ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

ESCUELA DE INGENIERÍA ELECTRÓNICA EN TELECOMUNICACIONES Y REDES

"DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO PARA PAGO DE SERVICIO DE TRANSPORTE PÚBLICO EN ESTACIONES A TRAVÉS DE UN TELÉFONO INTELIGENTE CON TECNOLOGÍA NFC"

TRABAJO DE TITULACIÓN

TIPO: PROPUESTA TECNOLÓGICA

Presentado para optar por el título de:

INGENIERO EN ELECTRÓNICA EN TELECOMUNICACIONES Y REDES

AUTOR: JAIME GABRIEL SUÁREZ RUIZ

TUTOR: ING. DIEGO VELOZ CHÉRREZ

Riobamba-Ecuador

2018

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

ESCUELA DE INGENIERÍA EN ELECTRÓNICA TELECOMUNICACIONES Y REDES

El Tribunal del Trabajo de Titulación certifica que: El trabajo de titulación: DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO PARA PAGO DE SERVICIO DE TRANSPORTE PÚBLICO EN ESTACIONES A TRAVÉS DE UN TELÉFONO INTELIGENTE CON TECNOLOGÍA NFC, de responsabilidad del Señor Jaime Gabriel Suárez Ruiz, ha sido minuciosamente revisado por los Miembros del Tribunal del Trabajo de Titulación, quedando autorizada su presentación.

ING. WASHINGTON LUNA
DECANO DE LA FACULTAD
DE INFORMÁTICA Y ELECTRÓNICA **……** ……..

ING. PATRICIO ROMERO
DIRECTOR DE LA ESCUELA DE
INGENIERÍA ELECTRÓNICA
TELECOMUNICACIONES Y REDES **……** ……..

ING. DIEGO VELOZ
DIRECTOR DEL TRABAJO
DE TITULACIÓN **……** ……..

ING. ALBERTO ARELLANO
MIEMBRO DE TRIBUNAL **……** ……..

Yo, Jaime Gabriel Suárez Ruiz soy responsable de las ideas, doctrinas y resultados expuestos en este trabajo de Titulación y el patrimonio intelectual del trabajo de Titulación pertenece a la ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

Jaime Gabriel Suárez Ruiz

AGRADECIMIENTO

A mis padres por todo el realizado a lo largo de mi vida, a los docentes por la formación profesional recibida y los consejos que me han guiado a la obtención de este objetivo en mi vida.

Jaime

TABLA DE CONTENIDOS

INDICE DE TABLAS

INDICE DE FIGURAS

ANEXOS

RESUMEN

El objetivo del presente trabajo de titulación fue de diseñar e implementar una plataforma de pagos de servicios de transporte público usando smartphones con soporte de NFC. Se estudiaron las arquitecturas que posee la tecnología NFC y las tramas del protocolo de comunicación para la creación de aplicaciones tanto de escritorio como móviles que operarán en smartphones compatibles con Android, las mismas que fueron desarrolladas en Java. De igual manera, se creó una base de datos en MySQL, alojada en un servidor, donde se guardarán todos los datos de usuarios, historiales y transacciones que realice la plataforma para control y administración del sistema. En cuanto al sistema de pagos, se evaluó la integridad y la confidencialidad de la comunicación para proteger la información personal de los usuarios y las transacciones estos realicen. Para proteger la comunicación se implementaron técnicas de encriptación con 3DES y SHA-1, sesiones de usuario, emparejamiento de aplicaciones para evitar intercepciones de datos y además se aprovechó la característica de proximidad de NFC que brinda una seguridad física al establecer una comunicación. Se realizaron ataques "Man in the middle" para determinar el impacto que tendría un atacante si lograse vulnerar el sistema. Los resultados de estas pruebas fueron satisfactorios, concluyéndose que la integridad y confidencialidad de los datos está protegida y que el esfuerzo para tratar de desencriptarlos por fuerza bruta tardaría incluso cientos de años si utiliza un ordenador promedio, mientras que si se utiliza un clúster de ordenadores o una supercomputadora disminuye el tiempo siendo el caso más crítico el de 7,7 segundos si se implementa 3DES solamente, mejorando significativamente con SHA-1. Se recomienda después de estas evaluaciones que se realicen segmentaciones de red y se establezcan redes privadas para las comunicaciones con el servidor para dificultar más aún cualquier intento de ataque.

Palabras clave: <TECNOLOGÍA Y CIENCIAS DE LA INGENIERÍA>, <TELECOMUNICACIONES>, <COMUNICACIÓN INALÁMBRICA (NFC)>, <APLICACIÓN ANDROID>, < ANALIZADOR DE PROTOCOLOS (SNIFFER)>, <ENCRIPTACIÓN>, <JAVA (LENGUAJE DE PROGRAMACIÓN)>, <ATAQUE INFORMÁTICO>.

SUMMARY

The aim of this certification work was to design and implement a payment platform for public transport services using smartphones with NFC support. The structural design that NFC technology possesses and the communication protocol frames were analyzed for the creation of both desktop and mobile applications that operate on Android compatible smartphones, the same ones that were developed in Java. In the same manner, a database was created in MySQL, hosted in a server, where all user data, history and transactions made in the platform for control and administration of the system will be saved. Regarding the payment system, the integrity and confidentiality of the communication was evaluated in terms of the protection of the personal information of the users and the transactions they perform. To protect communication, encryption techniques were implemented with 3DES and SHA-1, user sessions, application pairing to avoid data interceptions, and also took advantage of the proximity feature of NFC which provides physical security when establishing a communication. "Man in the middle" attacks were carried out to determine the impact an attacker would have if he succeeded in violating the system. The results of these tests were satisfactory, concluding that the integrity and confidentiality of the data is protected and that the effort to try to decrypt them by brute force would take even hundreds of years if you use an average computer, whereas if you use a cluster of computers or a supercomputer decreases the time being the most critical case 7.7 seconds if 3DES is implemented only, significantly improving with SHA-1. It is recommended after these evaluations that network segmentations be made, and private networks be established for communications with the server to further hinder any attack attempt.

Key Words: <TECHNOLOGY AND ENGINEERING SCIENCES> <TELECOMMUNICATIONS> <WIRELESS COMUNICATION (NFC)> < ANDROID APPLICATION > <PROTOCOL ANALYZER (SNIFFER).> < ENCRYPTION> <(PROGRAMMING LANGUAGE)> <INFORMATIC ATTACK>.

INTRODUCCIÓN

En la última década se ha desarrollado tecnologías que permitan que las comunicaciones converjan y se facilite el intercambio de datos, la tecnología NFC es ahora un componente vital en las comunicaciones ya que en la actualidad se encuentra en la mayoría de teléfonos móviles inteligentes además de otros dispositivos como electrodomésticos, computadores; además de que se está implementando esta tecnología en carnets de conducción, en posters inteligentes, DNI entre otros. Como aporte al desarrollo de esta tecnología, se propone realizar el diseño e implementación de un prototipo para pago de transporte público usando un teléfono inteligente y la tecnología NFC.

Esta tecnología ya es usada en muchos países desarrollados para realizar pagos como si se tratase de una tarjeta de débito ya que posee características y elementos que hace que estas transferencias sean seguras, por lo que se propone realizar el aplicativo móvil y el servidor donde se controlará cuentas, abonados entre otras variables, además de analizar a través de procedimientos de seguridad informática que tan vulnerable puede ser el sistema entero ante ataques informáticos.

A través de la investigación, se determinará la arquitectura y el funcionamiento de la tecnología NFC además de la seguridad que pueda tener la implementación de tecnologías desarrolladas para el propósito de pagos como lo es HCE, los cuáles beneficiarán a la realización de los aplicativos tanto móviles como de la aplicación de escritorio. Una vez realizados el diseño y la red en la cual funcionará esta plataforma se obtendrán resultados mediante ataques de hombre en el medio para probar la seguridad en el prototipo.

JUSTIFICACIÓN TEÓRICA

Las empresas de transporte público tienen en la actualidad problemas al momento de cobrar sus pasajes ya que no cuentan con un sistema inteligente que facilite esta labor y mejore su control, por tal razón se producen retrasos, desorden y descontento del usuario de este servicio.

Los actores involucrados en el transporte público reconocen las ventajas que presenta incorporar tecnología en el proceso de cobro de pasajes, las razones de esto son la agilidad del sistema y permitir una integración intermodal. Sin embargo, la necesidad de proteger la comunicación en esta plataforma es primordial, ya que tanto el usuario como las empresas de transporte todavía desconfían de la seguridad y robustez que pueden ofrecer estos sistemas de pago.

El sistema de pago electrónico mejora el expendio de pasajes, es rápido y cómodo para los usuarios de servicios de transporte público, además gracias al procesamiento de la información la gestión de las empresas mejoraría en gran porcentaje.

La presente investigación se enfoca en la seguridad al momento de realizar pagos con la tecnología NFC que poseen gran parte del mercado de teléfonos móviles inteligentes.

JUSTIFICACIÓN APLICATIVA

El diseño del prototipo se perfila en la aplicación de una tecnología inalámbrica de corto alcance que permite el intercambio de datos entre dos terminales a una corta distancia, el protocolo tiene mucha versatilidad y gran parte de esta se la debe a su corto alcance lo que la hace también segura, así como también es rápida al momento de gestionar la conexión lo que le permite ser ideal para transmitir pequeñas tramas de datos.

Los datos que son leídos por el lector NFC van a ser enviados a un servidor a través de una aplicación que va a ser diseñada, en dicho servidor se almacenará los datos de usuario y saldos, como también las llaves criptográficas del elemento seguro. Los certificados digitales también serán entregados por este servidor para establecer la conexión segura.

OBJETIVOS

OBJETIVO GENERAL

Diseñar e implementar un prototipo de pago de transporte público en estaciones usando un teléfono inteligente y un sensor NFC.

OBJETIVOS ESPECÍFICOS

- Investigar los conceptos básicos del protocolo NFC y sus funcionalidades.

- Diseñar e implementar un prototipo que permita realizar los pagos seguros desde el móvil a través de la tecnología NFC.

- Comprobar la seguridad que ofrece la tecnología NFC tanto para el usuario como para el operador de transporte.

CAPÍTULO 1

1 MARCO TEÓRICO

1.1 NFC (Comunicación de Campo Cercano)

1.1.1 Definición de la Tecnología NFC

NFC son las siglas de Near Field Communication que en español significa Comunicación de Campo Cercano. Funciona bajo el estándar RFID a una frecuencia de 13,56 MHz, esta frecuencia es libre y no tiene ninguna restricción legal (Albiñana, Cardona, & Piles, 2012,p.3).

Su desarrollo comenzó en el año 2002 cuando Philips y Sony intentaron conseguir un protocolo compatible con las tecnologías sin contacto existentes en ese momento, en el año siguiente este fue aprobado con el estándar ISO 18092, ya en el año 2004 se sumó a esta tecnología la compañía de teléfonos móviles Nokia con la cual crearon el NFC Forum consiguiendo que empresas como Google, Paypal, Visa entre otras se una al Foro y apoyen esta tecnología. (Universitat Politécnica de Valencia, 2012, http://histinf.blogs.upv.es/2012/11/21/nfc/)

Este foro de desarrollo tecnológico vio en la tecnología una oportunidad para el desarrollo y el futuro del comercio electrónico, aunque no es lo único que se desarrolló. Los principales objetivos que define el foro para esta tecnología son:

- Desarrollar estándares, especificaciones, interoperabilidad y arquitecturas para los dispositivos con esta tecnología.
- Desarrollo de dispositivos y pro ductos
- Asegurarse que los dispositivos NFC cumplan con los estándares y especificaciones del Foro.
- Inducción de esta tecnología a empresas y consumidores.

Esta tecnología en los últimos años se ha popularizado por lo que se encuentra en la mayoría de dispositivos móviles en especial en los teléfonos inteligentes. (Albiñana, Cardona, & Piles, 2012, p.4)

La tecnología NFC permite la interacción inalámbrica de corto alcance entre dispositivos que posean esta tecnología, el Foro NFC es el ente encargado de todos los aspectos en las especificaciones técnicas y actualizaciones de las mismas.

