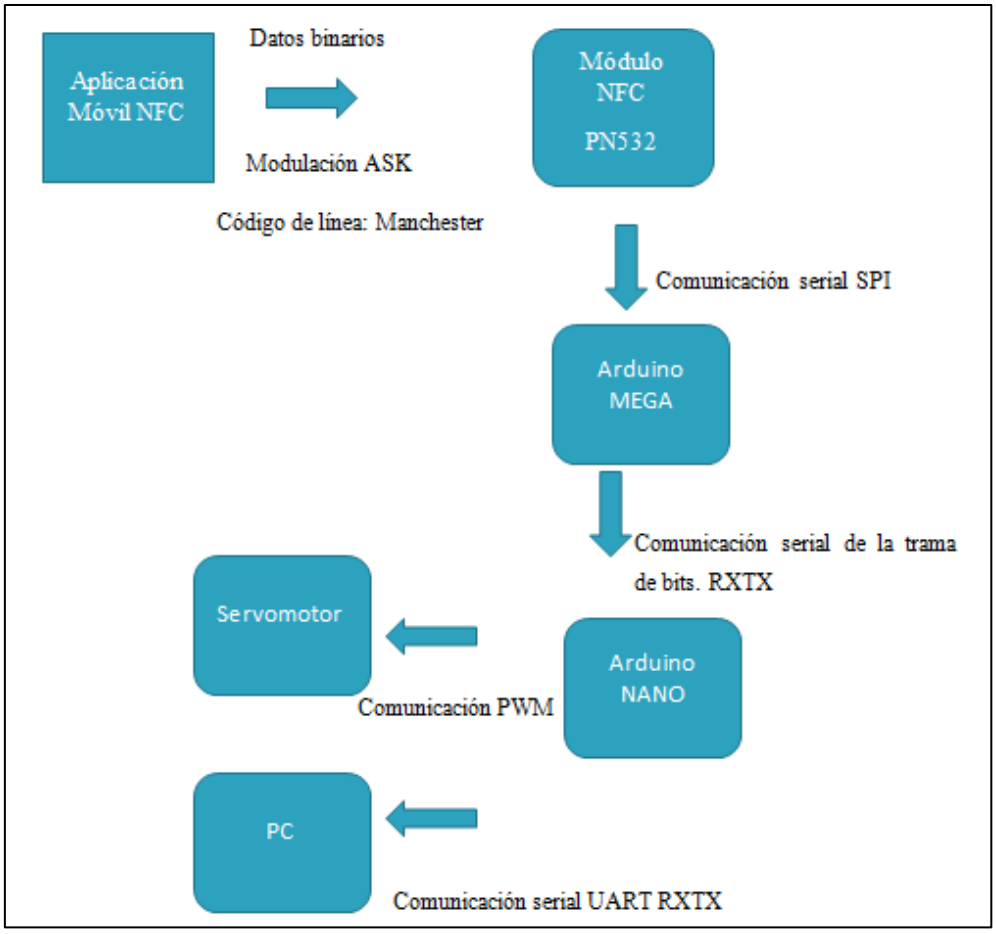


Figura. 1. 16. Comunicación entre dispositivos

Fuente: Elaborado por el autor

El formato para el intercambio de datos entre la aplicación móvil y el módulo NFC es el NDEF (Data Exchange Format) mediante el cual se guardan y transmiten elementos.

NDEF está conformado por tres procesos para su transmisión: Tipo, Longitud, Identificador.

El proceso “Tipo” identifica la clase de datos, los cuales pueden ser XML (*Stándart Generalised Mark-up Language*), Encriptados (transformación de los datos electrónicos en otra forma para que no puedan ser entendidos) o imágenes jpeg, gif, bpm.

Cuando se envía mediante una tarjeta, tag o aplicación, un conjunto de números ASCII hacia un receptor NFC, estos son convertidos a números binarios (uso de símbolos 1 y 0) como se observa en la tabla 7.

Tabla. 1. 7. Datos decimales convertidos a binarios

Números ASCII	Datos Binarios
331	101001011

Fuente: Elaborado por el autor

El proceso “Longitud” permite el registro de 8 octetos siendo transmitidos de izquierda a derecha y de arriba hacia abajo como se observa en la tabla 8. En la cual están ingresados los datos binarios que se están transmitiendo desde el más significativo al menos significativo. Los fragmentos sobrantes deben ser cero.

Tabla. 1. 8, Tabla de datos binarios transmitidos

MB	ME	CF	SR	IL	TNF	TNF	TNF
1	0	1	0	0	1	0	1
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Fuente: Elaborado por el autor

Las casillas MB, ME, CF, SR, IL, TNF, son banderas (indicadores) que cuando muestran 1 bit indican el inicio del mensaje NDEF, el final del mensaje NDEF, la mitad de una payload fragmentada (información que se transmite), los registros NDEF normales o cortos, si está presente en la cabecera o no, la estructura del valor de campo respectivamente.

El proceso “Identificador” se encarga de verificar si los payloads a transmitirse son URI (para vincular otras tecnologías), MIME (extensiones de correo de internet), NFC (dispositivos específicos de NFC).

Al finalizar este proceso de encapsulación de mensajes NDEF para el intercambio de datos entre dispositivos NFC y tener la señal en banda base binaria, se obtiene la modulación ASK (Modulador digital por desplazamiento de amplitud) de múltiples estados M-ASK (con Me>>2), en cuyo caso la amplitud de la portadora modulada presentará M valores diferentes y, cada uno de ellos constituirá un símbolo o estado de la señal modulada. En la figura

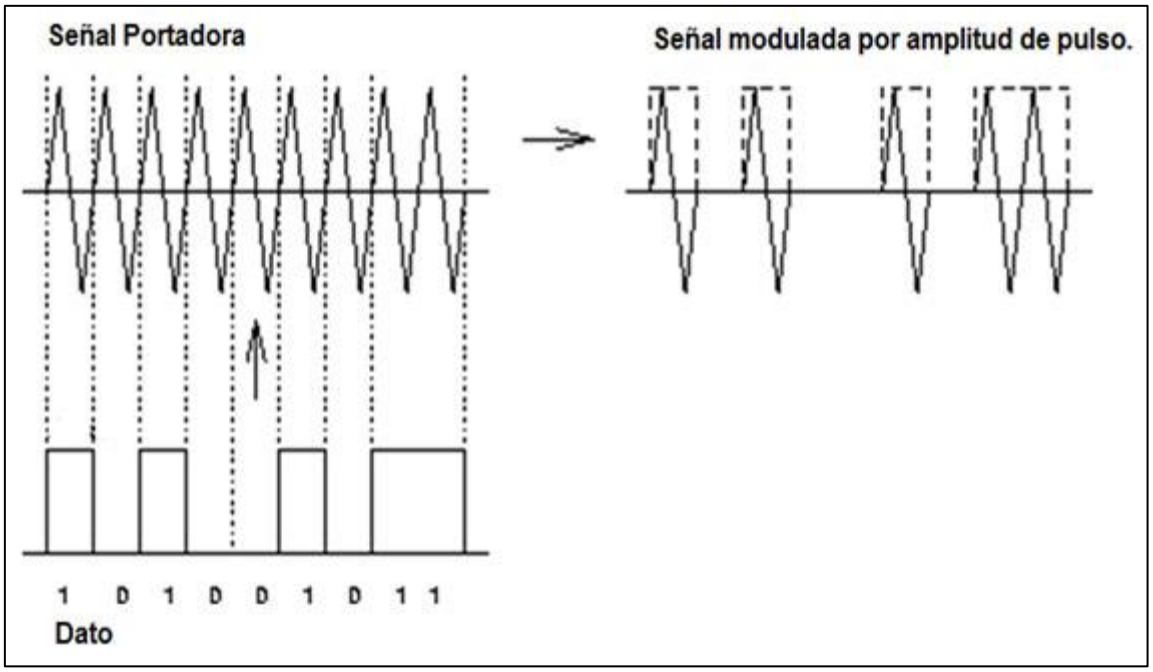


Figura. 1. 17. Señal modulada en amplitud, M=2

Fuente: (Tarifa Amaya & Del Risco Sánchez, 2012)

Para extraer los datos de la señal modulada ASK esta se muestrea, detectando cambios en la amplitud, donde la oscilación se interpreta como un uno lógico y la ausencia de señal se asocia con un cero lógico.

Para la codificación se emplea el tipo Manchester «bifase» L. En esta codificación el uno lógico equivale a un flanco de bajada en la mitad del período del bit y un cero lógico se asocia a un flanco de subida en el mismo instante.

Este código de línea está directamente asociado con la modulación ASK, de tal manera que para representar un uno lógico durante la primera mitad del tiempo de bit se transmite una señal de frecuencia 13,56 MHz y en la segunda mitad no se emite señal alguna. En el caso de un cero lógico en la primera mitad de bit no se emite señal y en la segunda mitad se transmite la señal de 13,56 MHz. En la f se muestra lo antes explicado.

Estas transiciones se generan mediante interrupciones de un temporizador en la programación del Arduino y por lo tanto cualquier secuencia de instrucciones que se ejecuten no afecta la precisión con que se genera la salida de información.

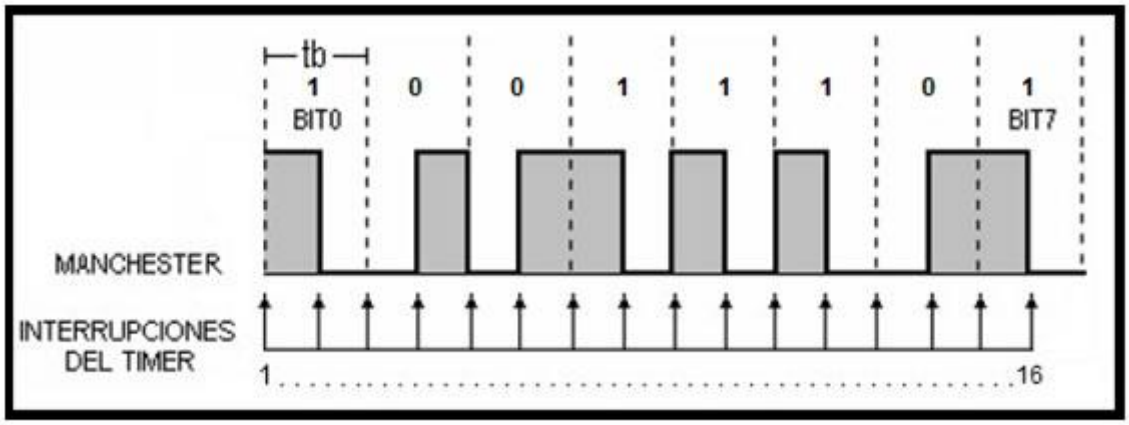


Figura. 1. 18. Codificación Manchester

Fuente: (Tarifa Amaya & Del Risco Sánchez, 2012)

El dato codificado llega al módulo NFC, el cual es trasmitido hacia el Arduino MEGA mediante comunicación serial SPI (transferencia de información entre circuitos integrados en equipos electrónicos.) para ser decodificado y cumplir con la programación del Arduino.

Después mediante una trama de bits viaja hacia el Arduino NANO por la comunicación serial RxTX para finalmente, trasladarse al servomotor y al computador.

La comunicación necesaria del servomotor es la PWM unidireccional (Modulación por ancho de pulsos), la cual mediante un pulso de 5v expresa un uno lógico y 0v para un cero lógico. El uno lógico realizará el movimiento necesario para activar el servomotor y el cero lógico no realizará ninguna acción.

La comunicación serial hacia la computadora es la UART (Transmisor-Receptor Asíncrono Universal, es el dispositivo que controla los puertos y dispositivos serie), la cual entrega la información ya decodificada para ser descrita en la misma.

2.2. Diseño de placa de control de los módulos NFC y servomotores

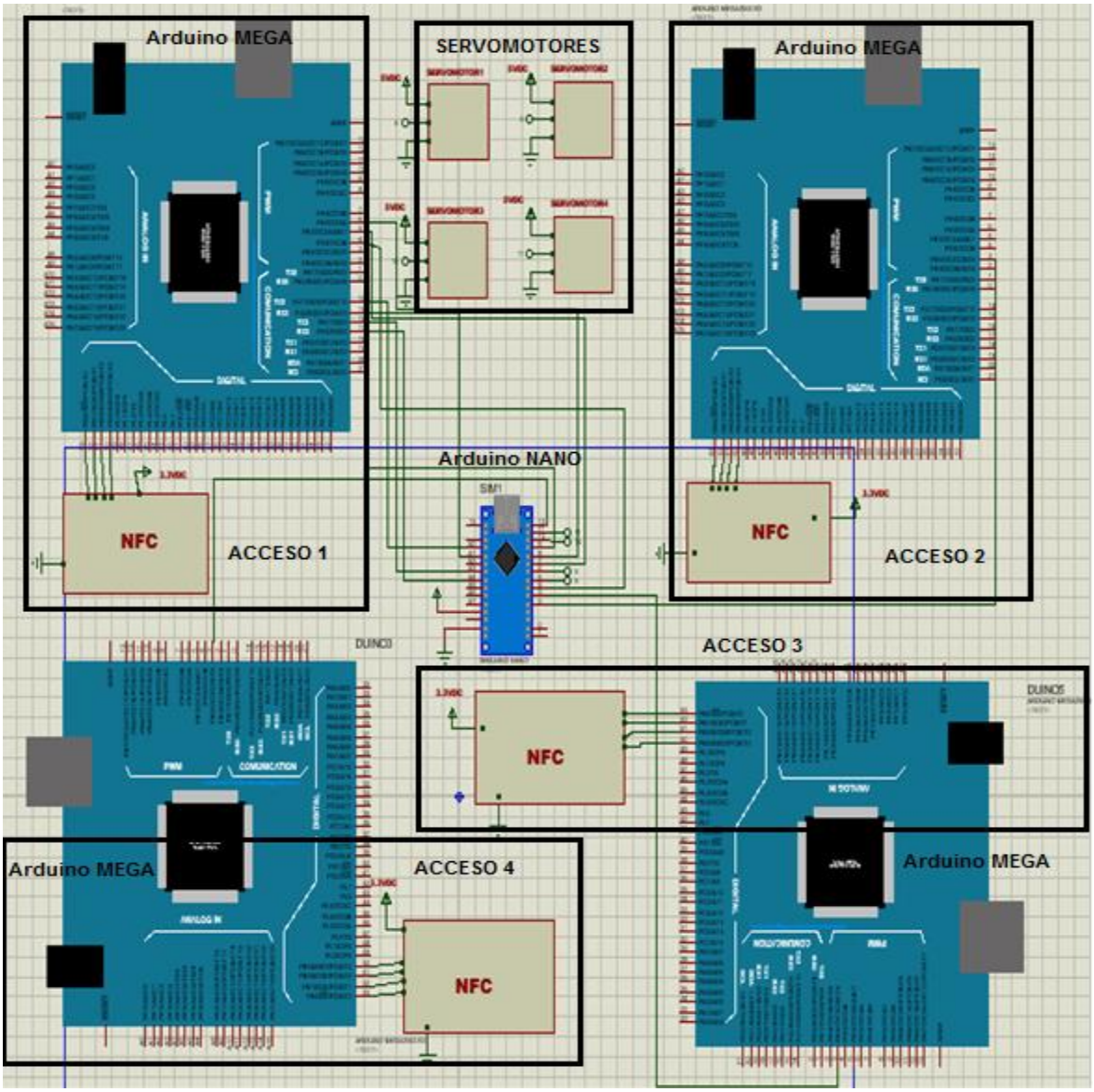


Figura. 2. 19. Diagrama esquemático del Sistema

Fuente: Elaborado por el autor

La placa diseñada para este proyecto consta de 4 accesos. Cada acceso está conformado por un Arduino MEGA y un lector o módulo NFC los cuales se encargan de recoger y transmitir la información adquirida hacia el Arduino NANO.

