FACULTAD DE INGENIERIA Y CIENCIAS APLICADAS

IMPLEMENTACIÓN DE UN SISTEMA ELECTRÓNICO DE SEGURIDAD PARA TARJETAS BANCARIAS CONTACTLESS

AUTOR

Andrés Rodrigo Pozo León
Diego Steven Pérez Real

AÑO

2019

FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

IMPLEMENTACIÓN DE UN SISTEMA ELECTRÓNICO DE SEGURIDAD PARA TARJETAS BANCARIAS CONTACTLESS

Profesor guía

Mgs. Iván Patricio Ortiz Garcés

Autor

Diego Steven Pérez Real

Andrés Rodrigo Pozo León

Año

2019

DECLARACIÓN DE PROFESOR GUÍA

"Declaro haber dirigido el trabajo, Implementación de un Sistema Electrónico de Seguridad para Tarjetas Bancarias Contactless, a través de reuniones periódicas con el estudiante Diego Steven Pérez Real y Andrés Rodrigo Pozo León, en el semestre 201920, orientando sus conocimientos y competencias para un eficiente desarrollo del tema escogido y dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación".

_____

Iván Patricio Ortiz Garcés

Magister en Redes de Comunicaciones

C.I.: 0602356776

DECLARACIÓN DE PROFESOR CORRECTOR

"Declaro haber revisado este trabajo, Implementación de un Sistema Electrónico de Seguridad para Tarjetas Bancarias Contactless, de Diego Steven Pérez Real y Andrés Rodrigo Pozo León, en el semestre 201920, dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación".

_____

José Julio Freire Cabrera

Magíster en Gerencia Empresarial

C.I. 1709731457

DECLARACIÓN DE AUTORÍA DEL ESTUDIANTE

“Declaramos que este trabajo es original, de nuestra autoría, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autores vigentes”

_____	_____
Diego Steven Pérez Real	Andrés Rodrigo Pozo León
C.I.: 1717920001	C.I.: 1709539967

AGRADECIMIENTOS

Agradezco a Dios, mi familia y a las personas que se encuentran en mi vida por brindarme su apoyo incondicional en cada uno de mis pasos.

Diego Pérez

AGRADECIMIENTOS

Expreso mi gratitud a Dios quien con su bendición llena siempre mi vida y a todas las personas que conozco por estar siempre allí presentes en todo momento que he necesitado.

Andrés Pozo

DEDICATORIA

Dedico esta tesis al pilar fundamental de mi vida, mi Madre Rosa Real, a mis familiares y amigos que me impulsaron todos los días y depositaron su esperanza en mí.

Diego Pérez

DEDICATORIA

Esta tesis está dedicado a mi madre, quien sin su ayuda no hubiese tenido el coraje de seguir adelante en todo momento, de igual forma a mi padre, quien me enseñó que el mejor conocimiento que se puede tener es el que se aprende por sí mismo.

Andrés Pozo

RESUMEN

Nuevas tecnologías se van incorporando día a día, RFID y NFC permiten a los usuarios realizar cualquier tipo de transacción de manera más rápida y ágil, sin necesidad de contacto físico con la tarjeta o firmas.

RFID y NFC utilizan señales de onda corta de manera pasiva, mismas que pueden ser obtenidas de manera remota con diferentes técnicas.

La arquitectura sobre la que está basada NFC posee tres diferentes modelos de operación, las cuales son: punto a punto, lectura-escritura y emulación de tarjeta.

Para el desarrollo de la presente tesis se pudo demostrar que hay diferentes formas de acceso a los datos a través de equipos especializados, y que con un poco de conocimiento permiten acceder a los mismos.

El equipo desarrollado permite tener de manera más segura la información en un “firewall” haciendo que sea decisión de cada uno cuando activar o desactivar el sistema implementado.

Este prototipo podría ser miniaturizado en un futuro para así servir de alerta en caso de hurto de tarjetas y además de tener una detección temprana de fraude.

De las pruebas realizadas con los equipos de escucha se logra determinar que no todos los equipos con tecnología NFC tienen un cifrado en el tráfico haciendo que dichas transferencias sean en texto plano, haciendo de estas inseguras para el usuario y que la información que es transmitida pueda ser interceptada.

ABSTRACT

New technologies are incorporated day by day, RFID and NFC allow users to perform any type of transaction more quickly and quickly, without the need for physical contact with the card or signatures.

RFID and NFC use passive shortwave signals, which can be obtained remotely with different techniques.

The architecture on which NFC is based has three different operation models, which are: point to point, read-write and card emulation.

For the development of this thesis it was possible to demonstrate that there are different forms of access to data through specialized teams, and that with a little knowledge they allow access to them.

The developed equipment allows to have more secure information in a "firewall" making it the decision of each one when to activate or deactivate the implemented system.

This prototype could be miniaturized in the future to serve as an alert in case of card theft and in addition to having an early fraud detection.

From the tests carried out with the listening equipment, it is possible to determine that not all the equipment with NFC technology has an encryption in the traffic making said transfers are in plain text, making these insecure for the user and that the information that is transmitted can be intercepted

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

1. Introducción

En este capítulo se dará a conocer todas las generalidades de la tesis, el problema que se intenta dar solución, motivos para el desarrollo de esta, su alcance, objetivos y una visión general a la problemática planteada.

1.1 Antecedentes

Las entidades financieras utilizan métodos de financiamiento económico con el fin de brindar comodidad a sus clientes obteniendo una comisión monetaria, es por esto por lo que las tarjetas bancarias son el medio mundial para obtener dinero de manera casi inmediata financiando o debitando el mismo. “A mediados del año 1914 las tarjetas eran utilizadas en ciertos establecimientos o utilizaban este método como una membresía a través de hojas impresas, en 1954 donde Frank McNamara implementa un método de financiamiento utilizando una sola tarjeta para varios establecimientos. Visa y MasterCard a la par, utilizando tarjetas plásticas que solo poseen números únicos y nombre del titular.” (BBVA, Historia de las tarjetas de crédito).

La primera generación de tarjetas o tarjetas de banda magnéticas utilizan un código único para identificarlas. En las tarjetas de banda magnética puede haber tres pistas o tracks, conocidas como 1, 2 y 3. La Pista 3 es la menos utilizada, y a veces no se encuentra ni siquiera físicamente presente en la tarjeta al emplearse una banda magnética más estrecha.

Los lectores de los TPV (terminales punto de venta) o POS (Point of sale), siempre leen las pistas 1 o 2: la información mínima necesaria para realizar una transacción se encuentra en ambas pistas. La pista 1 tiene una mayor densidad de bits, es la única que puede contener caracteres alfanuméricos, y por tanto es la única que puede contener el nombre del portador de la tarjeta.

La segunda generación de tarjetas son las tarjetas inteligentes o también llamadas Smart card, o TCI (tarjeta con circuito integrado), es una tarjeta del tamaño con circuitos integrados, que permite la ejecución de cierta lógica programada.

Aunque existe un diverso rango de aplicaciones, hay dos categorías principales de TCI. Las tarjetas de memoria solo contienen componentes de memoria no volátil y cierta lógica de seguridad. Las tarjetas micro procesadoras contienen memoria y microprocesadores.

Las aplicaciones de las tarjetas inteligentes incluyen su uso como tarjeta de crédito, SIM (Servicios integrales para la movilidad) para telefonía móvil, tarjetas de autorización para televisión por pago, identificación de alta seguridad, tarjetas de control de acceso.

“La actual vulnerabilidad del popular método de pago con tarjetas de débito y crédito que poseen un chip y un número pin ha sido expuesta por investigadores de la Universidad de Cambridge.

Los hallazgos, presentados en una conferencia de criptografía en Leuven, Bélgica, muestran que las tarjetas aún pueden ser clonadas, a pesar de las promesas de los bancos que aseguran que los dispositivos están protegidos de cualquier amenaza. Según los expertos, la razón es la escasa implementación de métodos de criptografía.” (finanzaspersonales, s.f.).

La tercera generación son las tarjetas contactless el sistema funciona gracias al NFC (near field communications), una tecnología inalámbrica de corto alcance que permite la transmisión instantánea de datos entre dispositivos que se encuentren a unos cuantos centímetros de distancia. Su tecnología deriva de las

etiquetas RFID (identificación por radiofrecuencia), que son las que llevan los datos de transporte a algunos sistemas de seguridad para apertura de puertas.

Al incorporar esta tecnología en las tarjetas de crédito y los terminales de venta se crea un canal de comunicación que sirve para enviar y recibir la información relativa a la autorización del pago de forma segura. Así, gracias al NFC es posible pagar de forma prácticamente instantánea al situar la tarjeta sobre el TPV, sin necesidad de introducirla o pasarla por el lector de banda.

Según un informe de Visa Europa, en 2016 ya había 3,2 millones de terminales de venta y 165 millones de tarjetas compatibles con tecnología 'contactless' activos en Europa. Para poner las cifras en perspectiva, el informe de Visa destaca que desde 2013 a 2016, la proporción de pagos realizados por esta vía han pasado a ser uno de cada 60, a uno de cada cinco. (BBVA, s.f.)

1.2 **Alcance**

Se estudiará diferentes soluciones tecnológicas para la identificación por radiofrecuencia, rango en que operan estas tecnologías, sus vulnerabilidades y sus fortalezas.

Se aplicarán técnicas de obtención de información en tarjetas contactless con tecnología NFC y RFID para poder medir el nivel de seguridad que se tiene en estas tecnologías.

Se desplegará un prototipo en un dispositivo electrónico que permitirá bloquear e informar a través de canales electrónicos al usuario en su smartphone y a la entidad bancaria sobre ataques que se generen en un tiempo adecuado.

El procedimiento de pruebas contempla en primera instancia la lectura y escritura de la información por RFID y NFC desde diferentes equipos, se verificará la capacidad de bloqueo y alerta temprana con el equipo Raspberry.

1.3 **Justificación**

Nuevas tecnologías se van incorporando día a día, RFID y NFC permite a los usuarios realizar cualquier tipo de transacción de manera más rápida y ágil, sin necesidad de contacto físico con la tarjeta o firmas.

RFID y NFC utilizan señales de onda corta de manera pasiva, mismas que pueden ser obtenidas de manera remota con diferentes técnicas.

Es por esta razón que bloquear y alertar este tipo de ataques es muy necesario para los usuarios y entidades bancarias, ya sea con una alerta, mensaje o correo para acciones inmediatas.

Con el desarrollo de este prototipo se puede aportar con nuevas técnicas de alerta temprana en caso de intento de robo de información con la tecnología contactless.

1.4 **Objetivo general**

Implementación de un prototipo electrónico de seguridad para las tarjetas contactless que permita determinar a través de medios de comunicación electrónico cuándo pretende ser clonada una tarjeta.

1.5 Objetivos específicos

- Estudiar las diferentes soluciones tecnológicas RFID y NFC que existen en el mercado.
- Modelar un dispositivo electrónico para el envió de alertas vía Wireless al usuario en un entorno de aplicación móvil Android para una posterior alerta a la entidad bancaria.
- Enviar una alerta a la entidad bancaria a través de la aplicación móvil.
- Proteger las tarjetas contactless con la programación en el dispositivo electrónico.

2. Marco teórico

En el capítulo a continuación se presentará una descripción de la tecnología NFC, además de los equipos que se van a utilizar, sus características técnicas y su modalidad de trabajo.

2.1 NFC (Near Field Communication)

2.1.1 Tecnología NFC

La tecnología de comunicación de campo cercano funciona bajo el estándar ISO 14443 (RFID), generalmente funciona a una distancia de 10 cm o menos entre dos dispositivos de comunicación que utilizan la misma tecnología.

En una configuración básica de NFC, existe un elemento activo que se encuentra alimentado de energía eléctrica y el elemento pasivo que no tiene una fuente propia de alimentación.

El elemento activo genera un campo de radio frecuencia (RF) donde trasfiere al elemento pasivo para su lectura y escritura de datos. (Sabetti, Qian, & Bush, 2019)

Las transferencias de datos manejan velocidades 106, 212, 424 o 848 Kbit/s, esto se debe a que no está orientado a la trasferencia masiva de datos, su uso apropiado es para comunicaciones pequeñas entre dispositivos.

La banda de frecuencia utilizada para su funcionamiento es de 13,56 MHz, esta frecuencia pertenece al conjunto de bandas de radio ISM (Industrial, Científica y Medica). Para el uso de esta banda no es necesario una licencia, pero esta no debe producir interferencias entre dispositivos. (INTECO,2013)

2.1.2 **Diseño NFC**

Existen diferentes modos de comunicación:

- Pasivo, el dispositivo que no genera su campo de comunicación es energizado por el campo de radiofrecuencia de otro dispositivo.
- Activo, en donde los dos dispositivos generan campos electromagnéticos para comunicarse. (Albiñana, 2016, p.15)

Su conjugación puede ser varias como se puede observar en la figura 1:

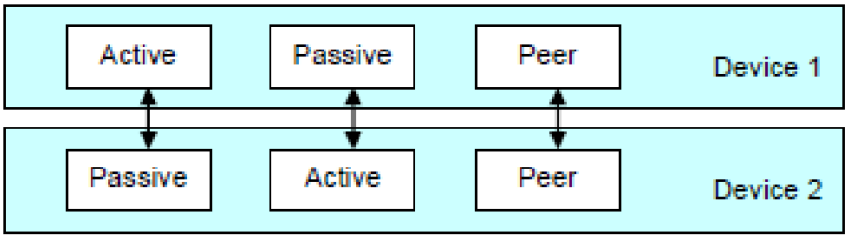

Figura 1. Modo de operación NFC

Tomado de (Minihold, 2011).

La tecnología NFC es apto y eficiente en los modelos de operación como tarjeta inteligente, modo lectura/escritura y comunicación punto a punto.

2.1.2.1 **Tarjeta Inteligente (Pasiva)**

Las tarjetas inteligentes son denominadas de esta manera por sus propiedades de seguridad que son implementadas en las entidades emisoras, cada tarjeta cuenta con varias capas para que estos sean un elemento seguro como se puede observar en la figura 2.

Figura 2. Capas tarjeta inteligente modo NFC
Tomado de (NFC-Forum, 2015).

La capa contactless y los campos RF son el transporte para el standard de la tarjeta inteligente (ISO/IEC 7816). Para las aplicaciones de pago NFC este modo se basa en el estándar EMV (Europay MasterCard VISA) y las especificaciones de las tarjetas PIN.

El componente que genera la seguridad es un chip que cuenta con un procesador, un cripto-procesador, que incluye una memoria EPROM; el elemento seguro recibe mensaje y envía respuestas a través de sus interfaces de entrada/salida.

2.1.3 **Componentes NFC**

Los dispositivos NFC pasivos pueden contener estructuras de información. Estas son pequeñas espirales de metal a las que se les complementa con dispositivos de memoria y comunicación.

Su diseño plano es estético e ideal para formatos como tarjetas de visita, adhesivos, llaveros o pulseras.

Su funcionamiento es similar al código de barras y códigos QR, pues un dispositivo NFC activo (POSNFC, Smartphone, etc) actúa como lector. El campo de radiofrecuencia del lector activa el elemento NFC pasivo y hace que transmita la información almacenada.

Estos datos de transferencia dependen de otros factores como el tipo de elemento NFC, ya que existen varias clases en función de su memoria, los modos de interacción y su tasa de transferencia de datos. (INTECO,2013).

Estos tipos fueron estandarizados por el NFC Fórum y se pueden observar en la tabla 1 presentada a continuación:

Tabla 1.

Tipos de etiquetas NFC

Tipo	Estándar	Modo	Memoria	Velocidad
Tipo 1	ISO14443 Tipo A	Sólo lectura Lectura/Escritura	96 bytes Ampliable a 2 kbytes	106 kbit/s

Tipo 2	ISO14443 Tipo A	Sólo lectura Lectura/Escritura	48 bytes Ampliable a 2 kbytes	106 kbit/s
Tipo 3	Sony FeliCa	Sólo lectura	2 kbytes	212 kbit/s
Tipo 4	ISO14443 Tipo A y B	Sólo lectura Lectura/Escritura	32 kbytes	106 kbit/s 424 kbit/s

2.1.3.1 **Protocolo de Control de Enlace Lógico (LLCP)**

El protocolo LLCP (LogicalLink Control Protocol) permite comunicaciones multiplexadas entre dos dispositivos NFC, donde uno de los dispositivos envía información en cualquier instante de manera asíncrona.

La parte final de comunicación se denomina SAP (Service Access Point) o puntos de acceso al servicio y son guiados mediante un identificador numérico de 6 bits.

El tamaño de direcciones de los SAP se clasifica en 3 partes: direcciones entre 0 y 15 identificados como un servicio conocido, entre 16 y 31 identificados como un servicio de registro del servicio local y entre 16 y 63 son utilizadas como dirección de origen del cliente o se conectan a servicios de pares. (HAO, CHANG, 2019)

2.1.3.2 **Mensaje NDEF (NFC Data Exchange Format)**

Un mensaje NDEF está compuesto por una serie de múltiples registros, mismos que contienen cabecera y cuerpo. La cabecera contiene información sobre el tipo y la longitud de los datos almacenados en el cuerpo. Estos se encuentran en el payload. (Reddy Arunan, Won, 2018)

Esta información tiene diferentes clases, por ejemplo, URIs o cualquiera de los tipos de datos específicos para NFC identificados por las definiciones de tipos de registros o RTD (Record Type Definition).

Los RTD son formatos optimizados para la transmisión entre dispositivos NFC. (INTECO,2013)

NDEF es capaz de encapsular una o más cargas útiles (payload) de diferentes tipos y tamaños dentro de la estructura del mensaje NDEF. El payload está constituida por:

Longitud (PAYLOAD_LENGTH): Este parámetro indica el número de octetos de carga útil (payload). El payload es encapsulada en un registro que se encuentra dentro de los primeros 8 octetos.

Para registros que contienen poca carga útil este parámetro es de un octeto establecido por un bit de bandera SR12 en 1, mientras que para registros normales este parámetro es de 4.

Tipo de carga útil (payload): Se encuentra el tipo de datos contenidos en el payload, estos pueden ser URIs, MIME o específicos NFC (NFC-specific).

Identificador de Payload: este identificador es adicional ya que viene en forma de URI absoluta o relativa, esto permitiendo que el payload de tipo URI enlace tecnologías con otros payload. (Veloz, 2010, pp.16-17)

Los tipos conocidos por NFC contienen un identificador NID el cual nadie puede utilizar.

En la figura 3 se puede observar la estructura de la carga que contiene la información y grabaciones de la estructura NDEF.

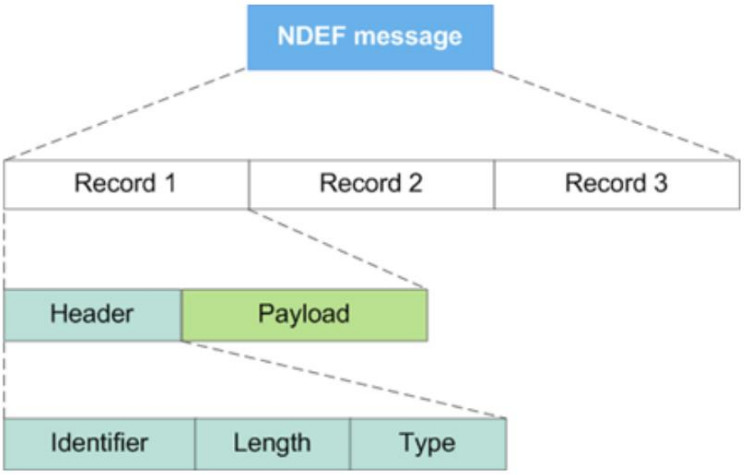

Figura 3. Estructura de carga de los mensajes NDEF

Tomado de (Lesas & Miranda, 2007)

Como se puede observar en la figura 4, cada registro contiene un encabezado y una carga útil.