NFC al igual que el RFID ISO/IEC 14443 hace la comunicación por inducción de campo magnético, funcionando en la banda ISM (Industrial, Scientific and Medical) a una frecuencia de radio de 13,56 MHz y un ancho de banda de casi 2 MHz. Esta tecnología es de plataforma abierta y fue estandarizada en la ISO/IEC 18092 y ECMA-340, donde se especifican la codificación, esquemas de modulación, velocidad de transferencia y formato de la trama de interfaz RF de dispositivos NFC, esquemas de inicialización, condiciones que se requieren para el control de colisión de datos. Es necesario que los dispositivos NFC estén a no más de 10 cm. de distancia de separación para que el intercambio de datos entre estos no resulte afectado, esta transmisión se realiza a velocidades de 106, 212, 424 u 848 Kbits/s. (Veloz, 2010, pp.14-15)

1.1.2 Arquitectura de la tecnología NFC

1.1.2.1 Modos de Funcionamiento

La tecnología NFC puede funcionar de dos modos que son:

- Pasivo, en donde el dispositivo que inicia la comunicación genera campos electromagnéticos para que el otro dispositivo aprovechándose de la modulación se comunique con este y es posible la transferencia de datos.
- Activo, en donde los dos dispositivos generan campos electromagnéticos para comunicarse. (Albiñana, 2016,p.15)

En la Figura 1-1 se observa los modos de operación que funcionan los dispositivos NFC.

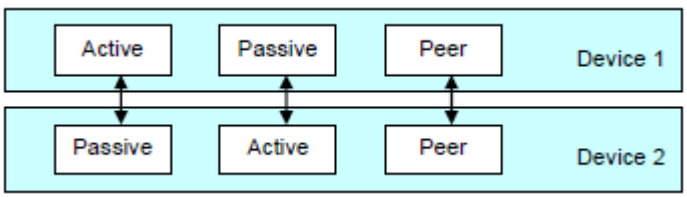


Figura 1-1 Modos de operación NFC

Fuente: (Minihold, 2011,p.10)

La robustez en la arquitectura de NFC le da una ventaja frente a la tecnología RFID, a pesar de sus características similares, por lo que NFC tiene la capacidad de ser más adaptable y eficiente que otras tecnologías, esta posee tres modelos de operación que son modo emulación de tarjeta inteligente, modo de comunicación punto a punto y modo lectura/escritura como se observa en la Figura 2-1.

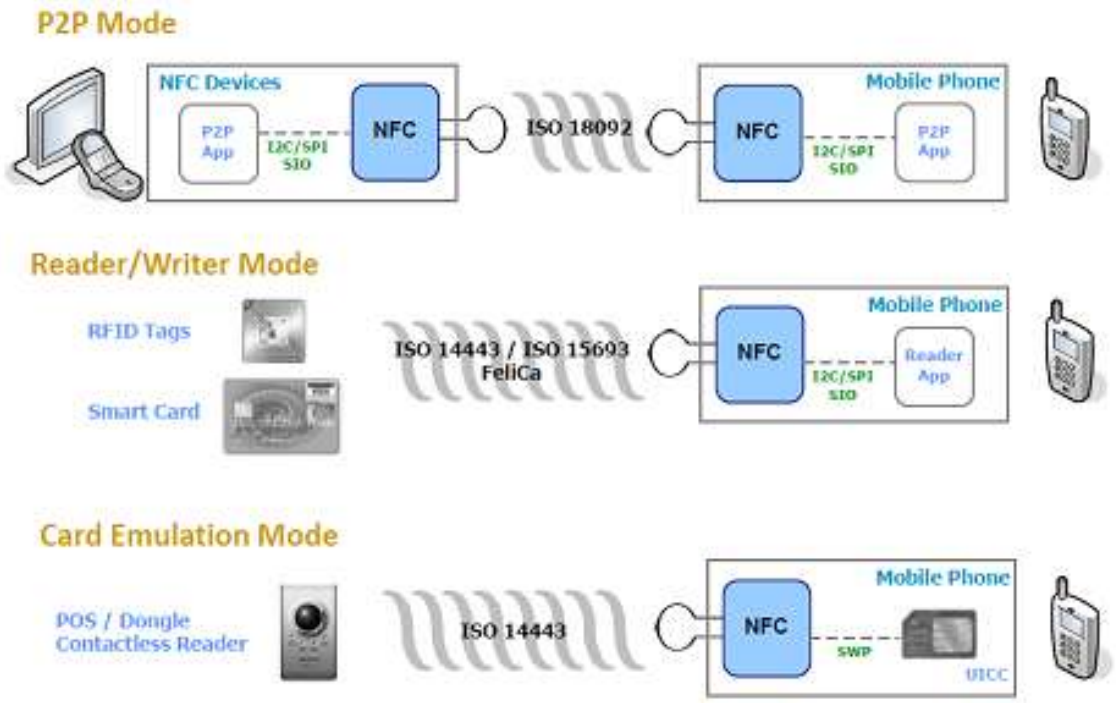


Figura 2-1 Arquitectura Tecnología NFC

Fuente: http://www.myti.it/blog/2013/11/5/nfc-operating-modes

1.1.2.1.1 Modo de tarjeta inteligente (modo pasivo)

En este modo el dispositivo se comporta como una etiqueta o tarjeta inteligente, por lo que se pueden utilizar características de seguridad avanzadas del elemento seguro ya que es utilizada como medio de pago, también para el almacenamiento y gestión de entradas y recibos, en la Figura 3-1 se observa las capas del modo tarjeta inteligente. (Albiñana, 2016,p.16)

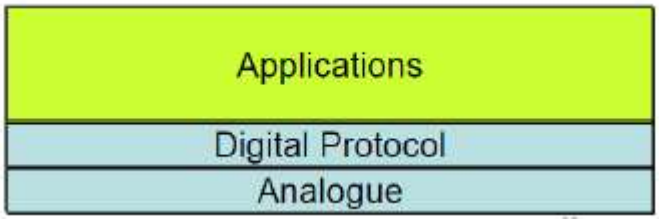


Figura 3-1 Modo de Tarjeta Inteligente

Fuente: http://www.nfc-forum.org/events/oulu_spotlight/technical_Architecture.pdf

En este modo la capa sin contacto (contactless layer) y los campos RF hacen de 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 y las especificaciones de las tarjetas PIN.

El componente que más seguridad le da a este modo es el elemento seguro, el cuál es un chip que cuenta con un procesador, un cripto-procesador, además de su propio sistema operativo llamado JavaCard, también incluye una memoria EPROM; el elemento seguro SE recibe mensaje y envía respuestas a través de sus interfaces de entrada/salida y a través de sus interfaces de fuente de energía. (Lesas & Miranda, 2007, p.23)

En la Figura 4-1 se observa la pila de protocolos que pueden ser usados en el modo tarjeta inteligente.

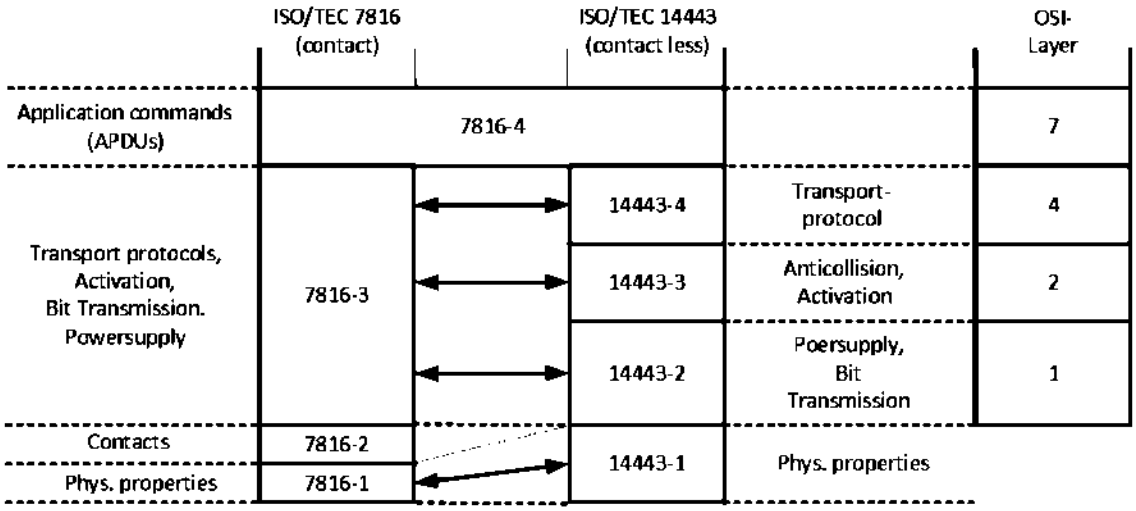


Figura 4-1 Protocolos de Modo de Tarjeta Inteligente

Fuente: (Lesas & Miranda, 2007, p.23)

El elemento seguro posee tres configuraciones posibles que son:

- En la tarjeta SIM o integrada en la placa del circuito manejada por un MNO.
- Embebida en el dispositivo (eSE) manejada por el fabricante del dispositivo.

- Externo y removible como las microSD.

En 5-1 se observa los diferentes tipos de configuraciones de elemento seguro en un móvil.

Se puede acceder al elemento seguro remotamente en la nube o emulando un servicio en segundo plano que actúe como Elemento seguro, este modo es llamado emulación de tarjeta basada en host (HCE). (Lesas & Miranda, 2007, p.24)

Figura 5-1 Configuraciones del Elemento Seguro

Fuente: (Lesas & Miranda, 2007, p.24)

1.1.2.1.2 Modo punto a punto

En este modo los dos dispositivos NFC intercambia información, se intercambian unos pocos kilobytes de datos. (Albiñana, 2016,p.16) En la Figura 6-1 se puede observar la arquitectura de este modo.

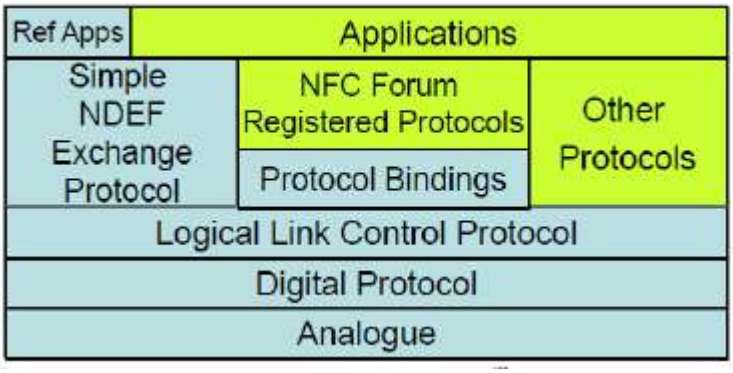


Figura 6-1 Modo Punto a Punto

Fuente: http://www.nfc-forum.org/events/oulu_spotlight/technical_Architecture.pdf

1.1.2.1.3 Modo de Lectura/escritura (modo activo)

En este modo el dispositivo NFC que está activo lee o escribe hacia otro dispositivo que se encuentra en modo pasivo. (Albiñana, 2016, p.16) En la Figura 7-1 se puede observar la arquitectura de este modo.

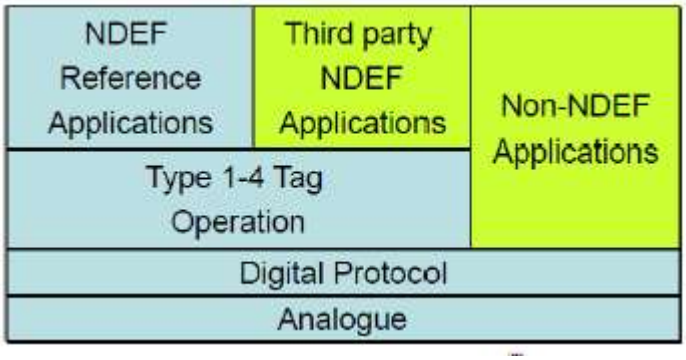


Figura 7-1 Modo Lectura/escritura

Fuente: http://www.nfc-forum.org/events/oulu_spotlight/technical_Architecture.pdf

Los tags NFC consisten en una antena y un circuito integrado, la energía es entregada por acoplamiento inductivo desde el dispositivo que inicia la comunicación. Los datos almacenados en el NFC tag deben cumplir con la definición de tipo de registro (RTD) y son almacenados en mensajes con el formato de intercambio de datos NFC (NDEF). (Lesas & Miranda, 2007, p.22)En la Figura 8-1 se puede observar como interactúa el modo de lectura/escritura con una etiqueta NFC.