El Arduino NANO es el encargado de verificar si la información es aceptada o negada, para así activar los servomotores los cuales simulan la apertura de puertas.

Mediante la aplicación en la PC se puede observar la hora, fecha actual y el registro del usuario, para evitar retrasos o falsos controles de acceso.

2.2.1. Placa Arduino MEGA 2560

La siguiente placa gracias a sus características únicas, permitirá la lectura de los códigos transmitidos por el módulo NFC Pn532, la placa funciona con un voltaje de 5v para cada una. En la figura se observa la placa Arduino MEGA 2560 en su parte superior.

Figura. 2. 20. Placa Arduino MEGA 2560

Fuente: Elaborado por el autor

La gran capacidad y protección ante imprevistos es una característica adecuada para el uso de esta placa en el sistema e ideal para los módulos NFC y servomotores entre otros, además de ser una de las únicas placas las cuales tienen las librerías para la comunicación con tecnología NFC.

2.2.2. Módulo NFC Pn532

Mediante las características anteriormente mencionadas los módulos NFC tienen muchos usos en los cuales reciben la señal de un código mediante tarjetas NFC o en este caso la aplicación móvil. Adicionalmente estos módulos trabajan a un voltaje de 3.3v adaptándose a la alimentación del Arduino mencionado.

Se escoge estos módulos por su tamaño y recepción para tener un óptimo funcionamiento ya que al ser un sistema de control de accesos no debe tener errores y debe estar siempre trabajando.

Figura. 2. 21. Módulo PN532

Fuente: Elaborado por el autor

En la figura 21 se muestra el módulo NFC que será utilizado para el reconocimiento, los cuales son colocados cerca de las puertas para tener un acceso fácil y cómodo.

2.2.3. Servomotores

El control del servomotor se basa en la placa Arduino NANO. La comunicación Arduino – Servomotor se realiza mediante comunicación serial PWM, para la transmisión de instrucciones de 1 lógicos (5 voltios) o 0 lógicos (o voltios) hacia los servomotores. Los cuales activarán o desactivarán el movimiento de los mismos. La conexión de los servos hacia el Arduino contiene una trama de terminales de 3 pines (Señal (instrucciones de 1 y 0), Voltaje, Tierra)

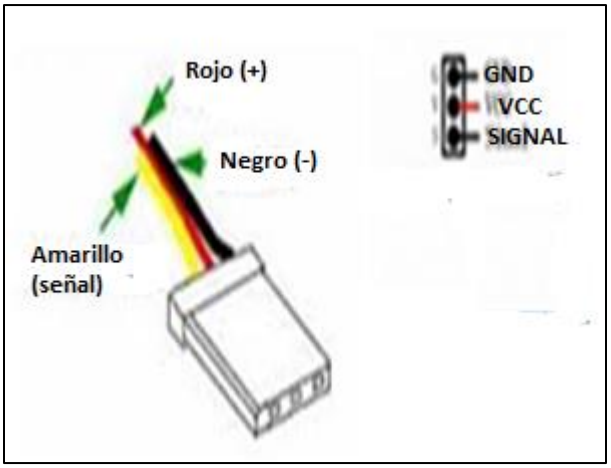


Figura. 2. 22. Conexiones Servomotor

Fuente: Elaborado por el autor

Son 4 servomotores ubicados en las diferentes puertas del prototipo por lo que se asigna un número de referencia a cada uno de ellos para la conexión a las terminales del Arduino NANO para su respectivo control.

Estos servomotores servirán para simular la apertura de la puerta en caso de que el sistema haya aceptado todas las condiciones establecidas.

2.2.4. Diseño Esquemático

El diseño esquemático permite conocer de una manera más ordenada y detallada los componentes electrónicos y sus conexiones antes de realizar una placa para así tener una mejor comprensión del sistema.

El diseño contiene 4 módulos NFC, 4 Arduinos MEGA, 4 servomotores, un Arduino NANO, los cuales son los elementos principales la estructura electrónica del prototipo como se lo puede observar en la figura 23.

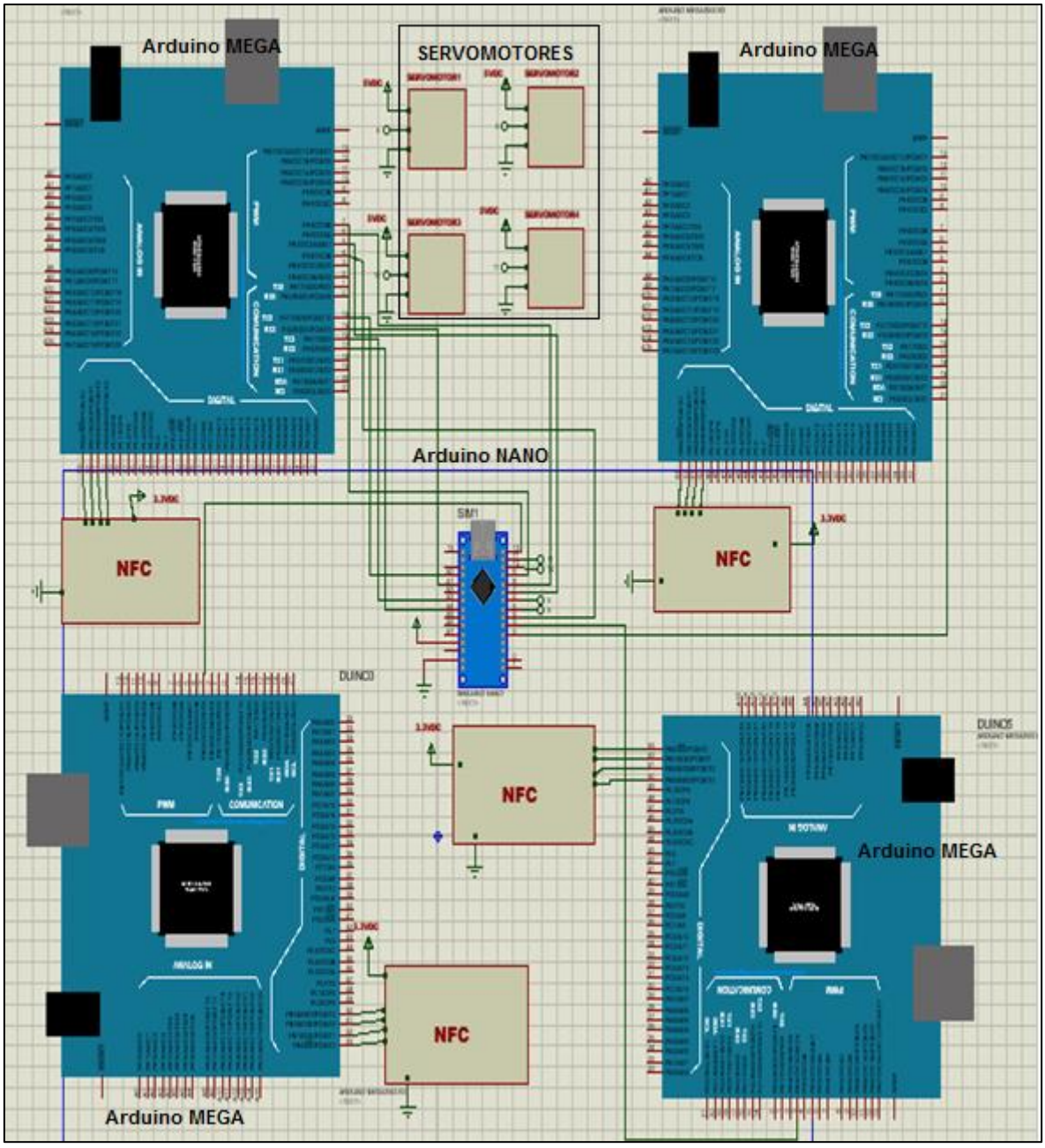


Figura. 2. 23. Diagrama de Conexiones Arduino Mega- NANO- Servomotor- Módulo NFC

Fuente: Elaborado por el autor

Una vez realizado el diseño esquemático es necesario dirigirse a realizar la placa PCB que es la que se encargara de contener los elementos y las conexiones. La placa PCB se puede realizar en el programa Proteus Profesional 8.6 SP2, el cual nos permite realizar las pistas de las conexiones del sistema de una forma más detallada como se observa en la figura 24.

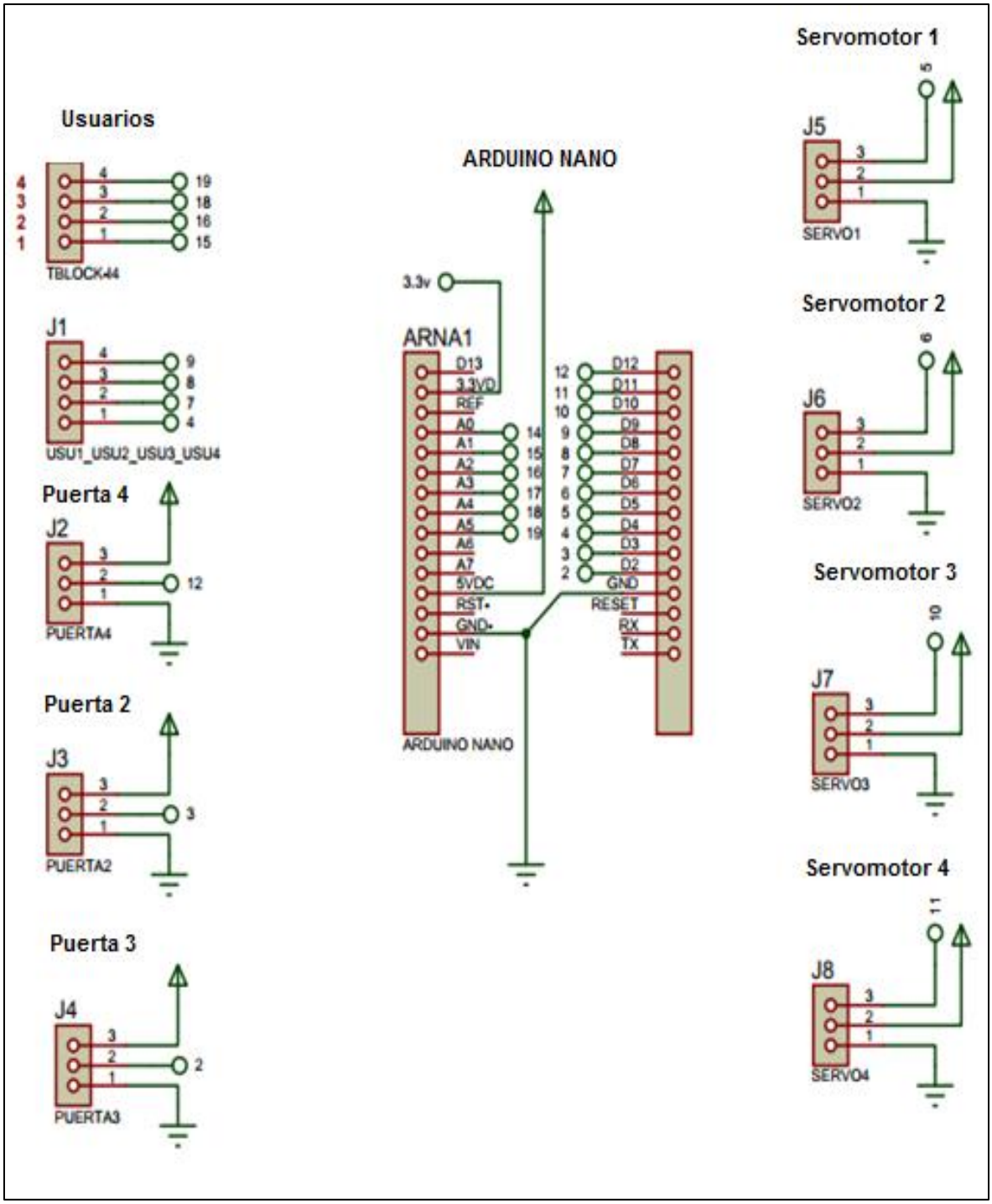


Figura. 2. 24. Diagrama de conexiones simplificado

Fuente: Elaborado por el autor

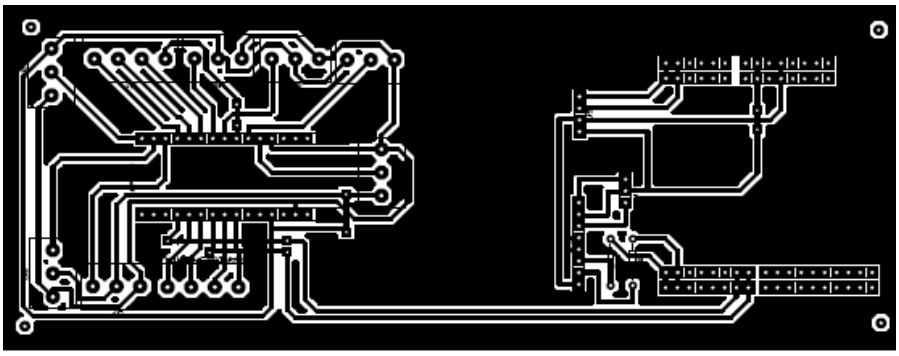


Figura. 2. 25. Diagrama Diseño de placa PCB

Fuente: Elaborado por el autor

2.2.5. Diagrama de flujo del sistema de acceso

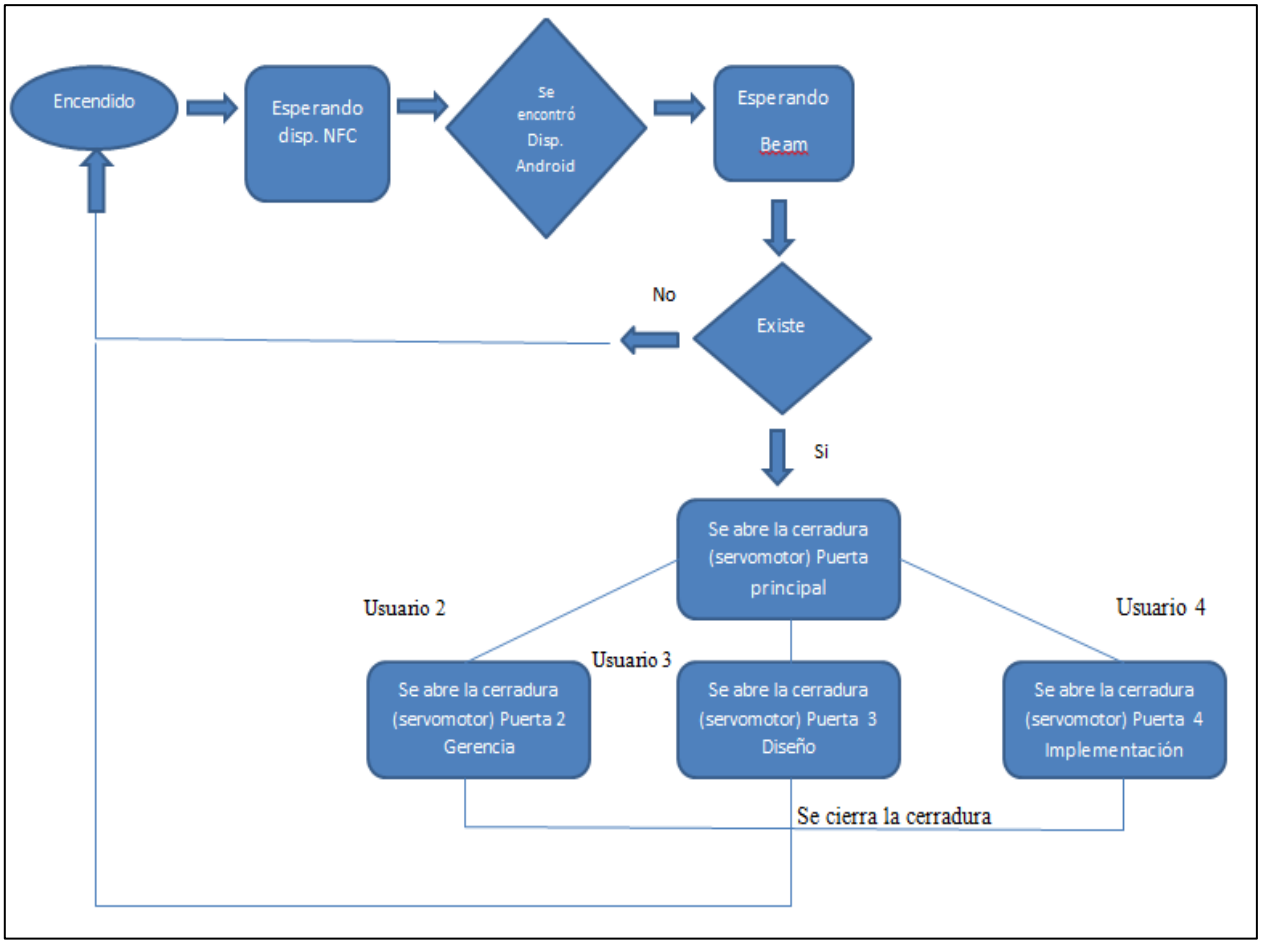


Figura. 2. 26. Diagrama de flujo del sistema

Fuente: Elaborado por el autor

En la figura 26 se observa el diagrama de flujo de la lógica para la programación del presente proyecto.