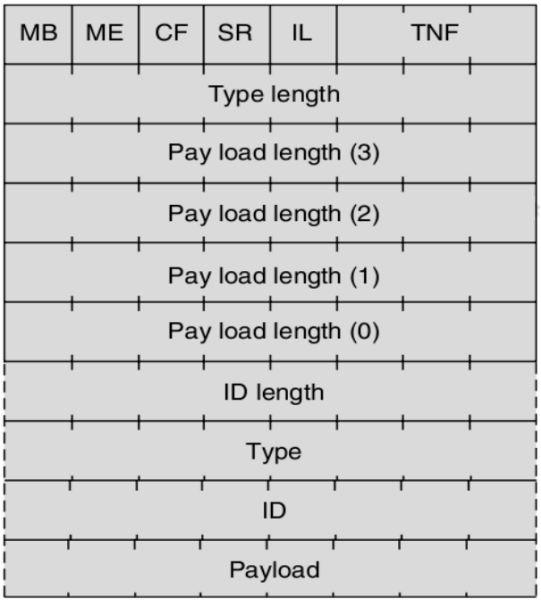

Figura 4. Estructura lógica de los mensajes NDEF

Tomado de (Lesas & Miranda, 2007)

En la figura 4, se encuentra estructurado el registro NDEF, el primer byte es usado como cabecera de la trama del registro en donde:

- MB (Message Begin): es el inicio del mensaje tomando el valor de 1 si el mensaje comienza, de manera contraria tomará el valor de 0.

- ME (MessageEnd): es el final del mensaje tomando el valor de 1 si el mensaje termina, de lo contrario tomará el valor de 0.

- CF: este bit indica si existe una carga útil partida en trozos.

- SR: indica un registro corto de 7 bytes en lugar de 10 bytes.

- IL: indica si los bytes de longitud de ID e ID deben ser leídos.

- TNF: formato de tipo de nombre en 3 bits el cual puede ser:

  - 0x00: registro vacío
  - 0x01: tipo NFC bien conocido definido por el Foro NFC
  - 0x02: tipo MIME (texto, multimedia, imagen, etc.)
  - 0x03: URI
  - 0x04: externo
  - 0x05: tipo desconocido
  - 0x06: sin alterar (para registros en trozos)
  - 0x07: reservado para uso futuro 14

- TypeLength (Longitud de Tipo): Longitud del campo PTYPE en 8 bytes.

- ID Length (Longitud de ID): tamaño del ID de la carga útil en 8 bytes.

- PayloadLength (Longitud de carga útil): especifica la longitud de la carga útil que está determinado por el campo SR.

- Type (Tipo): tipo de registro en hexadecimal.
- ID: tipo de ID o prefijo hexadecimal los cuales pueden ser:

  - 0x00: sin prefijo
  - 0x01: http://www
  - 0x02: https://www
  - 0x03: http://
  - 0x04: https://
  - 0x05: tel:
  - 0x06: mailto:
  - 0x1D: file://
  - 0x24…0xFF: Reservado para uso futuro.

- Payload (Carga útil): contiene la carga útil de tamaño determinado por el campo Payload Length.(Lesas & Miranda, 2007, p.33-34)

2.2 **Elementos de seguridad electrónica y fraude**

Existen varios métodos para mitigar las vulnerabilidades, sin embargo, ningún sistema es infalible, por lo que a nivel de seguridad electrónica se brinda una alta complejidad y dificulta el fraude por entes malintencionados.

2.2.1 **Lectura y sobre escritura NFC**

2.2.1.1 **uFR series**

Este dispositivo RFID está destinado principalmente a empresas y personas en desarrollo (profesionales y aficionados) para futuras aplicaciones y desarrollo de soluciones.

- Hardware:

La comunicación RF de este lector RFID de NFC se basa en el lector integrado y potente IC de NXP para comunicación sin contacto a 13.56 MHz, que admite todas las capas de protocolo de los estándares de comunicación ISO / IEC 14443 A, ISO / IEC 14443 B e ISO / IEC 18092.

Este dispositivo RFID también tiene incorporado el algoritmo Crypto1® y un almacenamiento de memoria interno no volátil seguro para una mejor seguridad. La comunicación con el host se proporciona mediante la interfaz compatible con USB 2.0 Full Speed.

Las velocidades de transferencia de datos están limitadas de 9600 baudios a 1 MBaud, por razones de integridad y seguridad. Este Lector RFID NFC también puede actuar como un dispositivo de interfaz en serie utilizando los controladores del puerto COM virtual (VCP) de FTDI.

En la tabla 2 se puede observar las características más importantes del equipo.

Tabla 2.

Características Hardware uFR

Característica	**Descripción**
Frecuencia de operación	13.56 MHz
Rango de lectura	Según la geometría de la antena y la configuración del lector, la distancia de operación es de 2-8 cm (0,78 "-3,15")

Velocidad de lectura / escritura	Hasta 424 kbps.
Corriente de suministro	150mA (en funcionamiento)
Tensión de alimentación	5V
Tarjetas y etiquetas soportadas	Tipo A (Hardware + Software), Tipo B (Hardware) Tarjetas compatibles: MIFAREClassic ® (4 bytes y 7 bytes), MIFAREUltralight ® / MIFAREUltralight ® C (Modo no seguro), NTAG203F
Conexión y alimentación eléctrica.	Puerto USB
Software	El software µFR contiene la biblioteca para trabajar con Java, Applet de Java, JavaScript, Lazarus, Delphi, C + + Builder, Microsoft® Visual C ++ .NET, Microsoft® Visual C #, Microsoft® Visual Basic .NET
Sistemas operativos compatibles	Microsoft® Windows ™, Linux®, OS X, Android
Interfaz de tarjeta inteligente	ISO14443A / B
El equipaje relacionado	Cable USB, descarga de software gratis

Equipamiento opcional	Zumbador, reloj de tiempo real (RTC), EEPROM
Peso	0.21kg nett
Dimensiones	16 cm x 12,6 cm x 3,2 cm (6,3 "x 5" x 1,2 ")
Accesorios	Tarjetas, llaveros o etiquetas

- Firmware:

El firmware tiene muchas funciones complejas integradas que se pueden llamar a través de las bibliotecas proporcionadas en la API.

El punto principal sobre el firmware es que todas las funciones del firmware también pueden llamarse a través del protocolo de comunicación.

Este hecho traslada el uso de este dispositivo a otra dimensión, lo que significa que este lector RFID NFC se puede usar en casi cualquier plataforma que tenga interfaz en serie, simplemente mediante el uso del protocolo de comunicación.

Por lo tanto, se puede utilizar libremente en PC, tableta, teléfono inteligente, sistemas integrados, Raspberry Pi, Beagle Board, placas MIPS, PLC y otras plataformas.

En la figura 5 se puede observar el dispositivo que puede ser utilizado en posibles ataques de duplicación de datos NFC y RFID.

Figura 5. uFR para clonación de tarjetas NFC

Tomado de (DLOGIC, 2019)

2.2.1.2 **ETEKJOYhanahela**

ETEKJOY es un dispositivo con fines investigativos y de uso simple, trabaja en varias bandas de comunicación donde permite leer y escribir en nuevos dispositivos RFID y NFC.

Es compatible 10 frecuencias: 125 Khz, 250 Khz, 375 Khz, 500 Khz, 625 Khz, 750 Khz, 875 Khz, 1000 Khz, 13,56 Mhz (ISO1443A/B), 125 Khz Prox.

Lecturas compatibles: EM4100/EM4200, MifareClassic, UIDcards, Ultralight, Ntag203, 1386/1326/1346; Soporte de escritura: 1386/1326/1346, T5577, EM4305, MifareUID.

En la figura 6 se puede observar el dispositivo portátil que puede clonar tecnología RFID y NFC de diferente rango de frecuencia.

Figura 6. ETEKJOY para clonación de tarjetas NFC y RFID

Tomado de (ETEKJOY, 2019)

2.2.2 **Raspberry pi y módulo NFC**

Es un ordenador reducido de placa simple, en el 2009 nace la fundación Raspberry PI impulsado por el desarrollo informático.

2.2.2.1 **Software**

El software es totalmente libre, su sistema operativo tiene gran variedad de entornos sin embargo su sistema madre es Raspbian basado en debían de la distribución Linux.

2.2.2.2 **Hardware**

Raspberry pi posee diferentes características técnicas como:

- Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
- 1 GB de RAM
- BCM43438 LAN inalámbrica y Bluetooth Low Energy (BLE) a bordo
- 100 Base Ethernet
- GPIO extendido de 40 pines
- 4 puertos USB 2
- Salida de 4 polos estéreo y puerto de video compuesto
- HDMI de tamaño completo
- Puerto de cámara CSI para conectar una cámara Raspberry Pi
- Puerto de visualización DSI para conectar una pantalla táctil Raspberry Pi
- Puerto microSD para cargar su sistema operativo y almacenar datos
- Fuente de alimentación Micro USB conmutada actualizada de hasta 2.5 A

2.3 **Tarjetas bancarias contactless**

Diferentes estadísticas estiman más de 500 millones de usuarios alrededor del mundo usando NFC como método de pago primario en 2019. (Juniper, 2018)

En la figura 7 se puede observar a nivel interno como se encuentra estructurada una tarjeta contacless.

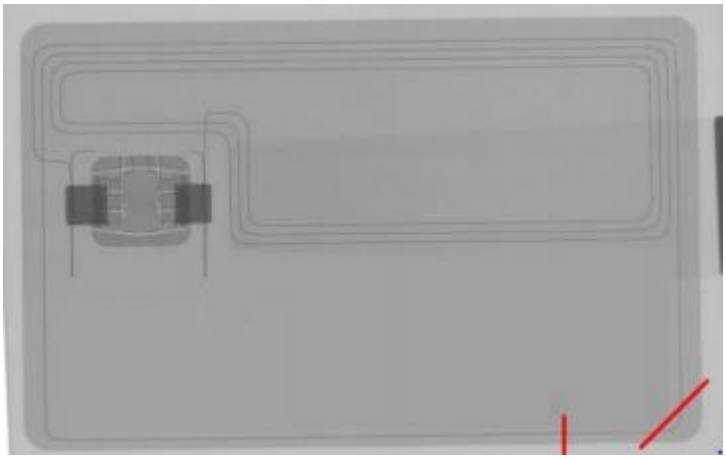

Figura 7. Estructura interna de una tarjeta contactless
Tomado de (CN-CERT, 2016)

El flujo de pago con tarjetas contactless es sencilla, únicamente acercando la tarjeta a un POS se produce el cobro, dependiendo de la entidad bancaria es necesario introducir el pin de la tarjeta.

La tecnología NFC sufre diferentes problemas de seguridad, mismos que están identificados como escucha secreta, alteración de transmisión o ataques de retrasmisión. (CCN-CERT, 2016)

2.3.1 **Funcionamiento de tarjetas bancarias contactless**

Las tarjetas bancarias necesitan un proveedor como visa, MasterCard etc., una entidad bancaria y de manera opcional un switch transaccional.

El funcionamiento de las tarjetas contactless emitidas por entidades financieras va de la siguiente forma como se puede ver en la figura 8.

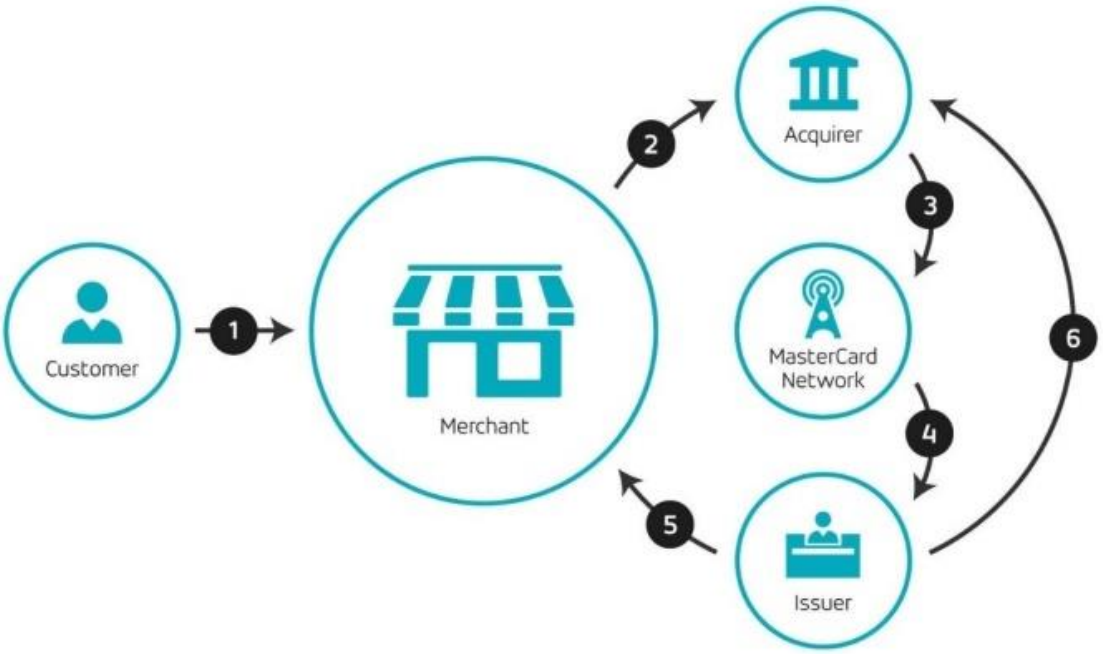

Figura 8. Flujo de transacciones en tarjetas bancarias.
Tomado de (MasterCard, 2017)

El switch transaccional hace referencia a un intermediario que comisiona por el servicio de transacciones, su función principal es interconectar a otras entidades financieras como Banred.

2.3.1.1 **Vulnerabilidades de tarjetas contactless**

La extracción de información personal comúnmente es para el beneficio de compras en línea o duplicación de la información en compras físicas.

Existen varios métodos identificados como principales vulnerabilidades de las tarjetas contactless.

- Modificación de información: Una vez que la tarjeta pudo ser leída o activada por un elemento activo, es posible modificar la información que se está transmitiendo.

Las tarjetas contactless pueden sufrir ataques dentro de la modificación de servicios como la denegación de este, pues un atacante podría desvincular o comprometer la información original generando falsas lecturas.

- **Retransmisión:** La retransmisión o relay utiliza un canal de comunicación de retransmisión, pues de manera lógica, realiza un incremento de rango en la comunicación.

Este tipo de ataque requiere un elemento que interactúe directamente con la tarjeta contactless emulando ser la tarjeta y otro emulando ser el lector. Este método se lo conoce como “mafia fraud”.

Como se puede ver en la figura 9 el ataque se produce entre dispositivos con capacidad de leer NFC.

Figura 9. Simulación de ataque mafia fraud

Tomado de (CN-CERT,2016)

Desde la versión 4.4 KITKAT de Android, se estableció una nueva funcionalidad de emulación de tarjeta a nivel de usuario, con el cual es posible el ataque con la retransmisión de dispositivos usando Android, sin ninguna modificación adicional.

El ataque únicamente se lo puede realizar solo si se mantiene una latencia baja en la red de comunicación. (CCN-CERT, 2016)

La fórmula de Frame Waiting Time (FWT), definida en ISO/IEC 14443-4 permite al configurar y delimitar el tiempo de respuesta entre dispositivos NFC.

Como se puede ver en la fórmula a continuación se delimita el tiempo de respuesta para mitigar problemas de ataque por retransmisión.

$$FWT = 256.\left(\frac{16}{fc}\right).2^{FWI}$$

$$0 \leq FWI \leq 14,$$

$$donde$$

$$fc = 13.56 = 13.56\,MHz$$

Es decir, FWT varía entre 500µs a 5s, Un ataque de transmisión en NFC es posible cuando el retraso del canal de retraso del canal de retransmisión es inferior a 5 segundos.

Resistir a los tipos de ataques más clásicos, como el fraude a distancia y la mafia fraud. el dispositivo del medio intenta demostrar con un verificado que el

dispositivo activo está en la proximidad del verificador, a pesar de que el mismo está mucho más lejos.

Es fácil darse cuenta de la importancia de la limitación de protocolos de distancia si tomamos en consideración aplicaciones del mundo real, como tarjetas bancarias y control de acceso a automóviles RFID.

Los protocolos son utilizados frecuentemente por los fabricantes de automóviles para el sistema de bloqueo / desbloqueo en vehículos.

Sin embargo, se ha demostrado que estos protocolos son susceptibles a ataques.

Por lo tanto, hay una creciente necesidad de utilizar protocolos de límite de distancia segura para lograr integridad y confiabilidad en aplicaciones del mundo real.

2.3.1.2 **Ataque a tarjeta contactless aplicado**

- **Escucha secreta (eavesdropping):** consiste en escuchar una conversación privada de terceros sin consentimiento, misma que puede estar cifrada o no.

Este problema aparece en cualquier sistema de comunicación por ondas de radio para su transmisión.

Los atacantes utilizan los POSNFC y realizan transacciones mínimas para evitar el uso del código o pin de la tarjeta vulnerada una vez obtenida la información. (CCN-CERT, 2016)

La vulnerabilidad en NFC reside en que para poder “escuchar” las emisiones de radiofrecuencia entre un dispositivo lector NFC y una tarjeta o tag, estas deben estar lo suficientemente cerca como para detectar los paquetes enviados de información.

La distancia normada para la comunicación por NFC son de 10 cm, que está basada en el estándar ISO/IEC 14443-3A.

A pesar de esto, una tarjeta contactless transmite su información a través del protocolo NFC sin verificación de quien la está leyendo, este problema es heredado de la tecnología en la que se basa por lo que no se puede comprobar la autenticidad del lector.

Para probar dicho concepto se procede a realizar el ataque. Esta descripción de ataque y ejecución fue desarrollada únicamente con fines investigativos y cumpliendo con todos los permisos y autorizaciones de la entidad que permitió este ataque.

El ataque no mostrará códigos completos.

Para el presente ataque se utilizará el lector NFC descrito en la figura 5, mismo que brinda las capacidades de ser configurado en un dispositivo con Sistema Operativo Windows y del cual se podrá obtener algunos datos de la tarjeta atacada.

Para lo cual se desarrolla el siguiente diagrama de flujo explicado en la figura 10.

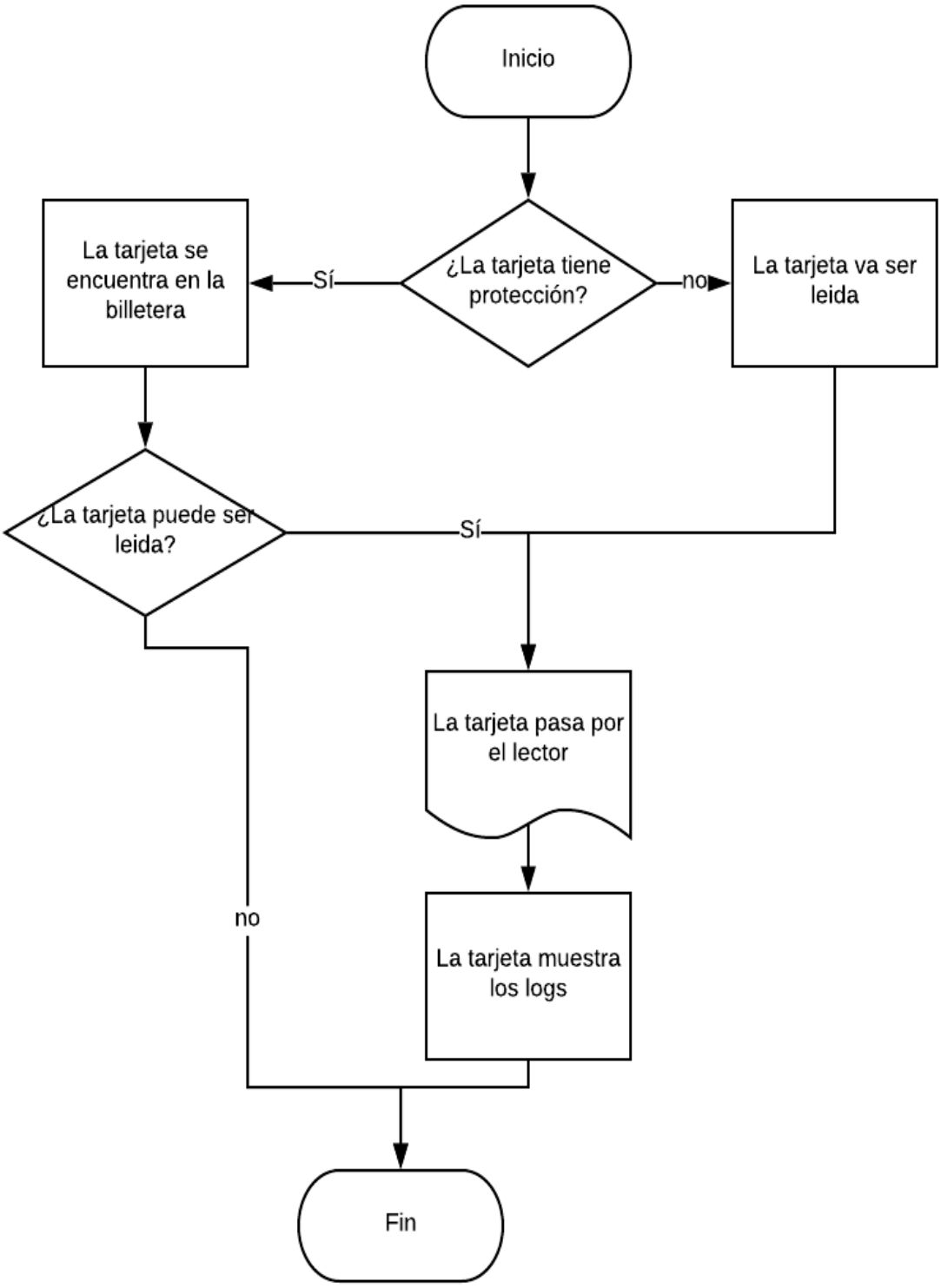

Figura 10. Diagrama de flujo ataque tarjetas.