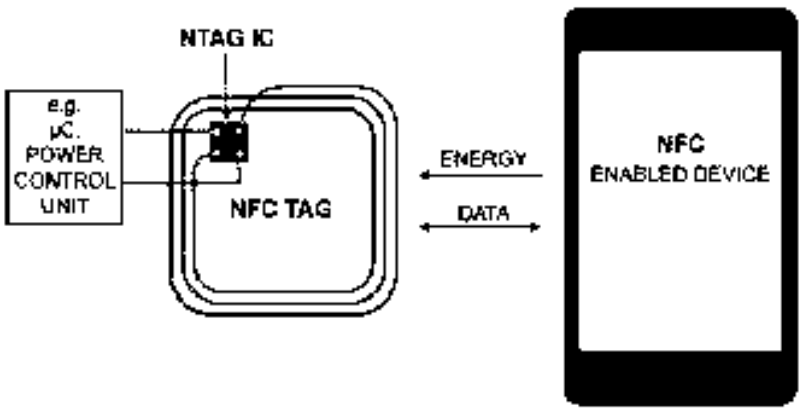


Figura 8-1 Modo Lectura/Escritura

Fuente: (Lesas & Miranda, 2007, p.22)

1.1.2.2 Especificación técnica del protocolo de enlace lógico NFC

Define un protocolo OSI de capa 2 para el soporte de la comunicación punto a punto entre los dos dispositivos NFC. Esta especificación define dos tipos de servicios que son sin conexión y orientados a la conexión, los mismos que se organizan en tres clases de servicios de enlace que son: servicio sin conexión solamente; servicio orientado a la conexión solamente; y servicio sin conexión y orientado a la conexión. (NFC Forum, 2017, https://nfc-forum.org/our-work/specifications-and-application-documents/specifications/nfc-forum-technical-specifications/)

El servicio sin conexión solamente no ofrece fiabilidad ni garantías de control de flujo ya que tiene una configuración muy básica; el servicio orientado a la conexión por el contrario ofrece confiabilidad, tiene control de flujo.

1.1.2.3 Protocolo de Control de Enlace Lógico (LLCP)

Este protocolo permite comunicaciones multiplexadas entre dos dispositivos NFC, donde uno de ellos envía datos de protocolo en cualquier instante (modo equilibrado asíncrono). Los puntos finales de la comunicación se denominan puntos de acceso al servicio (SAP) y son direccionados mediante un identificador numérico de 6 bits. Las unidades de datos de protocolo se intercambian entre dos puntos de acceso al servicio en donde el un punto se convierte en la fuente (SAAP "Source Service Access Point") hacia un destino (DSAP "Destination Service Access Point"). El espacio de direcciones de los SAP se divide en 3 partes: una dirección entre 0 y 15 que identifica a un servicio conocido, una dirección entre 16 y 31 que identifica un servicio que está registrado en el entorno del servicio local y las direcciones entre 16 y 63 que son utilizadas como una dirección de origen por las aplicaciones del cliente que envían o se conectan a servicios de pares, en la Figura 9-1 se observa una comparación entre el modelo OSI y el Protocolo de control de enlace lógico y en la Figura 10-1 el formato de PDU de LLCP.(Tiedemann, 2009, http://nfcpy.readthedocs.io/en/latest/topics/llcp.html)

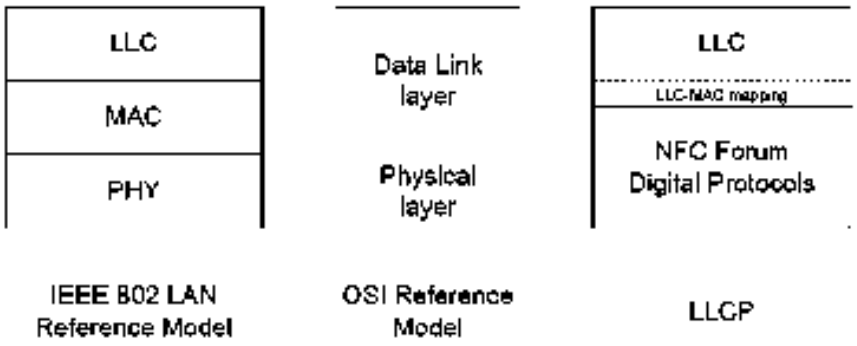
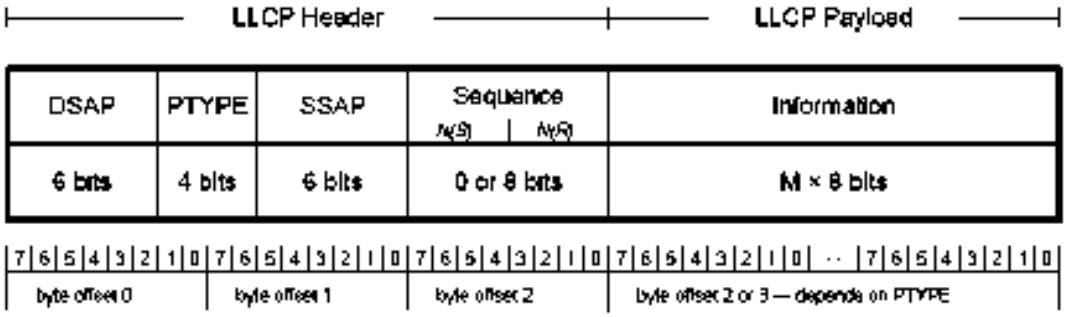


Figura 9-1 Modelo OSI y Protocolo de Control de Enlace Lógico

Fuente: (Lesas & Miranda, 2007, p.30)

Figura 10-1 Formato de PDU de Protocolo de Control de Enlace Lógico



Fuente: (Lesas & Miranda, 2007, p.30)

1.1.2.4 Especificación técnica del protocolo digital NFC

Esta especificación armoniza las tecnologías integradas, especifica las opciones implementación y limita la interpretación de las normas, es decir garantiza la interoperabilidad global entre dispositivos NFC y entre dispositivos NFC y otras tecnologías sin contacto. (NFC Forum, 2017, https://nfc-forum.org/our-work/specifications-and-application-documents/specifications/nfc-forum-technical-specifications)

Para configurar el protocolo de comunicación se describen bloques de construcción, los cuales son llamados actividades, las mismas que se combinan en perfiles y cada uno de estos contiene los parámetros de configuración específicos. La combinación de perfiles y actividades definen un comportamiento predecible para el dispositivo NFC, pero no limita a estos dispositivos a la aplicación para usar otros bloques de construcción. (Albiñana, 2016,p.27)

1.1.2.5 Especificaciones del intercambio de datos NFC (NDEF)

NFC Data Exchange Format (NDEF) o por su traducción al español Formato de intercambio de datos NFC, es una especificación que define como se van a encapsular los mensajes cuando se va a hacer un intercambio de datos entre dispositivos NFC, además de las reglas para construir un mensaje NDEF válido y también de una cadena ordenada de registros NDEF. (Veloz, 2010,p.16) NDEF tiene la capacidad de encapsular una o más cargas útiles (payload) de diferente tipo y tamaño dentro de la estructura del mensaje NDEF. La carga útil (payload) está conformada por:

- **Longitud (PAYLOAD_LENGTH):** Este parámetro indica la cantidad de octetos de carga útil (payload) en otras palabras, la longitud de carga útil (payload) encapsulada en un registro misma que se encuentra dentro de los primeros 8 octetos del mismo. Para registro pequeños este parámetro es de un octeto y esto se indica estableciendo el bit de la bandera SR^{12} en 1, mientras que para registros normales este parámetro es de cuatro.
- **Tipo de carga útil(payload):** Indica que tipo de datos se encuentran en el payload de un registro los cuales pueden ser URIs, MIME o específicos NFC (NFC-specific).
- **Identificador de Payload:** este identificador es opcional y viene en forma de URI absoluta o relativa, permitiendo que el payload de tipo URI vincule tecnologías de referencia con otros payload. (Veloz, 2010, pp.16-17)

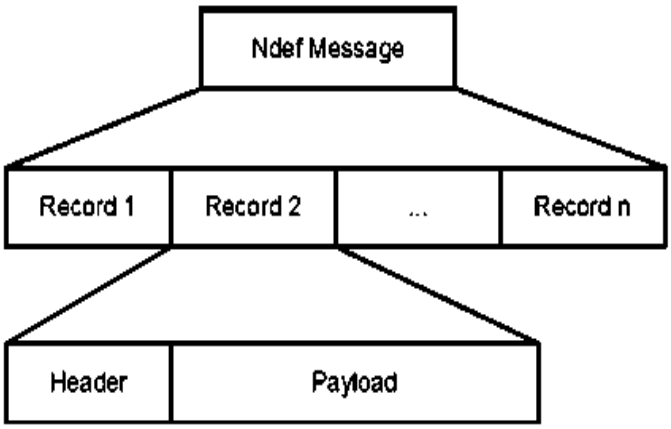


Figura 11-1 Estructura del Mensaje NDEF

Fuente: (Lesas & Miranda, 2007, p.31)

En la Figura 11-1 se observa que el mensaje NDEF está conformado por uno o más registros en los cuales se encuentra la carga útil(payload).

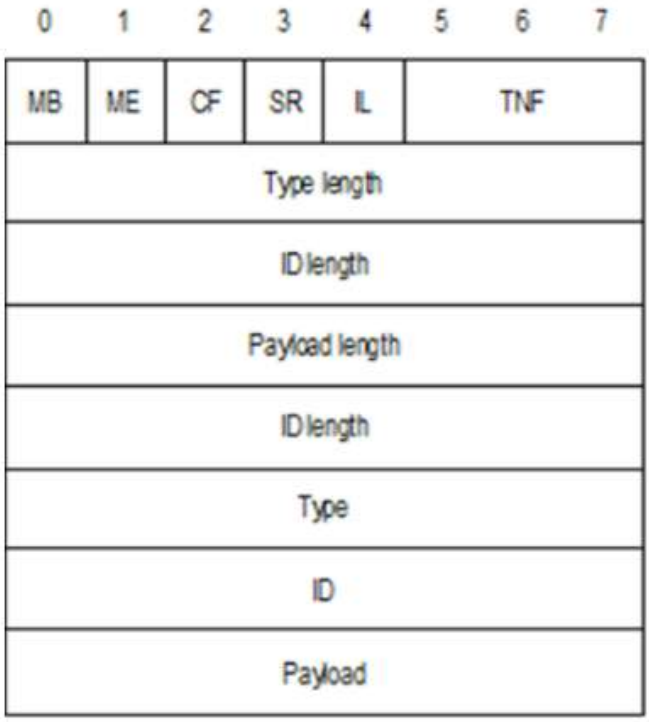


Figura 12-1 Estructura del Registro NDEF

Fuente: (Lesas & Miranda, 2007, p.32)

En la Figura 12-1 se puede observar cómo 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 (Message End): 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

- Type Length (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.
- Payload Length (Longitud de carga útil): especifica la longitud de la carga útil que esta determinado por e l 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)

NDEF incluye una RTN (Records Type Name) que es el nombre del tipo de Registro el cual determina algunos formatos, de los RTN existen dos tipos que son los globales/locales que son determinados por el Foro NFC ("NFC Forum"), en este tipo el identificador NID tiene un valor de 0X01 en el registro TNF; este tipo de RTN deben comenzar con mayúsculas mientras que las de tipo local si pueden empezar con minúsculas o números. En cambio, los RTN de tipo externo son usados por empresas para sus necesidades.(Albiñana, 2016, p.22)

1.1.2.5.1 Registro Tipo URI

Este tipo de registro es utilizado para receptar la URI que se encuentre almacenada en una etiqueta NFC, una URI es una URL almacenada como registro, en este caso el tipo de RTN puede ser considerado como extensión de un tipo NFC bien conocido.(Albiñana, 2016, p.23)

La conformación de la estructura del Registro URI se observa en la Tabla 1-1.