2.2.6. Diseño del Sistema

El sistema diseñado para este proyecto se representa simplificado en la figura. 27.

El sistema consiste en:

a. Abrir la aplicación en el celular
b. Desplegar el móvil por el módulo NFC colocado alado de cada puerta.
c. Esperar mientras se procesa el código según el tipo de usuario que sea. Observar en la tabla 9 los diferentes usuarios y sus códigos.
d. El Arduino NANO entrega el registro a la PC y se activa el servomotor, para abrir la puerta deseada.
e. Finalmente, se observa en la PC: el usuario, la hora y fecha de entrada y salida.

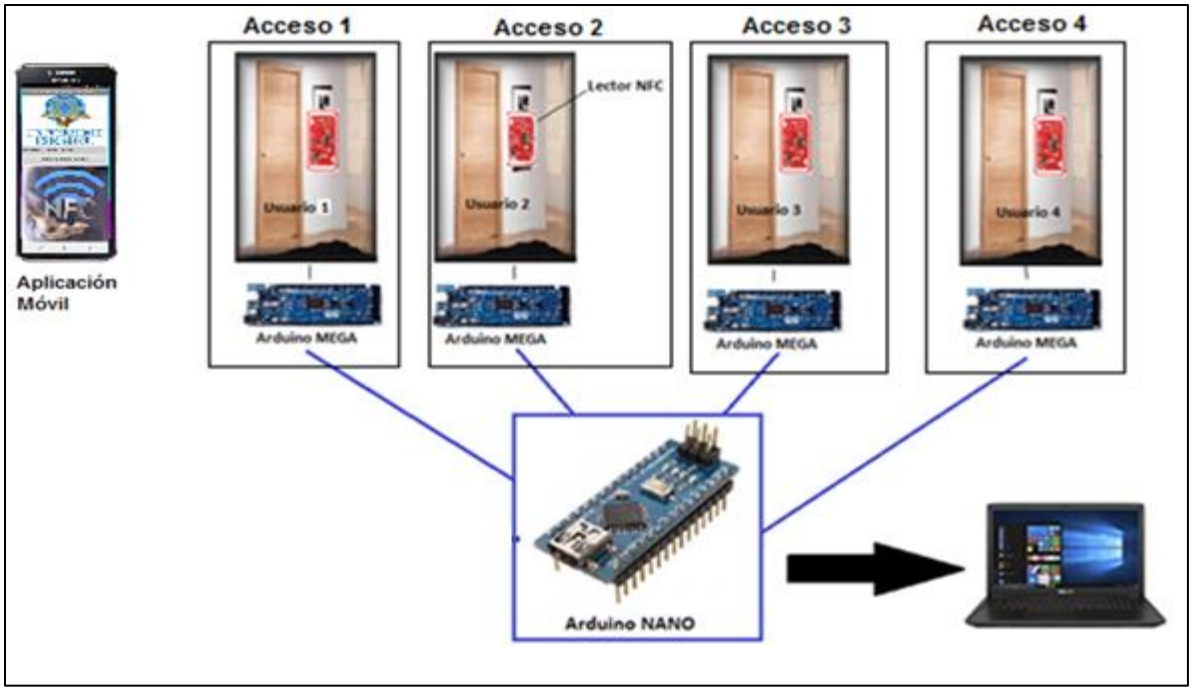


Figura. 2. 27. Diagrama General del Prototipo

Fuente: Elaborado por el autor

Tabla. 2. 9. Códigos ocultos de acceso

Códigos Secretos en Aplicación Móvil de Acceso	
1234	Usuario 1 (Administrador)
abcd	Usuario 2 (Diseñador)
efgh	Usuario 3 (Gerente)
ijkl	Usuario 4 (Implementador)

Fuente: Elaborado por el autor

El lector NFC es el principal medio encargado de receptar los datos de la comunicación los cuales procesan toda esta información para posteriormente ser enviada y manipulada según los requerimientos del proyecto, para la comunicación entre el lector y el Arduino 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 Arduino poseen la información de la aplicación NFC de un móvil, se envía dicha información a un concentrador (Arduino NANO) donde se conectan todos los accesos del prototipo y este pueda re direccionarla a la aplicación de Visual Basic en donde administra esta información logrando visualizar y controlar los registros de acceso de las puertas mediante Microsoft Excel.

2.2.7. Descripción de las etapas de funcionamiento

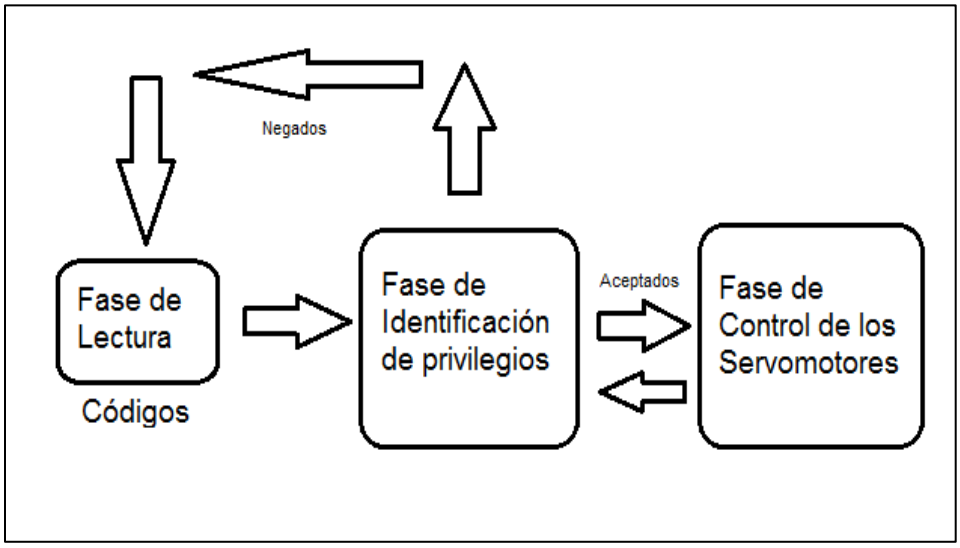


Figura. 2. 28. Diagrama en bloques de las Etapas de funcionamiento

Fuente: Elaborado por el autor

El sistema está compuesto por 3 fases principales:

Fase de Lectura: 1 módulo NFC colocado en la puerta principal y 3 en las puertas secundarias de las oficinas principales que permiten la lectura del código entregado por la aplicación de Android

Fase de Identificación de Privilegios: Cuatro Arduinos Mega los cuales contendrán la programación que identifica el código entregado por el módulo NFC.

Fase de Control de Servomotores: El Arduino NANO permitirá la unión de los cuatro programas de los Arduinos Mega otorgando los permisos para los usuarios designados y el resultado para la identificación y así activar o no los servomotores.

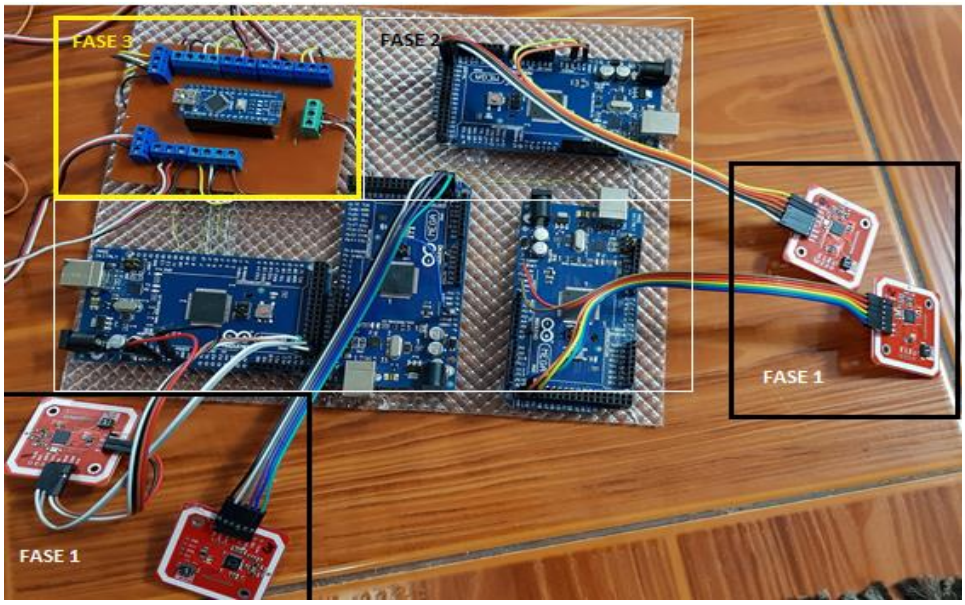


Figura. 2. 29. Descripción de Fases

Fuente: Elaborado por el autor

2.3. Diseño del Prototipo

El prototipo está basado en oficinas de una empresa genérica de reparación de computadoras, en donde es necesaria la implementación de un control de accesos para evitar pérdidas de equipos y verificar la asistencia de los trabajadores.

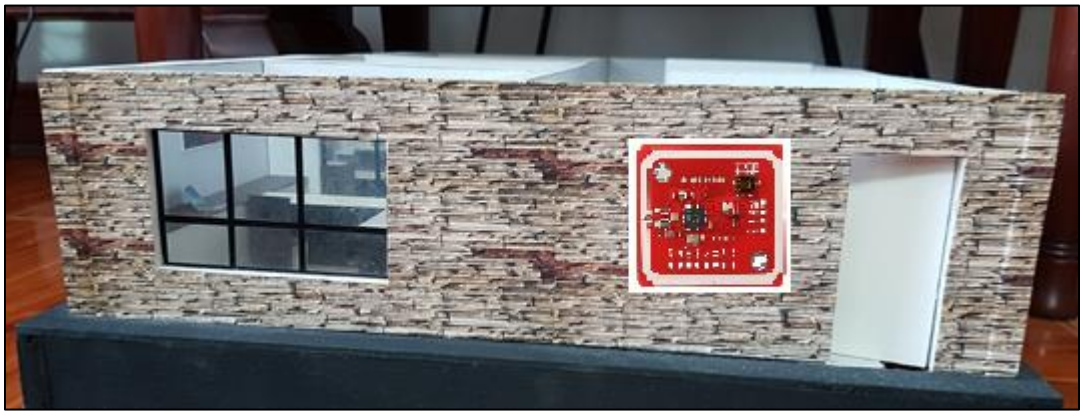


Figura. 2. 30. Parte frontal del Prototipo

Fuente: Elaborado por el autor

Figura. 2. 31. Parte izquierda del Prototipo

Fuente: Elaborado por el autor

El prototipo como consta en la figura está conformado por 4 puertas principales:

Puerta 1: Es la puerta principal para el ingreso a la empresa y sin acceso a esta no se puede abrir ninguna otra puerta

Puerta 2: Es la Gerencia en donde se administra la empresa.

Puerta 3: Es la oficina de Diseño en donde se crean nuevos equipos.

Puerta 4: Es la oficina de Implementación en donde los equipos son elaborados.

Estas son las principales oficinas en donde es necesario el control de acceso.

Figura. 2. 32. Diseño del Prototipo

Fuente: Elaborado por el autor

2.4. Diseño de la aplicación móvil

La aplicación fue diseñada en la plataforma App Inventor 2, la cual permite crear aplicaciones sencillas pero óptimas para personas las que interactúan mucho con diferentes tecnologías en el campo de la programación pudiendo ser para Android y algunas para Ios; en la figura 24, se muestra el diseño de la aplicación sencilla para un fácil uso.

2.4.1. Interfaz Visible

Es el contenido que se muestra cada vez que en el teléfono se abre la aplicación, que además cuenta con los permisos específicos de cada usuario.

Figura. 2. 33. Interfaz Visible del Administrador

Fuente: Elaborado por el autor

La aplicación solo necesita ser abierta y colocarla frente al módulo NFC para que pueda ser leída, además de tener un manual que es donde se indica como colocar el móvil.

2.5. Diseño de aplicación en la PC

Se desarrolló una aplicación dentro del programa Visual Basic para que el usuario pueda observar la hora, fecha y el número de puerta que fue abierta para un mejor control.

Mediante esta aplicación se podrá bloquear a usuarios no permitidos mediante un botón colocado en la interfaz de la aplicación, además de poder observar los pulsos de NFC que estén por registrarse.

La aplicación creada en Visual Basic funciona fácilmente en muchas computadoras y es sencilla de usar para el registro de entradas y salidas.

2.5.1. Interfaz gráfica

Interfaz Visible de reconocimiento de Códigos proporcionados por la aplicación móvil

Es el contenido que se muestra cada vez que se abre la aplicación

El cable USB está conectado con el Arduino central (NANO).

Además, muestra la fecha y la hora exacta para ser añadida en los registros.

Permite bloquear o desbloquear accesos según el usuario indicado.

El registro de usuarios que ingresaron y salieron se imprime directamente en Microsoft Excel.