El equipo con la lectura correspondiente se puede apreciar en la figura 11.

Figura 11. Lectura de tarjeta Contactless en equipo.

- Se desarrolla una pasarela de pagos que tendrá como características previas comunicarse con el equipo a través del puerto USB-2 y realizará un testeo de conectividad como se indica en la figura 12.

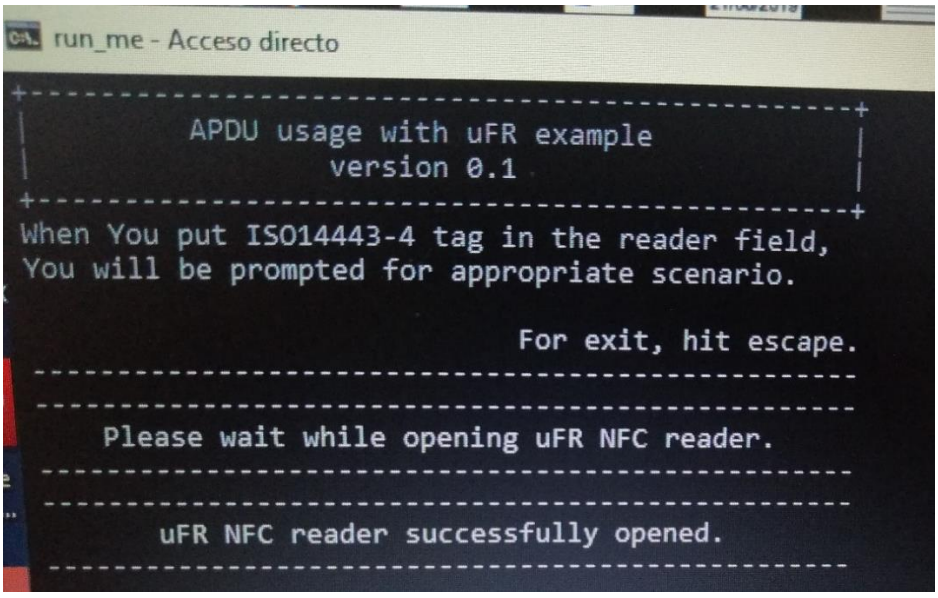

Figura 12. Testeo de conexión usb 2 hacia uFR.

- Se procede a revisar los paquetes de trasmisión y paquetes de información denominados APDUs como se indica en la figura 13.

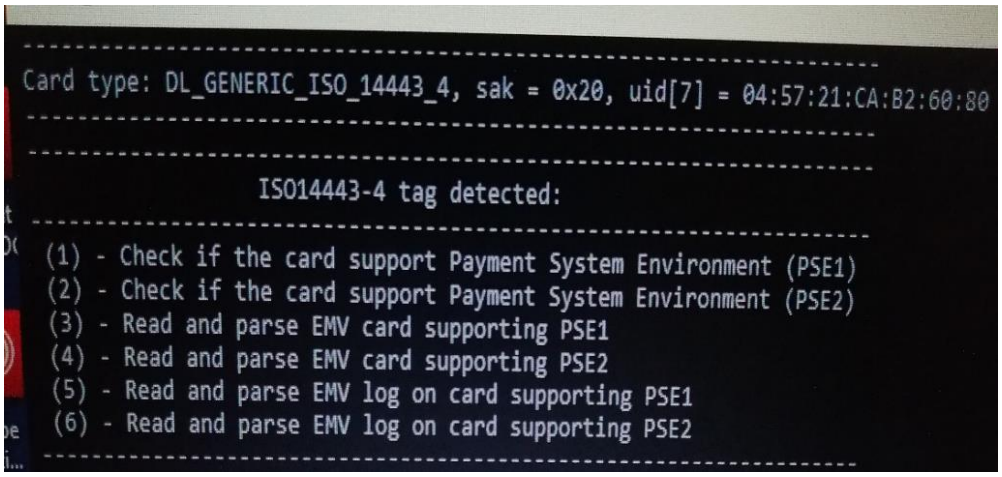

Figura 13. Pasarela de pagos escucha oculta tarjetas contacless.

- Se valida un histórico de transacciones relacionadas a la tarjeta que fácilmente indica el promedio de dinero que maneja la cuenta y la frecuencia en la que es utilizada. Esta información podría dar una perspectiva al atacante del dinero y los montos manejados en la cuenta tal como se visualiza en la figura 14.

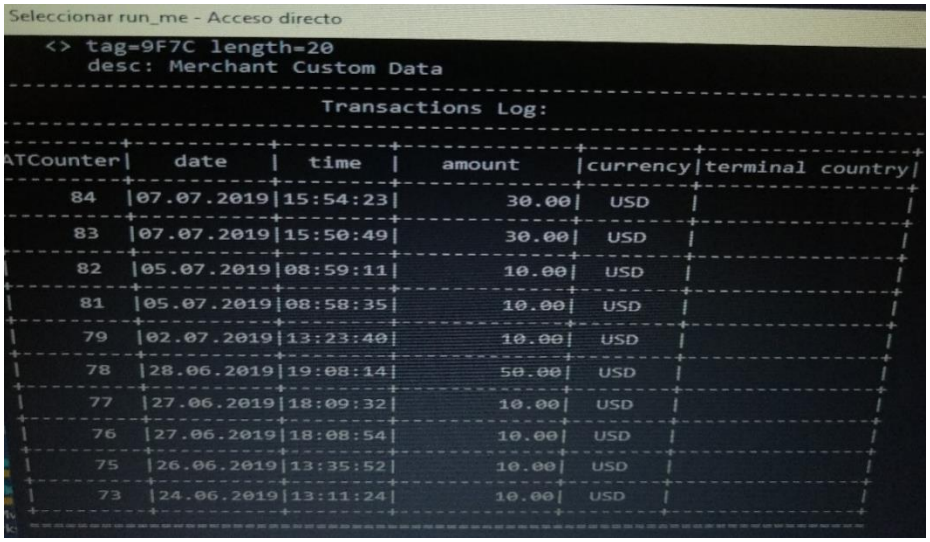

Figura 14. Escucha oculta, tabla de transacciones registradas.

- **Duplicación:** Este tipo de ataque, duplica la información necesaria para mantener en dos instancias iguales el código de la tarjeta contactless, en la prueba detallada a continuación se muestra la lectura y la escritura de la tarjeta

contactless emitida por una entidad bancaria del Ecuador hacia un Tag NFC funcional en la misma frecuencia.

En la figura 15 se puede identificar un diagrama de flujo para este tipo de ataque:

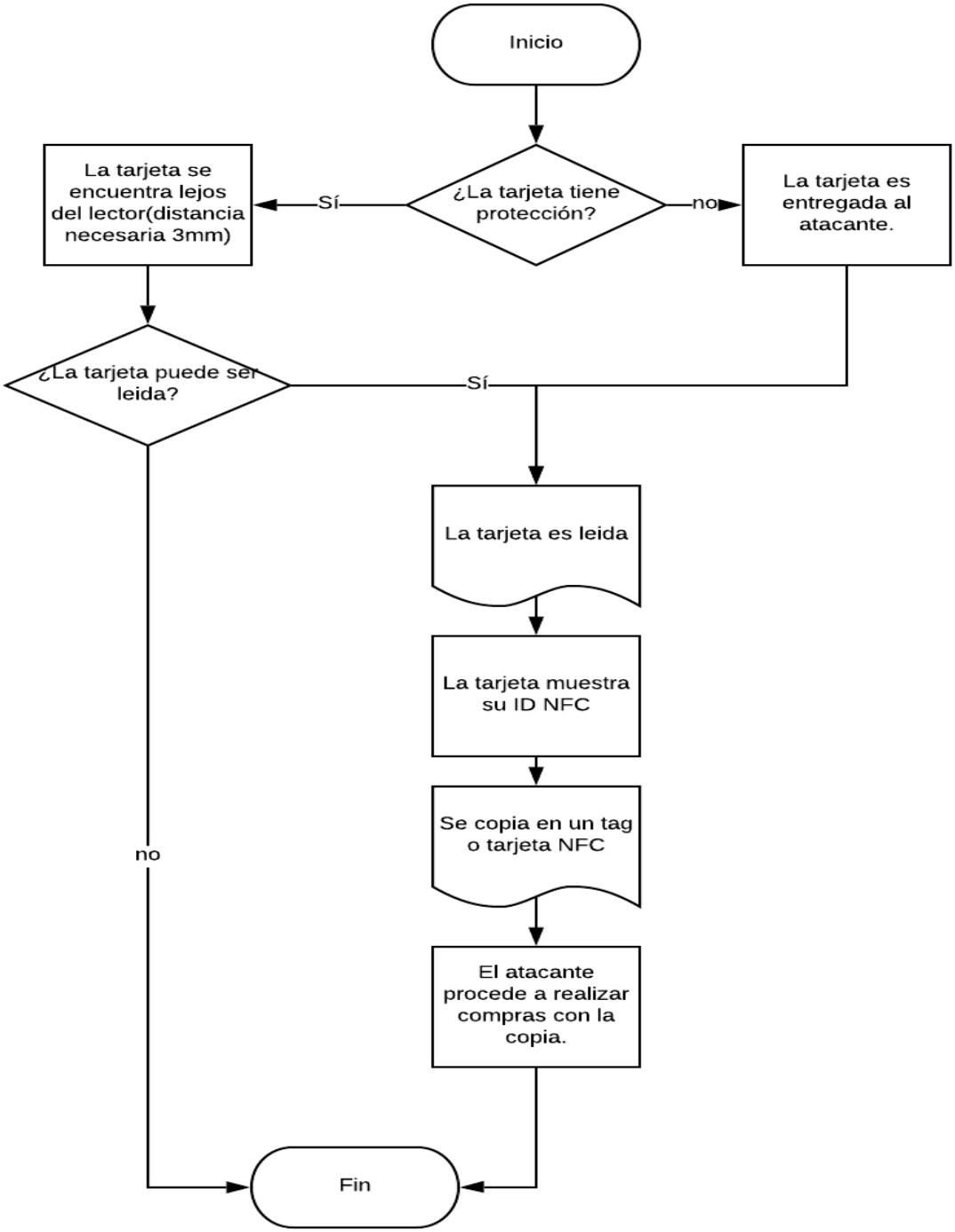

Figura 15. Escucha oculta, Diagrama de flujo

Como se puede visualizar en la figura 16 la tarjeta pasa por el lector y este obtiene de manera efectiva el ID del NFC mismo que puede ser utilizado en cualquier espacio público.

Figura 16. Lectura de tarjeta contactless en entidad bancaria.

El ID detectado puede ser plasmado en otro elemento como tarjeta o tag, que maneje las mismas características y frecuencias como se puede visualizar en la figura 17.

Figura 17. Escritura de tarjeta contactless a un tag.

2.3.2 **Mejoras en tarjetas contactless**

Las contramedidas para las vulnerabilidades detectadas en tarjetas contactless son definidas como recomendaciones, en el caso de la vulnerabilidad de escucha secreta, se recomienda el uso de carteras o tarjeteros que bloqueen las señales.

Otra medida adicional es el uso de mecanismos de cifrado en la comunicación, en este caso el uso de esquemas criptográficos como RSA, 3DES, etc.

Para evitar los ataques de retransmisión es necesario utilizar protocolos de cota de distancia donde se establece una cota superior a la distancia física entre dos dispositivos utilizando el RTT (Round-Trip-Time).

RTT sirve para conocer el tiempo de ida y vuelta de un paquete de información emitido entre dos dispositivos. (Heder, 2014)

2.3.3 **Seguridad MasterCard**

MasterCard introdujo las tarjetas sin contacto por primera vez en 2003 para ofrecer a los consumidores una forma segura y sencilla de pagar.

La tecnología contactless fue desarrollada por MasterCard con la mentalidad de nunca sacrificar la seguridad por conveniencia.

Las tarjetas y los dispositivos contienen un chip integrado y una antena de radiofrecuencia (RFID) que proporcionan un enlace inalámbrico con el lector contactless.

Cuando se toca la tarjeta o el dispositivo contra el lector, la información se transmite de manera muy segura en una fracción de segundo.

Los pagos con tarjetas contactless requieren información diferente a la que se realiza por teléfono o en línea.

El nombre del titular de la tarjeta, el código de seguridad de tres dígitos en el reverso de la tarjeta y la información de facturación, como el código postal, nunca se transmiten.

Cuando se realiza una transacción contactless, la tarjeta o el dispositivo le proporciona al lector un número único y dinámico que identifica de forma única y segura cada transacción específica.

2.3.4 **Raspberry pi y módulo NFC**

Es un ordenador reducido de placa simple de bajo costo, en el 2009 nace la fundación Raspberry PI impulsado por el desarrollo informático con el incentivo de estimular la enseñanza de la informática en instituciones educativas a pesar de que su comercialización empezó en el año 2012.

Este equipo guarda en su interior un gran poder computacional que es aprovechado para diferentes proyectos a un bajo coste.

2.3.4.1 **Software**

El software es totalmente libre, su sistema operativo tiene gran variedad de entornos sin embargo su sistema madre es Raspbian basado en debían de la distribución Linux.

2.3.4.2 **Hardware**

Raspberry pi posee diferentes características técnicas como:

- Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
- 1 GB de RAM
- BCM43438 LAN inalámbrica, Bluetooth Low Energy (BLE) a bordo
- 100 Base Ethernet, GPIO extendido de 40 pines
- 4 puertos USB 2, Salida estéreo de audio
- Puerto de video compuesto, HDMI de tamaño completo
- Puerto de cámara CSI para conectar una cámara Raspberry Pi
- Puerto de visualización DSI para conectar una pantalla táctil Raspberry Pi
- Puerto microSD para cargar su sistema operativo y almacenar datos
- Fuente de alimentación Micro USB conmutada actualizada de hasta 2.5A.

3. Soluciones tecnológicas RFID y NFC

El presente capítulo describe cómo ha evolucionado la tecnología RFID sus diferentes frecuencias de operación, su aporte para la tecnología NFC, con una breve comparación entre ambas tecnologías.

Además, presenta las frecuencias en que operan estas tecnologías, sus posibles usos en diferentes ámbitos y futuros desarrollos.

3.1 **Tecnología RFID**

3.1.1 **Origen.**

El RFID se originó en la segunda guerra mundial por el desarrollo del radar donde el ejército alemán se vio en la necesidad de reflejar la señal dando origen o un RFID pasivo.

Para 1990 en IBM sus ingenieros desarrollan y patentan un sistema de frecuencia ultra elevada UHF.

Como se observa en la figura 18 se puede ver el rango de frecuencias en que opera el RFID, su uso, aplicaciones, etc.

	LF	HF	UHF	Active
Frequency	125 – 134.2 KHz	13.56 MHz	850 – 960 MHz	100 KHz – 2.45GHz
Range	0.2 – 2m	Up to 1m	Up to 3m	Up to 100m
Cost	Typ. 3 GBP	(Typ. 0.50 GBP)	(Typ. 0.30 GBP)	(Typ. 20 GBP)
Memory	Typ. 64 bits	Typ. 2048 bits	Typ. 96 bits	Typ. 32 bits
Penetration of Materials	V. Good	Good	Poor	V. Good
Data Rate	Slow	Fast	Fast	Fast
Reader Cost	50 – 500 GBP	50 – 3000 GBP	1000- 3000 GBP	200-600 GBP
Read Multiple Tags	Poor	Good	Very Good	Good
Applications	Animal Tags, Vehicle Immobilisers, Industrial Applications	Item Tracking, Access Control, Smart Labels	Box and Pallet tracking Some Item Tracking	Industrial Applications. Asset Tagging Location Systems

Figura 18. Banda de Frecuencias que opera RFID

Adoptado de (advancedmobilegroup,2019)

Para los años posteriores esta tecnología evolucionó en forma de uso cotidiano siendo por ejemplo el uso de acceso a edificios, entre otros.

El sistema RFID ha ido evolucionando y esta se ve reflejada en la tabla 3 que se muestra a continuación:

Tabla 3.

Historia del RFID

Década	**Eventos**
1940-1950	Uso del radar durante la segunda guerra mundial, descubrimiento RFID en 1948
1950-1960	RFID primeros experimentos bajo condiciones controladas.
1960-1970	Desarrollo de la teoría RFID, al igual que aplicaciones para diferentes áreas de aplicación.
1970-1980	Repunta el desarrollo del RFID, aumento en su velocidad, primeras muestras en uso industrial.
1980-1990	Surgen aplicaciones comerciales.
1990-2000	Salen los estándares internacionales.

Tomado de: (Landt, 2019)

La tabla 4 muestra algunas patentes estadounidenses en lo que se refiere al RFID, pedidas a lo largo de la historia.

Tabla 4.

Patentes solicitadas para RFID

Numero de Patente	**Titulo**
3,713,148	Aparato y sistema de transpondedor.
3,745,569	Transponder accionado remotamente.

3,852,755	Transpondedor accionado remotamente que tiene una matriz de antena dipolo.
4,001,822	Matricula electrónica para vehículos de motor.
4,068,232	Codificador pasivo de transpondedor de microondas.
4,096,477	Sistema de identificación mediante transpondedores pasivos codificados.
4,114,151	Sistema de identificación mediante transpondedores pasivos codificados.
4,123,754	Sistema electrónico de detección e identificación.
4,242,663	Sistema de identificación electrónica
4,345,146	Aparato y método para un sistema electrónico de identificación, actuación y registro.
4,354,099	Sistema de identificación electrónica
4,463,353	Sistema de seguimiento.
4,473,825	Sistema de identificación electrónica con bloqueo de entrada / salida de potencia y mayores capacidades.
4,481,428	Divisor de frecuencia portátil, sin batería, útil como transpondedor de radiación electromagnética.
4,490,718	Aparato de radar para detectar y / o clasificar un objetivo reflectante agitado.
4,494,545	Sistema de telemetría de implantes.
4,510,495	Sistema de identificación pasiva a distancia.
4,525,713	Sistema electrónico de identificación de etiquetas.
4,546,241	Sistema electrónico de identificación de proximidad.

Tomado de: (Landt, 2019)

Por su facilidad de uso y por su gran utilidad esta tecnología se expandió rápidamente hacia muchos sectores como son el industrial y uso en casa. La figura 19 muestra el interior de una tarjeta RFID.