Tabla 1-1 Estructura de Registro URI

Nombre	Offset	Tamaño	Valor
ID	0	1 Byte	URI ID
URI	1	N	UTF-8 String

Elaborado por: SUÁREZ, Jaime. 2018

1.1.2.5.2 Registro de tipo Texto

Este tipo de registro es de texto plano y se combina con otros campos para tener información extra en una etiqueta, su especificación recomienda utilizarlo para propósitos informativos y no vincularlo con otra acción.(Albiñana, 2016, p.24)

1.1.2.5.3 Registro tipo Póster Inteligente (Smart Poster)

Define la cabida de agregar datos que están ocultos en un documento y contienen datos e información agregada del mismo documento a una URI, por lo que el registro ya no va a guardar una simple URL, sino que además puede realizar acciones como abrir una ubicación en un mapa o abrir otras aplicaciones según como se la desarrolle en el póster inteligente. Los datos del Póster inteligente se envían en conjuntos de mensajes de registros NDEF (bandera SR=0), los campos que puede tener la etiqueta del Póster Inteligente son: Título que es una instancia de Text_RTD, URI que es un URI pero añadido metadatos este no puede ser repetido, Action realiza acciones como emparejar bluetooth o abrir el mapa, Icon se usa para incluir imágenes MIME, Size es usada para el tamaño del contenido, Type determina si el dispositivo puede acceder a un objeto externo.(Albiñana, 2016, pp.24-25)

1.1.2.6 Señal Análoga y transposición digital NFC

La comunicación sin contacto modula y demodula datos en binario en una señal en este caso analógica enviando una onda electromagnética que se la conoce como onda portadora a través de la creación de la variación en amplitud, frecuencia y fase; cuando la señal es recibida se transpone digitalmente. (Lesas & Miranda, 2007, p.19)

Los dispositivos NFC utilizan la modulación y demodulación ASK al igual que BPSK de acuerdo a su especificación y tipo, como también la codificación NRZ, Manchester y Miller.(Lesas & Miranda, 2007, p.20)

Tabla 2-1 Especificaciones Técnicas de los Estándares NFC

Standard NFC-Forum	Tasa de Bits	Frecuencia de Portadora
NFC -A	106 kb/s	13.56 MHz
	106 kb/s	13.56 MHz ± 848 kHz subportadora
NFC -B	106 kb/s	13.56 MHz
	106 kb/s	13.56 MHz ± 848 kHz subportadora
NFC -C	212/424 kb/s	13.56 MHz
	212/424 kb/s	13.56 MHz sin subportadora

Elaborado por: SUÁREZ, Jaime. 2018

En la tabla 2-1 se observa la tasa de bits y frecuencia de portadora que utilizan los diferentes estándares NFC.

1.2 HCE (Host-Based Card Emulation o Emulación de Tarjetas basadas en Host)

La emulación de tarjetas basadas en Host es un tipo de emulación de tarjeta inteligente que la implemento el sistema operativo móvil Android desde su versión 4.4. se diferencia de la emulación de tarjeta inteligente común en donde la tarjeta es emulada por un microchip llamado elemento seguro, HCE no involucra al elemento seguro lo que permite a los desarrolladores de aplicaciones emular una tarjeta inteligente NFC y comunicarse directamente con el lector.

1.2.1 Emulación de tarjeta con elemento seguro

Cuando se realiza la emulación de tarjeta con elemento seguro, esta tarjeta estará guardada en el chip llamado elemento seguro a través de una aplicación; por lo que al momento de que el usuario inicie una comunicación con un lector NFC, el controlador NFC del dispositivo móvil enrutará todos los datos hacia el elemento seguro, por lo que la aplicación no está involucrada en la transacción como se observa en la Figura 13-1. (Android,2014, https://developer.android.com/guide/topics/connectivity/nfc/hce.html)

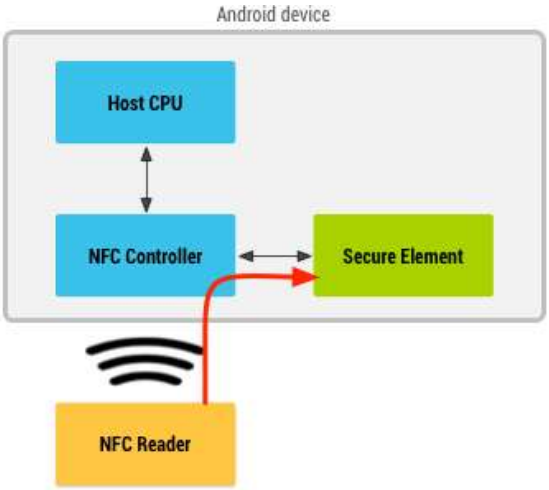


Figura 13-1 Emulación de tarjeta inteligente con elemento seguro

Fuente: (Android, 2014, https://developer.android.com/guide/topics/connectivity/nfc/hce.html)

1.2.2 Emulación de tarjetas basada en host

En la emulación de tarjetas basadas en host los datos son enrutados a la Unidad Central de Proceso del dispositivo anfitrión en donde se está ejecutando la aplicación como se observa en la Figura 14-1, es decir el teléfono inteligente se comporta como una tarjeta inteligente. La principal ventaja que presenta este tipo de emulación de tarjeta inteligente es que no requiere cooperación con las operadoras móviles ni los fabricantes de teléfonos y chips, ya que su implementación depende del desarrollador de la aplicación.(Android,2014,https://developer.android.com/guide/topics/connectivity/nfc/hce.html)

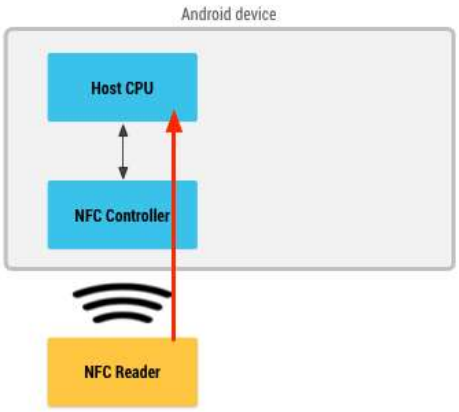


Figura 14-1 Emulación de tarjeta inteligente basada en host

Fuente: (Android, 2014, https://developer.android.com/guide/topics/connectivity/nfc/hce.html)

1.2.2.1 Pila de Protocolos de HCE

Para emular estas tarjetas basadas en host, deben estar basadas en la especificación ISO-DEP de NFC-Forum (ISO/IEC 14443-4) la cual Android exige emular en la parte superior de la tecnología NFC tipo A, en la tecnología NFC tipo B es opcional; además deben procesar unidades de datos de protocolo de aplicación (APDU) definidos en la especificación ISO/IEC 7816-4, esto se observa en la Figura 15-1. (Android, 2014, https://developer.android.com/guide/topics/connectivity/nfc/hce.html)

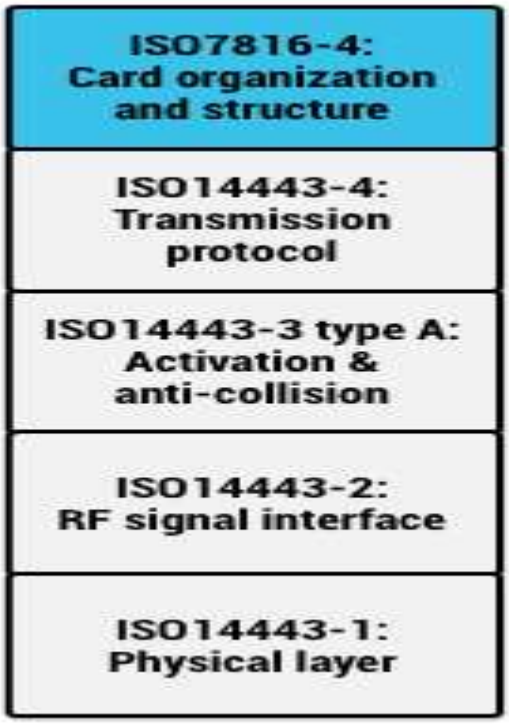


Figura 15-1 Pila de Protocolos de HCE en Android.

Fuente: (Android, 2014, https://developer.android.com/guide/topics/connectivity/nfc/hce.html)

1.2.2.2 Estructura de mensaje de Unidad de Datos de Protocolo de Aplicación (APDU)

Una APDU tiene en su interior un mensaje de comando o un mensaje de respuesta, el primero es enviado desde la aplicación del dispositivo hacia una entidad receptora la cual devuelve el mensaje de respuesta; a cada comando específico corresponde una respuesta específica esto es

denominado par de comando y respuesta, estos pueden contener datos cada uno quedando así cuatro casos de par comando-respuesta con datos como se observa en la tabla 3-1:

Tabla 3-1 Datos en el par comando-respuesta

DATOS DE COMANDO	DATOS DE RESPUESTA ESPERADOS
Datos	Datos
Datos	Sin datos
Sin datos	Sin datos
Sin datos	Datos

Elaborado por: SUÁREZ, Jaime. 2018

1.2.2.2.1 APDU de Comando

Está conformada por un encabezado de 4 bytes obligatorio y un cuerpo condicional de longitud variable, El encabezado consta de la Clase (CLA), Instrucción (INS) y dos parámetros (P1 y P2); el cuerpo en cambio consta de los parámetros Longitud de comando (Lc field), Datos (Data field) y Longitud de datos esperados (Le field). (Jacquinot Consulting, 2018, http://cardwerk.com/smart-card-standard-iso7816-4-section-5-basic-organizations/#chap5_3) En la Figura 16-1 se puede ver como esta estructurado el mensaje APDU de comando.

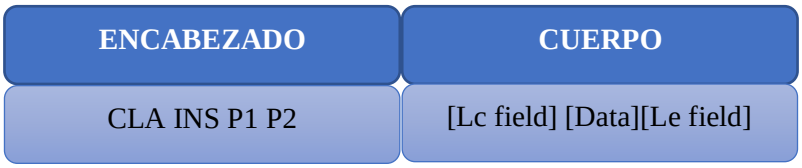


Figura 16-1 Estructura de la APDU de comando

Elaborado por: SUÁREZ, Jaime. 2018

El byte Clase (CLA) es utilizado para indicar en que proporción el comando y la respuesta cumplen con la ISO/IEC 7816, además de si el formato es de mensaje seguro y el número de canal lógico. Este byte como los demás del encabezado tienen una codificación especial que se la puede ver en la tabla 4-1 y 5-1.

Tabla 4-1 Codificación del byte Clase (CLA)

CÓDIGO	SIGNIFICADO
0X	Estructura y codificación de comando-respuesta
9X, 8X	Codificación y significado del comando-respuesta son patentados
B0 a CF	Estructura y codificación de comando-respuesta
10 a 7F	Reservado para uso futuro
D0 a FE	Estructura propietaria y codificación de comando y respuesta
FF	Reservado para PTS

Fuente: (Jacquinot Consulting, 2018, http://cardwerk.com/smart-card-standard-iso7816-4-section-5-basic-organizations/#table6)

En la tabla 4-1 se puede observar que los dos primeros códigos tienen una X, esta tiene una codificación especial que se puede observar en la tabla 5-1.

Tabla 5-1 Codificación de X cuando CLA= 0x, 8x, 9x

BITS				SIGNIFICADO
b1	b2	b3	b4	
x	x	-	-	Formato de mensaje seguro
0	x	-	-	Mensaje no seguro
0	0	-	-	Indicación de sin Mensaje Seguro
0	1	-	-	Formato Propietario de Mensaje Seguro
1	0	-	-	Encabezado de comando no autenticado
1	1	-	-	Encabezado de comando autenticado
-	-	x	x	Número de canal lógico

Fuente: (Jacquinot Consulting, 2018, http://cardwerk.com/smart-card-standard-iso7816-4-section-5-basic-organizations/#table6)

El byte Instrucción (INS) permite codificar para permitir la transmisión con uno de los protocolos ISO/IEC 7816-3, esto cuando el valor del byte CLA se encuentra dentro del rango '00' a '7F', para los otros valores deben ser asignados por la ISO/IEC JTC 1 SC17.(Jacquinot Consulting, 2018, http://cardwerk.com/smart-card-standard-iso7816-4-section-5-basic-organizations/)

En la tabla 6-1 se puede observar los códigos con cada una de sus descripciones del byte INS.

Tabla 6-1 Codificación del byte Instrucción (INS)

CÓDIGO	SIGNIFICADO
0E	Borrar Binario
20	Verificar
70	Administrar Canal
82	Autenticación Externa
84	Obtener Desafío
88	Autenticación Interna
A4	Seleccionar archivo
B0	Leer Binario
B2	Leer Registros
C0	Obtener Respuesta
C2	Envelope
CA	Obtener Datos
D0	Escribir Binario
D2	Escribir Registro
D6	Actualizar Binario
DA	Poner Datos
DC	Actualizar Datos
E2	Apéndice de Registro

Fuente: (Jacquinot Consulting, 2018, http://cardwerk.com/smart-card-standard-iso7816-4-section-5-basic-organizations/#table6)

Los dos bytes siguientes que son P1 y P2 es decir de parámetros pueden tener cualquier valor, si ninguno de estos aporta una valoración adicional estos se establecen en '00'.