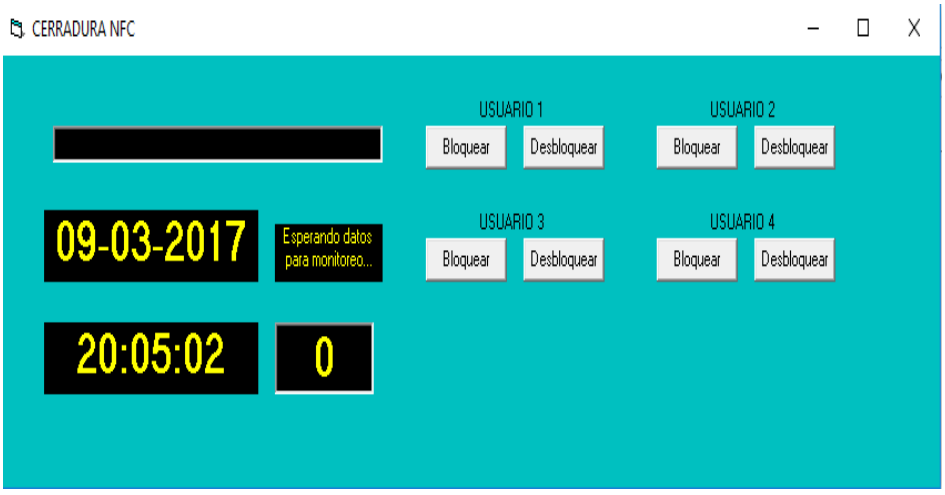


Figura. 2. 34. Aplicación en Visual Basic

Fuente: Elaborado por el autor

	A	B	C
1	DATOS	HORA	FECHA
2	#1 REGISTRA SALIDA	15:34:50	03/08/2018
3	#1 REGISTRA ENTRADA	15:34:55	03/08/2018
4	#1 REGISTRA SALIDA	15:35:05	03/08/2018
5	#1 REGISTRA ENTRADA	16:01:05	03/08/2018
6	#1 REGISTRA SALIDA	16:01:05	03/08/2018
7	#1 REGISTRA ENTRADA	16:05:05	03/08/2018
8	#1 REGISTRA SALIDA	16:08:05	03/08/2018
9	#1 REGISTRA ENTRADA	16:15:05	03/08/2018
10	#1 REGISTRA SALIDA	16:20:05	03/08/2018
11	#1 REGISTRA ENTRADA	16:30:05	03/08/2018

Figura. 2. 35. Registros de entradas y salidas en Microsoft Excel

Fuente: Elaborado por el autor

2.6. Programación Arduino NANO

La programación en el Arduino NANO es la fundamental para el funcionamiento del sistema ya que mediante este programa se controlarán los códigos que ingresaron a los 4 arduinos MEGA.

Este programa identifica mediante los códigos, que tipo de usuario es el ingresado y así otorgar permisos y restricciones para las puertas.

2.7. Programación Arduino MEGA

La programación en el Arduino MEGA es exclusiva para la recepción del código que el módulo NFC recibió, y así mediante las librerías que solo el Arduino MEGA posee, leer el código y aceptarlo o negarlo (librería: PN532_SPI.h, snep.h, NdefMessage.h). En la figura se puede visualizar el código 1234 perteneciente a la primera puerta (entrada) y sin este código no es posible ingresar a ninguna otra puerta.

La librería SPI (del inglés *Serial Peripheral Interface*), es el medio de comunicación en la programación del dispositivo NFC y el Arduino MEGA.

```
#include "SPI.h"                //Librería de comunicación serial
#include "PN532_SPI.h"         // Comunicación con el módulo NFC
#include "snep.h"              // Librería para el módulo NFC
#include "NdefMessage.h"       // Librería para el módulo NFC
int outusuariol = 3;           // Nombres de los pines
//int outusuario2 = 4;
int SW = 12;                   // Pin Arduino MEGA que permite que el módulo NFC sea transmisor o receptor
bool R_state = 1;              // Variable
bool G_state = 1;              // Variable
PN532_SPI pn532spi(SPI, 53);   // COnfiguración de pines
SNEP nfc(pn532spi);            // Inicializando
uint8_t ndefBuf[128];          // valores de las librerías
```

Figura. 2. 36. Descripción Inicialización del programa

Fuente: Elaborado por el autor

```
String readMsg( NdefRecord record ) {
  int payloadLength = record.getPayloadLength();
  byte payload[payloadLength];
  record.getPayload(payload);
  String payloadAsString = "";
  for (int c = 0; c < payloadLength; c++) {
    payloadAsString += (char)payload[c];
  }
  return payloadAsString.substring(3);
}
```

Proceso de recepción y almacenamiento de datos en un espacio de memoria

Figura. 2. 37. Almacenamiento de datos

Fuente: Elaborado por el autor

```
void setup() {
  Serial.begin(9600);                    // Comunicación al PC
  pinMode(outusuario1, OUTPUT);          // Salida Arduino MEGA
  //pinMode(outusuario2, OUTPUT);
  Serial.println("NFC Peer to Peer");    // Imprimir
```

Figura. 2. 38. Comunicación con la PC

Fuente: Elaborado por el autor

```
void loop() {

  if ( digitalRead(SW) == 0) {
    getMsgFromAndroid();
  } else {
    SendMsgToAndroid();
  }
  delay(3000);
}
```

NFC
Mediante el comando "if" y la opción 0 se visualizará si es receptor, caso contrario es emisor

Figura. 2. 39. Opción de recepción módulo NFC

Fuente: Elaborado por el autor

2.8. Croquis del prototipo de las oficinas

El prototipo está basado en una empresa genérica de reparación de computadoras y creación de páginas web en donde existen 3 oficinas, la Gerencia, el Diseño y la Implementación y cada oficina contiene en el control de acceso en la parte derecha de cada puerta, incluido la puerta principal de entrada.

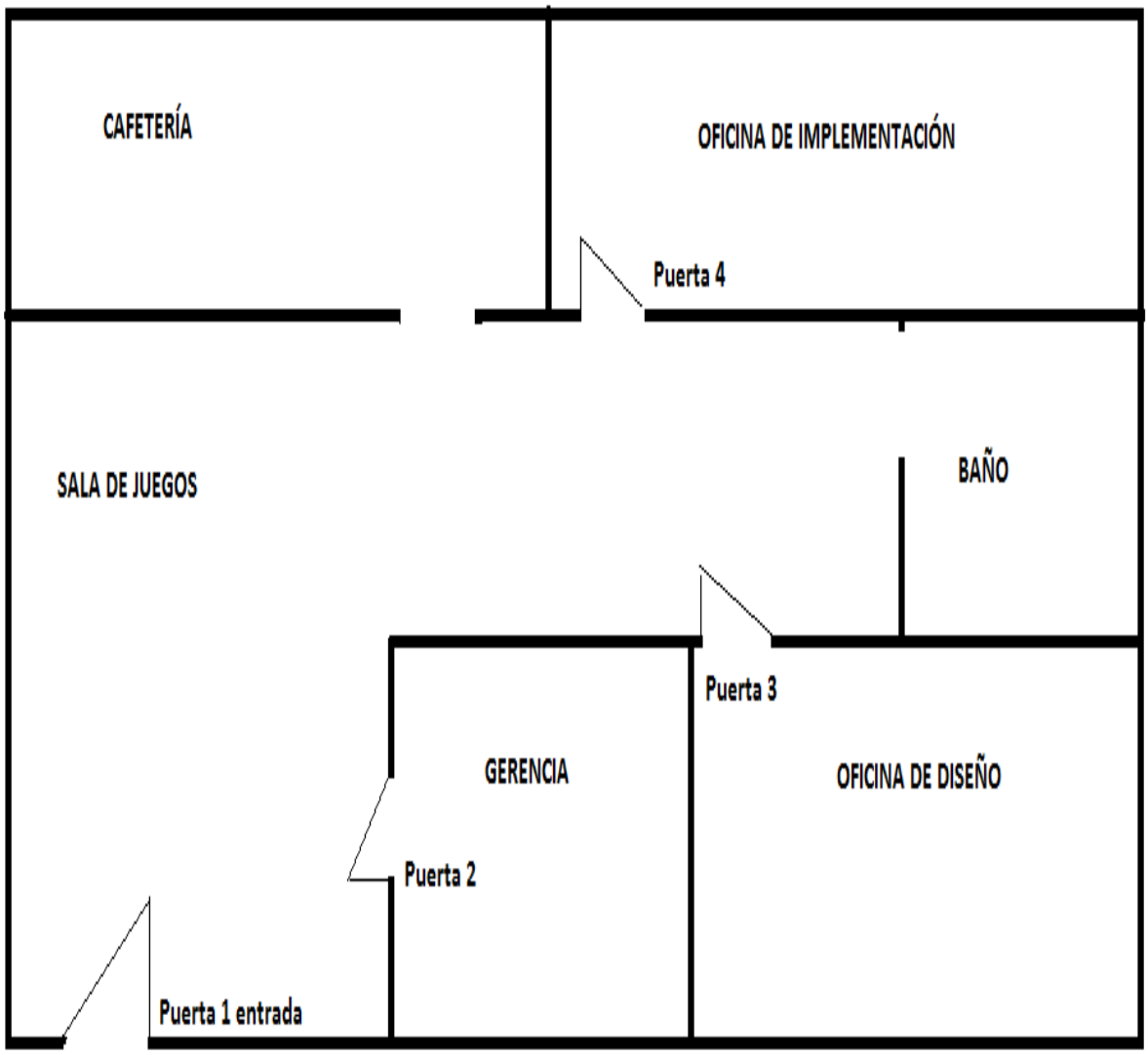
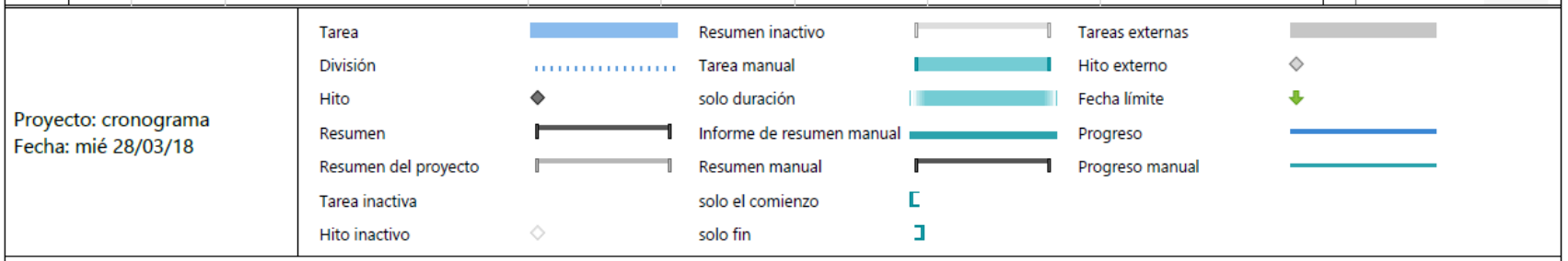


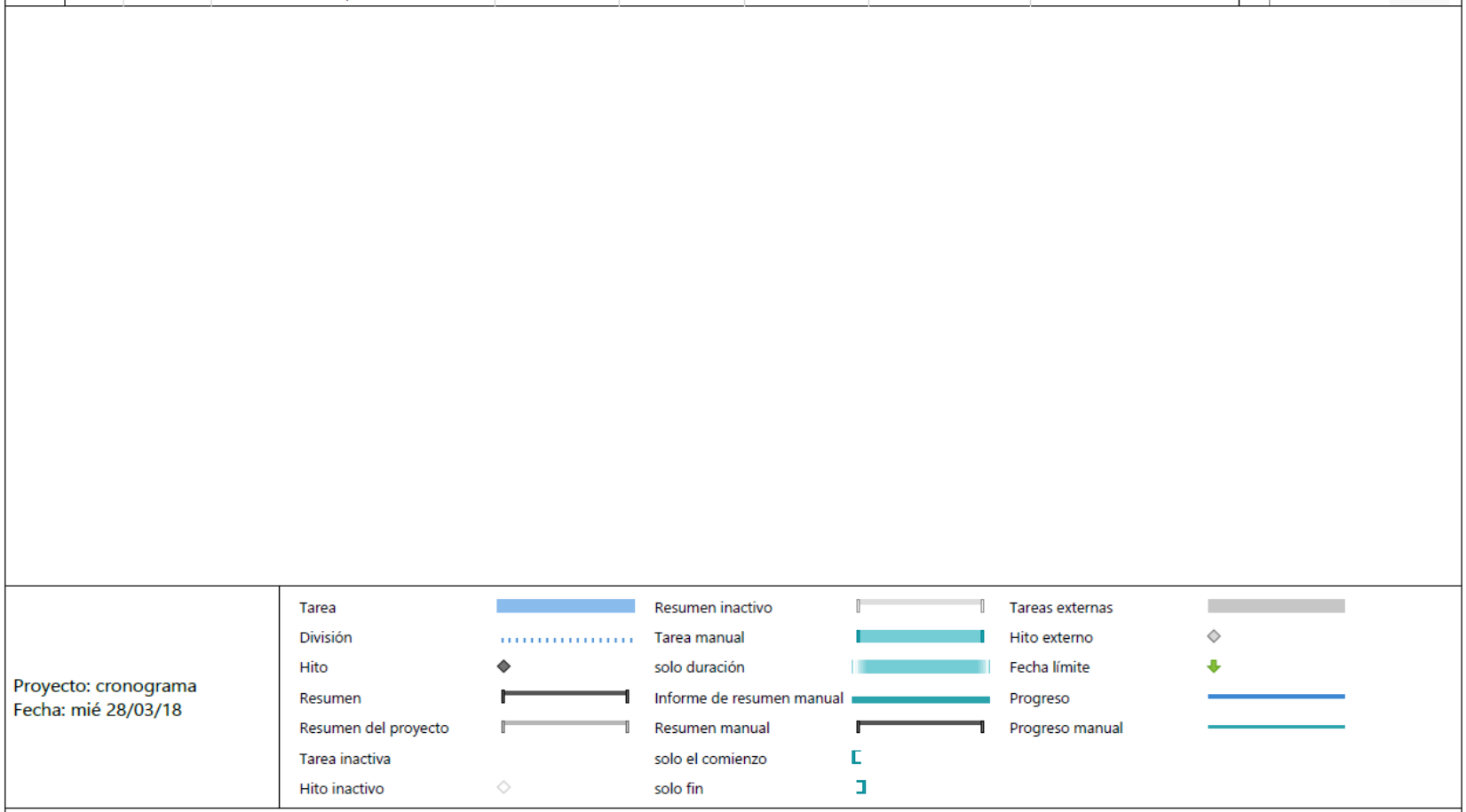
Figura. 2. 40. Croquis del prototipo

Fuente: Elaborado por el autor

Id	Modo de tarea	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Nombres de los recursos
1		**Plan de Proyecto**	**34 días**	**mié 15/11/17**	**sáb 30/12/17**		
2		Investigar el tema de	4 días	mié 15/11/17	sáb 18/11/17		
3		Elaboración del documento	32 horas	dom	mié 22/11/17		
4		Rectificación y Corrección del	184 horas	jue 23/11/17	lun 25/12/17		
5		Firma y Entrega del plan de	24 días	sáb 25/11/17	mié 27/12/17		
6		Cronograma de Actividades a	4 días	lun 27/11/17	jue 30/11/17		
7		**Investigacion de tecnologías**	**30 días**	**mar**	**lun 15/01/18**		
8		Investigación de la tecnología	5 días	mar 05/12/17	dom		
9		Investigación de la	6 días	dom	vie 15/12/17		
10		**Elaboración del Prototipo para**	**42 días**	**sáb 06/01/18**	**lun 05/03/18**		
11		Compra de materiales	2 días	sáb 06/01/18	lun 08/01/18		
12		Instalación de software y	3 días	lun 08/01/18	mié 10/01/18		
13		Programación del Arduino	12 días	jue 11/01/18	vie 26/01/18		
14		Programación del los	7 días	sáb 27/01/18	lun 05/02/18		
15		Elaboración de la maqueta de	7 días	mar 06/02/18	mié 14/02/18		
16		Programación del Arduino	6 días	jue 15/02/18	jue 22/02/18		
17		Diseño de una placa de	3 días	jue 22/02/18	sáb 24/02/18		
18		Diseño de una aplicación en	4 días	dom	mié 28/02/18		
19		Implementación del sistema	2 días	mié 28/02/18	jue 01/03/18		
20		Correciones del prototipo	2 días	jue 01/03/18	vie 02/03/18		
21		Pruebas Finales	2 días	vie 02/03/18	lun 05/03/18		
22		**Elaboración del documento**	**84 días**	**mié 15/11/17**	**sáb 10/03/18**		
23		Introducción	19 días	mié 15/11/17	lun 11/12/17		



Id	Modo de tarea	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Nombres de los recursos
24		Desarrollo del prototipo	52 días	lun 11/12/17	mar 20/02/18		
25		Justificación	6 días	mar 20/02/18	mar 27/02/18		
26		Conclusiones,	10 días	mar 27/02/18	sáb 10/03/18		



3. CAPÍTULO III IMPLEMENTACIÓN

3.1. Desarrollo

3.1.1. Fabricación de la placa

La placa PCB se realizó de acuerdo al diseño de la figura. 25 mediante una impresora láser en papel Transfer. Al tener ya impreso el diseño se procede a calentar la impresión sobre la baquelita de cobre.