Figura 19. Tarjeta RFID
Adoptado de (LOGYCA, 2015)

3.1.2 **Funcionamiento.**

Este sistema básicamente se basa en la interacción entre tres elementos una antena, transmisor y un transponder, el transmisor está compuesto con un decodificador, el transmisor por una antena y un chip que electrónicamente es programado con cierta información.

RFID se puede clasificar en dos: emisor y receptor, los cuales tienen unas diferencias entre sí y radica en el que tiene una fuente de alimentación o no.

Las etiquetas pasivas en caso de poseer una alimentación propia tienen un alcance limitado de 5 metros, si lleva o cuenta con una fuente de alimentación propia.

En la figura 20 se muestra la base del funcionamiento del sistema RFID.

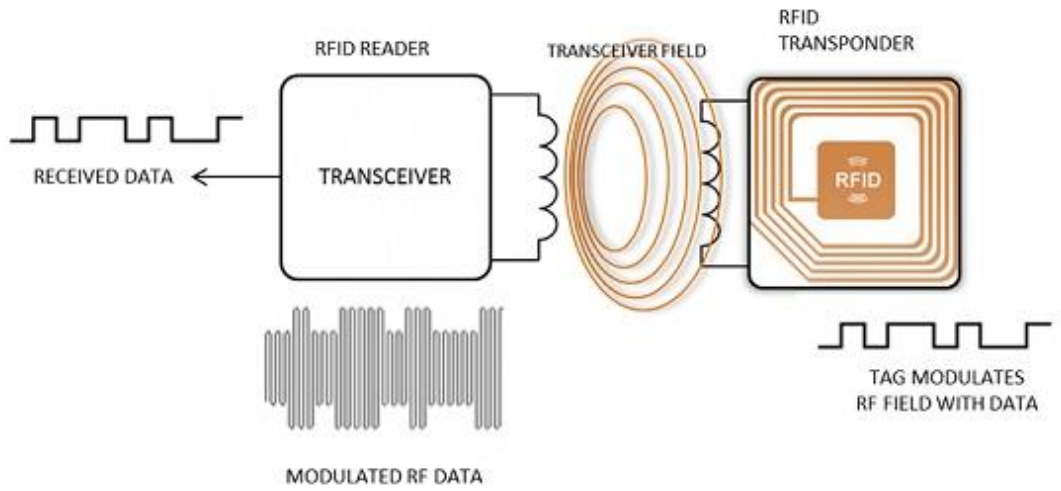

Figura 20. Tarjeta RFID funcionamiento

Adoptado de: (Seeburger, s.f.)

3.1.3 **Conceptos Básicos.**

RFID tiene tres componentes básicos los cuales son:

- Controlador

Es el equipo anfitrión que puede ser la PC en la que se tiene el software controlador.

- Tag

Es la antena con un circuito integrado encargado de transmitir el número de serie único de acuerdo con cada petición.

- Lector

Puede ser de lectura o escritura compuesto por un circuito integrado de radiofrecuencia y un módulo de control.

3.1.4 **Clasificación**

- Activos

Integra en sus componentes elementos más sofisticados además de baterías lo que hace que su capacidad de almacenar datos sea mayor, y hace posible que la distancia de lectura sea mayor. (hasta 100m)

- Pasivos

No tienen baterías, para almacenar la energía utiliza un capacitor que está en el interior de las tarjetas, por lo cual su rango de lectura es muy limitado.

- Semi activos

Tienen una fuente integrada para la lectura, pero en él envió utiliza la potencia emitida por el lector.

3.1.5 **Lectores RFID**

Dispositivo mediante el cual permite leer y escribir a un tag compatible su lector se comprende de las siguientes partes:

- Antena

Está conectado al emisor y transmisor

- Transmisor

Envía el ciclo del reloj a través de su antena al tag.

- Receptor

Recibe cualquier señal proveniente del tag y los verifica con el microprocesador.

- Microprocesador

Encargado de verificar y convertir todas las señales para su lectura.

- Memoria

Aquí se almacena información necesaria para que funcione todo el sistema, a si no exista comunicación.

- Controlador

Hace posible controlar todas las funcionalidades del equipo.

- Canales de entrada/salida

Permiten al lector tener la posibilidad de colocar sensores de diferentes tipos.

- Alimentación

Provee de energía eléctrica a todos los componentes.

- Fuente de comunicación

Es el protocolo que utiliza para su comunicación siendo posible r232, rj45, etc. La conectividad es un aspecto muy necesario para tomar en cuenta ya que de este depende la velocidad de conexión, calidad de lectura y escritura.

- RS232

Comunicación de corto alcance que es muy confiable, tiene la limitante de tener una baja velocidad de comunicación que máximo llega a 115.2 kbps y su alcance está limitado a 30 metros.

- Rs-485

Es una mejora del sistema r232 limitando a 1200 metros y llegando hasta una velocidad de 2.5 Mbps.

- Ethernet

Comunicación estable y eficiente que cubre todas las demandas que la tecnología requiere para una rápida lectura y escritura.

- Wireless 802.11

Reduce los cables y es utilizado en los RFID inalámbricos.

- USB

Puerto que algunos fabricantes lo incorporan con la desaparición del puerto serial.

La imagen 21 muestra como está conformada una tarjeta RFID con sus componentes internos.

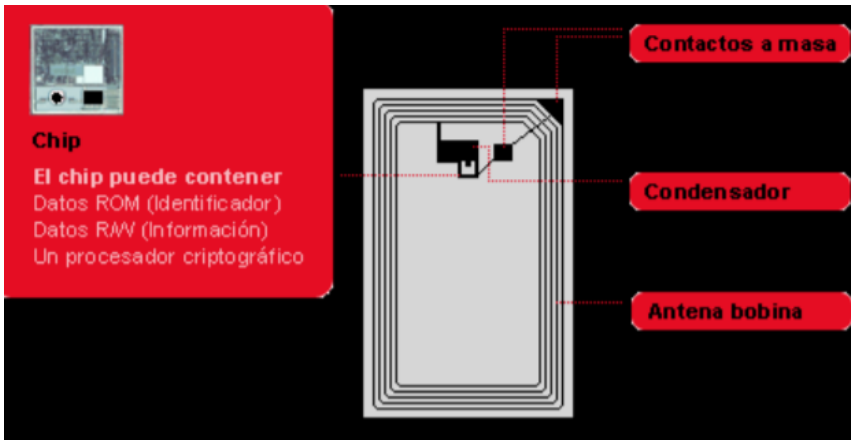

Figura 21. Componentes de un RFID

Adoptado de (Aidacentre, s.f.)

3.1.6 **Frecuencias**

Las frecuencias de RFID tiene cuatro rangos:

- Microondas

(2.45-5.8Ghz), puede llegar a cubrir hasta un área de 6 metros donde los tags no están muy separados del lector.

- Ultra alta frecuencia

(0.3-1.2GHz), cubre hasta 4 metros, pero esto dependiendo el fabricante y de las condiciones ambientales.

- Alta frecuencia

(13.56MHz), cubre una distancia desde 1cm hasta los 1.5 metros, y son de tipo pasivo.

- Baja frecuencia

(9-135 KHz), tiene una distancia de lectura de corto alcance y tiene como desventaja la lectura de un equipo al tiempo.

3.1.7 **Equipos que cuenta con RFID**

Los equipos han evolucionado con el tiempo tanto de su uso comercial como en uso personal ya sea en teléfonos, sistemas de seguridad e identificación, en desarrollo para determinar la propagación de señales en diferentes rangos de frecuencia y como de tecnología.

La tabla 5 que está a continuación muestra los diferentes tipos de equipos que cuenta con la tecnología de transmisión de datos de corto alcance y su respectiva descripción.

Tabla 5.

Equipos RFID

Tipo de RFID	**Descripción**
Lectores y antes fijas	Antes fijas en frecuencia UHF que pueden operar en condiciones extremas
Dispositivos móviles	Ofrecen movilidad a través de tecnología inalámbrica bluetooth, wifi, etc.

Etiquetas pasivas	Etiquetas para fácil aplicación y que permiten una lectura rápida.
Etiquetas activas	Estas etiquetas pueden ser leídas hasta una distancia de 50 metros.
Lectores de área	Permiten monitorear un área y son ideales para puertas inteligentes.
Impresoras RFID	Impresora de alto rendimiento que incorpora un codificador RFID.

3.2 **Tecnología NFC**

El sistema NFC se basa en el sistema RFID, siendo una evolución de este último además de presentar mayor seguridad en cuanto a su operabilidad.

La tecnología NFC aparece en el año 2004 por un acuerdo de Nokia, Philips y Sony pero que se basa en tecnología patentada por Charles Walton quien fue el pionero de los sistemas RFID.

Uno de los primeros fabricantes en introducir la tecnología fue Nokia con el modelo 6131 y el Nexus S fue uno de los primeros con sistema Android en colocarlo, ahora son cada vez más fabricantes de diferentes empresas los que colocan esta tecnología en sus teléfonos.

Su sistema se basa en el mismo sistema RFID solo que es una extensión al estándar ISO 14443 de RFID (Estándar que define las tarjetas por proximidad y pagos que se realizan con esta tecnología), hoy en día es implementado en la mayoría de los dispositivos electrónicos móviles.

En la tabla 6 se muestra algunas aplicaciones de esta tecnología.

Tabla 6.

Aplicaciones NFC

Año	**Descripción**
Equipos celulares	Utilizados para pagos, identificación de los teléfonos y transferencia de datos.
Smartwatchs inteligentes	Utilizado para pagos, identificación y transferencia de datos.
Tarjetas de crédito	Utilizado para pagos, es la última tecnología utilizada en la mayoría de las tarjetas de crédito.
Tarjetas de acceso a terminales de buses.	Para pagos y recargas mediante esta tecnología un claro ejemplo es SITP en Bogotá.
Collares de mascotas	Para información de dueño de la mascota y fácil localización de este.

3.3 **Diferencias entre NFC y RFID**

- NFC es una evolución del sistema RFID otorgando mayor seguridad.
- RFID puede operar hasta unos metros mientras que NFC solo puede operar a máximo unos 10 cms.
- NFC cada vez tiene un uso más común por la popularización e integración en teléfono móviles.

3.4 **Futuro del NFC**

NFC viene evolucionado y actualmente en la mayoría de los teléfonos de hoy en día es posible ya contar con esta tecnología, y en un futuro no muy lejano no habrá equipo que no cuente con esta tecnología.

Esto hace que se pueda pensar en varios usos de esta tecnología y con sus beneficios, como es rapidez de lectura en transacciones, envió y recepción de información más efectiva.

Pero el futuro de la tecnología se centra en el consumidor, donde este tiene la prioridad y donde toda transacción se podría simplificar con esta tecnología como por ejemplo en ingresos a aeropuertos reduciendo tiempos de espera, pagos de todo tipo de transacción, etc.

3.5 **Tabla comparativa de NFC y RFID**

En la tabla 7 se presenta una comparación entre las tecnologías NFC Y RFID.

Tabla 7.

Tabla comparativa NFC y RFID

CARACTERÍSTICA	**RFID**	**NFC**
Tipo de Red	Multipunto	Punto a punto
Frecuencia	Hasta 868 MHz	13.56 MHz
Cifrado	No	Si tiene
Tiempo requerido para establecimiento de conexión	>5s	<0.1s
Distancia de operación	Hasta 10 metros	Máximo 10 cms

4. Modelamiento de dispositivo

En el presente capítulo se presenta la solución a implementarse para el problema planteado, los equipos a utilizar, las características de estos, sus diferentes funcionamientos, y la implementación del prototipo como tal.

4.1 Equipos por utilizar

4.1.1 Raspberry PI

Placa de computadora con bajos costos que fue desarrollado en Reino Unido en 2011 aunque fue hasta el año 2012 cuando empezó su comercialización.

Esta ‘’pc de bajo costo’’ se da quitando todos los accesorios que no son necesario para poder funcionar un computador sin afectar su funcionamiento básico.

En la figura 22 se muestra el diseño de un Raspberry PI con sus principales componentes.

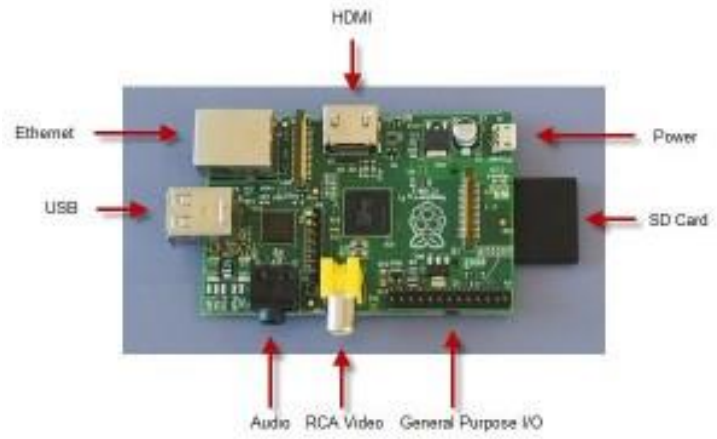

Figura 22. Componentes de un Raspberry

Raspberry ha evolucionado desde su lanzamiento y han lanzado diferentes modelos desde su fundación de los cuales son:

- Raspberry Pi 1 modelo A

Fue el primer modelo, el cual no poseía conector rj45 ni wifi por lo que requería de un adaptador USB wifi para conexión, entre sus características destacan 256 Mb en RAM procesador Broadcom BCM2835 single Core a 700 MHz y gráfica Broadcom Video Core IV.

Además de esto en sus conexiones posee 26 conectores GPIO, actualmente este modelo se encuentra descontinuado junto con el modelo Raspberry Pi 1 modelo B y B+.

- Raspberry Pi 1 modelo B y B+

Fue una variante del modelo anterior con mejoras en sus características como 512 Mb en RAM, conector RJ45, un puerto USB más, el modelo B+ incluyo 4 puertos USB e incluye el uso de la microSD.

- Raspberry Pi 2 modelo B

Modelo que vio la luz en 2014 destaca su cambio de procesador por el modelo BCM2836 de cuatro núcleos y 900 MHz, posee 1Gb de RAM.

- Raspberry Pi 3 modelo B

Este modelo e incluye wifi y bluetooth además de incrementar su velocidad de 900 a 1.2 GHz

- Raspberry Pi 3 modelo B+

El procesador pasa de 1.2 a 1.4 GHz, conexión inalámbrica de doble canal 2.4 y 5.8.

Cabe destacar que este modelo es el que será utilizado para el desarrollo del prototipo a implementar.

La imagen 23 muestra un Raspberry Pi 3 modelo B+ lanzado en marzo del 2018.

Figura 23. Raspberry Pi 3 modelo B+

- Raspberry Pi 3 modelo A+

Modelo de bajo costo del B+ con 256MB en RAM, un puerto USB sin conexión ethernet.

4.1.2 **Módulo de mensajes PUSH**

El mensaje PUSH es una forma moderna de comunicación implementada en dispositivos, que a diferencia de un mensaje de texto tiene una interacción con el usuario.

4.2 **Implementación del prototipo**

A continuación, se presenta la solución a implementarse para el problema planteado, código implementado, pruebas de simulación y pruebas de funcionamiento además de presentar las diferentes conexiones entre equipos.

4.3 **Tecnologías por implementarse**

4.3.1 **MQTT**

Que son las siglas de Message Queue Telemetry Transport que significa protocolo de transporte de mensajes, es un protocolo usado para comunicación de maquina a máquina a través del internet, es un protocolo que se expandido rápidamente debido a su bajo consumo de ancho de banda y que puede ser utilizado en casi todo dispositivo.

La implementación que se va a realizar es tipo estrella ya que tendremos un servidor o nodo central que hace de bróker, a modo de pruebas nos creamos una cuenta en clodmqtt.com y al no ocupar más de 5 usuarios concurrentes el servidor permite crear una cuenta gratuita.

4.3.2 **Firebase**

Otro protocolo que se utilizara en el proyecto es Firebase, el cual permite tener la aplicación móvil trabajando con datos de la nube y sincronizando con el bróker constantemente.

4.4 **Conexión de Equipos**

Los equipos que serán utilizados para la implementación del modelo son:

- Raspberry
- Módulo NFCPN532
- Memoria microSD 64gb clase 10

Para poder implementar el modelo se procede a descargar el software correspondiente. (Raspberry,2019).

Realizado esto se procede con la programación en Thonny Python IDE que permite la gestión de los pines GPIO proporcionados en la misma placa, por defecto Phyton viene instalado en el sistema operativo Raspbian por lo que no hay que instalar nada adicional.

El módulo NFC PN532 tiene los siguientes pines de configuración tal y como se describe en la imagen 24:

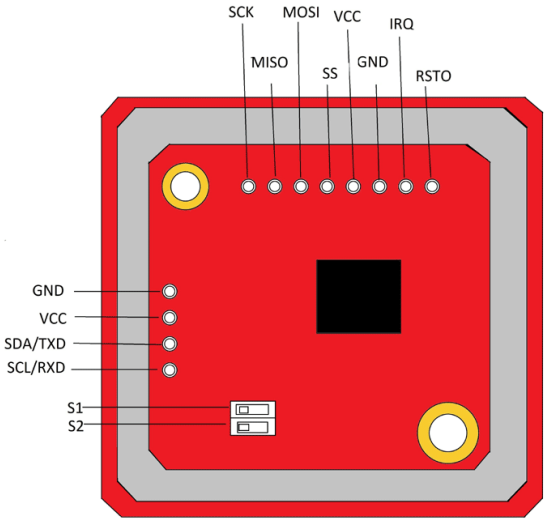

Figura 24. Módulo NFC, pines de conexión.

En la Tabla 8 se puede observar el detalle de los pines que serán utilizados para el desarrollo del proyecto.

Tabla 8.

Pines utilizados en modulo NFC

Abreviación	**Significado**
GND	Tierra
VCC	Fuente de poder (5V)
TXD	Pin de Trasmisión

RXD	Pin de Recepción

4.5 **Conexiones**

En la figura 25 se puede apreciar la conexión entre el módulo NFC y el Raspberry para las pruebas correspondientes de lectura y escritura.

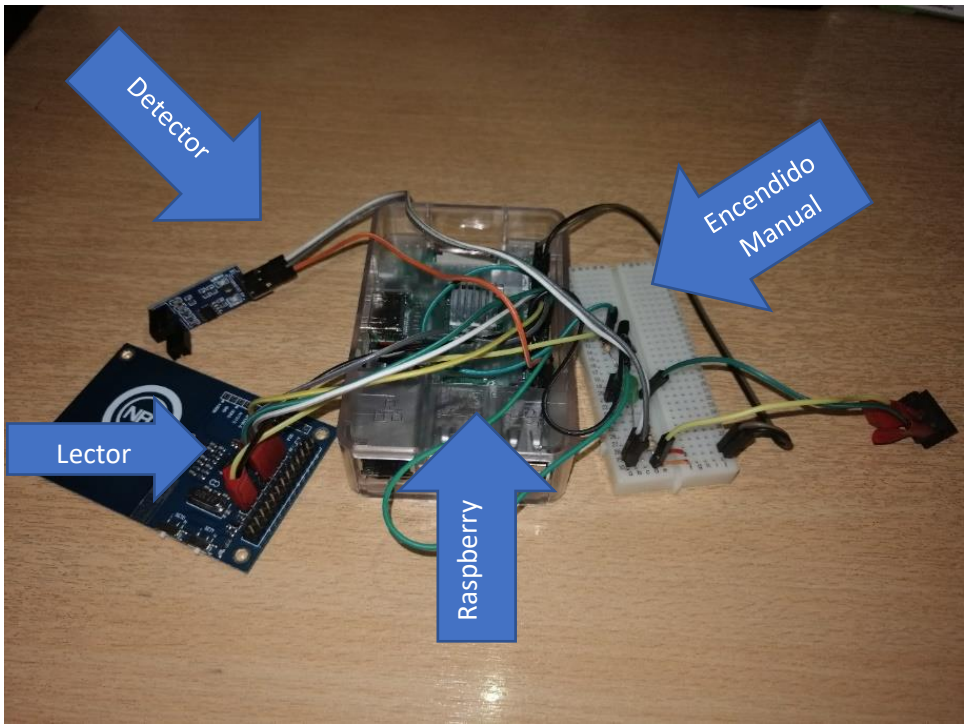

Figura 25. Pruebas de conexión módulo NFC y Raspberry.