El número de bytes del campo datos (DATA field) está definido por el byte Longitud de comando (Lc field), el campo de longitud de datos esperados (Le field) contiene el número máximo de bytes que se espera en el campo datos de la APDU de respuesta, si Le es solo ceros está solicitando la máxima cantidad de bytes de datos.(Jacquinot Consulting, 2018, http://cardwerk.com/smart-card-standard-iso7816-4-section-5-basic-organizations/#table6)

La APDU de comando puede tener 4 estructuras las cuales están descritas en la tabla 7-1:

Tabla 7-1 Estructuras de APDU de Comando

Lc	Datos	Le	Significado
Nulo	Nulo	Nulo	El cuerpo no contiene datos está vacío.
Nulo	Nulo	No Nulo	El cuerpo consiste en el campo Le
No Nulo	No Nulo	Nulo	El cuerpo consiste en Le seguido de Datos
No Nulo	No Nulo	No Nulo	El cuerpo está completo con todos los campos

Elaborado por: SUÁREZ, Jaime. 2018.

1.2.2.2.2 Decodificación de APDU de Comando

Al denotar las 4 estructuras del comando APDU podemos observar que en los casos 2,3 y 4 el cuerpo de la APDU de comando tiene longitudes variables, para decodificarlas el cuerpo del APDU de comando se lo trata como una cadena de L bytes expresada por B1 a BL, en la Figura 17-1 se puede apreciar como es el cuerpo del APDU de comando.(Jacquinot Consulting, 2018, http://cardwerk.com/smart-card-standard-iso7816-4-section-5-basic-organizations/#chap5_3)

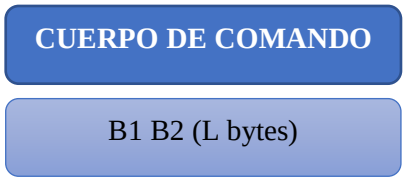


Figura 17-1 Cuerpo de la APDU de comando

Elaborado por: SUÁREZ, Jaime. 2018.

El campo Lc y el campo Le cambia de acuerdo a las capacidades de la tarjeta que se va a emular y puede ser corto (short) cuando el valor es de 1 byte o tiene un valor predeterminado, también puede ser extendido el cual es una declaración explícita, por lo que los casos 2,3 y 4 son cortos cuando cada campo de longitud es de 1 byte o extendidos cuando B1 es valorado con '00' y el valor de cada longitud se codifica en otros 2 bytes.

Cuando el valor Le está codificado en 1 o 2 bytes y todos los bits no son nulos, entonces Le obtiene el valor del byte o bytes que se encuentren en el rango de 1 a 255, si el valor de todos los bits es nulo significa el valor máximo de Le que es de 256. En la tabla 8-1 se denota cada caso si es corto o extendido.(Jacquinot Consulting, 2018, http://cardwerk.com/smart-card-standard-iso7816-4-section-5-basic-organizations/#chap5_3)

Tabla 8-1 Convenciones de decodificación

CASO	CORTO	CASO	EXTENDIDO
1. L=0	Lc valorado en 0 No hay bytes de datos Le valorado en 0	1. L=0	No existe
2. L=1	Lc valorado en 0 No hay bytes de datos B1 codifica Le valorado de 1 a 256	2. L=3 y (B1=0)	Lc valorado en 0 No hay bytes de datos Le son 3 bytes donde es codificado por B2 y B3 y tiene un valor de 1 a 65536
3. L=1+(B1) Y (B1)! =0	Código B1 Lc(=0) valorado de 1 a 255 B2 a Bl son los bytes Lc del campo de datos Le valorado en 0	3. L=3+(B2 ‖B3), (B1)=0 y (B2 ‖B3)=0	Lc consta de los 3 primeros bytes B2 y B3 Lc(!=0) tiene un valor de 1 a 65536 B4 y B2 son los bytes Lc del campo datos Le valorado en 0
4. L=2+(B1) Y (B1)! =0	Códigos B1 Lc(!=0= valorados de 1 a 255 B2 a Bl-1 son los bytes Lc del campo de datos Bl codifica Le de 1 a 256	4. L=5+(B2 ‖B3), (B1)=0 y (B2 ‖B3)=0	Lc consta de los 3 primeros bytes donde es codificado por B2 y B3Lc(!=0) tiene un valor de 1 a 65535 B4 a Bl-2 son los bytes Lc del campo datos Le son los 2 últimos bytes Bl-1 y Bl que codifican Le desde 1 a 65536

Elaborado por: SUÁREZ, Jaime. 2018.

1.2.2.2.3 APDU de Respuesta

La APDU de respuesta consiste en cuerpo condicional que tiene longitud variable en donde van los datos y un tráiler mandatorio (SW1 SW2) que tiene 2 bytes como se observa en la Figura 18-1. El número de bytes que tiene el campo de Datos en la APDU de respuesta es denotado por Lr, la función del tráiler es de codificar el estado de la entidad receptora luego de procesar el par comando-respuesta. Si el comando es cancelado la APDU de respuesta es un tráiler codificado una condición de error en 2 bytes.(Jacquinot Consulting, 2018, http://cardwerk.com/smart-card-standard-iso7816-4-section-5-basic-organizations/#chap5_3) En la Figura 18-1 se observa la estructura del mensaje APDU de respuesta.

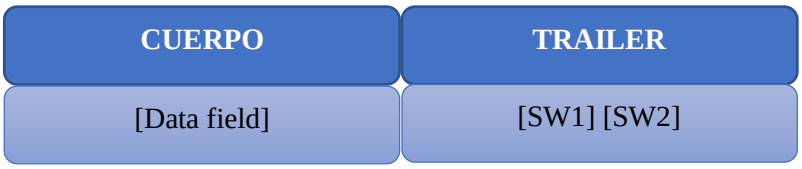


Figura 18-1 APDU de Respuesta

Elaborado por: SUÁREZ, Jaime. 2018.

SW1 y SW2 también son conocidos como bytes de estado y señalan el estado de procesamiento en la tarjeta. SW1 y SW2 pueden tomar algunos valores por ejemplo cuando SW1= ‘63’ o ‘65’ el estado cambia a memoria no volátil o cuando SW1=’6X’ excepto ‘63’ y ‘65’ no cambia el estado de la memoria no volátil. En las especificaciones de ISO/IEC 7816 define los siguientes valores para SW1-SW2:

‘60XX’

‘67XX’,’6BXX’, ‘6DXX’, ‘6EXX’, ‘6FXX’; en cada caso sí ‘XX’ no es igual a ‘00’

‘9XXX’, si ‘XXX’ no es igual a ‘000’

Y los siguientes valores se definen según el protocolo que se utilice:

Si un comando es cancelado y en la respuesta el byte SW1=’6C’, entonces SW2 indicará el valor que se le asignará a la longitud de datos esperados (Le) al volver a emitir el mismo comando.

Si un comando de estructura 2 o 4 se procesa con una respuesta en la cual SW1=’61’, entonces SW2 indicara el valor máximo que se asignará a Le en un comando GET RESPONSE. (Jacquinot Consulting, 2018, http://cardwerk.com/smart-card-standard-iso7816-4-section-5-basic-organizations/#chap5_3)

La codificación de los bytes de estado es la siguiente:

PROCESAMIENTO NORMAL

‘9000’ Sin más calificación

‘61XX’ SW2 indica cuantos bytes de respuesta se tienen aún disponibles

PROCESOS DE ADVERTENCIA

‘62XX’ Estado de la memoria no volátil sin cambios

‘63XX’ Estado de la memoria no volátil cambiado

ERRORES DE EJECUCIÓN

‘64XX’ Estado de la memoria no volátil sin cambios (SW2=00, otros valores son reservados para uso futuro).

‘65XX’ Estado de la memoria no volátil cambiado

‘66XX’ Reservado para problemas relacionados con la seguridad

COMPROBACIÓN DE ERRORES

‘6700’ Longitud incorrecta

‘68XX’ Las funciones en el byte Clase (CLA) no es compatible

‘69XX’ Comando no permitido

‘6AXX’ ‘6B00’ Parámetros P1-P2 incorrectos

‘6CXX’ Longitud incorrecta de Le: SW2 indica la longitud exacta

‘6D00’ Código de instrucción (INS) no compatible o no válido

‘6E00’ Clase (CLA) no admitida

‘6F00’ Sin un diagnóstico preciso. (Jacquinot Consulting, 2018, http://cardwerk.com/smart-card-standard-iso7816-4-section-5-basic-organizations/#chap5_3)

1.2.2.3 Seguridad en Emulación de Tarjetas Basadas en Host (HCE)

La seguridad es muy importante al utilizar HCE ya que están comprometidos los datos y credenciales de los usuarios del servicio desarrollado, ya que la comunicación entre el lector NFC y el dispositivo móvil puede ser interceptada por aplicaciones malware, y este riesgo aumenta

cuando el dispositivo móvil esta realizado una explotación, rooting o jailbreaking.(Alliance, 2014, p.19)

Para eliminar las amenazas externas de robo de información y datos se utilizan varias medidas de seguridad que son:

- Criptografía de caja blanca (White box)
- Factores Biométricos
- Servicio de Enlace NFC (Bind NFC Service)
- Ejecuciones en ambiente seguro
- Tokenización

1.2.2.3.1 Criptografía de Caja Blanca

La criptografía de Caja Blanca permite realizar operaciones criptográficas sin revelar los datos confidenciales, sin esto el atacante puede tomar la información de claves secretas de la implementación binaria, de la memoria o interceptar datos al momento que se ejecute una aplicación de software. Para que la criptografía de caja blanca funcione es necesario implementar un algoritmo criptográfico en el software para que lo activos criptográficos sigan teniendo esa seguridad incluso cuando estén bajo un ataque de caja blanca.

A la criptografía de Caja blanca se la puede ver como un generador de códigos que convierte un cifrado en una representación robusta en la cual estas operaciones se combinan con datos y códigos aleatorios para que los datos aleatorios no se distingan de la información clave como se observa en la Figura 19-1. (Wyseur, 2012, http://www.whiteboxcrypto.com/)

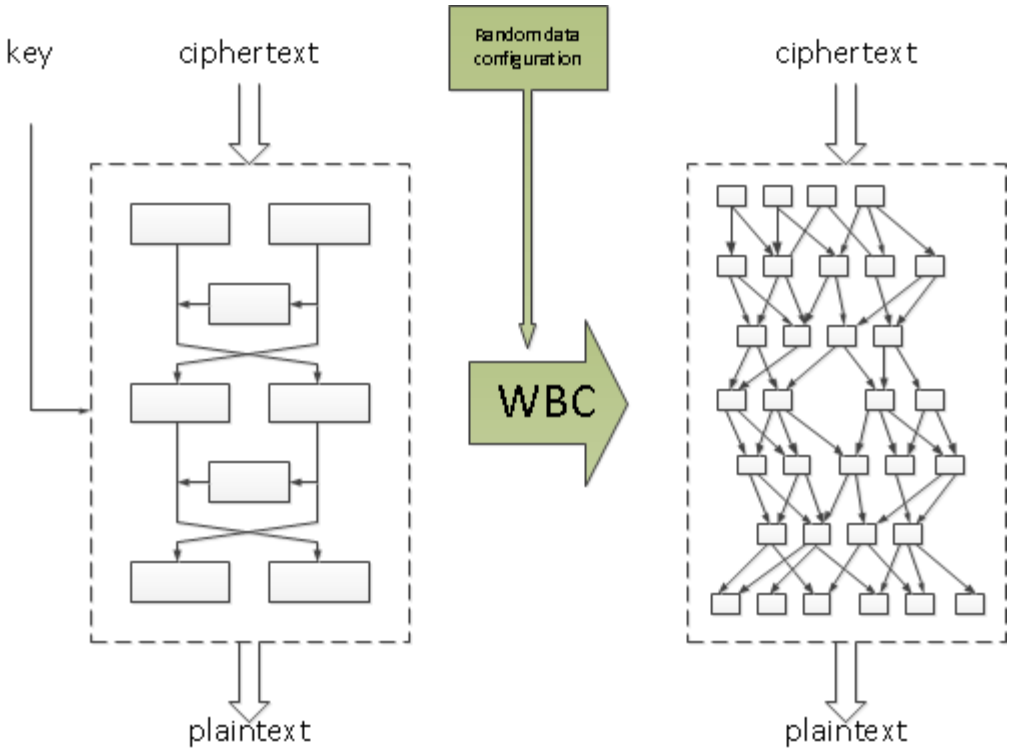


Figura 19-1 Representación del Cifrado de Caja Blanca

Fuente: http://www.whiteboxcrypto.com/img/overview.png

1.2.2.3.2 Factores Biométricos

Los factores biométricos son usados en HCE para fortalecer la seguridad de las transacciones y en la autenticación, la ventaja que tiene el uso de los factores biométricos es que es más amigable al usuario en lugar de tener varias contraseñas. En la actualidad los teléfonos inteligentes pueden usar tres factores biométricos que son la huella dactilar, el reconocimiento de voz y el reconocimiento facial, la privacidad y seguridad. (Alliance, 2014, p.20)