Figura. 3. 41. Calentamiento de la impresión sobre la Baquelita

Fuente: Elaborado por el autor

Al haber realizado el proceso anterior, se obtendrá las pistas electrónicas dibujadas en la baquelita. Se debe asegurar de que todas las pistas hayan sido plasmadas en la placa para que al sumergirla en agua y ácido cloruro férrico deje totalmente limpio el diseño.

Figura. 3. 42. Impresión del diagrama en la baquelita

Fuente: Elaborado por el autor

Se procede a realizar los huecos correspondientes a los elementos que se van a soldar en la placa según su primer diseño

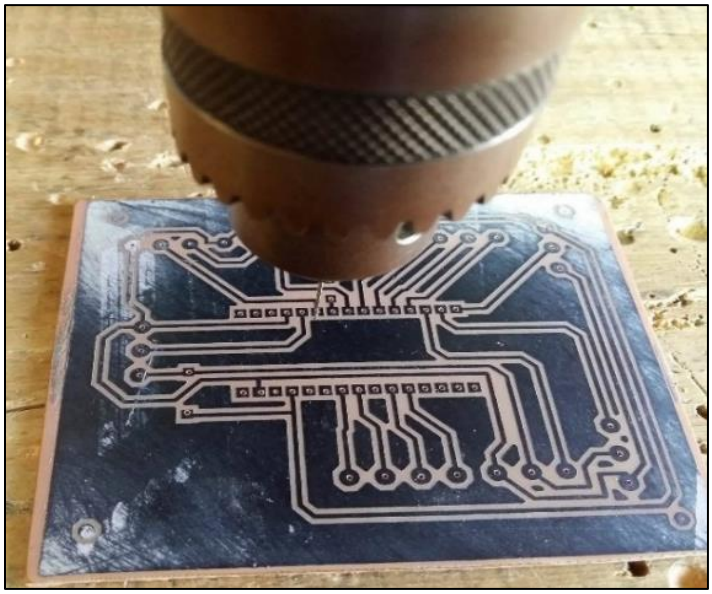


Figura. 3. 43. Perforación de la placa

Fuente: Elaborado por el autor

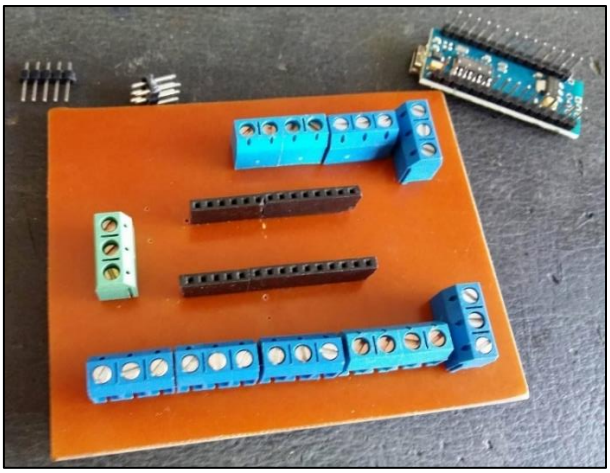


Figura. 3. 44. Elementos colocados en la placa

Fuente: Elaborado por el autor

Al integrar todos los componentes en la placa mencionada se encuentra terminada la baquelita.

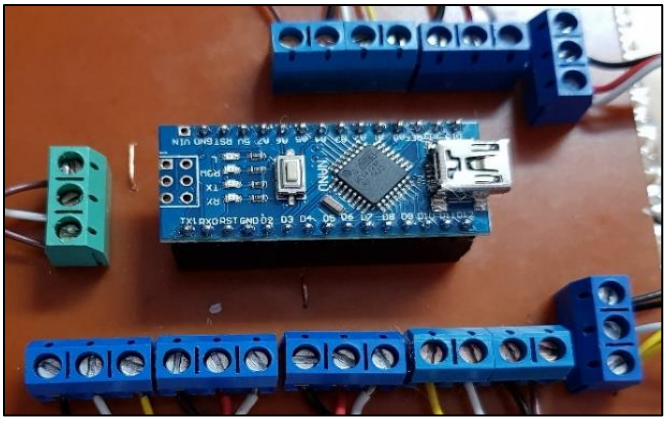


Figura. 3. 45. Placa Terminada

Fuente: Elaborado por el autor

3.2. Implementación del Sistema de Acceso

Al obtener el conocimiento del procedimiento para realizar el control de accesos se procede a la implementación de todos los componentes y aplicaciones para el funcionamiento final del prototipo.

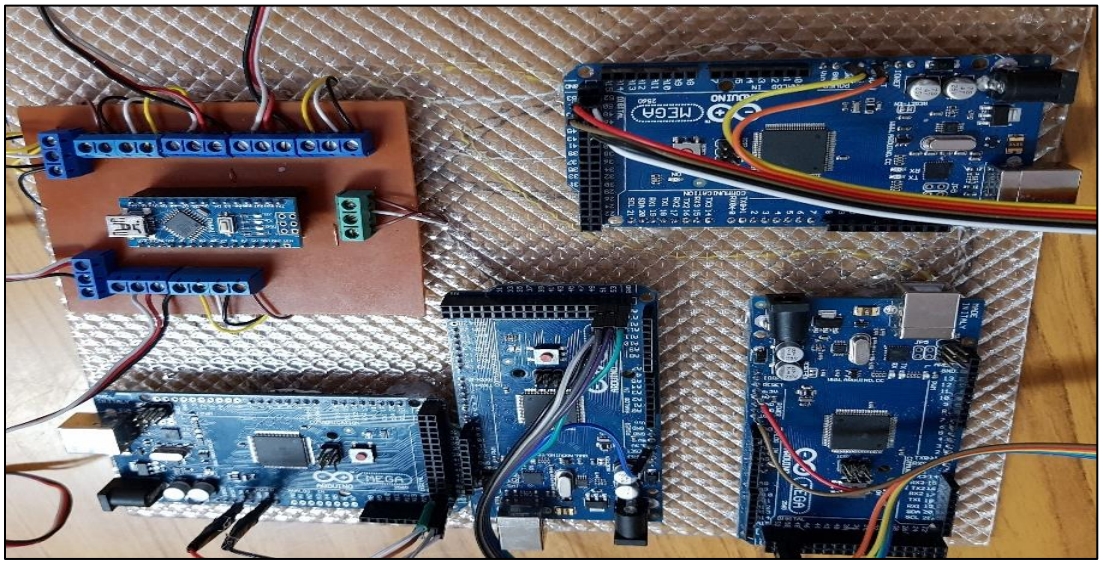


Figura. 3. 46. Implementación de los Arduinos MEGA con el arduino NANO

Fuente: Elaborado por el autor

Al tener todas las conexiones se añaden los módulos NFC y los servomotores los cuales realizan el proceso de recibir los códigos entregados por la aplicación de celular.

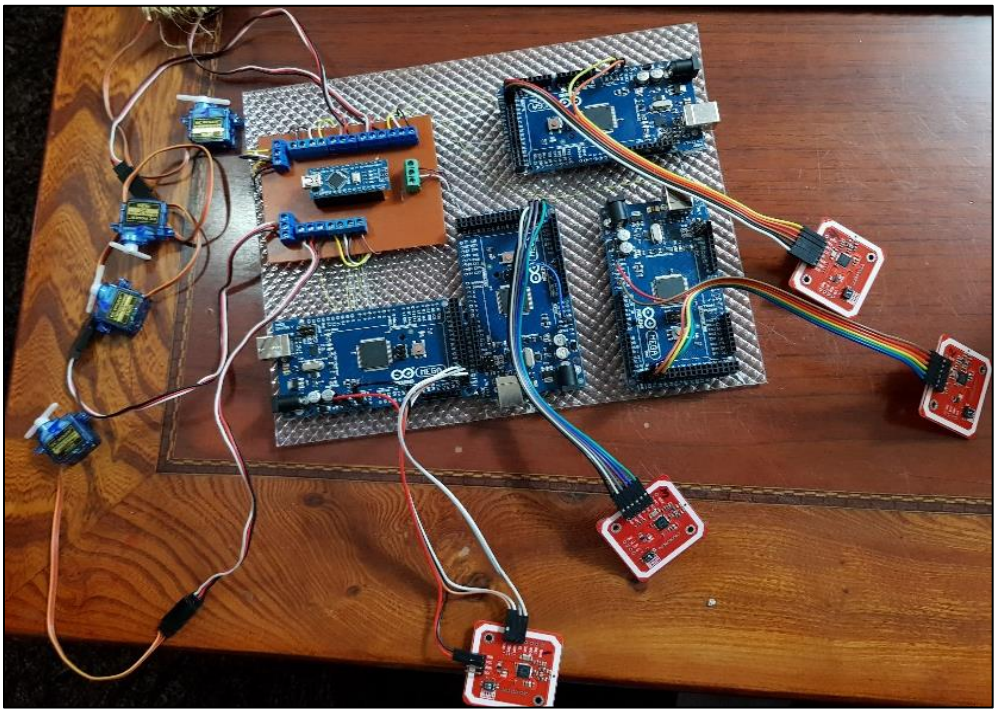


Figura. 3. 47. Sistema completo para el funcionamiento del prototipo

Fuente: Elaborado por el autor

Los módulos NFC y los servomotores no pueden ser insertados en la placa ya que tienen sus funciones en la parte visible del prototipo siendo los módulos NFC los principales transmisores de códigos y los servomotores se encargarán de simular el movimiento de las puertas.

3.3. Implementación de aplicación móvil

3.3.1. Inicio de aplicaciones

Se elaboraron 4 aplicaciones móviles las cuales contienen los códigos secretos de acceso, además de la descripción para cada usuario.

El proceso para un correcto funcionamiento de la aplicación móvil es el siguiente:

a) En el móvil: Dirigirse a AJUSTES, CONEXIONES, NFC Y PAGO, activar la opción de encendido de NFC y la de Android Beam en caso de tenerla. Como se muestra en la figura 48.

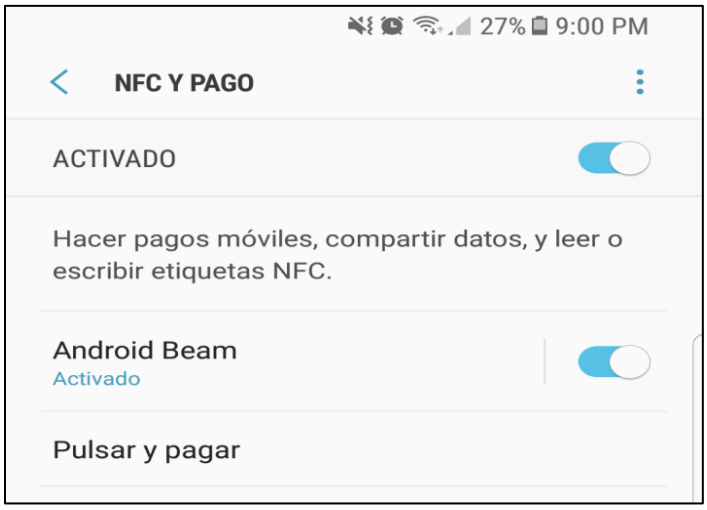


Figura. 3. 48. Encendido del NFC en el móvil

Fuente: Elaborado por el tutor

b) El archivo que contiene la aplicación será facilitado por el administrador el cual debe ser instalado en el teléfono.

c) Después de la instalación de la aplicación móvil aparecerá un icono en el teléfono como se muestra en la figura 49. Cada aplicación según el usuario que fue provisto por el administrador.

Figura. 3. 49. Aplicaciones Instaladas

Fuente: Elaborado por el autor

d) Al abrir la aplicación se muestra el logo de la Universidad Israel, además del logo NFC el cual representa en que se basa la aplicación. A continuación en las figuras 50, 51, 52,53 se observa las 4 aplicaciones abiertas de los 4 usuarios, y el código oculto que cada una contiene.

Código secreto: Es una combinación de símbolos que, en el marco de un sistema ya establecido, cuenta con un cierto valor. Cada uno de los caracteres tiene un código digital equivalente. Esto se denomina código ASCII.

El código ASCII básico representa caracteres utilizando 7 bits (para 128 caracteres posibles, enumerados del 0 al 127). Por lo tanto una cadena de caracteres son enviados para recibir la misma cadena de caracteres como por ejemplo: 1234 es enviado en código ASCII para recibir los mismos caracteres.

Figura. 3. 50. Interfaz visible del Administrador

Fuente: Elaborado por el autor

Figura. 3. 51. Interfaz Visible del Gerente

Fuente: Elaborado por el autor

Figura. 3. 52. Interfaz Visible del Diseñador

Fuente: Elaborado por el autor

Figura. 3. 53. Interfaz Visible de la oficina de Implementación

Fuente: Elaborado por el autor

Para iniciar la aplicación se debe descargar desde App Inventor 2 y después instalarlo en el móvil sin necesidad de usar un usuario o contraseña ya que la aplicación contiene ya la información del usuario asignado.

Para la transmisión de datos es necesaria la capa física en modelo OSI (en inglés, *Open System Interconnection*) la cual se encarga de la transmisión y recepción de una secuencia no estructurada de bits a través de un medio.

- Frecuencia de trabajo: 13.56 Mhz: Frecuencia libre de uso
- Protocolo de comunicación: NFCIP-1: Combina dos protocolos de comunicación que pertenecen al RFID
- Tipo de codificación: Manchester

  Es una codificación auto sincronizada, ya que en cada bit se puede obtener la señal de reloj, lo que hace posible una sincronización precisa del flujo de datos.

- Modulación: ASK (*Amplitude-shift keying*) , modulación por desplazamiento en amplitud.

  Es una modulación de amplitud donde la señal moduladora (datos) es digital.

- NFC *Exchanged Format* (NDEF), especificación de un formato común para el intercambio de datos.

3.3.2. Aplicación en App Inventor 2

Mediante este programa se puede realizar diferentes tipos de aplicaciones para personas con o sin experiencia ya que es fácil de programar y su interfaz es amigable.