En la figura 25 se observa por partes el módulo NFC, el detector de ingreso de la tarjeta, el Raspberry y el circuito de prioridad para encendido manual o por aplicación.

1. Lector, se encuentra conectado directamente al Raspberry, mediante el cual recibe la carga eléctrica y recibe o entrega en forma de pulsos eléctricos la información para ser interpretada.

2. Detector de tarjetas, es un módulo compuesto de dos sensores que perciben cualquier alteración entre sus dos polos, enviando la información de interrupción de un objeto en su campo, similar a un detector de movimiento.

3. Raspberry, es el equipo que permite que se procese de manera adecuada la información y controla los equipos conectados hacia él.

4. Circuito de encendido manual o automático a través de un transistor donde permite realizar este cambio de manual a automático.

En la figura 26 se realizan pruebas de bloqueo implementando el equipo Arduino para emitir la señal y poder simular un ataque.

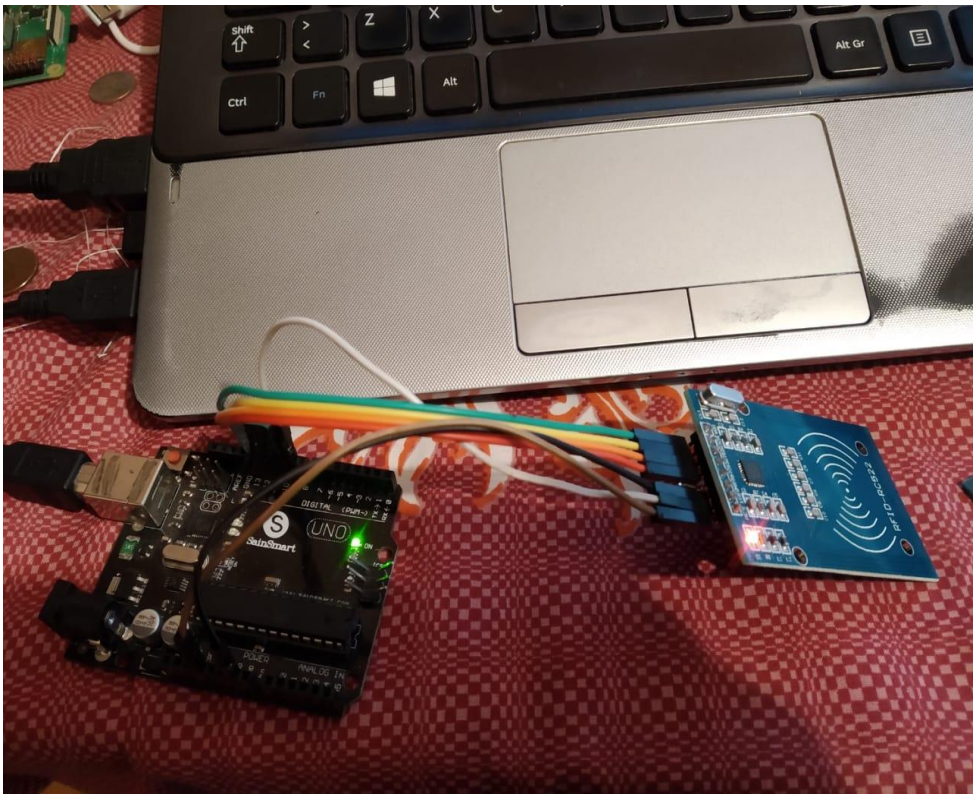

Figura 26. Pruebas para ataques NFC

En la figura 27 se procede a ejecutar la aplicación desde el servidor implementado en el Raspberry.

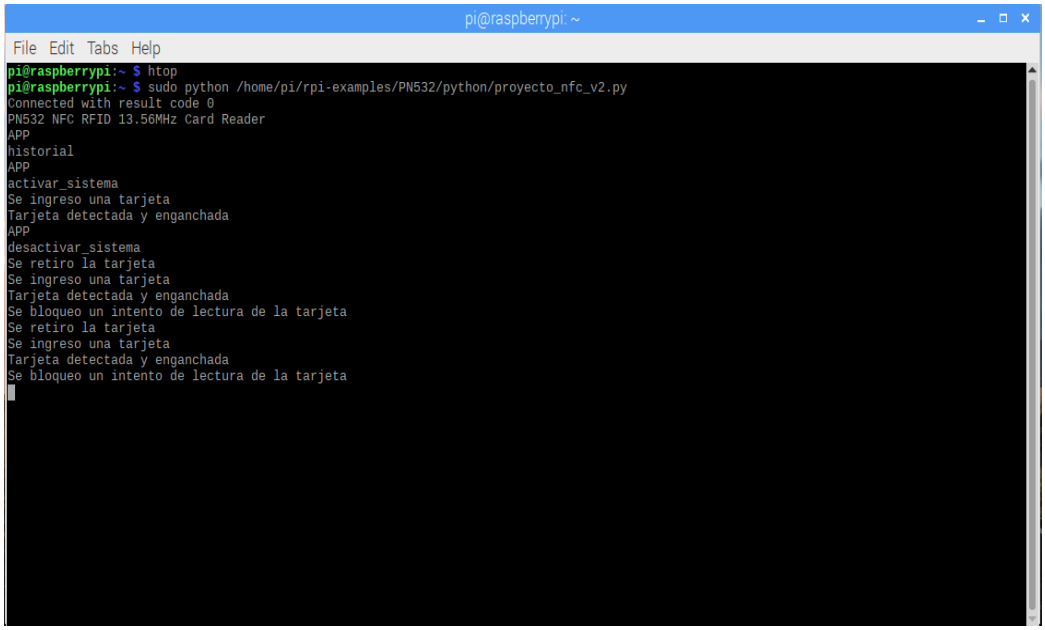

Figura 27. Ejecución de programa en Raspberry

En la figura 28 se observa la detección, bloqueo de intento de lectura y pedido de historial en la aplicación.

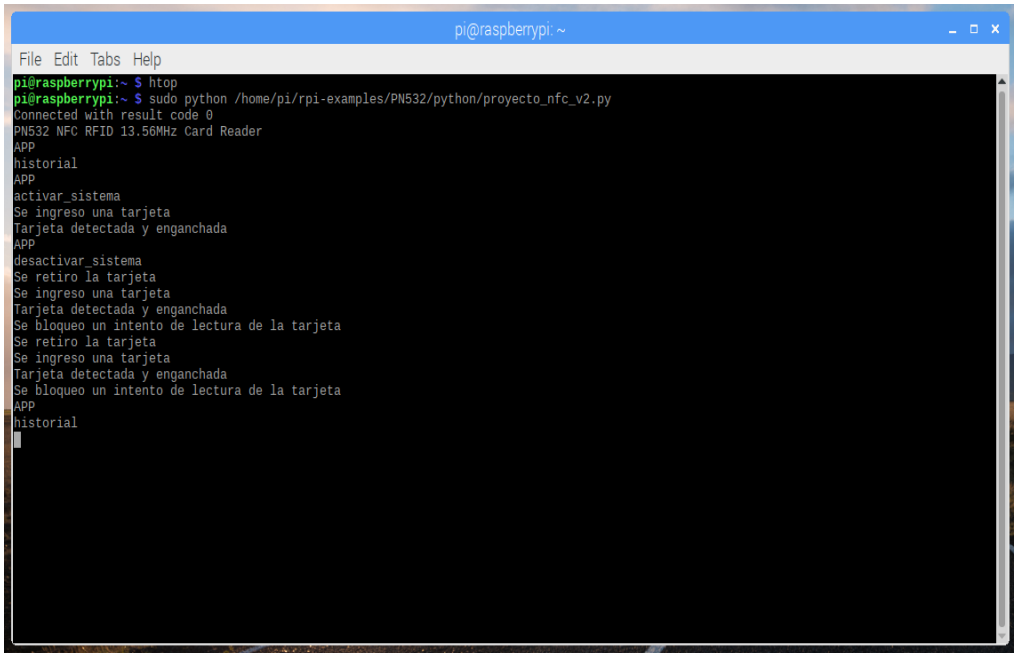

Figura 28. Pruebas de bloqueo y lectura de tarjeta.

En la figura 29 se observa el equipo que se utiliza para detectar cuando es introducida una tarjeta, el cual solo envía un 1 o 0 lógico.

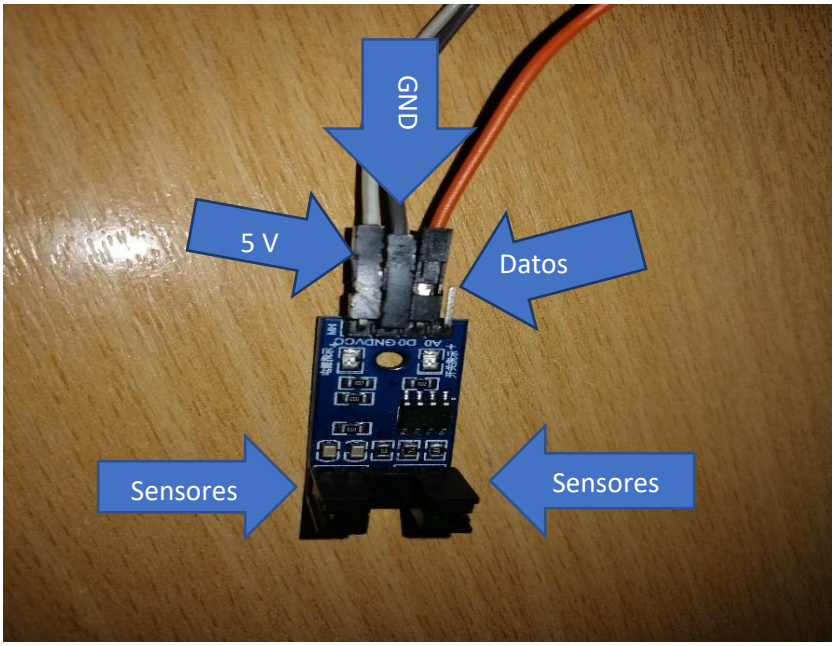

Figura 29. Equipo de detección de tarjeta.

La figura 30 muestra el equipo Raspberry conectado con la tarjeta NFC y a la red, la ip que se implementó al equipo es la 192.168.10.5

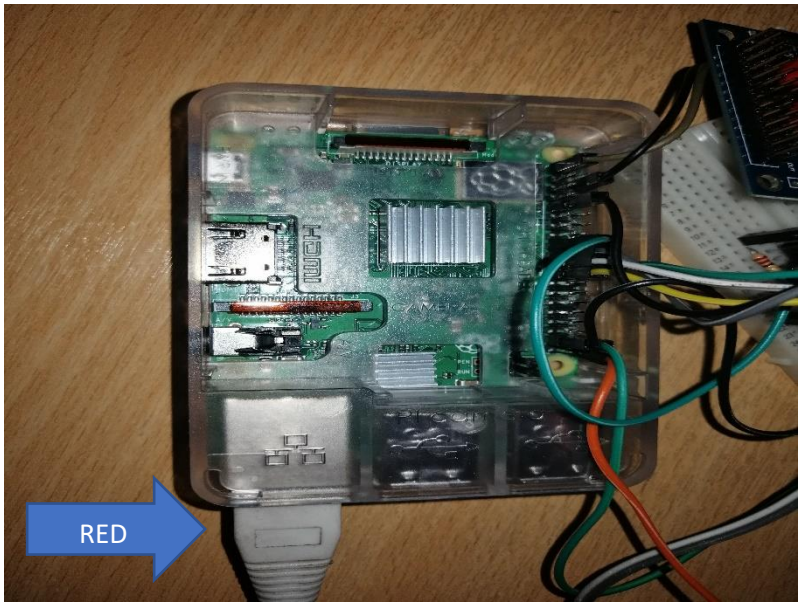

Figura 30. Equipo Raspberry conectado

La figura 31 indica el equipo NFC con las conexiones hacia el Raspberry.

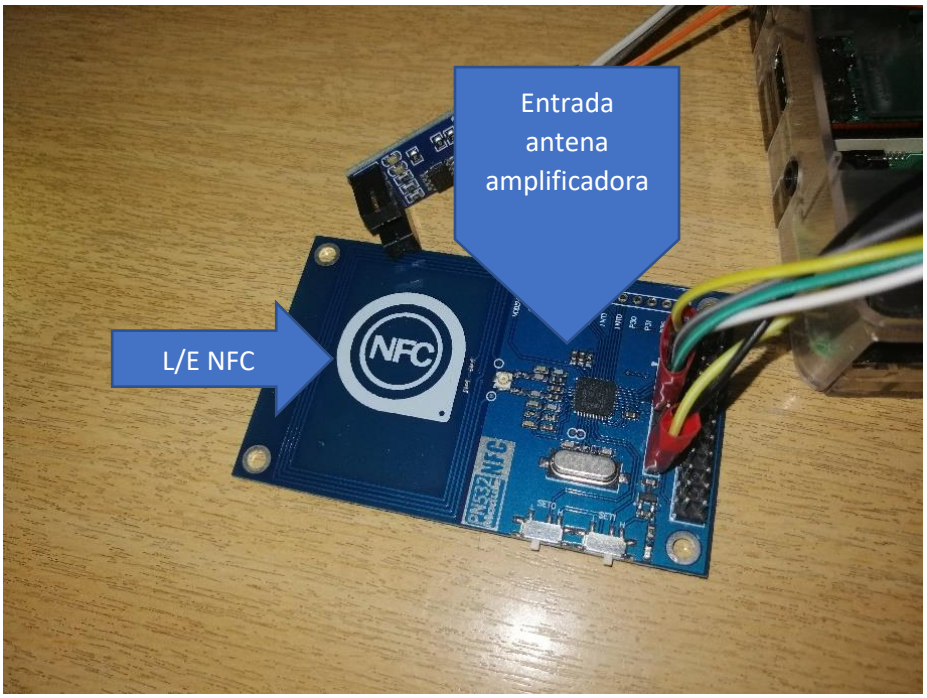

Figura 31. Módulo NFC PN532

4.6 **Diagrama**

La figura 32 muestra de manera general como esta implementado el enlace entre los diferentes protocolos implementados.

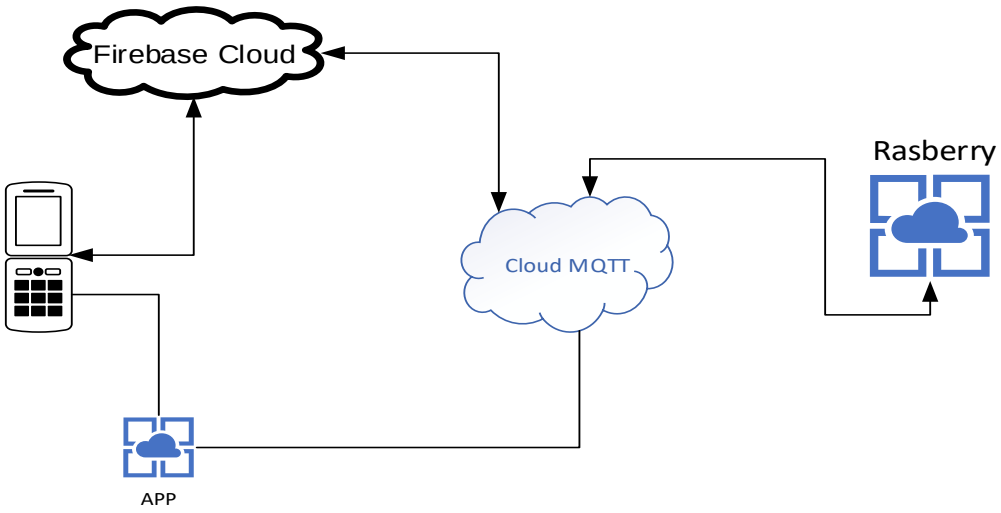

Figura 32. Conexión de Programa con cloud y Raspberry

En la Figura 32 se observa que utiliza el cloud MQTT y el Firebase Cloud para poder unir todo el proceso y tener una comunicación más fluida, además nos muestra de manera general como está todo el proceso de comunicación entre las diferentes instancias implementadas.

4.7 **Diagrama de Flujo aplicación móvil**

En la figura 33 se puede observar el diagrama de flujo de la aplicación móvil, su funcionamiento y características.

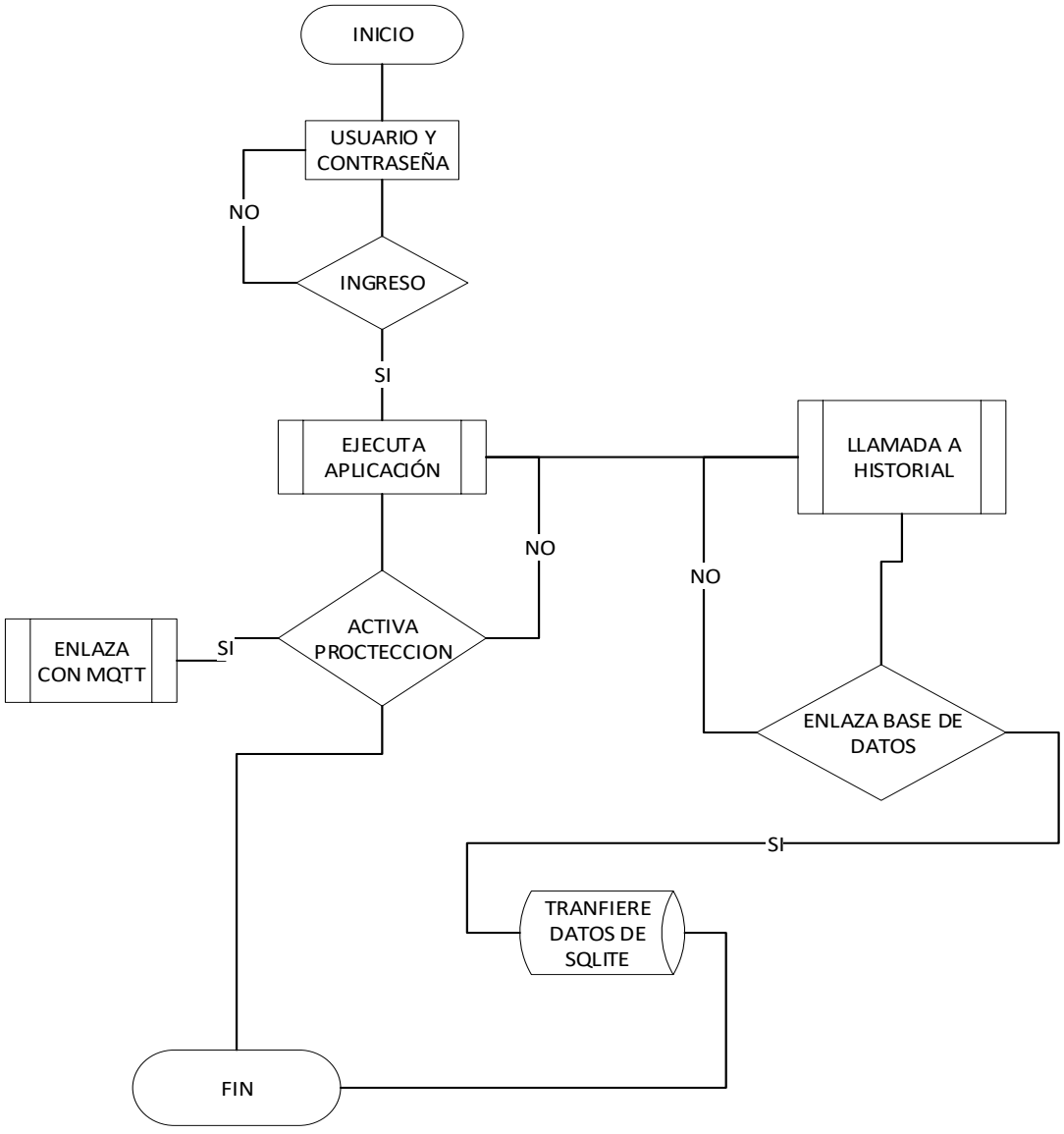

Figura 33. Diagrama de flujo aplicación móvil

4.8 **Funcionamiento del Software**

4.8.1 **Detección de encendido o apagado del sistema:**

Local: Interrupción o evento que se detecta por medio del pin GPIO20, configurado con pull up, se detectan los cambios de estado de alto a bajo (sistema activado) y de bajo a alto (sistema desactivado) La función se llama: sistema().