1.2.2.3.3 Servicio de Enlace NFC

Provee seguridad haciendo que el sistema operativo sea el único que pueda vincularse y comunicarse con el servicio NFC en el modo de emulación de tarjeta inteligente basado en host (HCE), garantizando que cualquier APDU que se envíe de regreso solo se irá al Sistema Operativo el mismo que reenviará directamente las APDU al controlador de NFC. Otra característica importante de Servicio de Enlace NFC es cuando obtiene los datos de la aplicación móvil en el

lector NFC, estos datos están desacoplados intencionalmente en el diseño de HCE por lo que no importa de dónde provengan los datos lo que importa es el transporte seguro de información hacia el lector y el controlador NFC. (Android, 2014, https://developer.android.com/guide/topics/connectivity/nfc/hce.html)

1.2.2.3.4 Ejecuciones en ambiente seguro (TEE)

Las Ejecuciones en ambiente seguro (TEE) se dan cuando existe un área segura en el procesador del dispositivo móvil o en su coprocesador en donde puede procesarse la información y almacenarse, está compuesta por hardware y software, asiste en el control de los derechos de acceso en las aplicaciones y ofrece seguridad contra ataques de software hechos desde el sistema operativo Rich (Rich OS). Las ejecuciones en ambiente seguro pueden proveer un nivel adicional de seguridad con:

- PIN o Contraseña: Las ejecuciones en ambiente seguro tienen la capacidad de obtener de forma segura el ingreso del PIN o Contraseña del dispositivo y que no pueda ser interceptada por malware en el Sistema Operativo del mismo.
- Almacenamiento Seguro de Credenciales: Ejecuciones en Ambiente Seguro permite almacenar llaves e implementar operaciones criptográficas.
- Protocolo de transferencia segura en punto final: en TEE es posible transmitir los comandos APDU entre el dispositivo móvil y el lector en un canal criptográfico seguro.(Alliance, 2014, p.20-21)

1.2.2.3.5 Tokenización

Es el proceso en el cual se sustituye un valor al azar por una credencial de valor alto, creando un valor equivalente bajo, es usada para enmascarar la identidad de una tarjeta y solo puede utilizarse una vez ya que es generado al momento de recibir información en la tarjeta. La Tokenización es un mecanismo que en la actualidad es necesario para proteger la identidad y las credenciales de pago contra fraudes y falsificación, en la figura 20-1 se puede observar el proceso de Tokenización.(Alliance, 2014, p.21)

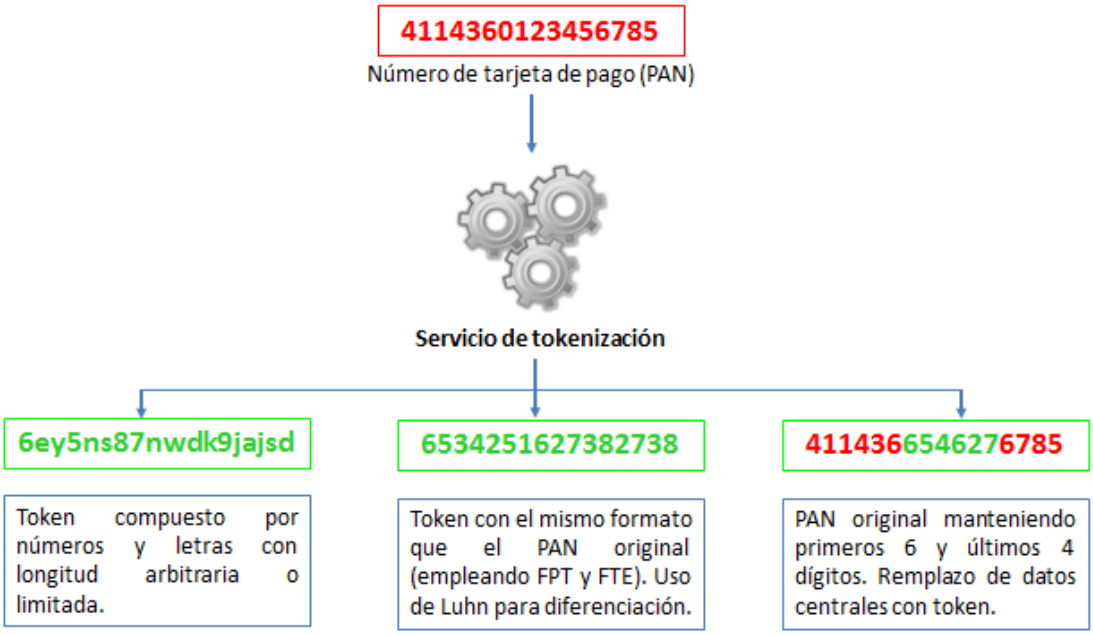


Figura 20-1 Tokenización

Fuente: https://www.pcihispano.com/el-concepto-de-tokenizacion-y-su-aplicabilidad-en-pci-dss/

1.3 Seguridad en los pagos móviles con NFC

La seguridad en este tipo de pagos es muy importante ya que están comprometidos datos sensibles de usuarios por lo que es esencial configurar con los protocolos de seguridad adecuados principalmente en las conexiones a los servidores.

1.3.1 Función Hash

“ Una función HASH o también llamados resúmenes, tiene por objetivo lograr un extracto (siempre de igual tamaño) de cualquier tipo de archivo binario, con el propósito que se puede generar una relación vinculante EN UN SOLO SENTIDO desde el documento hacia el HASH.”(Corletti Estrada, 2011, p.447)

El hash (h) aplicado a un archivo (M) quedando h(M) tiene algunas propiedades para poder decir que es seguro, estas se las describe en la Tabla 9-1.

Tabla 9-1 Propiedades de la función Hash

PROPIEDAD	DESCRIPCIÓN
Unidireccionalidad	Teniendo h(M) debe ser computacionalmente inalcanzable descubrir M desde un h.
Compresión	h(M) de ser de una longitud fija, normalmente menor que el mensaje M.
Facilidad de cálculo	Debe ser fácil de calcular h(M) desde un mensaje M
Difusión	h(M) debe ser una función compleja de todos los bits del mensaje M
Colisión Simple	Será computacionalmente imposible conocido M, encontrar otro M' tal que h(M)=h(M')
Colisión Fuerte	Será computacionalmente difícil encontrar un par (M,M') de forma que h(M)=h(M')

Elaborado por: SUÁREZ, Jaime. 2018.

1.3.1.1 MD5 (Message Digest versión 5)

Fue creado por Ron Rivest en 1992 mejorando a los anteriores MD4 y MD2, su Hash es de 128 bits de longitud fija, también es conocido como Huella Digital ya que es casi improbable que se produzca dos mensajes que posean la misma recopilación ni regenerar el mensaje a través de la recopilación. Genera un h(M) de 2^{128}, mientras que para generar dos h(M) aleatorios distintos son de 264 operaciones, lo que lo hace más rápido que SHA-1 pero menos seguro.(Corletti Estrada, 2011, p.448)

1.3.1.2 SHA-1 (Standard Hash Algorithm Version 1)

Es uno de los tipos de resúmenes o Hash más usados en la actualidad creado por el National Institute of Standards and Technology en 1994, tiene muchas similitudes a MD5 solo que su Hash es de 160 bits de longitud fija, en las versiones más actuales se encuentra el SHA-256 y el SHA-512. La dificultad de generar un h(M) es de 2^{160} para SHA-1, 2^{256} para SHA-256 y 2^{512} para SHA-512, mientras que para generar dos h(M) aleatorios distintos son de 280 operaciones en

SHA-1 por lo que lo convierte en un algoritmo más seguro ante ataques de fuerza bruta pero ya que realiza más operaciones es más lento y consume más recursos de hardware. (Corletti Estrada, 2011, p.448)

1.3.2 Cifrado Triple DES (3DES)

Triple DES fue creado para mejorar al cifrado DES que ya presentaba fallas, este solo extiende el tamaño de clave que en DES era de 56 bits aplicando el algoritmo de tres veces en sucesión con tres llaves diferentes, por lo que el tamaño de DES aumentó a 168 bits, existe una variación en donde se reduce el tamaño de la clave eficaz a 112 bits utilizando solo las llaves K1 y K3. Es utilizada masivamente en la industria de pagos electrónicos. (Medina et al, 2015, p.19)

1.4 Panorama de los pagos móviles en los últimos años

En los últimos años los dispositivos han dado pasos gigantes tecnológicamente hablando especialmente los dispositivos móviles, por lo que hoy en día la mayoría de teléfonos inteligentes poseen tecnología NFC, esto ha desencadenado que los desarrolladores de software enfoquen esfuerzos en explotar esta tecnología.

Según el Foro NFC esta tecnología puede ser usada en cualquier comunicación de forma activa al realizar intercambios de datos o pasiva leyendo etiquetas, en la Figura 21-1 se observan los ámbitos en donde se puede implementar la tecnología NFC.

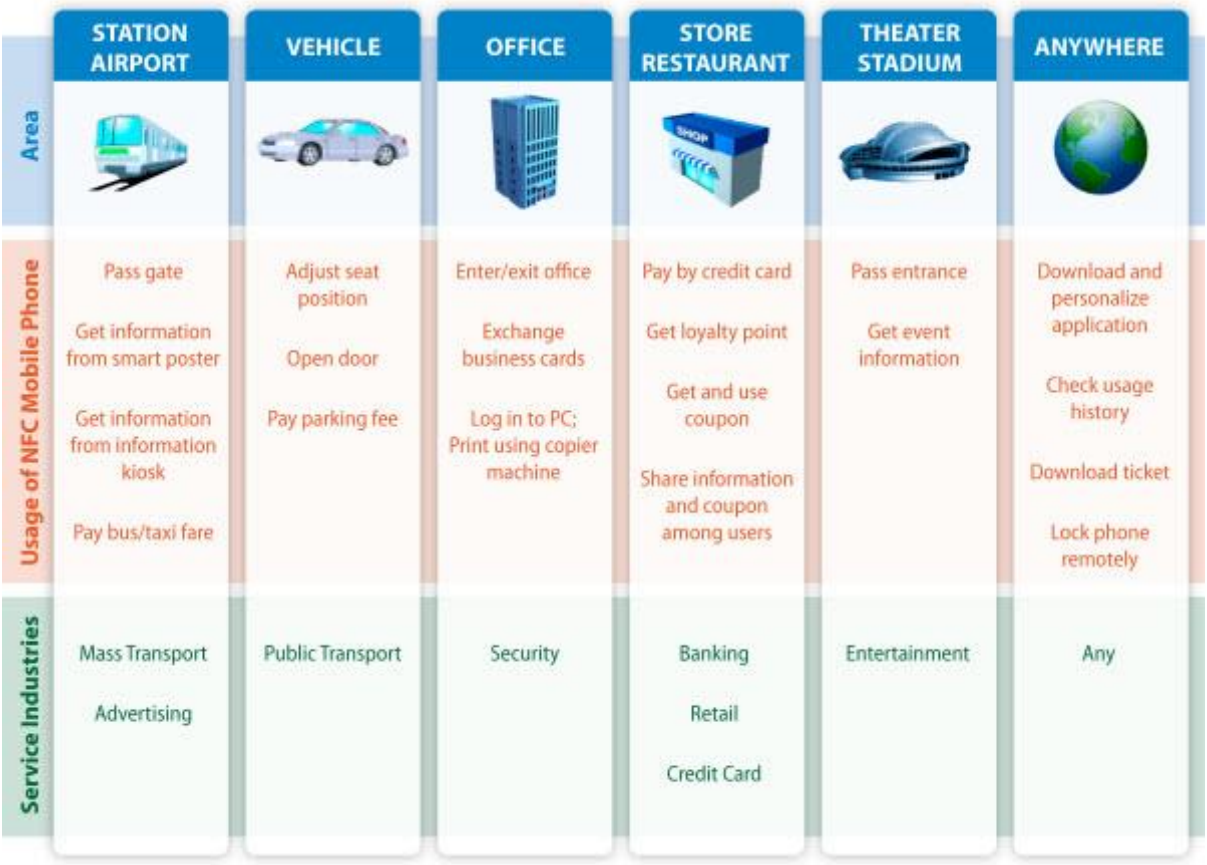


Figura 21-1 Ámbitos donde se puede usar NFC

Fuente: http://members.nfc-forum.org/aboutnfc/nfc_in_action/

Hoy en día la mayor parte del uso de NFC es para realizar pagos y transacciones bancarias, el país donde mayormente se usa NFC para realizar dichas transacciones es Japón donde se lo implementó ya a finales del 2009 nombrando a la tecnología Mobile FeliCa la cual es propiedad de Sony y pertenecen al tipo NFC-F o tipo 3, este funciona en modo de emulación de tarjeta inteligente y se pueden almacenar desde tarjetas para pagos de transporte hasta tarjetas de crédito. El problema que tiene FeliCa es que este estándar no contempla los tipos NFC-A y NFC-B por lo que cualquier dispositivo NFC puede leer una tarjeta FeliCa pero una tarjeta FeliCa no puede leer otro tipo de tarjeta NFC ya que Sony incluye una capa de cifrado y autenticación propietaria de la marca. (INTECO, 2013, pp.12-13)