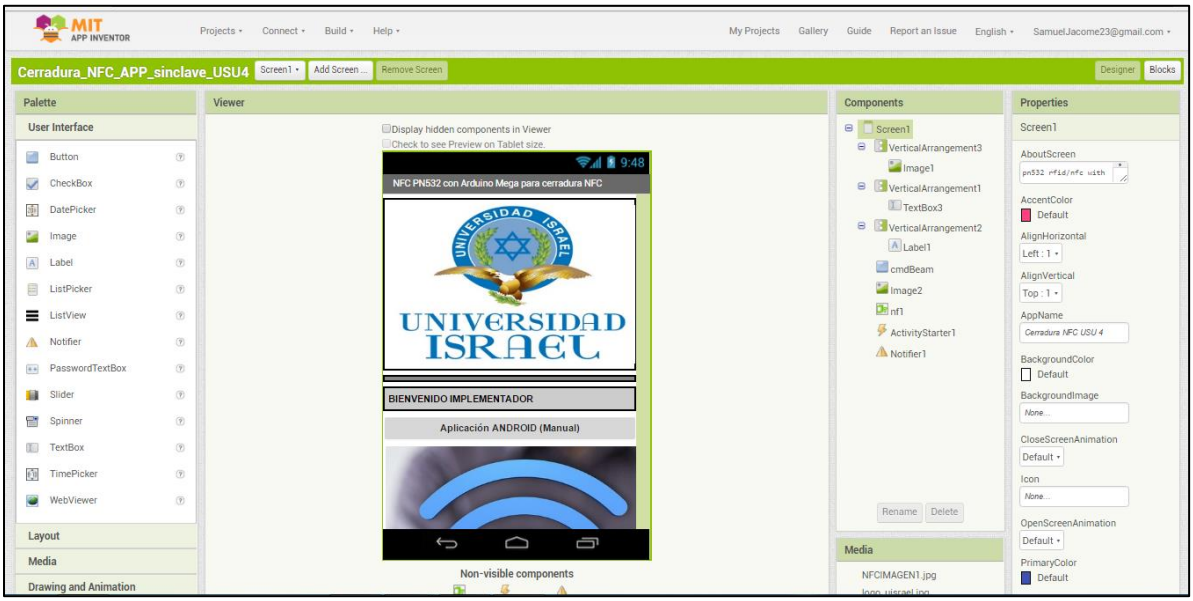


Figura. 3. 54. Programación y diseño de la aplicación

Fuente: Elaborado por el autor

En la figura 54 se muestra la interfaz que se observará en el móvil permitiendo aquí diseñar de una manera muy cómoda añadiendo imágenes, textos e iconos de acuerdo a las necesidades requeridas.

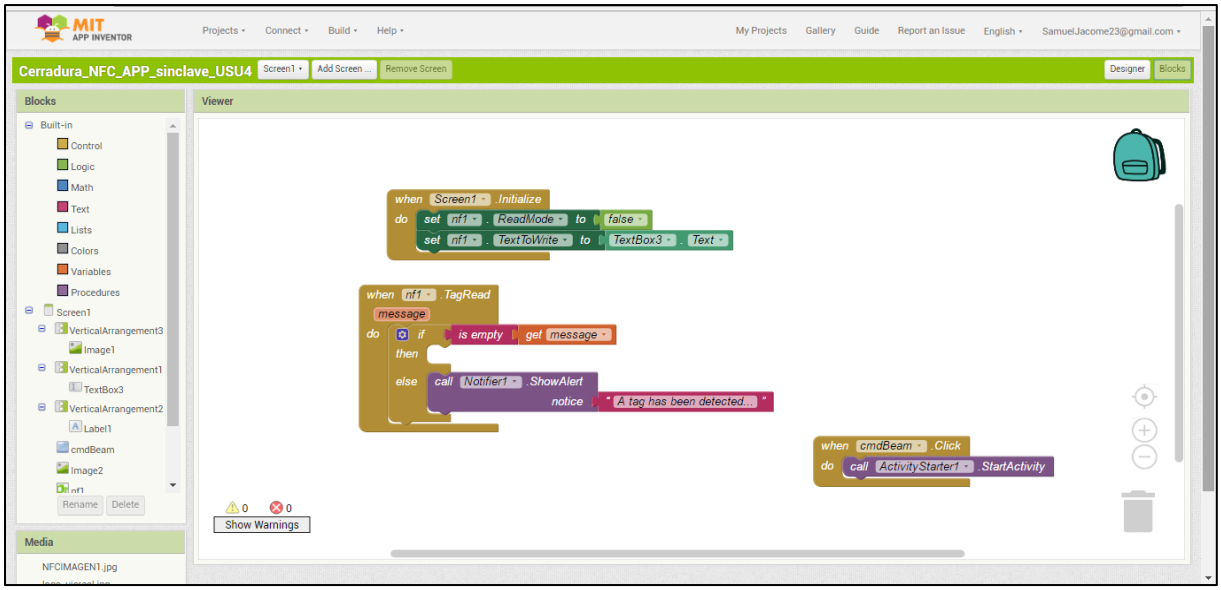


Figura. 3. 55. Programación de la aplicación y sus diferentes opciones

Fuente: Elaborado por el autor

La figura 55 muestra la programación por medio de bloques, los cuales envían el código ingresado para que todo el sistema funcione. Y así cumplir los objetivos propuestos para una mayor seguridad, mediante la encriptación de las contraseñas, evitando así un hackeo o intento de robo.

3.4. Pruebas de Funcionamiento

Para una mejor garantía del sistema se realizaron pruebas de funcionamiento, en las cuales se probó el estado de los módulos NFC, los Arduinos y los servomotores. La tabla 10 muestra la función principal de los equipos, la verificación de su funcionamiento en el circuito del proyecto y varias observaciones sobre su uso.

Tabla. 3. 10. Pruebas de funcionamiento

EQUIPOS	Función	Verificación	Observaciones
Módulos NFC	Detección de tarjetas y aplicaciones NFC	El equipo funcionó correctamente al leer los códigos entregados por las aplicaciones móviles.	No se debe deslizar el equipo a una distancia mayor a 5cm, porque su rango no es tan amplio.

Arduinos MEGA, NANO	Guardar la programación deseada y hacerla correr en un circuito o sistema	Los equipos funcionaron correctamente al hacer correr los programas cargados en las placas.	No se debe presionar el botón de reset de las placas ya que se borrará toda la programación insertada.
Servomotores	Moverse en ángulos fijos en respuesta a una señal de control	El equipo funcionó satisfactoriamente al moverse en diferentes ángulos	No se debe utilizar los servomotores para mover objetos de gran peso, ya que los servo son muy frágiles

Fuente: Elaborado por el autor

3.5. Análisis de Resultados

Después de la investigación de las mejores tecnologías para los sistemas de acceso, se procedió al prototipo donde se instaló dicho sistema de control de accesos, sin dejar de lado el estudio estratégico de la ubicación de los equipos, lo adquirido para el prototipo fue en base a las necesidades de las empresas.

El cableado se realizó mediante el cable UTP para el sistema conectado punto a punto, y así evitar inconvenientes como ruido o interferencias de señales.

La PC es de vital importancia para observar los movimientos sucedidos durante la jornada de trabajo, pero no es necesario tener computadoras sofisticadas ya que el sistema no requiere grandes características para funcionar, siendo sencillo de usar solo conectando el cable USB

Los módulos NFC PN532 fueron los más adecuados para ser instalados ya que su funcionamiento es óptimo al momento de recoger códigos y permite que el sistema no contenga errores.

En la parte de apertura de puertas se instaló servomotores SG90 los cuales se encargan de abrir y cerrar las puertas para el ingreso del personal, mediante un movimiento de 45 grados.

Finalmente, después de la observación de todos los elementos que conforman el sistema se puede establecer que el presente prototipo ha culminado satisfactoriamente dando por cumplidos los objetivos propuestos.

3.6. Implementación Final

3.6.1. Módulos NFC instaladas en el Prototipo

Figura. 3. 56. Implementación frontal final

Fuente: Elaborado por el autor

Figura. 3. 57. Implementación Superior Final

Fuente: Elaborado por el autor

3.7. Presupuesto

Tabla. 3. 11. Presupuesto

Equipos	Cantidad	Valor Unitario	Total
Módulo NFC	4	30.00	120.00
Arduino MEGA	4	25.00	100.00
Servomotor	4	8.00	32.00
Arduino NANO	1	12.00	12.00
Total			264.00

Fuente: Elaborado por el autor

En la tabla se muestra los materiales usados con sus respectivos precios para el proyecto realizado.

4. CONCLUSIONES

- Se eligió el módulo que más rápido estableció la comunicación con los Arduinos, para la implementación del sistema de control de registros de accesos mediante tecnología NFC en el prototipo.

- Se implementó el módulo NFC PN532 el cual permite la lectura de todos los códigos que ingresan y un correcto funcionamiento del sistema.

- Se desarrolló la aplicación móvil para el administrador y sus trabajadores, la cual transmite los códigos hacia los receptores NFC, para ser identificados y distribuidos de acuerdo al puesto de trabajo.

- Se desarrolló la aplicación en la PC para el administrador, el mismo que puede monitorear el sistema de control de accesos y registros, creando permisos o negando accesos.

- Se realizó varias pruebas de funcionamiento y validación de datos, al tener en cuenta esta implementación se puede afirmar que el equipo está totalmente operativo para cualquier adversidad que se presente, por ende, el prototipo de control de accesos se encuentra protegido ante cualquier evento a suscitarse.

5. RECOMENDACIONES

- Para el correcto funcionamiento del sistema, se debe verificar que los cables encargados de la alimentación no se encuentren doblados o rotos, para evitar inconvenientes al encender el sistema.

- Las pruebas de funcionamiento están establecidas para los equipos mencionados, si se requiere adherir más equipos se debería realizar una modificación en la programación para evitar fallas en el sistema.

- El administrador principal encargado del control de accesos y registros, será responsable del manejo de las claves colocadas en las aplicaciones móviles.

- Se requiere de un mantenimiento de los dispositivos colocados cada cierto tiempo para un mejor funcionamiento y así evitar falsos códigos y deterioro de los dispositivos.

6. REFERENCIAS BIBLIOGRÁFICAS

Arduino. (2017). *MANTECH.* Obtenido de http://www.mantech.co.za/datasheets/products/A000047.pdf

Arroyo Briones , B. A., Contreras Bernal , G. A., & Espíritu de la Paz, E. A. (25 de 02 de 2016). *Tesis Instituto Politécnico Naiconal*. Obtenido de http://tesis.ipn.mx/jspui/bitstream/123456789/17226/1/tesis_control%20de%20acceso%20mediante%20nfc.pdf

Contreras, L. (21 de Noviembre de 2012). *Historia de la Informática*. Obtenido de http://histinf.blogs.upv.es/2012/11/21/nfc/

Crespo, E. (24 de Noviembre de 2014). *Aprendiendo Arduino*. Obtenido de https://aprendiendoarduino.wordpress.com/2016/09/25/que-es-arduino/

Del Campo García, M. (13 de Enero de 2016). *Mi Arduino UNO tiene un Blog*. Obtenido de https://miarduinounotieneunblog.blogspot.com/2016/01/programar-posiciones-en-un-micro-servo.html

Electronilab. (s.f.). Obtenido de https://electronilab.co/tienda/arduino-nano-v3-atmega328-5v-cable-usb/

ETOLOCKA. (15 de Octubre de 2015). *PROFE TOLOCKA*. Obtenido de http://www.profetolocka.com.ar/2015/10/15/lcd-y-keypad-shield-para-arduino/

García Gonzáles, A. (23 de Enero de 2013). *Panamahitek.* Obtenido de http://panamahitek.com/arduino-mega-caracteristicas-capacidades-y-donde-conseguirlo-en-panama/

GeekFactory. (s.f.). *GeekFactory*. Obtenido de https://www.geekfactory.mx/tienda/motores-y-controladores/servo-sg90-tower-pro/

GeekFactory. (s.f.). *GeekFactory*. Obtenido de https://www.geekfactory.mx/

Igoe, T. (2014). *Beginning NFC.* Estados Unidos: O´Reilly Media .

internacional, Forum. (18 de 2 de 2017). *NFC Forum.* Obtenido de http://nfc-forum.org/about-us/nfc-forum-member-meetings/

JADIAZ. (21 de Enero de 2016). *MiArduino*. Obtenido de http://www.iescamp.es/miarduino/2016/01/21/placa-arduino-uno/

Llamas, L. (18 de Octubre de 2016). *Ingeniería, informática y diseño*. Obtenido de https://www.luisllamas.es/reloj-y-calendario-en-arduino-con-los-rtc-ds1307-y-ds3231/

Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, and David Mellis. (2017). *ARDUINO.* Obtenido de https://www.arduino.cc/

Patagoniatec. (07 de febrero de 2013). *patagoniatec*. Obtenido de http://saber.patagoniatec.com/arduino-nano-328-arduino-atmega-clon-compatible-arduino-argentina-ptec/

Penalva, J. (25 de 01 de 2011). *Xataka*. Obtenido de https://www.xataka.com/moviles/nfc-que-es-y-para-que-sirve

Penalva, J. (15 de Junio de 2017). *Xataka*. Recuperado el 25 de Enero de 2011, de Xataka: https://www.xataka.com/moviles/nfc-que-es-y-para-que-sirve

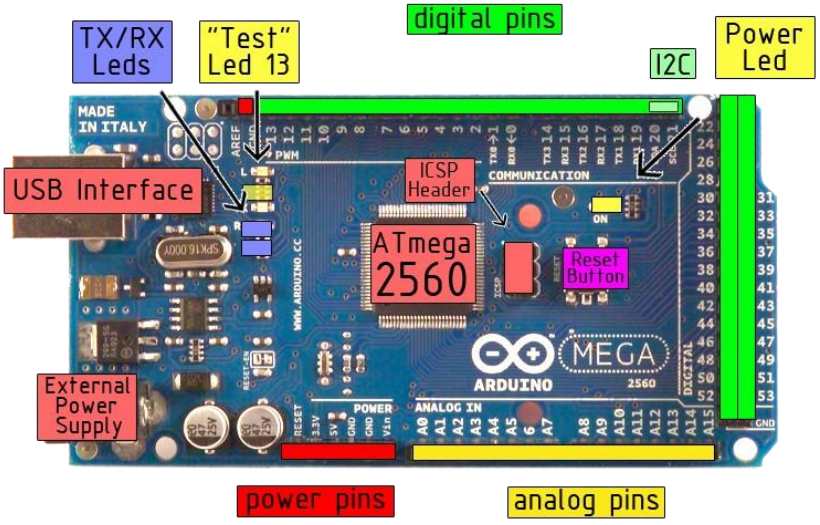
Simpson, C. (01 de 03 de 2017). *APP INFORMERS*. Obtenido de http://appinformers.com/what-is-nfc-and-how-is-it-used/6738/

Thayer Ojeda, L. (s.f.). *Arduino.cl*. Obtenido de http://arduino.cl/arduino-mega-2560/

7. ANEXOS

Anexo A: Arduino Mega

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 (datasheet). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.