Remota: Interrupción o evento mqtt, se recibe un mensaje mqtt desde la aplicación móvil, esta envía un mensaje "activarsistema" o "desactivarsistema" y reenvía un mensaje de confirmación hacia la misma. La función se llama: on_message(client, userdata, msg):

4.8.2 **Detección de presencia de la tarjeta:**

Se realiza por medio del sensor de presencia en forma de U, el sensor se encuentra conectado al GPIO 21, el sensor envía a su salida un 0L cuando se detecta la presencia de la tarjeta y envía un 1L cuando no se detecta la tarjeta.

4.8.3 **Comunicación de la tarjeta:**

Para la comunicación de la tarjeta primero se debe cumplir la condición de activo del sistema (de forma manual o remota) y después el sensor de presencia haya detectado la misma; dadas estas condiciones el módulo PN532 está conectado con la tarjeta.

4.8.4 **Detección de un intento de lectura de la tarjeta:**

Cuando un dispositivo externo al sistema intenta leer la tarjeta, el sistema bloquea el intento, lo registra y notifica al usuario.

4.8.5 **Registro en base de datos:**

Para la base de datos se usa SQLite, sirve para guardar el registro de las alarmas que se producen mientras el sistema se encuentra encendido.

Las alarmas pueden ser: activación o desactivación manual del sistema, un retiro de la tarjeta cuando el sistema estaba encendido y un bloqueo de intento de lectura de la tarjeta.

Se guarda la alarma y la fecha y hora en la que se produjo, la función es: registrar_bdd(msg)

4.8.6 **Notificaciones hacia la aplicación:**

Para el sistema de notificaciones se usa el servicio de Google Firebase Cloud Messaging, el método para él envió de las notificaciones de forma individual es Publicador/Suscriptor.

El programa en Python publica una notificación a la aplicación en el tópico “app_test”, mientras que la aplicación móvil recibe la publicación suscribiéndose al mismo tópico.

La función se llama: enviarNotificacion(msg)

En la figura 34 se muestra la ventana de notificaciones que es enviada hacia la aplicación móvil.

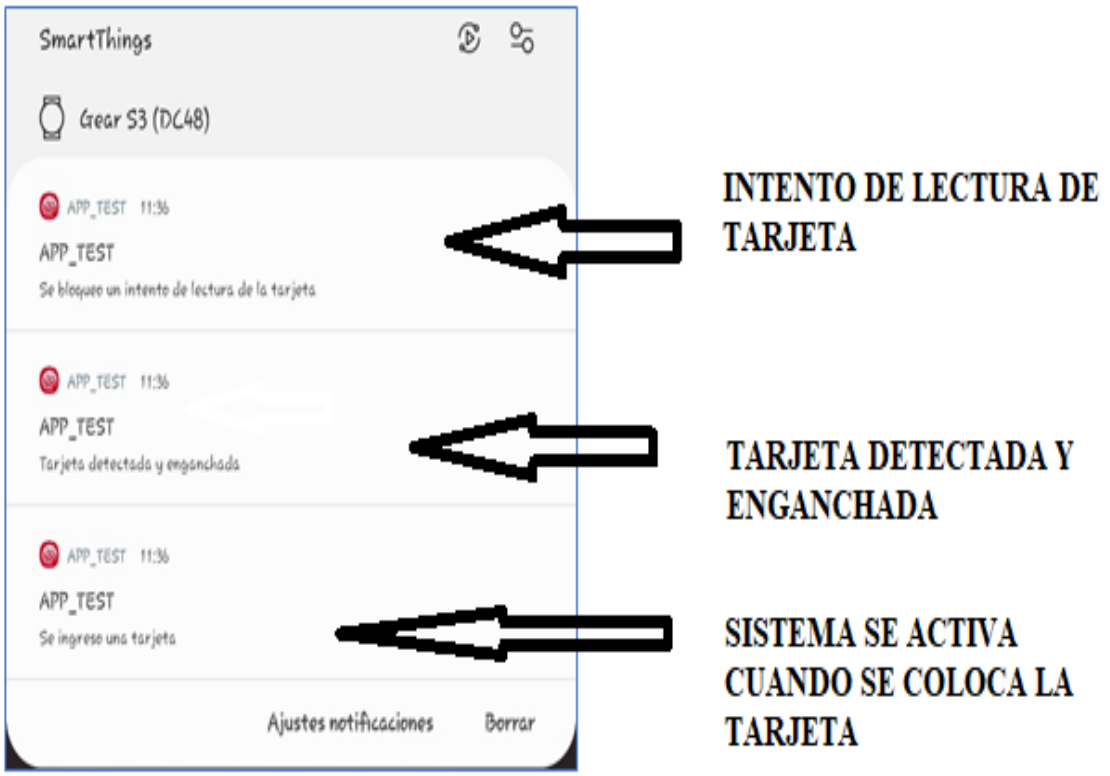

Figura 34. Notificaciones aplicación móvil

4.8.7 **Sistema de mensajería remota:**

Para el sistema de mensajes entre la aplicación y la Raspberry pi 3 se usa el servicio MQTT.

Este servicio permite comunicación bidireccional entre los dispositivos, el tópico en el cual la aplicación publica sus datos es: “APP”, mientras que la Raspberry

publica sus datos en los tópicos “RPI” Y “RPI_h” siendo el último en el que se publica el historial de actividades registrado en la base de datos.

El servidor que se usa para este caso es CloudMQTT.

4.8.8 **Sincronización del historial:**

Usando el protocolo de comunicación MQTT, se lee la información de la base de datos (sqlite), se organiza y se envía hacia la aplicación móvil.

Para una correcta sincronización se envía un mensaje de inicio y de fin. La función se llama sincronizar_historial()

4.9 **Paso a Paso**

Para tener el programa operando se debe realizar el siguiente procedimiento:

- Encender el equipo Raspberry
- Conectar el cable de red a la computadora, la ip designada el equipo Raspberry es la 192.168.10.5.
- Por escritorio remoto ingresamos el usuario: pi y contraseña Raspberry
- En terminal ejecutamos la aplicación desarrollada, con esto automáticamente se pedirá que se ingrese la tarjeta y el sistema estará listo para operar.
- Aquí se puede activar o desactivar el sistema manualmente con el switch instalado en el equipo o a través de la aplicación.
- Para terminar el o apagar todo el sistema hacemos todo el proceso contrario y se apagara el sistema operativo de manera correcta.

En la figura 35 se puede observar el botón de encendido o apagado del firewall a través de la aplicación.

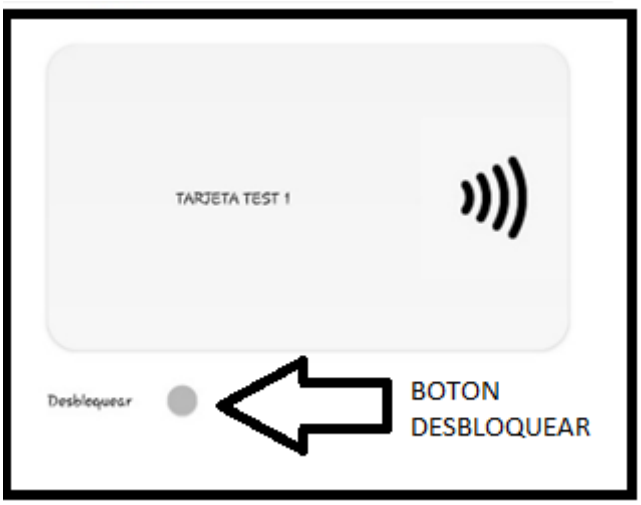

Figura 35. Botón de encendido en aplicación móvil.

4.10 **Pantallas Aplicación móvil**

La aplicación debe tener conexión a internet para poder comunicarse con el sistema, en la ventana de loggin se pide al usuario y contraseña, usuario: “admin” contraseña:”1234”

Al ingresar esos datos se pasa a la ventana principal.

4.10.1 **Ventana principal:**

En la ventana principal se tiene el control para activar o desactivar el sistema de manera remota y además se tiene la opción de abrir el historial de registro de alarmas.

Para lo primero se usa el servicio MQTT, el servicio se incluye en la aplicación al iniciar esta ventana, se usa el servicio en la nube Cloud MQTT. Al activar el switch la aplicación envía un mensaje a la Raspberry a través del tópico “APP” donde se indica si se debe activar o desactivar el sistema.

Luego se debe esperar el mensaje de confirmación de la Raspberry y cambia el estado del switch al valor correspondiente, ya sea para bloquear o desbloquear.

En la figura 36 se puede observar el ingreso a la aplicación desde un celular Android.

Figura 36. Menú principal aplicación móvil

4.10.2 **Ventana histórica:**

Al iniciar esta ventana se hace envía un mensaje MQTT para iniciar la sincronización del historial, la Raspberry responde con un mensaje “inicio” y después envía de manera ordenada cada alarma registrada y la fecha y hora

correspondiente, al final envía un mensaje “fin_sincronizacion” con el cual el sistema cierra la sincronización.

Mientras los datos van llegando, se guardan en una base de datos local, para mantener el registro y posteriormente mostrar los datos en el historial de la aplicación.

En la figura 37 se puede observar un ejemplo de cómo guarda los datos en la base de datos y como son estos observados a través de la aplicación.

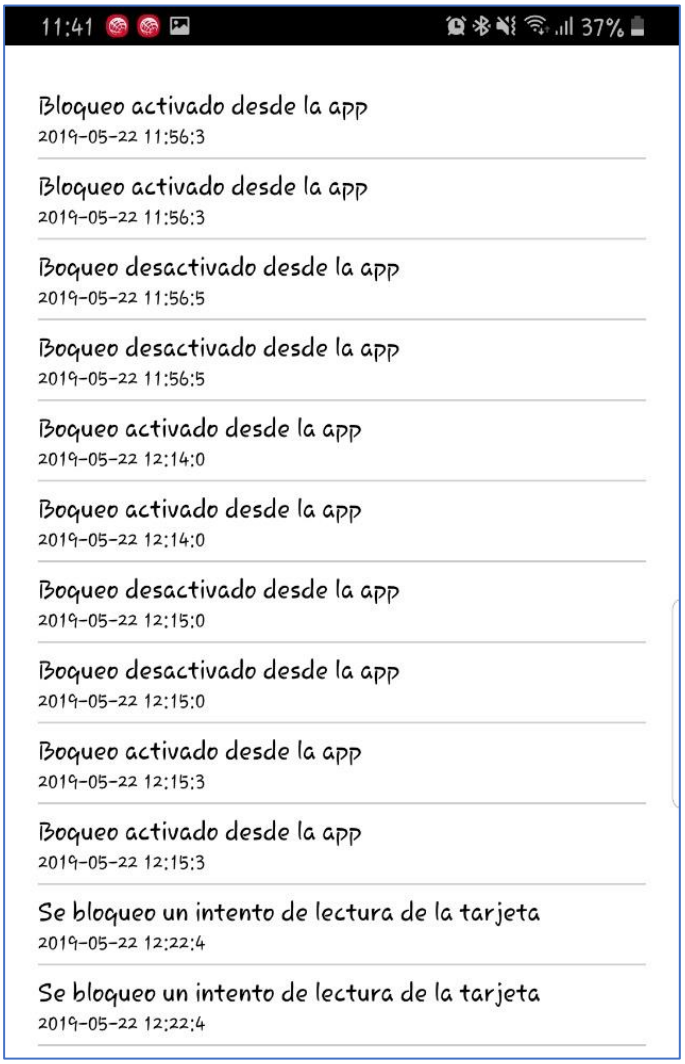

Figura 37. Pantalla de historial de sucesos

4.11 **Análisis de resultados**

Como se puede observar las tarjetas contactless pueden ser un problema de seguridad en caso de ser expuestas por el usuario, si no se tiene medidas de seguridad ni precaución de los establecimientos en los cuales son entregados, estos pueden ser vulnerados fácilmente sin ser un experto en el tema.

Varios estudios relacionados, demuestran que se debe poseer un ambiente propicio y adecuado para proceder con el ataque, en específico.

La escucha oculta puede ser efectuada en presencia del usuario sin que este mismo se percate del fraude, como se demostró en el ataque efectuado.

4.11.1 **Seguridad Física**

El equipo desarrollado, permite bloquear y mostrar alertas tempranas para evitar posibles fraudes con el fin de cumplir las características que ofrece contactless.

Sin embargo, el prototipo no es viable por su robustez y dificultad para ser transportado, pero puede ser modificado y adaptado a la necesidad de manera estética.

Se puede considerar que existen algunas ventajas y desventajas del prototipo implementado que nombraremos a continuación:

Ventajas

a. El usuario protege sus datos e información.
b. Bloquea señales que puedan captar o distorsionar al momento de un ataque.
c. Permite alertar al usuario en caso de requerirlo
d. No permite manipular al atacante de manera física la tarjeta contactless.
e. Permite informar de manera oportuna a la institución financiera el ataque intentado.
f. Permite aprovechar de mejor manera la tarjeta contactless.

Desventajas

a. No es de fácil de portar.
b. La batería tiene que ser recargada al día.
c. El equipo puede ser destruido en caso de robo para la obtención de los datos.
d. El equipo debe estar conectado a internet.

4.11.2 **Tiempos detectados**

Los tiempos detectados para él envió de información fue visto de manera local hacia el aplicativo y de manera institucional, donde se validó de manera general los tiempos en confirmar la transacción.

Estos tiempos son solo pruebas en ambientes controlados, por lo que en un ambiente de producción no serían los mismos, aunque nos dan una idea del funcionamiento de todo el sistema.

En la figura 38, se detalla el tiempo y los flujos detectados como protocolos que se utilizaron en este tipo de comunicación:

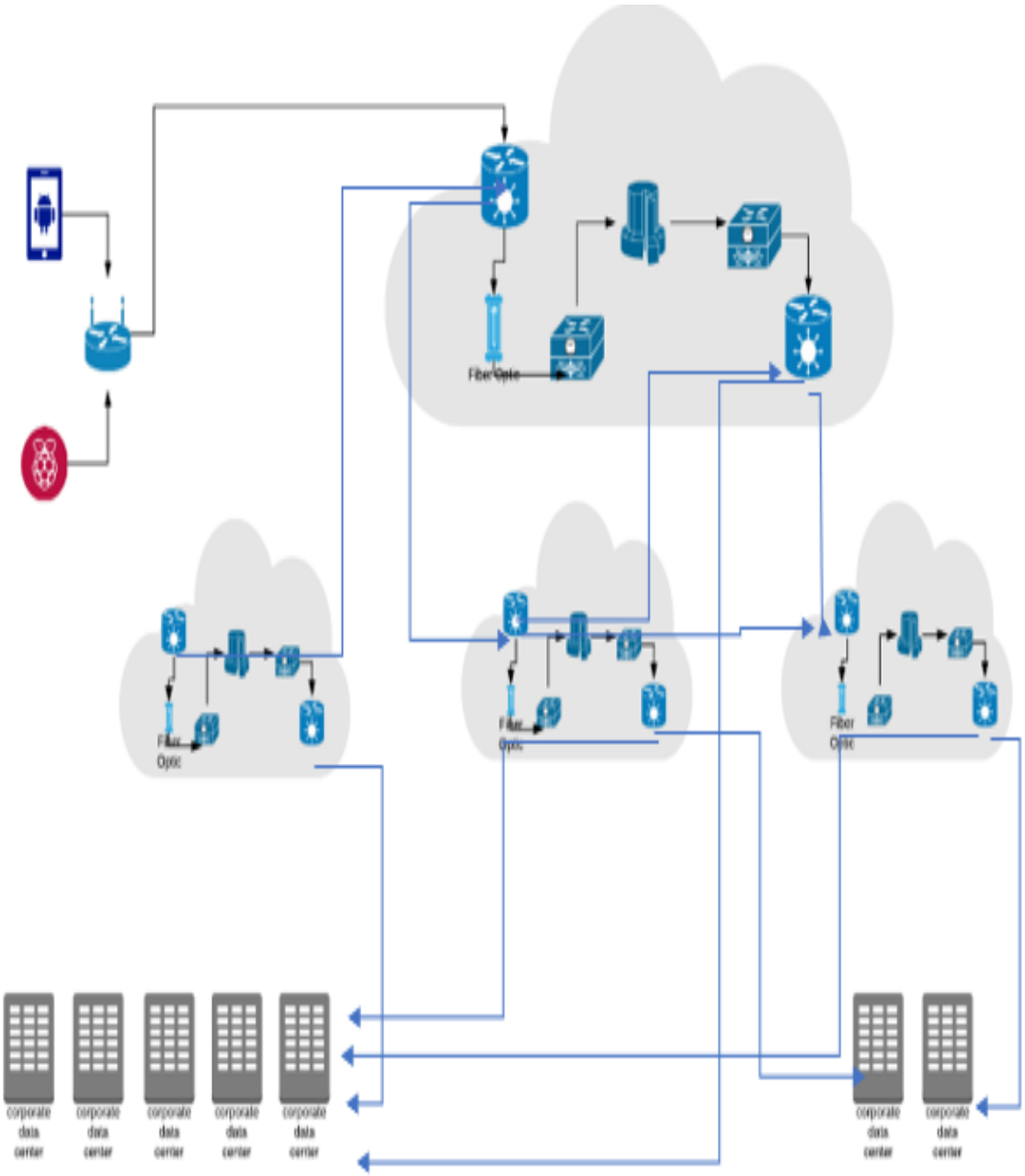

Figura 38. Distribución lógica del proveedor de Internet.

Adoptado de (ProCredit, 2019)

El equipo puesto en producción produjo los siguientes valores que son indicados en la figura 39:

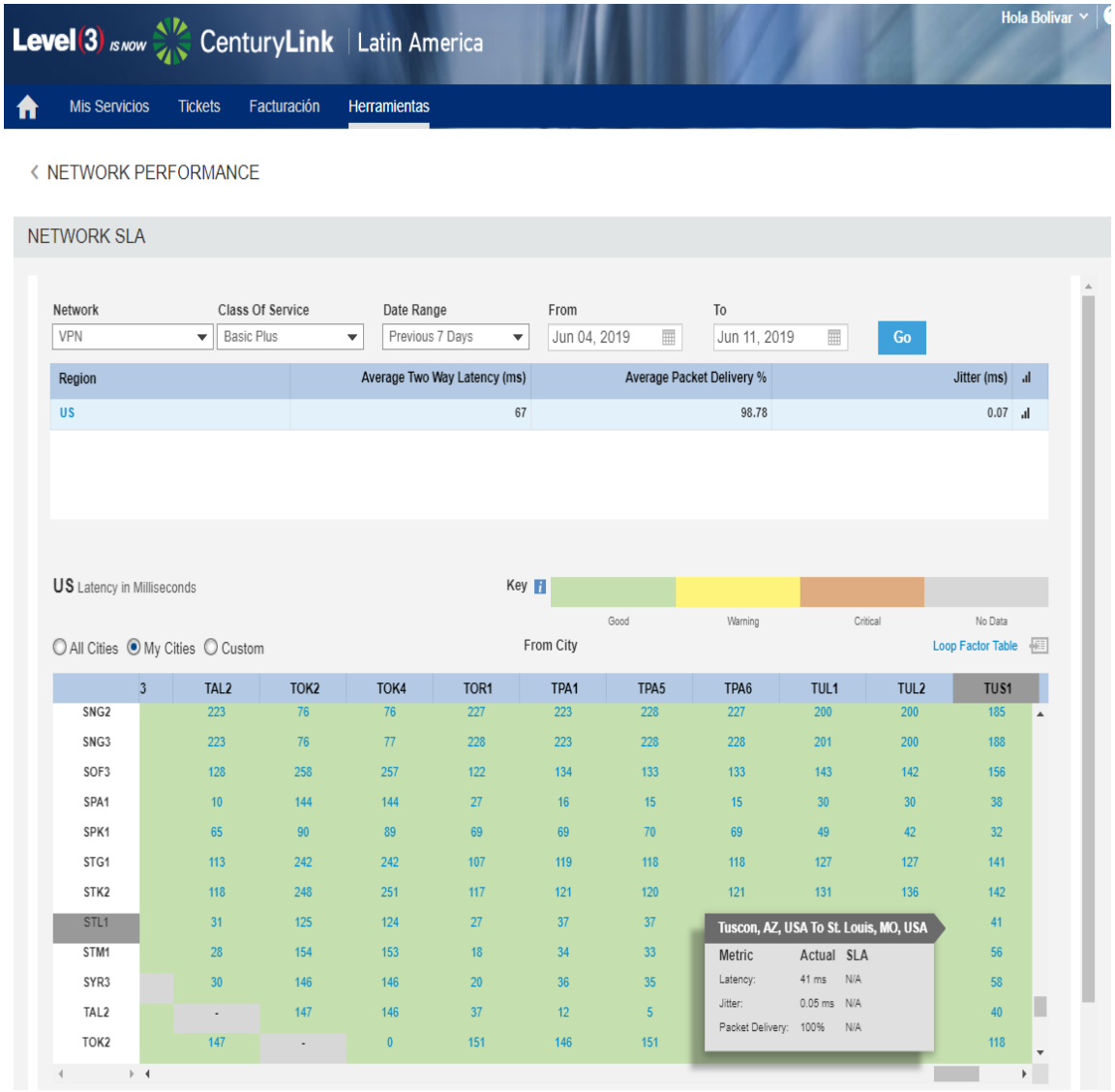

Figura 39. Centurylink provider, Network Performance

Adoptado de (ProCredit, 2019)

El tiempo final por el enlace dedicado fue de 41 ms con un jitter de 0.05 ms

Este paquete fue enviado desde la matriz hacia el servidor principal en Alemania.