Otra potencia por así decirlo que lidera las transacciones bancarias es Estados Unidos y uno de los servicios que se está imponiendo en estos tiempos es Google Wallet, este fue presentado en 2011 por Google y usa estrictamente la tecnología NFC. Para el uso de este servicio Google creó una red segura para que los usuarios puedan usarlo, además este servicio utiliza el elemento seguro y este debe ser compatible con Wallet.(INTECO, 2013, pp.13)

Los países africanos en este último tiempo se están destacando como pioneros en productos y servicios de pagos móviles en el que destaca M-Pesa implementando en Kenia, haciendo por mes movimientos en promedio de $415 millones de dólares, teniendo una percepción positiva de usuarios del 95%.(Cerón et al , 2015, p.79)

En Latinoamérica más puntualmente en Colombia, los pagos móviles están siendo pensados para el sector rural y agrícola, ya que este tipo de pagos no necesariamente necesita de una cuenta vinculada a un banco, y puede resultar una forma fácil e inmediata de enviar dinero comparándola con oficinas postales, bancos y empresas de transporte, además de evitar el desplazamiento poblacional rural hacia las grandes ciudades lo que conlleva a una reducción del impacto ambiental de 0,4 megatoneladas menos de CO2 para el 2020. Los pagos móviles también ofrecerían una identidad financiera para la gente que habita en las zonas rurales y obtendrían beneficios tales como rapidez y comodidad, cobertura en zonas donde no se encuentren entidades financieras. (Cerón et al, 2015, p.80)

CAPÍTULO 2

2 MARCO METODOLÓGICO

2.1 INTRODUCCIÓN

Este capítulo explica la elaboración del prototipo de pagos de transporte público propuesto, a partir de la equiparación de características de las comunicaciones inalámbricas que existen en la actualidad en teléfonos inteligentes, mediante el método deductivo se realiza la elección apropiada de la tecnología inalámbrica con la cual se van a realizar los pagos. En la tabla 1-2 se observa las características de las tecnologías inalámbricas que se puede encontrar en un teléfono inteligente.

Para la selección de la tecnología inalámbrica se utilizó la escala de Likert, el cual nos permite medir el grado de conformidad en este caso de la tecnología a usarse, a partir de un criterio subjetivo, este método mide tanto el grado positivo como neutral y negativo de cada enunciado o característica.(Educalab, 2014) Por lo tanto se asignó valores de 1 a 3, siendo 3 la opción adecuada, 2 la neutral y 1 no adecuada de forma cualitativa, para el análisis se ponderan estos valores en porcentajes de forma cuantitativa como se observa en la Tabla 1-2

Tabla 1-2 Cuadro Comparativo de características de las tecnologías inalámbricas en teléfonos inteligentes.

TECNOLOGÍA	TIPO DE CONEXIÓN	RANGO DE COMUNICACIÓN (metros)	TIEMPO REQUERIDO DE COMUNICACIÓN (segundos)	VELOCIDAD DE TRANSMISIÓN (Kilobits por segundo)	CONSUMO DE ENERGÍA
Near Field Communication (NFC)	Punto a punto Lectura/escritura Emulación de tarjeta inteligente	<0.1	<0.1	424	Mínimo
Bluetooth (BLE)	WPAN	Hasta 30	< 6	24000	Medio
WIFI	Ad-hoc	Hasta 100	< 20	6000000	Alto

Elaborado por: SUÁREZ, Jaime. 2018

En la tabla 2-2 se puede observar los valores que se les dio a cada tecnología de forma cualitativa.

Tabla 2-2 Cuadro Comparativo de características de las tecnologías inalámbricas en teléfonos inteligentes.

TECNOLOGÍA	TIPO DE CONEXIÓN	RANGO DE COMUNICACIÓN	TIEMPO REQUERIDO DE COMUNICACIÓN	VELOCIDAD DE TRANSMISIÓN	CONSUMO DE ENERGÍA
Near Field Communication (NFC)	3	3	3	1	3
Bluetooth (BLE)	2	1	1	3	2
WIFI	2	1	1	3	1

Elaborado por: SUÁREZ, Jaime. 2018

En la tabla 3-2 se puede observar el valor total según el criterio subjetivo y su ponderación a porcentaje de forma cuantitativa.

Tabla 3-2 Ponderación en porcentaje de los valores de elección de tecnología inalámbrica

TECNOLOGÍA	TOTAL	PORCENTAJE
Near Field Communication (NFC)	13	86.67 %
Bluetooth (BLE)	9	60 %
WIFI	8	53 %

Elaborado por: SUÁREZ, Jaime. 2018

De acuerdo a la tabla 3-2 se determina que la mejor tecnología para ser implementada en el prototipo es la de NFC, por lo que se realizará la selección de modo de comunicación adecuadas de la tecnología NFC para el diseño y la implementación de las aplicaciones, tomando en cuenta la operabilidad, seguridad en los datos sensibles y costo, para obtener un sistema que ofrezca robustez y confianza. Para esta selección del modo adecuado se utilizará el método de prototipo de sistemas el cual ayudará a que el desarrollo de las aplicaciones de forma rápida y se centrará en los aspectos que serán visibles para el usuario final.

El proyecto se lo ha planteado como una alternativa tecnológica que permite realizar los cobros de servicios de transporte público con la seguridad de que los datos sensibles no van a ser modificados o alterados por agentes externos a las empresas.

Por otra parte, también se trata de establecer un punto de partida para el Internet de las Cosas en este caso aplicado al transporte público y con la tecnología del NFC la cual se encuentra en la actualidad en auge ya que está incorporada en la mayoría de teléfonos inteligentes del mercado.

2.2 DESARROLLO DE PROTOTIPO

La metodología que se utiliza en el desarrollo del prototipo es basada en una investigación experimental, en donde se aplica el método deductivo para recopilar información necesaria que permita determinar el modelo de arquitectura de red más adecuado para que las transacciones se realicen de manera correcta.

Después de realizado el análisis por el método deductivo se consideró que la arquitectura más adecuada es la de cliente-servidor. La elección de esta arquitectura se acopla al prototipo por que un servidor puede dar servicio a varios clientes, además de que puede ser escalable de forma independiente, por lo que si se realizan cambios en las plataformas de los clientes o en el servidor estos van a ser transparentes para el usuario final.

La topología diseñada para el prototipo se puede visualizar en la Figura 1-2, en la cual se muestra cómo va interactuar la aplicación móvil en el teléfono celular con el lector y la aplicación de escritorio, en la cual se controla todas las transacciones, dichas transacciones se las va a almacenar en el servidor con una base de datos en donde se implementa la seguridad necesaria para que no exista ningún robo de datos, devolviendo saldos y registros a la aplicación móvil.

Figura 1-2 Topología del prototipo

Elaborado por: SUÁREZ, Jaime. 2018

2.2.1 Aplicación Móvil

Este es el punto de partida del sistema, el teléfono inteligente con la aplicación instalada va a iniciar la comunicación con el lector/escritor NFC, es amigable y de fácil manejo por parte del usuario.

2.2.2 Lector/Escritor NFC

Realiza la modulación y demodulación de los pulsos obtenidos del teléfono inteligente, este lee estos datos y los transmite hacia la aplicación de escritorio y viceversa.

2.2.3 Aplicación de Escritorio

Aquí se realiza la administración de saldos y el control de pagos que son realizados por los usuarios, está conectada directamente con el servidor de base datos.

2.2.4 Servidor/Base de Datos

En la base de datos se encuentran almacenados los usuarios, operadores, administradores, contraseñas, etc., mientras que el servidor tiene características de seguridad que no permiten que los datos de la base de datos sean alterados por agentes externos.

2.3 Requerimientos del diseño del prototipo

Basándose en la investigación que se realizó en el capítulo anterior, estamos en capacidad de plantear los requerimientos que va tener el diseño del prototipo para pago de transporte público en estaciones a través de un teléfono móvil con tecnología NFC, estas son:

- Ser de fácil operación y costo moderado.
- La comunicación entre el servidor y el software del operador en la estación debe ser segura.
- La aplicación móvil debe ser robusta para evitar el robo de información.
- Los usuarios no deben necesitar abrir la aplicación móvil para su uso.

2.4 Requerimientos de software

2.4.1 Elección del software de aplicación de estación

Los lenguajes de programación para realizar aplicaciones son muy variados, así como los entornos de programación por lo que se debe tomar en cuenta algunas variables importantes para la elección del software como son: el tipo de aplicación a desarrollar, las plataformas de funcionamiento, la dificultad de desarrollo, la compatibilidad con el lector/escritor NFC, entre otras.

Por lo que al estudiar estas variables se eligió el lenguaje de programación Java, ya que este es multiplataforma, es el más utilizado para el desarrollo de software, y tiene compatibilidad con Android el cual es el sistema operativo más utilizado en los teléfonos inteligentes.
Para el desarrollo de la aplicación de escritorio los entornos de programación más utilizados son:

- Netbeans IDE
- Eclipse
- Intellij IDEA

2.4.1.1 Comparación entre los entornos

Mediante la equiparación de características y compatibilidad con los componentes necesarios de hardware de los entornos de programación, también mediante el método de observación se realiza la elección más apropiada del software a ser usado en el proyecto, en la tabla 4-2 se puede ver las características de cada uno de los entornos de programación.

Es muy importante esta selección ya que la aplicación de escritorio es una parte fundamental en el prototipo y su desarrollo debe ser sencillo y poseer los elementos que permitan la comunicación entre aplicaciones a través del lector/escritor NFC.

Para la selección del entorno se utilizó de igual forma la escala de Likert, por lo tanto, se asignó valores de 1 a 3, siendo 3 la opción adecuada, 2 la neutral y 1 no adecuada de forma cualitativa, para el análisis se ponderan estos valores en porcentajes de forma cuantitativa como se observa en la Tabla 5-2.

Tabla 4-2 Cuadro Comparativo de características de los entornos de programación Java

ENTORNO	COMPATIBILIDAD	API NFC	USO DE RECURSOS DE HARDWARE	COMPLEJIDAD
NetBeans IDE 	Compatible con todos los sistemas operativos que utilicen Java. Es de código abierto.	SI	Medio	Medio Intuitivo Configuración sencilla
Eclipse IDE 	Compatible con todos los sistemas operativos que utilicen Java. Código abierto	SI	Alto	Media-Alta Su configuración es complicada.
Intellij IDEA 	Compatible con todos los sistemas operativos que utilicen Java. No está basado totalmente en software libre. Plugins importantes son de pago y costo altos.	SI	Muy Alto	Medio

Elaborado por: Suárez, Jaime. 2018

Tabla 5-2 Cuadro Comparativo de entornos de programación Java

ENTORNO	COMPATIBLIDAD	API NFC	USO DE RECURSOS	COMPLEJIDAD	TOTAL	PORCENTAJE
NetBeans IDE	3	3	2	2	10	83,33%
Eclipse IDE	3	3	1	1	8	66,66%
Intellij IDEA	3	3	1	2	9	75%

Elaborado por: SUÁREZ, Jaime. 2018

De acuerdo a la tabla 3-2 se determina que la mejor opción para el desarrollo de la aplicación de escritorio es NetBeans IDE ya que el uso de recursos y su complejidad es el más apto para el proyecto.