Microcontroller	**ATmega2560**	
Operating Voltage	5v Input Voltage (recommended)	7-12V Input
Voltage (limits)	6-	20V
Digital I/O Pins	54 (of which 14 provide PWM output)	
Analog Input Pins	16	
DC Current per I/O Pin	40 mA	
DC Current for 3.3V Pin	50 mA	
Flash Memory	256 KB of which 8 KB used by bootloader	
SRAM	8 KB	
EEPROM	4 KB	
Clock speed	16Mhz	

The Arduino Mega2560 can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

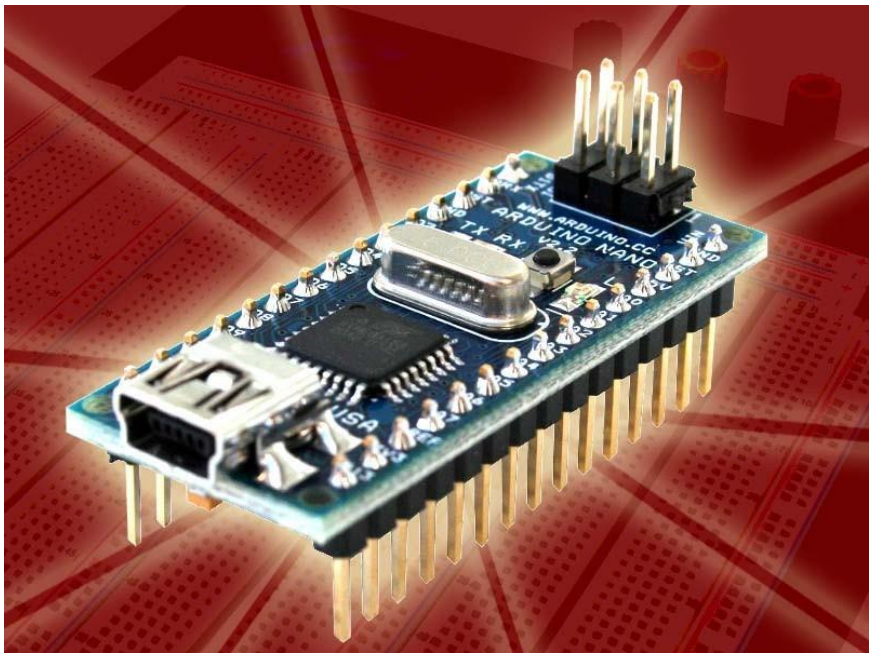
The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the EEPROM library).

Anexo B: Arduino Nano



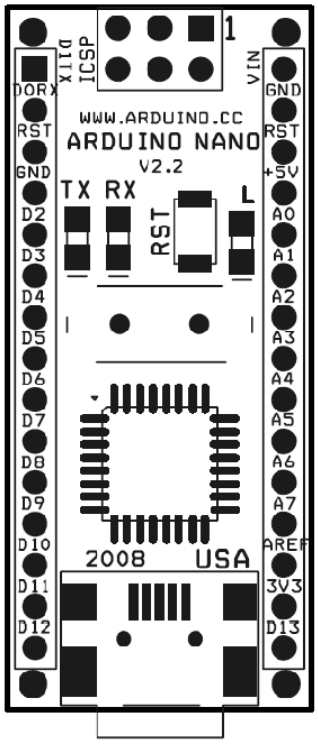
Arduino Nano Pin Layout

D1/TX(1)
D0/RX(2)
RESET(3)
GND(4)
D2(5)
D3(6)
D4(7)
D5(8)
D6(9)

(30) VIN
(29) GND
(28) RESET
(27) +5V
(26) A0
(25) A1
(24) A2
(23) A3
(22) A4

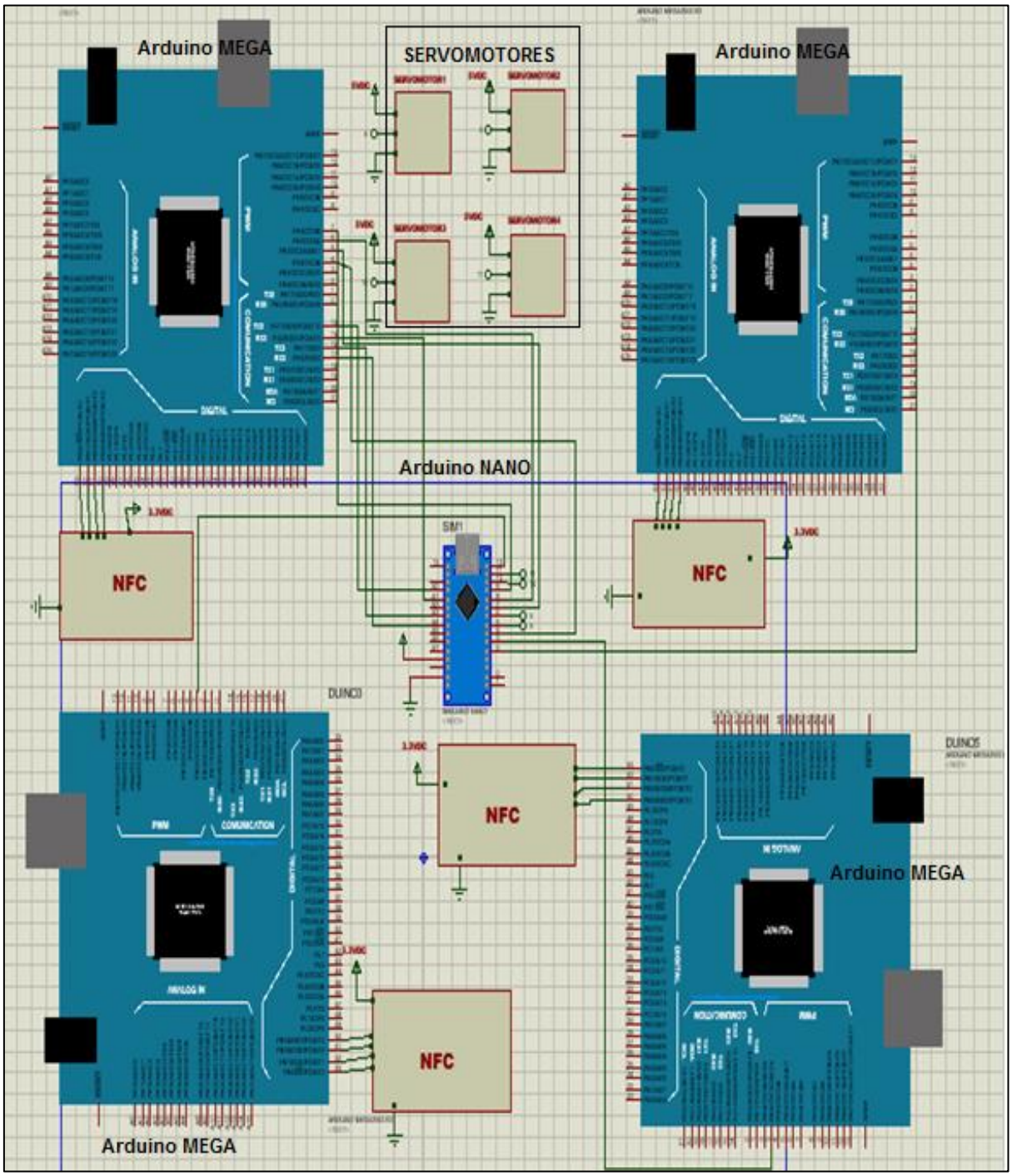
D7 (10)
D8 (11)
D9 (12)
D10 (13)
D11 (14)
D12 (15)

(21) A5
(20) A6
(19) A7
(18) AREF
(17) 3V3
(16) D13



Pin No.	Name	Type	Description
1-2, 5-16	D0-D13	I/O	Digital input/output port 0 to 13
3, 28	RESET	Input	Reset (active low)
4, 29	GND	PWR	Supply ground
17	3V3	Output	+3.3V output (from FTDI)
18	AREF	Input	ADC reference
19-26	A7-A0	Input	Analog input channel 0 to 7
27	+5V	Output or	+5V output (from on-board regulator) or
30	VIN	PWR	Supply voltage

Anexo C: Esquemático de la tarjeta de control



Anexo D: Programación Arduino MEGA 1

```
//Se está probando en IDE Arduino 1.6.12

// Receive a NDEF message from a Peer

// Requires SPI. Tested with Seeed Studio NFC Shield v2

//MODULO NFC   ARDUINO MEGA

//   VCC          3VDC

#include "SPI.h"

#include "PN532_SPI.h"

#include "snep.h"

#include "NdefMessage.h"

int outusuario1 = 3;

//int outusuario2 = 4;

int SW = 12;

bool R_state = 1;

bool G_state = 1;

PN532_SPI pn532spi(SPI, 53);

SNEP nfc(pn532spi);

uint8_t ndefBuf[128];



String readMsg( NdefRecord record ) {

  int payloadLength = record.getPayloadLength();

  byte payload[payloadLength];

  record.getPayload(payload);
```

```
  String payloadAsString = "";
  for (int c = 0; c < payloadLength; c++) {
    payloadAsString += (char)payload[c];
  }
  return payloadAsString.substring(3);
}

void setup() {
  Serial.begin(9600);
  pinMode(outusuario1, OUTPUT);
  //pinMode(outusuario2, OUTPUT);
  Serial.println("NFC Peer to Peer");
}
void loop() {
  if ( digitalRead(SW) == 0) {
    getMsgFromAndroid();
  } else {
    SendMsgToAndroid();
  }
  delay(3000);
}
void SendMsgToAndroid() {
  Serial.println("Send a message to Peer");
  NdefMessage message = NdefMessage();
```

```
  String sendAndroid;
  sendAndroid = "RLED:" + String(digitalRead(outusuario1));
  //sendAndroid += ", GLED:" + String(digitalRead(outusuario2));
  message.addTextRecord(sendAndroid);
  //message.addUriRecord("http://shop.oreilly.com/product/mobile/0636920021193.do");
  //message.addUriRecord("http://arduino.cc");
  //message.addUriRecord("https://github.com/don/NDEF");
  int messageSize = message.getEncodedSize();
  if (messageSize > sizeof(ndefBuf)) {
    Serial.println("ndefBuf is too small");
    while (1) {
    }
  }
  message.encode(ndefBuf);
  if (0 >= nfc.write(ndefBuf, messageSize)) {
    Serial.println("Failed");
  } else {
    Serial.println("Success");
  }
  Serial.println(digitalRead(SW));
}

void getMsgFromAndroid() {
  Serial.println("Waiting for message from Peer");
```

```
int msgSize = nfc.read(ndefBuf, sizeof(ndefBuf));
if (msgSize > 0) {
  NdefMessage msg  = NdefMessage(ndefBuf, msgSize);
  msg.print();
  int recordCount = msg.getRecordCount();
  NdefRecord record = msg.getRecord(0);  //read 1 record
  String myLED = readMsg(record);
  if (myLED == "abcd") {
    digitalWrite(outusuario1, HIGH);
    delay (1000);
    digitalWrite(outusuario1, LOW);
    //R_state = !R_state;
  }
  if (myLED == "1234") {
    digitalWrite(outusuario1, HIGH);
    delay (10);
    digitalWrite(outusuario1, LOW);
    //R_state = !R_state;
  }
} else {
  Serial.println("Failed");
}
Serial.println(digitalRead(SW));
                                                    }
```

Anexo E: Programación Arduino MEGA 2

```
//Se está probando en IDE Arduino 1.6.12

// Receive a NDEF message from a Peer

// Requires SPI. Tested with Seeed Studio NFC Shield v2

//MODULO NFC   ARDUINO MEGA

//   VCC          3VDC

#include "SPI.h"

#include "PN532_SPI.h"

#include "snep.h"

#include "NdefMessage.h"

int outusuario1 = 3;

//int outusuario2 = 4;

int SW = 12;

bool R_state = 1;

bool G_state = 1;

PN532_SPI pn532spi(SPI, 53);

SNEP nfc(pn532spi);

uint8_t ndefBuf[128];


String readMsg( NdefRecord record ) {

  int payloadLength = record.getPayloadLength();

  byte payload[payloadLength];

  record.getPayload(payload);
```

```
  String payloadAsString = "";
  for (int c = 0; c < payloadLength; c++) {
    payloadAsString += (char)payload[c];
  }
  return payloadAsString.substring(3);
}

void setup() {
  Serial.begin(9600);
  pinMode(outusuario1, OUTPUT);
  //pinMode(outusuario2, OUTPUT);
  Serial.println("NFC Peer to Peer");
}
void loop() {
  if ( digitalRead(SW) == 0) {
    getMsgFromAndroid();
  } else {
    SendMsgToAndroid();
  }
  delay(3000);
}
void SendMsgToAndroid() {
  Serial.println("Send a message to Peer");
  NdefMessage message = NdefMessage();
```

```
  String sendAndroid;
  sendAndroid = "RLED:" + String(digitalRead(outusuario1));
  //sendAndroid += ", GLED:" + String(digitalRead(outusuario2));
  message.addTextRecord(sendAndroid);
  //message.addUriRecord("http://shop.oreilly.com/product/mobile/0636920021193.do");
  //message.addUriRecord("http://arduino.cc");
  //message.addUriRecord("https://github.com/don/NDEF");
  int messageSize = message.getEncodedSize();
  if (messageSize > sizeof(ndefBuf)) {
    Serial.println("ndefBuf is too small");
    while (1) {
    }
  }
  message.encode(ndefBuf);
  if (0 >= nfc.write(ndefBuf, messageSize)) {
    Serial.println("Failed");
  } else {
    Serial.println("Success");
  }
  Serial.println(digitalRead(SW));
}
void getMsgFromAndroid() {
  Serial.println("Waiting for message from Peer");
  int msgSize = nfc.read(ndefBuf, sizeof(ndefBuf));
```

```
if (msgSize > 0) {
  NdefMessage msg  = NdefMessage(ndefBuf, msgSize);
  msg.print();
  int recordCount = msg.getRecordCount();
  NdefRecord record = msg.getRecord(0);  //read 1 record
  String myLED = readMsg(record);
  if (myLED == "efgh") {
    digitalWrite(outusuario1, HIGH);
    delay (1000);
    digitalWrite(outusuario1, LOW);
    //R_state = !R_state;
  }
  if (myLED == "1234") {
    digitalWrite(outusuario1, HIGH);
    delay (10);
    digitalWrite(outusuario1, LOW);
    //R_state = !R_state;
  }
} else {
  Serial.println("Failed");
}
Serial.println(digitalRead(SW));
                                                    }
```

Anexo F: Programación Arduino MEGA 3

```
//Se está probando en IDE Arduino 1.6.12
// Receive a NDEF message from a Peer
// Requires SPI. Tested with Seeed Studio NFC Shield v2
//MODULO NFC   ARDUINO MEGA
//   VCC          3VDC
#include "SPI.h"
#include "PN532_SPI.h"
#include "snep.h"
#include "NdefMessage.h"
int outusuario1 = 3;
//int outusuario2 = 4;
int SW = 12;
bool R_state = 1;
bool G_state = 1;
PN532_SPI pn532spi(SPI, 53);
SNEP nfc(pn532spi);
uint8_t ndefBuf[128];

String readMsg( NdefRecord record ) {
  int payloadLength = record.getPayloadLength();
  byte payload[payloadLength];
  record.getPayload(payload);
```

```
  String payloadAsString = "";
  for (int c = 0; c < payloadLength; c++) {
    payloadAsString += (char)payload[c];
  }
  return payloadAsString.substring(3);
}

void setup() {
  Serial.begin(9600);
  pinMode(outusuario1, OUTPUT);
  //pinMode(outusuario2, OUTPUT);
  Serial.println("NFC Peer to Peer");
}
void loop() {
  if ( digitalRead(SW) == 0) {
    getMsgFromAndroid();
  } else {
    SendMsgToAndroid();
  }
  delay(3000);
}
void SendMsgToAndroid() {
  Serial.println("Send a message to Peer");
  NdefMessage message = NdefMessage();
```

```
  String sendAndroid;
  sendAndroid = "RLED:" + String(digitalRead(outusuario1));
  //sendAndroid += ", GLED:" + String(digitalRead(outusuario2));
  message.addTextRecord(sendAndroid);
  //message.addUriRecord("http://shop.oreilly.com/product/mobile/0636920021193.do");
  //message.addUriRecord("http://arduino.cc");
  //message.addUriRecord("https://github.com/don/NDEF");
  int messageSize = message.getEncodedSize();
  if (messageSize > sizeof(ndefBuf)) {
    Serial.println("ndefBuf is too small");
    while (1) {
    }
  }
  message.encode(ndefBuf);
  if (0 >= nfc.write(ndefBuf, messageSize)) {
    Serial.println("Failed");
  } else {
    Serial.println("Success");
  }
  Serial.println(digitalRead(SW));
}
void getMsgFromAndroid() {
  Serial.println("Waiting for message from Peer");
  int msgSize = nfc.read(ndefBuf, sizeof(ndefBuf));
```

```
if (msgSize > 0) {
  NdefMessage msg  = NdefMessage(ndefBuf, msgSize);
  msg.print();
  int recordCount = msg.getRecordCount();
  NdefRecord record = msg.getRecord(0);  //read 1 record
  String myLED = readMsg(record);
  if (myLED == "ijkl") {
    digitalWrite(outusuario1, HIGH);
    delay (1000);
    digitalWrite(outusuario1, LOW);
    //R_state = !R_state;
  }
  if (myLED == "1234") {
    digitalWrite(outusuario1, HIGH);
    delay (10);
    digitalWrite(outusuario1, LOW);
    //R_state = !R_state;
  }
} else {
  Serial.println("Failed");
}
Serial.println(digitalRead(SW));
                                                    }
```