Como se puede visualizar en la figura 40, los protocolos que más predominan es ESP y UDP en la transmisión de la prueba.

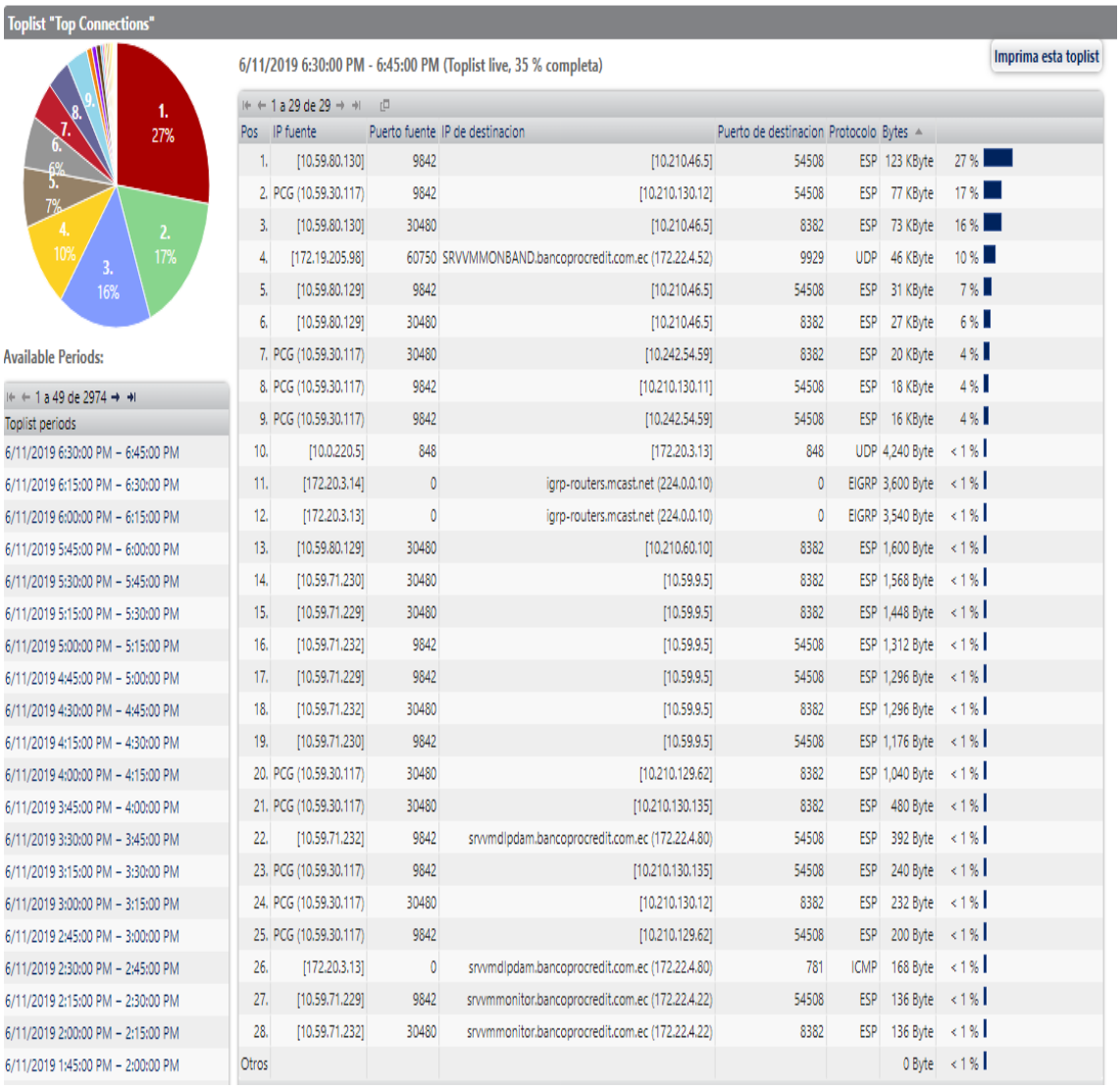

Figura 40. PRGT, Network Performance.

Adoptado de (ProCredit, 2019)

Carga de seguridad encapsulada (ESP):128 Kbyte utilizando el 27% de la red de prueba.

UDP:46 Kbyte utilizando un 10% de la red destinada para la prueba.

Como se puede visualizar en la figura 41, el tiempo de conexión entre dispositivo móvil y Raspberry tiene un tiempo aproximado de 6.57 ms.

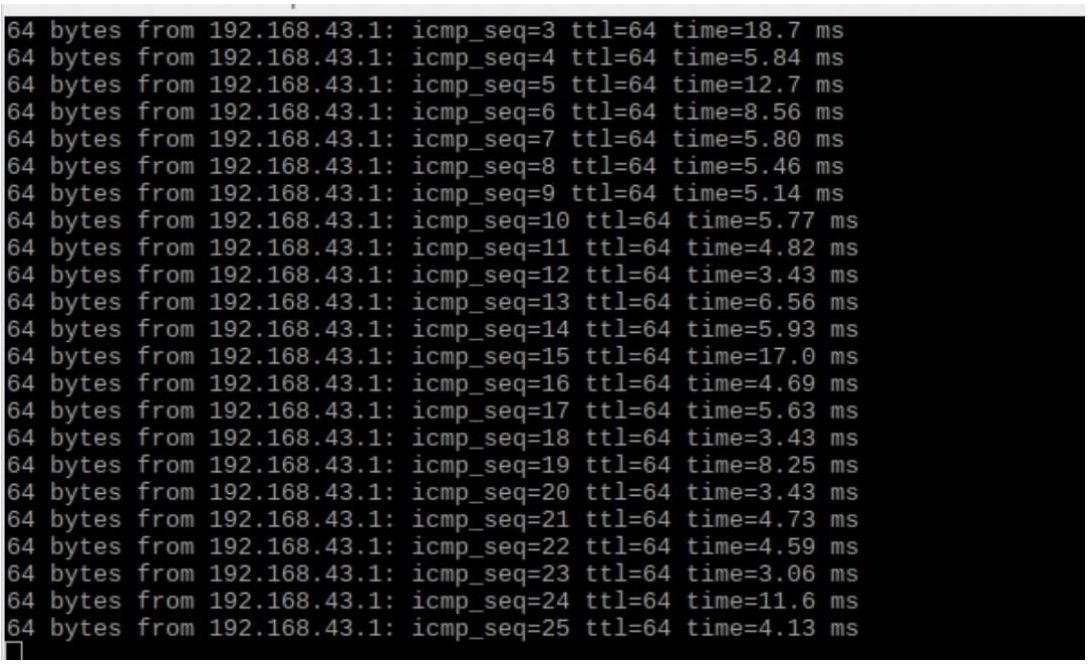

Figura 41. Raspberry consola.

5. CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

El estándar ISO/IEC 14443 (Estándar de tarjetas para identificación y pagos por proximidad) para tarjetas contactless muestra una grave falencia al momento de proteger los logs internos de la tarjeta que, si bien podrían estar enmascarados, no deberían ser mostrados bajo ninguna circunstancia.

Los equipos y las tarjetas con las que se realizaron las pruebas de lectura, se logra determinar ciertas fallas en la seguridad específicamente de las tarjetas, lo que permite alertar a las entidades que presentan estos inconvenientes con el objeto de solventar una mayor seguridad para el usuario.

El número de operaciones en transacciones de compra está delimitado por tiempo, y de acuerdo con la entidad bancaria que pertenezca la tarjeta, estos tiempos son transparentes para el usuario, permitiendo así mitigar posibles robos en sitios públicos, pero a pesar de esto como ya se comprobó hay que tener ciertas precauciones adicionales, ya que nada es infalible.

El sistema desarrollado es un prototipo que podría ser miniaturizado en un futuro para así servir de alerta en caso de hurto de la tarjeta de crédito y también como detección temprana de fraude.

Varios estudios relacionados, demuestran que debe existir un ambiente propicio y adecuado para proceder con el ataque, en específico, la escucha oculta puede ser efectuada en presencia del usuario sin que este mismo se percate del fraude, como se demostró en el ataque efectuado.

De las pruebas realizadas con los equipos de escucha se logra determinar que no todos los equipos con tecnología NFC tienen un cifrado en el tráfico con sus transferencias en texto plano, lo que provoca inseguridad para el usuario y la posibilidad de que la información transmitida pueda ser interceptada.

5.2 **Recomendaciones**

La autenticación de dos vías es algo que se debe activar para que toda transacción sea más segura y llegue cualquier notificación así será más difícil para cualquier persona no autorizada obtener la información y realizar compras.

El uso del NFC no solo se limita a pagos de productos, también es muy útil para transferencia de información entre equipos que incorporen esta tecnología, pero lo que se recomienda es una socialización masiva a todo tipo de usuario.

Analizar a profundidad los estándares internacionales que regulan la emisión de tarjetas, para prevenir al usuario final de posibles ataques.

Es importante que cada entidad bancaria invierta en áreas destinadas al monitoreo y análisis de datos de sus clientes para así prevenir en segunda instancia los fallos físicos.

Las nuevas generaciones y tecnologías permiten que la era de la tecnología NFC sea trasportada al dispositivo móvil inteligente, por lo tanto, una manera óptima sería el virtualizar las tarjetas para evitar el contacto físico y repotenciar las prestaciones que brinda NFC, es decir, una billetera electrónica.

REFERENCIAS

Aidacentre (s.f.) Componentes de un RFID, Recuperado el 15 de Agosto de 2016 de: http://www.aidacentre.com/rfid_002.php

Bbva. (s.f.). Nuevo estudio expone la vulnerabilidad de las tarjetas de débito y crédito. Recuperado el 18 Noviembre de 2017 de: https://www.bbva.com/es/tecnologia-contactless-pago-contacto/

Bruce eckel (2004), Piensa en java, ed. Prentice hall, 4ta edición

C. Busold, a. Taha, c. Wachsmann, a. Dmitrienko, h. Seudié, m. Sobhani, and a.-r. Sadeghi, (2013) Smart keys for cyber-cars: secure smartphone-based nfcenabled car immobilizer, in proceedings of the third acm conference on data and application security and privacy, pp. 233–242

Ccn-cert, españa. (s.f.) "informe de amenazas ccn-cert ia-05/16 comunicación de campo cercano (near field communication – nfc). Vulnerabilidades"

DLOGIC (2019) uFR para clonacion de tarjetas NFC, Recuperado el 20 de Octubre de 2018 de: https://www.d-logic.net/nfc-rfid-reader-sdk/products/ufr-classic/

D. Oswald, t. Kasper and c. Paar. (2011) Side-channel analysis of cryptographic rfids with analog demodulation. Springer lncs. In proceedings of rfidsec.

Emv. Communication (2007) contactless specifications for payment systems. Version 2.0, Recuperado el 20 de Febrero de 2007 de: http://www.emvco.com/

Erl thomas,(2004) Service-oriented architecture, prentice hall, 1era edición.

ETEKJOY (2019) ETEKJOY para clonacion de tarjetas NFC Y RFID, Recuperado el 21 de Marzo de 2019 de: https://guatemaladigital.com/Accesorios+-+teclados+de+control+de+acceso/ETEKJOY+Handheld+10-Frequency+RFID+NFC+Card+Reader+Writer+Copier+Duplicator+Programmer+for+ID+IC+Cards+w%2F+5X+125kHz+Cards%2C+5X+125kHz+k

eyfobs+%26+5X+13.56MHz+UID+Key+fobs/Producto.aspx?Codigo=480 5915

Finanzaspersonales. (s.f.). Nuevo estudio expone la vulnerabilidad de las tarjetas de débito y crédito. Recuperado el 12 Septiembre de 2012 de: https://www.finanzaspersonales.co/ahorro-e-inversion/articulo/nuevo-estudio-expone-vulnerabilidad-tarjetas-debito-credito/47101

Francis, l.; hancke, g.; mayes, k. Y markantonakis, k. (2011). Practical relay attack on contactless transactions by using nfc mobile phones. Recuperado el: 23 de Diciembre de 2015 de: http://eprint.iacr.org/2011/618.pdf

G. P. Hancke. (2010) Practical eavesdropping and skimming attacks on high-frequency rfid tokens. Journal of computer security - 2010 workshop on rfid security (rfidsec'10 asia), volume 19, issue 2 (april 2011), 259-288, 2011

Garfinkel, s.l., juels, a., pappu, r., rfid (2005) privacy: an overview of problems and proposed solutions,& security and privacy magazine, ieee volume 3(3):34-43.

Hid global (s.f.) Access control solutions. Recuperado el 15 de Agosto de 2019 de: http://www.hidglobal.com/main/id-cards/iclass-standard-credentials/

Igoe t., coleman d jepson b. (2014). Beginning nfc: near field communication with arduino, android, and phonegap.o'reilly. United states of america. Isbn: 978-1-449-37206-4.

International civil aviation organization (icao). (2005) Document 9303 machine readable travel documents (mrtd). Part 1: machine readable passports.

Contactless (2019) Iso/iec 15693. Identification cards – contactless integrated circuit cards – vicinity cards. Recuperado el 9 de junio de 2019 de: http://www.iso.org/

Card (2005) Iso/iec 7501 Identification cards - machine readable travel documents, Recuperado el 11 de enero de 2019 de: http://www.iso.org/

J. Axelson. (1998) Serial port complete: programming and circuits for rs-232 and rs-485 links and networks"; madison, wi:lakeview research, 1998.14

Javier eguíluz pérez,(s.f.) introducción a ajax, tomado el 12 de Febrero de 2019 de: www.librosweb.es

Juels, d. Molnar and d. Wagner. (2005). Security and privacy issues in e-passports. In proceedings of the first international conference on security and privacy for emerging areas in communications networks (securecomm '05). Ieee computer society, washington, dc, usa, 74-88.

Kirschenbaum and a. Wool.(2006) How to build a low-cost, extended-range rfid skimmer. In proceedings of 15th usenix security symposium, pp 43–57

Landt Jeremy (2019) Shoruds of Time, The History of RFID an AIM Publication Tomado el 15 de Marzo del 2019 de: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&cad=rja&uact=8&ved=2ahUKEwjG7surmPTjAhXE1VkKHfOeCewQFjACegQIABAC&url=https%3A%2F%2Fwww.transcore.com%2Fwp-content%2Fuploads%2F2017%2F01%2FHistory-of-RFID-White-Paper.pdf&usg=AOvVaw1z9khT41JkiJyCqNglJeZz

Lesas & Miranda (s.f.) NFC Secure Applications on Top of Multiple Secure Elements, Recuperado el 15 de Marzo de 2019 de: https://www.semanticscholar.org/paper/WOLF%3A-a-Research-Platform-to-Write-NFC-Secure-on-of-Lesas-Miranda/48ea57779e4b01d106295634e0e571f537c8b331

M. Engelhardt, f. Pfeiffer, k. Finkenzeller, y e. Biebl. (2013) Extending iso/iec 14443 type a eavesdropping range using higher harmonics In proceedings of 2013 european conference on smart objects, systems and technologies (smartsystech), pp. 1–8. Ieee, 2013

Manual ,learn (s.f.) how bpel process manager enables soa

Mastercard (s.f.) Paypasstm, Recuperado el 10 de Abril del 2019 de: http://www.paypass.com/

Matthias hertel, (2007) Aspects of ajax, version 1.2 Recuperado Mayo 2007 de: https://api.jquery.com/category/version/1.2/

Nfc forum. (2014) Tag type technical specifications. Recuperado el 12 de Mayo de 2019 de: http://nfcforum.org/our-work/specifications-and-application-documents/specifications/tagtype-technical-specifications/

Octopus (s.f.) Octopus cards limited, hong kong. Recuperado el 20 de Mayo de 2019 de: http://www.octopus.com.hk/

Patrick j. Sweeney (2005) Rfid for dummies, wiley publishing, inc 2005

Philipose, m. Smith, j.r. jiang, b. Mamishev, a. Sumit roy sundara-rajan, k. (2005) Battery-free wireless identification and sensing, pervasive computing, ieee, volumen 4(1): 37-45, marzo 2005.

Pourghomi, p., & ghinea, g. (2013). A proposed nfc payment application. Arxiv preprint arxiv:1312.2828.

Prentice (2004) Architecture best practices 1era Edition

R. Weinstein (2005) Rfid: a technical overview and its application to the enterprise,& it professional,volumen 7(3): 27-33, junio 2005.

Ramos-de-luna, i., montoro-ríos, f., & liébana-cabanillas, f. (2016). Determinants of the intention to use nfc technology as a payment system: an acceptance model approach. Information systems and e-business management, 14(2), 293-314.

S.c. bono, m. Green, a. Stubblefield, a. Juels, a.d. rubin and m. Szydlo.(2005) Security analysis of a cryptographically-enabled rfid device. Proceedings of usenix security symposium, 2005.

Seeburger (s.f.) Tarjeta RFID Funcionando, Recuperado el 8 de mayo de 2019 de: http://www.seeburger.info/

Syed a., mohammad i. (2016). Rfid handbook: applications, technology, security, and privacy. Taylor & francis group. United states of america. Isbn-13: 978-1-4200-5499-6

T.s. heydt-benjamin, d.v. bailey, k. Fu, a. Juels and t. Ohare. (2007). Vulnerabilities in first-generation rfid-enabled credit cards. In proceedings of financial cryptography and data security, pp. 1–22, february, 2007.

Tan, g. W. H., ooi, k. B., chong, s. C., & hew, t. S. (2014). Nfc mobile credit card: the next frontier of mobile payment?. Telematics a nd informatics, 31(2), 292-307.

Technical reference (2002) texas instruments tag-it hf-i transponder inlay extended commands and options.

V. Daniel hunt, albert puglia, mike puglia,(2007) Rfid a guide to radio frequency identification. Ed. Wiley 2007.

Visa (s.f.) Paywavetm. Recuperado el 6 de Junio de 2019O de: http://www.visaeurope.com/en/cardholders/visa_paywave.aspx.