Anexo G: Programación Arduino MEGA 4

```
//Se está probando en IDE Arduino 1.6.12

// Receive a NDEF message from a Peer

// Requires SPI. Tested with Seeed Studio NFC Shield v2

//MODULO NFC   ARDUINO MEGA

//   VCC          3VDC

#include "SPI.h"

#include "PN532_SPI.h"

#include "snep.h"

#include "NdefMessage.h"

int outusuario1 = 3;

//int outusuario2 = 4;

int SW = 12;

bool R_state = 1;

bool G_state = 1;

PN532_SPI pn532spi(SPI, 53);

SNEP nfc(pn532spi);

uint8_t ndefBuf[128];



String readMsg( NdefRecord record ) {
```

```
  int payloadLength = record.getPayloadLength();
  byte payload[payloadLength];
  record.getPayload(payload);
  String payloadAsString = "";
  for (int c = 0; c < payloadLength; c++) {
    payloadAsString += (char)payload[c];
  }
  return payloadAsString.substring(3);
}

void setup() {
  Serial.begin(9600);
  pinMode(outusuario1, OUTPUT);
  //pinMode(outusuario2, OUTPUT);
  Serial.println("NFC Peer to Peer");
}
void loop() {
  if ( digitalRead(SW) == 0) {
    getMsgFromAndroid();
```

```
  } else {
    SendMsgToAndroid();
  }
  delay(3000);
}
void SendMsgToAndroid() {
  Serial.println("Send a message to Peer");
  NdefMessage message = NdefMessage();
  String sendAndroid;
  sendAndroid = "RLED:" + String(digitalRead(outusuario1));
  //sendAndroid += ", GLED:" + String(digitalRead(outusuario2));
  message.addTextRecord(sendAndroid);
  //message.addUriRecord("http://shop.oreilly.com/product/mobile/0636920021193.do");
  //message.addUriRecord("http://arduino.cc");
  //message.addUriRecord("https://github.com/don/NDEF");
  int messageSize = message.getEncodedSize();
  if (messageSize > sizeof(ndefBuf)) {
    Serial.println("ndefBuf is too small");
    while (1) {
```

```
    }
  }
  message.encode(ndefBuf);
  if (0 >= nfc.write(ndefBuf, messageSize)) {
    Serial.println("Failed");
  } else {
    Serial.println("Success");
  }
  Serial.println(digitalRead(SW));
}

void getMsgFromAndroid() {

  Serial.println("Waiting for message from Peer");
  int msgSize = nfc.read(ndefBuf, sizeof(ndefBuf));
  if (msgSize > 0) {
    NdefMessage msg  = NdefMessage(ndefBuf, msgSize);
    msg.print();
    int recordCount = msg.getRecordCount();
```

```
  NdefRecord record = msg.getRecord(0);  //read 1 record
  String myLED = readMsg(record);
  if (myLED == "mnop") {
    digitalWrite(outusuario1, HIGH);
    delay (1000);
    digitalWrite(outusuario1, LOW);
    //R_state = !R_state;
  }
  if (myLED == "1234") {
    digitalWrite(outusuario1, HIGH);
    delay (10);
    digitalWrite(outusuario1, LOW);
    //R_state = !R_state;
  }
} else {
  Serial.println("Failed");
}
Serial.println(digitalRead(SW));
                                                    }
```

Anexo H: Programación Arduino NANO

```
#include <Servo.h>

#include <Wire.h>

#include <EEPROM.h>


Servo myservo;  // Crear el objeto para controlar el servo


int usuario1p1=15;

int usuario2p1=16;

int usuario3p1=18;

int usuario4p1=19;


int usuario1=4;

int usuario2=7;

int usuario3=8;

int usuario4=9;

int dato;

int dato2;

int dato3;

int a;


int puerta2=3;

int puerta3=2;

int puerta4=12;


int entradapuerta2=0;

int entradapuerta3=0;
```

```
int entradapuerta4=0;

int autorizacion1;
int autorizacion2;
int autorizacion3;
int autorizacion4;

char recvChar;



void setup() {
  Serial.begin(115200);
  pinMode(usuario1, INPUT);
  pinMode(usuario2, INPUT);
  pinMode(usuario3, INPUT);
  pinMode(usuario4, INPUT);

  pinMode(usuario1p1, INPUT);
  pinMode(usuario2p1, INPUT);
  pinMode(usuario3p1, INPUT);
  pinMode(usuario4p1, INPUT);

  pinMode(puerta2, INPUT);
  pinMode(puerta3, INPUT);
  pinMode(puerta4, INPUT);
```

```
  Serial.println("Esperando dato...");

autorizacion1 = EEPROM.read(0);
autorizacion2 = EEPROM.read(1);
autorizacion3 = EEPROM.read(2);
autorizacion4 = EEPROM.read(3);

}

void loop() {

  if(Serial.available()){
  recvChar = Serial.read();
   if(recvChar == '1'){
     autorizacion1=1;
     EEPROM.write(0, autorizacion1);
     Serial.println("USUARIO 1 BLOQUEADO");
   }
   if(recvChar == '2'){
     autorizacion1=2;
     EEPROM.write(0, autorizacion1);
     Serial.println("USUARIO 1 DESBLOQUEADO");
   }
   if(recvChar=='3'){
     autorizacion2=3;
```

```
  EEPROM.write(1, autorizacion2);
  Serial.println("USUARIO 2 BLOQUEADO");
}
if(recvChar=='4'){
 autorizacion2=4;
  EEPROM.write(1, autorizacion2);
  Serial.println("USUARIO 2 DESBLOQUEADO");
}
if(recvChar=='5'){
  autorizacion3=5;
  EEPROM.write(2, autorizacion3);
  Serial.println("USUARIO 3 BLOQUEADO");
}
if(recvChar=='6'){
 autorizacion3=6;
  EEPROM.write(2, autorizacion3);
  Serial.println("USUARIO 3 DESBLOQUEADO");
}
if(recvChar=='7'){
 autorizacion4=7;
  EEPROM.write(3, autorizacion4);
  Serial.println("USUARIO 4 BLOQUEADO");
}
if(recvChar=='8'){
 autorizacion4=8;
  EEPROM.write(3, autorizacion4);
  Serial.println("USUARIO 4 DESBLOQUEADO");
```

```
   }
  }


  dato = digitalRead(usuario1);
if ((dato == HIGH) && (autorizacion1 == 2)){
    myservo.attach(5);  // Enlazar el servo en el pin 5 al objeto servo "Digital 5"
    myservo.writeMicroseconds(2000);
    //Serial.print("# Usuario 1");
    //delay(2000);
    //Serial.println(" ");
    dato = digitalRead(usuario1p1);
    if(dato==HIGH){
    Serial.print("#1 registra entrada");
    delay(2000);
    Serial.println(" ");
    }
    if(dato==LOW){
    Serial.print("#1 registra salida");
    delay(2000);
    Serial.println(" ");
    }
    myservo.attach(5);  // enlazar el servo en pin 5 al objeto servo "Digital 5"
    myservo.writeMicroseconds(1000);
    delay(2000);
}
```

```
  dato = digitalRead(usuario2);
if ((dato == HIGH) && (autorizacion2 == 4)){
    myservo.attach(5);  // enlazar el servo en pin 5 al objeto servo "Digital 5"
    myservo.writeMicroseconds(2000);
    dato = digitalRead(usuario2p1);
    if(dato==HIGH){
    Serial.print("#2 registra entrada");
    delay(2000);
    Serial.println(" ");
    }
    if(dato==LOW){
    Serial.print("#2 registra salida");
    delay(2000);
    Serial.println(" ");
    }
    myservo.attach(5);  // enlazar el servo en pin 5 al objeto servo "Digital 5"
    myservo.writeMicroseconds(1000);
    delay(2000);
}

dato = digitalRead(usuario3);
if ((dato == HIGH) && (autorizacion3 == 6)){
    myservo.attach(5);  // enlazar el servo en pin 5 al objeto servo "Digital 5"
    myservo.writeMicroseconds(2000);
     dato = digitalRead(usuario3p1);
    if(dato==HIGH){
```

```
    Serial.print("#3 registra entrada");
    delay(2000);
    Serial.println(" ");
    }
    if(dato==LOW){
    Serial.print("#3 registra salida");
    delay(2000);
    Serial.println(" ");
    }
    myservo.attach(5);  // enlazar el servo en pin 5 al objeto servo "Digital 5"
    myservo.writeMicroseconds(1000);
    delay(2000);
}

dato = digitalRead(usuario4);
if ((dato == HIGH) && (autorizacion4 == 8)){
    myservo.attach(5);  // enlazar el servo en pin 5 al objeto servo "Digital 5"
    myservo.writeMicroseconds(2000);
     dato = digitalRead(usuario4p1);
    if(dato==HIGH){
    Serial.print("#4 registra entrada");
    delay(2000);
    Serial.println(" ");
    }
    if(dato==LOW){
    Serial.print("#4 registra salida");
    delay(2000);
```

```
    Serial.println(" ");
    }
    myservo.attach(5);  // enlazar el servo en pin 5 al objeto servo "Digital 5"
    myservo.writeMicroseconds(1000);
    delay(2000);
}



dato = digitalRead(puerta2);
  if(dato==HIGH){
       Serial.println("Detecto senal");
       delay(100);
       dato = digitalRead(puerta2);
       if(dato==HIGH){
        //Serial.println("Detecto nivel alto");
        entradapuerta2=1;
        a=1;
       }
        dato = digitalRead(puerta2);
        if(dato==LOW){
        //Serial.println("Detecto nivel bajo");
        entradapuerta2=2;
        a=1;
        }
  }
```

```
if(entradapuerta2==1){
 dato2 = digitalRead(usuario1p1);
 if(dato2==LOW){
  if(autorizacion1==2 && a==1 ){

    myservo.attach(6);  // enlazar el servo en pin 5 al objeto servo "Digital 5"
    myservo.writeMicroseconds(2000);
    Serial.print("#1 registra apertura en puerta 2");
    delay(2000);
    Serial.println(" ");
    entradapuerta2=0;
    a=0;
    myservo.attach(6);  // enlazar el servo en pin 5 al objeto servo "Digital 5"
    myservo.writeMicroseconds(1000);
 }
 }
 delay(2000);
 entradapuerta2=0;
 a=0;
}
 if(entradapuerta2==2){
 dato3 = digitalRead(usuario2p1);
 if(dato3==LOW){
 if((autorizacion2==4)&&(a==1)){
    myservo.attach(6);  // enlazar el servo en pin 5 al objeto servo "Digital 5"
    myservo.writeMicroseconds(2000);
    Serial.print("#2 registra apertura en puerta 2");
```

```
    delay(2000);
    Serial.println(" ");
    a=0;
    entradapuerta2=0;
    myservo.attach(6);  // enlazar el servo en pin 5 al objeto servo "Digital 5"
    myservo.writeMicroseconds(1000);
 }
 }
 delay(2000);
 entradapuerta2=0;
 a=0;
}

dato = digitalRead(puerta3);
if(dato==HIGH){
     Serial.println("Detecto senal");
     delay(100);
     dato = digitalRead(puerta3);
     if(dato==HIGH){
      //Serial.println("Detecto nivel alto");
      entradapuerta3=1;
      a=1;
     }
      dato = digitalRead(puerta3);
      if(dato==LOW){
      //Serial.println("Detecto nivel bajo");
      entradapuerta3=3;
```

```
      a=1;
      }
 }

if(entradapuerta3==1){
 dato2 = digitalRead(usuario1p1);
 if(dato2==LOW){
  if(autorizacion1==2 && a==1 ){

    myservo.attach(10);  // enlazar el servo en pin 5 al objeto servo "Digital 5"
    myservo.writeMicroseconds(2000);
    Serial.print("#1 registra apertura en puerta 3");
    delay(2000);
    Serial.println(" ");
    entradapuerta3=0;
    a=0;
    myservo.attach(10);  // attaches the servo on pin 5 to the servo object "Digital 5"
    myservo.writeMicroseconds(1000);
 }
 }
 delay(2000);
 entradapuerta3=0;
 a=0;
}
 if(entradapuerta3==3){
 dato3 = digitalRead(usuario3p1);
 if(dato3==LOW){
```

```
if((autorizacion3==6)&&(a==1)){
    myservo.attach(10);  // enlazar el servo en pin 5 al objeto servo "Digital 5"
    myservo.writeMicroseconds(2000);
    Serial.print("#3 registra apertura en puerta 3");
    delay(2000);
    Serial.println(" ");
    a=0;
    entradapuerta3=0;
    myservo.attach(10);  //
    myservo.writeMicroseconds(1000);
 }
 }
 delay(2000);
 entradapuerta3=0;
 a=0;
}

dato = digitalRead(puerta4);
if(dato==HIGH){
     Serial.println("Detecto senal");
     delay(100);
     dato = digitalRead(puerta4);
     if(dato==HIGH){
      //Serial.println("Detecto nivel alto");
      entradapuerta4=1;
      a=1;
     }
```

```
      dato = digitalRead(puerta4);
      if(dato==LOW){
      //Serial.println("Detecto nivel bajo");
      entradapuerta4=4;
      a=1;
      }
 }

if(entradapuerta4==1){
 dato2 = digitalRead(usuario1p1);
 if(dato2==LOW){
  if(autorizacion1==2 && a==1 ){

    myservo.attach(11);  // enlazar el servo en pin 5 al objeto servo "Digital 5"
    myservo.writeMicroseconds(2000);
    Serial.print("#1 registra apertura en puerta 4");
    delay(2000);
    Serial.println(" ");
    entradapuerta4=0;
    a=0;
    myservo.attach(11);  // enlazar el servo en pin 5 al objeto servo "Digital 5"
    myservo.writeMicroseconds(1000);
 }
 }
 delay(2000);
 entradapuerta4=0;
 a=0;
```

```
 }
  if(entradapuerta4==4){
  dato3 = digitalRead(usuario4p1);
  if(dato3==LOW){
  if((autorizacion4==8)&&(a==1)){
     myservo.attach(11);  // enlazar el servo en pin 5 al objeto servo "Digital 5"
     myservo.writeMicroseconds(2000);
     Serial.print("#4 registra apertura en puerta 4");
     delay(2000);
     Serial.println(" ");
     a=0;
     entradapuerta4=0;
     myservo.attach(11);  //
     myservo.writeMicroseconds(1000);
   }
   }
   delay(2000);
   entradapuerta4=0;
   a=0;
  }

}//llave void loop
```