Yassine Naija. (2019), Secured Digital Architectures for Low Cost Full-fledged HF RFID Tags Recuperado el 2 de Febrero de 2019 de: Https://www.researchgate.net/profile/yassine_naija/publication/330545779_secured_digital_architectures_for_low_cost_full-fledged_hf_rfid_tags/links/5c477f36299bf12be3dc651c/secured-digital-architectures-for-low-cost-full-fledged-hf-rfid-tags.pdf

ANEXOS

CÓDIGO EN RASPBERRY

```
# Requires Adafruit_Python_PN532

import binascii

import socket

import time

import signal

import sys

import Adafruit_PN532 as PN532

from gpiozero import Button

#cliente mqtt comunicacion con la app

import paho.mqtt.client as mqtt

import paho.mqtt.publish as publish

#registro bdd

import sqlite3

from datetime import datetime

#notificaciones push

from pyfcm import FCMNotification

def sistema():

    if switch.value == 1:

msg_push="Sistema activado manualmente"

        print(msg_push)

        print("Ingrese una tarjeta...")

    else:

        msg_push="Sistema desactivado manualmente"
```

```
        print(msg_push)


    enviarNotificacion(msg_push)
    registrar_bdd(msg_push)


def tarjeta():
    global activacion_remota
if switch.value == 1 or activacion_remota:
if sp.value == 0:
            msg_push="Se ingreso una tarjeta"
            print(msg_push)
            enviarNotificacion(msg_push)
            registrar_bdd(msg_push)
            ##enganche nfc
            uid_tarjeta = None
            tarjeta_enganchada = False
while (not tarjeta_enganchada) and (switch.value==1 or activacion_remota) and
sp.value==0:
                uid_tarjeta = pn532.read_passive_target()
                if uid_tarjeta is None:
time.sleep(0.5)
                    print("Buscando tarjeta")
                else:
                    tarjeta_enganchada=True
```

```
                msg_push="Tarjeta detectada y enganchada"
                print(msg_push)
                enviarNotificacion(msg_push)
                registrar_bdd(msg_push)
            uid=uid_tarjeta
while (switch.value==1 or activacion_remota) and tarjeta_enganchada and sp.value==0:
                uid_tarjeta = pn532.read_passive_target()
if uid_tarjeta!=uid:
                uid=uid_tarjeta
if uid_tarjeta is None:
msg_push="Se bloqueo un intento de lectura de la tarjeta"
print(msg_push)
                    registrar_bdd(msg_push)
                    enviarNotificacion(msg_push)
                else:
                    if not (uid_tarjeta is None):
time.sleep(0.1)
                        #print('Card UID 0x{0}'.format(binascii.hexlify(uid_tarjeta)))
time.sleep(0.5)
        else:
            msg_push="Se retiro la tarjeta"
            print(msg_push)
            enviarNotificacion(msg_push)
```

```
            registrar_bdd(msg_push)


#define si el sistema esta activado o no
switch = Button(20)
switch.when_pressed = sistema
switch.when_released = sistema


##define si se encuentra o no la tarjeta
sp = Button(21)
sp.when_pressed = tarjeta
sp.when_released = tarjeta
activacion_remota=False
##CONFIGURACION MQTT
def on_connect(client, userdata, flags, rc):
    print("Connected with result code "+str(rc))
    client.subscribe("APP")


def on_message(client, userdata, msg):
global activacion_remota
print (msg.topic)
print (msg.payload)


    if msg.topic=="APP":
```

```
        if str(msg.payload)=="activar_sistema":
activacion_remota=True
            client.publish("RPI","sistema_activado",2)
            registrar_bdd("Bloqueo activado desde la app")

        elif str(msg.payload)=="desactivar_sistema":
            activacion_remota=False
            client.publish("RPI","sistema_desactivado",2)
            registrar_bdd("Boqueo desactivado desde la app")

elif str(msg.payload)=="historial":
            client.publish("RPI_h","inicio",2)
            sincronizar_historial()
            client.publish("RPI_h","fin_sincronizacion",2)

def on_publish(mosq, obj, mid):
    print("mid: " + str(mid))

client = mqtt.Client()
client.username_pw_set("ixezgnub","54CPi0Dq0vWG")
client.on_connect = on_connect
client.on_message = on_message
client.connect("m11.cloudmqtt.com", 14228, 60)
```

```
client.loop_start()


##base de datos

#TABLA REGISTRO Historico

conn = sqlite3.connect('/home/pi/rpi-examples/PN532/python/REGISTRO_NFC.db')

cursor = conn.cursor()

sql = "CREATE TABLE IF NOT EXISTS historico (ID integer PRIMARY KEY, registro text, fecha text)"

cursor.execute(sql)

conn.commit()

conn.close()


def registrar_bdd(msg):

    f = datetime.now()

    f = str(f)

    f = f[0:19]

    conn = sqlite3.connect('/home/pi/rpi-examples/PN532/python/REGISTRO_NFC.db')

cursor = conn.cursor()

    cursor.execute("INSERT INTO historico (registro, fecha) VALUES ('"+msg+"','"+f+"')")

    conn.commit()

    conn.close()

#print ("Registro guardado")
```

```
def sincronizar_historial():
conn = sqlite3.connect('/home/pi/rpi-examples/PN532/python/REGISTRO_NFC.db')
    cursor = conn.cursor()

    sql="SELECT * FROM historico"
    cursor.execute(sql)
    results = cursor.fetchall()
    num_filas = len(results)
    cont=0

    while cont<num_filas:
        client.publish("RPI_h",str(results[cont][1])+"//"+str(results[cont][2]),2)
time.sleep(0.1)
        cont+=1

    conn.commit()
conn.close()

#funcion para enviar notificaciones
def enviarNotificacion(msg):
#time.sleep(0.1)
try:
```

```
        push_service =
FCMNotification(api_key="AIzaSyBu7aYeIBIjNdHpsuKWF_8T3OLiuivVM60")
        result = push_service.notify_topic_subscribers(topic_name="app_test",
message_body=msg)
    except Exception as e:
        print(e)
        pass


# PN532 configuration for a Raspberry Pi GPIO:
# GPIO 18, pin 12
CS   = 18
# GPIO 23, pin 16
MOSI = 23
# GPIO 24, pin 18
MISO = 24
# GPIO 25, pin 22
SCLK = 25


# Configure the key to use for writing to the MiFare card.  You probably don't
# need to change this from the default below unless you know your card has a
# different key associated with it.
CARD_KEY = [0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF]
# Number of seconds to delay after reading data.
DELAY = 0.5
```

```
# Prefix, aka header from the card

HEADER = b'BG'

def close(signal, frame):

        sys.exit(0)


signal.signal(signal.SIGINT, close)


# Create and initialize an instance of the PN532 class

pn532 = PN532.PN532(cs=CS, sclk=SCLK, mosi=MOSI, miso=MISO)

pn532.begin()

pn532.SAM_configuration()


print('PN532 NFC RFID 13.56MHz Card Reader')


while True:

    time.sleep(0.5)
```

CÓDIGO EN ANDROID

Historico

```
package com.example.fredd.app_test;

import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
```

```
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.ListView;

import org.eclipse.paho.client.mqttv3.MqttException;

import java.util.ArrayList;

public class Historico extends AppCompatActivity {

    ListView historico;
    public ArrayList<Entidad> listItems = new ArrayList<>();
    private Adaptador adaptador;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_historico);
        historico=(ListView) findViewById(R.id.Lv_historico);
        adaptador = new Adaptador(this,GetArrayItems());
        historico.setAdapter(adaptador);
    }

    @Override
    public void onBackPressed() {
        Intent i = new Intent(Historico.this, MainActivity.class);
        startActivity(i);
        finish();
    }

    @Override
    public void onPause(){
```

```
        super.onPause();
        finishAffinity();
    }



    private ArrayList<Entidad> GetArrayItems(){
        listItems.clear();
        SQLiteDatabase myDB = openOrCreateDatabase("my.db",
MODE_PRIVATE, null);
        Cursor myCursor = myDB.rawQuery("select * from historico", null);
        while(myCursor.moveToNext()) {
            listItems.add(new
Entidad(myCursor.getString(1),myCursor.getString(2)));
        }
        myCursor.close();
        myDB.close();
        return listItems;
    }
}
```

Login

```
package com.example.fredd.app_test;

import android.content.Intent;
import android.support.design.widget.TextInputLayout;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.view.Window;
import android.view.WindowManager;
import android.widget.Button;
```

```
import android.widget.Toast;

public class Login extends AppCompatActivity {
    Button btn_ingresar;

    private TextInputLayout txtInputUsuario,txtInputContrasena;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        requestWindowFeature(Window.FEATURE_NO_TITLE);

getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
WindowManager.LayoutParams.FLAG_FULLSCREEN);
        setContentView(R.layout.activity_login);

        txtInputUsuario = findViewById(R.id.txt_input_usuario);
        txtInputContrasena = findViewById(R.id.txt_input_contr);

        btn_ingresar=(Button) findViewById(R.id.Btn_Iniciar_Sesion);
        btn_ingresar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
              /* Intent i = new Intent(Login.this, MainActivity.class);
               startActivity(i);*/
               if(txtInputUsuario.getEditText().getText().toString().equals("admin")
&& txtInputContrasena.getEditText().getText().toString().equals("1234") ){
                   Intent i = new Intent(Login.this, MainActivity.class);
                   startActivity(i);
               }
               else {
                   Toast.makeText(Login.this, "Datos incorrectos",
```

```
Toast.LENGTH_LONG).show();
                }
            }
        });
    }
}
```

Main Activity

```
package com.example.fredd.app_test;

import android.content.ContentValues;
import android.content.Intent;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.os.Handler;
import android.support.annotation.NonNull;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.CompoundButton;
import android.widget.Switch;
import android.widget.TextView;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.messaging.FirebaseMessaging;

import org.eclipse.paho.client.mqttv3.IMqttDeliveryToken;
import org.eclipse.paho.client.mqttv3.MqttCallbackExtended;
import org.eclipse.paho.client.mqttv3.MqttException;
import org.eclipse.paho.client.mqttv3.MqttMessage;
```

```
public class MainActivity extends AppCompatActivity {
    Button btnTarjeta;
    Switch swBloqueo;
    TextView txtEstado;

    MqttHelper mqttHelper;
    Boolean s_historico;

    SQLiteDatabase myDB;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        myDB = openOrCreateDatabase("my.db", MODE_PRIVATE, null);
        myDB.execSQL("CREATE TABLE IF NOT EXISTS historico (ID INTEGER
PRIMARY KEY AUTOINCREMENT, registro TEXT, fecha TEXT)");



        txtEstado=(TextView) findViewById(R.id.txt_estadosw);
        swBloqueo = (Switch) findViewById(R.id.sw_bloqueo);
        swBloqueo.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(CompoundButton buttonView, boolean
isChecked) {
                String msg="";
                if(isChecked){
                    txtEstado.setText("NC");
                    msg="activar_sistema";
```

```
                swBloqueo.setEnabled(false);
            }
            else {
                txtEstado.setText("NC");
                msg="desactivar_sistema";
                swBloqueo.setEnabled(false);
            }
            try {
                mqttHelper.mqttAndroidClient.publish("APP", (msg).getBytes(), 2,
false);
            } catch (MqttException e) {
                e.printStackTrace();
            }
        }
    });
    btnTarjeta=(Button) findViewById(R.id.btn_Tarjeta);
    btnTarjeta.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            s_historico=false;
            try {
                mqttHelper.mqttAndroidClient.publish("APP",
("historial").getBytes(), 2, false);
            } catch (MqttException e) {
                e.printStackTrace();
            }
            Intent i = new Intent(MainActivity.this, Historico.class);
            startActivity(i);
        }
    });

    //inicia el servicio mqtt
```

```
        startMqtt();

        //suscripcion al topico "app_test" para recibir las notificaciones desde la rpi
        FirebaseMessaging.getInstance().subscribeToTopic("app_test")
                .addOnCompleteListener(new OnCompleteListener<Void>() {
                    @Override
                    public void onComplete(@NonNull Task<Void> task) {
                        String msg = "";
                        if (task.isSuccessful()) {

                            msg = "Suscripción completa";
                        }
                        else{
                            msg = "Suscripción fallida, vuelva a intentar";
                        }
                        Log.d("Firebase", msg);
                        //Toast.makeText(Nuevo_equipo.this, msg,
Toast.LENGTH_SHORT).show();
                    }
                });
    }

    @Override
    public void onPause(){
        super.onPause();
        finishAffinity();
    }

    @Override
    public void onBackPressed() {
        Intent i = new Intent(MainActivity.this, Login.class);
        startActivity(i);
```

```
        finishAffinity();
    }

    private void startMqtt() {
        mqttHelper = new MqttHelper(getApplicationContext());
        mqttHelper.mqttAndroidClient.setCallback(new MqttCallbackExtended() {
            @Override
            public void connectComplete(boolean b, String s) {
                Log.w("Debug","Connected");
            }
            @Override
            public void connectionLost(Throwable throwable) {
            }
            @Override
            public void messageArrived(String topic, MqttMessage mqttMessage)
throws Exception {
                Log.w("Debug", mqttMessage.toString());
                Log.w("Debug", topic);
                if(topic.equals("RPI")){
                    if(mqttMessage.toString().equals("sistema_activado")){
                        txtEstado.setText("Desbloquear");
                        swBloqueo.setEnabled(true);
                    }
                    if(mqttMessage.toString().equals("sistema_desactivado")){
                        txtEstado.setText("Bloquear");
                        swBloqueo.setEnabled(true);
                    }
                }
                if(topic.equals("RPI_h")){
                    if(mqttMessage.toString().equals("inicio")){
                        myDB.delete("historico",null,null);
                    }
```

```
                if(mqttMessage.toString().equals("fin_sincronizacion")){
                    myDB.close();
                    mqttHelper.mqttAndroidClient.disconnect();
                    Intent i = new Intent(MainActivity.this, Historico.class);
                    startActivity(i);
                    finish();
                }
                else{
                    int ind= mqttMessage.toString().indexOf("//");
                    ContentValues row = new ContentValues();
                    row.put("registro", mqttMessage.toString().substring(0,ind));
                    row.put("fecha",
mqttMessage.toString().substring(ind+2,mqttMessage.toString().length()-1));
                    myDB.insert("historico", null, row);
                }
            }

        }

        @Override
        public void deliveryComplete(IMqttDeliveryToken iMqttDeliveryToken) {
        }
    });


  }

}
```

MqttHelper

```
package com.example.fredd.app_test;


import android.content.Context;
import android.content.SharedPreferences;
import android.preference.PreferenceManager;
import android.util.Log;

import org.eclipse.paho.android.service.MqttAndroidClient;
import org.eclipse.paho.client.mqttv3.DisconnectedBufferOptions;
import org.eclipse.paho.client.mqttv3.IMqttActionListener;
import org.eclipse.paho.client.mqttv3.IMqttDeliveryToken;
import org.eclipse.paho.client.mqttv3.IMqttToken;
import org.eclipse.paho.client.mqttv3.MqttCallbackExtended;
import org.eclipse.paho.client.mqttv3.MqttConnectOptions;
import org.eclipse.paho.client.mqttv3.MqttException;
import org.eclipse.paho.client.mqttv3.MqttMessage;


public class MqttHelper {
    public MqttAndroidClient mqttAndroidClient;

    public String serverUri = "tcp://m11.cloudmqtt.com:14228";
    final String clientId = "APP_ADMIN";
    final String subscriptionTopic1 = "RPI";
    final String subscriptionTopic2 = "RPI_h";
    final String username = "ixezgnub";
    final String password = "54CPi0Dq0vWG";

    public MqttHelper(Context context){
        final SharedPreferences myPreferences =
PreferenceManager.getDefaultSharedPreferences(context);
```

```
        final SharedPreferences.Editor myEditor = myPreferences.edit();
        String sBroker = myPreferences.getString("SBroker", "unknown");
        if(!(sBroker.equals("unknown"))){
            serverUri = sBroker;
        }

        mqttAndroidClient = new MqttAndroidClient(context, serverUri, clientId);
        mqttAndroidClient.setCallback(new MqttCallbackExtended() {
            @Override
            public void connectComplete(boolean b, String s) {
                Log.w("mqtt", s);
            }

            @Override
            public void connectionLost(Throwable throwable) {

            }

            @Override
            public void messageArrived(String topic, MqttMessage mqttMessage)
throws Exception {
                Log.w("Mqtt", mqttMessage.toString());
            }

            @Override
            public void deliveryComplete(IMqttDeliveryToken iMqttDeliveryToken) {

            }
        });
        connect();
    }
```

```
    public void setCallback(MqttCallbackExtended callback) {
        mqttAndroidClient.setCallback(callback);
    }

    private void connect(){
        MqttConnectOptions mqttConnectOptions = new MqttConnectOptions();
        mqttConnectOptions.setAutomaticReconnect(true);
        mqttConnectOptions.setCleanSession(false);
        mqttConnectOptions.setUserName(username);
        mqttConnectOptions.setPassword(password.toCharArray());

        try {

            mqttAndroidClient.connect(mqttConnectOptions, null, new
IMqttActionListener() {
                @Override
                public void onSuccess(IMqttToken asyncActionToken) {

                    DisconnectedBufferOptions disconnectedBufferOptions = new
DisconnectedBufferOptions();
                    disconnectedBufferOptions.setBufferEnabled(true);
                    disconnectedBufferOptions.setBufferSize(100);
                    disconnectedBufferOptions.setPersistBuffer(false);
                    disconnectedBufferOptions.setDeleteOldestMessages(false);
                    mqttAndroidClient.setBufferOpts(disconnectedBufferOptions);
                    subscribeToTopic();
                }

                @Override
                public void onFailure(IMqttToken asyncActionToken, Throwable
exception) {
                    Log.w("Mqtt", "Failed to connect to: " + serverUri +
```

```
exception.toString());
                }
            });


        } catch (MqttException ex){
            ex.printStackTrace();
        }
    }


    private void subscribeToTopic() {
        try {
            mqttAndroidClient.subscribe(subscriptionTopic1, 2, null, new
IMqttActionListener() {
                @Override
                public void onSuccess(IMqttToken asyncActionToken) {
                    Log.w("Mqtt","Subscribed!");
                }

                @Override
                public void onFailure(IMqttToken asyncActionToken, Throwable
exception) {
                    Log.w("Mqtt", "Subscribed fail!");
                }
            });

        } catch (MqttException ex) {
            System.err.println("Exception whilst subscribing");
            ex.printStackTrace();
        }
```

```
        try {
            mqttAndroidClient.subscribe(subscriptionTopic2, 2, null, new
IMqttActionListener() {
                @Override
                public void onSuccess(IMqttToken asyncActionToken) {
                    Log.w("Mqtt","Subscribed!");
                }

                @Override
                public void onFailure(IMqttToken asyncActionToken, Throwable
exception) {
                    Log.w("Mqtt", "Subscribed fail!");
                }
            });

        } catch (MqttException ex) {
            System.err.println("Exception whilst subscribing");
            ex.printStackTrace();
        }
    }
}
```

MyFirebaseMessagingService

```
package com.example.fredd.app_test;

import android.content.SharedPreferences;
import android.preference.PreferenceManager;
import android.util.Log;
import android.widget.Toast;

import com.google.firebase.messaging.FirebaseMessagingService;
import com.google.firebase.messaging.RemoteMessage;
```

```
public class MyFirebaseMessagingService extends FirebaseMessagingService
{
    private static final String TAG = "FCM Service";
    @Override
    public void onMessageReceived(RemoteMessage remoteMessage ) {
        // TODO: Handle FCM messages here.
        // If the application is in the foreground handle both data and notification
messages here.
        // Also if you intend on generating your own notifications as a result of a
received FCM
        // message, here is where that should be initiated.

        Log.d(TAG, "From: " + remoteMessage.getFrom());
        String topico= remoteMessage.getFrom().replaceAll("/topics/","");

        Log.d(TAG, "Topico: "+topico);
        Log.d(TAG, "Notification Message Body: " +
remoteMessage.getNotification().getBody());

        /*final SharedPreferences myPreferences =
PreferenceManager.getDefaultSharedPreferences(getApplicationContext());
        final SharedPreferences.Editor myEditor = myPreferences.edit();
        myEditor.putString("Notificacion", topico);
        myEditor.putString("Codigo", topico);
        myEditor.commit();
        Toast.makeText(getApplicationContext(),topico,
Toast.LENGTH_SHORT).show();*/
    }
}
```