2.4.1.2 NetBeans IDE

Entorno de programación principalmente desarrollado para Java, aunque soporta otros lenguajes como C/C++, este entorno es de código abierto y es desarrollado por Sun Microsystems.
Es el entorno preferido por los desarrolladores de aplicaciones de escritorio, NetBeans permite que las aplicaciones sean desarrolladas por módulos, estos contienen las clases que van a interactuar con las API's de NetBeans. (DIARLU, 2016, https://www.diarlu.com/mejores-ide-programar-java/)

Existen millones de desarrolladores que utilizan esta plataforma ya que el consumo de recursos es bajo comparado con otros además de que al ser desarrollado por Sun Microsystems tiene las últimas actualizaciones y APIS de Java por lo que lo hace muy interesante, el entorno de Netbeans se lo puede observar en la Figura 2-2.

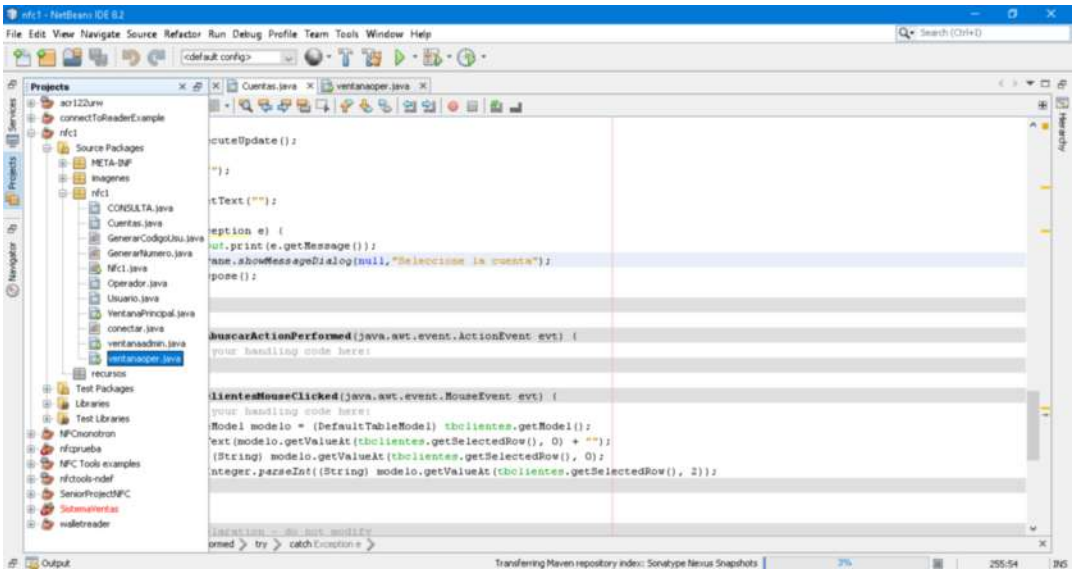


Figura 2-2 Entorno de NetBeans en Windows

Elaborado por: SUÁREZ, Jaime. 2018

2.4.1.3 Características de NetBeans IDE

La tabla 6-2 indica las características principales del entorno de desarrollo NetBeans descrito con anterioridad.

Tabla 6-2 Cuadro de características principales de NetBeans IDE

CARACTERÍSTICAS	DESCRIPCIÓN
Compatibilidad con ECMAScript 6	Nuevos Hexa, binarios y literales octales
Node.js	Declaración de devolución en contexto global
Soporte de Oracle JET	Plantillas Oracle JET y distribución base de las mismas
Mejoras de Perfiles SQL	Resultados de perfiles basados en filtros definidos por el usuario
Mejoras C/C++	Rediseñados para gestionar configuraciones de Run o Debug Detección automatizada del compilador utilizado. Contenedor de herramientas en Windows y Mac OSX
Aplicaciones web Java EE y Java	Incluidas tecnologías Enterprise JavaBeans, Java Persistence, Struts, GWT, Spring
Integración con herramientas y servicios externos	Incluida la integración con base de datos y uso de sistemas de administración de código fuente.

Continua

Requisitos mínimos de hardware	Procesador: Intel Pentium III o equivalente a 800MHz Memoria: 512MB Espacio en Disco: 750 MB de espacio libre en el disco
Requisitos mínimos de software	JDK 8 JavaFX SDK Controlador JavaDB Controlador Oracle Controlador PostgreSQL Controlador MySQL Sistemas Operativos: Windows Linux MacOSX Solaris Open VMS

Elaborado por: SUAREZ, Jaime. 2018

2.4.2 Elección software aplicación móvil

Al igual que para el software de escritorio para la parte móvil existen varios entornos de programación, pero al realizar esta aplicación para el sistema operativo Android únicamente se escogió Android Studio ya que este cuenta con soporte oficial del sistema operativo móvil, además de tener las librerías adecuadas para la comunicación NFC.

2.4.2.1 Android Studio

Android Studio es el entorno de programación de aplicaciones para Android más completo ya que viene listo para usar con los SDK (Kits de desarrollo de software) necesarios para el desarrollo de aplicaciones de manera nativa, está basado en Intellij IDEA ya que junto con Google lo desarrollaron.

El entorno está diseñado específicamente para el desarrollo de aplicaciones para la plataforma móvil Android y Android Wear por lo que es preciso el haber elegido este entorno ya que las prestaciones que ofrece para el desarrollo de la aplicación móvil son muy grandes.

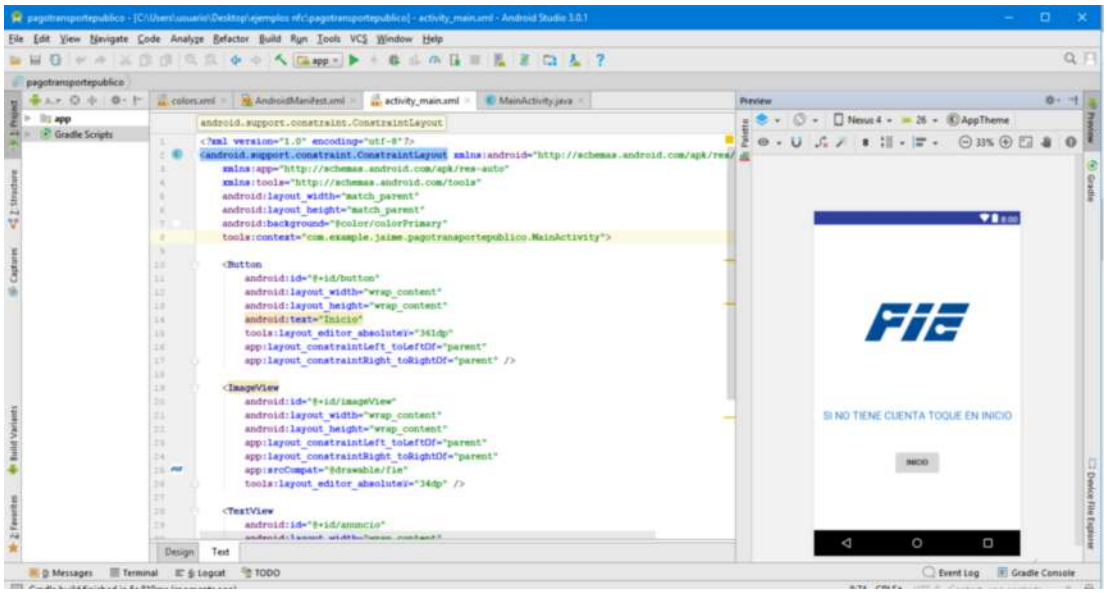


Figura 3-2 Entorno de Android Studio en Windows

Elaborado por: SUÁREZ, Jaime. 2018

2.4.2.2 Características de Android Studio

En la tabla 7-2 se describen las características principales de Android Studio.

Tabla 7-2 Características de Android Studio

CARACTERÍSTICAS	DESCRIPCIÓN
Editor de código inteligente	El entorno es capaz de completar códigos avanzados, analizarlos y refactorizarlos automáticamente.
Emulador de dispositivos Android	Se puede ejecutar las aplicaciones desarrolladas de forma rápida en el emulador que incluso tiene soporte para Android Wear.
Perfiladores de rendimiento	Proporcionan estadísticas en tiempo real de la CPU, memoria y actividad de red de la aplicación desarrollada.
Requisitos mínimos de hardware	Memoria RAM: 2GB Procesador: Intel 64 bits Espacio en Disco: 2GB espacio libre en disco
Requisitos de software	Sistema Operativo: Windows 7/8/10, Mac OSX 10.8.5 o superior, Linux GNOME o KDE Desktop Java 8

Elaborado por: SUÁREZ, Jaime, 2018

2.5 Requerimientos de hardware

2.5.1 Elección lector/escritor NFC

La selección de este hardware es muy importante por su participación en el proyecto, además de ser un punto crítico para la seguridad de todo el sistema.

En el país no existe una variedad de estos por lo que es casi nulo el mercado de lectores/escritores NFC por lo se tiene que recurrir al mercado internacional y mediante la comparación de características de hardware como de software para seleccionar el lector/escritor adecuado, al igual que para las aplicaciones también se utiliza el método de observación.

2.5.1.1 Comparación entre los lectores/escritores NFC

Para identificar cuál de estos lectores/escritores se utiliza igualmente el método de escala de Likert donde a cada una de los lectores/escritores se le asigna valores del 1 al 3, en donde 1 representa no adecuado, 2 neutral y 3 muy adecuado, el mayor valor se pondera a la opción más recomendable de forma cualitativa, también es importante la parte cuantitativa por lo que estos valores son ponderados a porcentajes, esto se puede observar en la tabla 9-2, mientras que en la tabla 8-2 se observa las características de los principales Lectores/escritores existentes en el mercado.

Tabla 8-2 Cuadro comparativo de lectores/escritores NFC

Lector/escritor NFC	**SDK**	**Conexión**	**Tipos de tarjeta soportada**	**Precio (Incluido envíos, impuestos y nacionalización)**
ACS ACR122U	Incluido, fácil entendimiento	PCSC USB	Mifare NFC Tipo A NFC Tipo B FeLica NFC(ISO/IEC18092)	$209,58
Digital-Logic uFR Classic	No incluido, descargable, difícil entendimiento	USB	Mifare NFC Tipo A NFC Tipo B NFC(ISO/IEC18092)	$218,25

Continua

HID Omnikey 3121 USB Card Reader	No	USB, Serial	Mifare NFC Tipo A NFC Tipo B FeLica RFID	$119,69
OEM NFC RFID Reader Writer DL-533N CS	No incluido	USB	Mifare NFC Tipo A NFC Tipo B NFC(ISO/IEC18092)	$126,47

Elaborado por: SUÁREZ, Jaime, 2018.

Tabla 9-2 Cuadro comparativo de lectores/escritores NFC

LECTOR/ESCRITOR NFC	SDK	CONEXIÓN	TIPOS DE TARJETA	PRECIO	TOTAL	PORC
ACS ACR122U	3	3	3	2	11	91,67%
Digital-Logic uFR Classic	2	1	3	2	8	66,67%
HID Omnikey 3121 USB Card Reader	1	2	3	3	9	75%
OEM NFC RFID Reader Writer DL-533N CS	2	1	3	2	8	66,67%

Elaborado por: SUÁREZ, Jaime, 2018

De acuerdo a los requisitos del proyecto, la opción más indicada de lector/escritor para el desarrollo del mismo es el ACS ACR122U, debido a su capacidad de lectura de tipos de tarjetas

NFC, además que cuenta con conexión PCSC la cual facilita el intercambio de datos entre el lector/escritor y la aplicación de escritorio.

2.5.1.2 Lector/Escritor NFC ACS ACR122U

El lector/Escritor NFC ACS ACR122U que se lo puede ver en la Figura 4-2, es el encargado en leer datos en la interacción del usuario con el sistema en el prototipo, primordialmente es usado en la lectura de tramas que son enviadas desde el teléfono móvil y enviar datos igualmente hacia el teléfono por lo que es uno de los puntos principales del prototipo.

Para que sean recibidos los datos de la aplicación móvil por el lector/escritor NFC ACS ACR122U se escriben las tramas de datos en una APDU, los valores de estos están especificados en el API del lector/escritor, es importante que según la arquitectura NFC escogida se escriba el APDU.

Figura 4-2 Lector/Escritor ACS ACR122U

Fuente: https://www.acs.com.hk/en/products/3/acr122u-usb-nfc-reader/

2.5.1.2.1 Datos Técnicos del lector/escritor ACS ACR122U

La tabla 10-2 muestra las características técnicas del lector/escritor NFC antes descrito.

Tabla 10-2 Datos Técnicos del Lector/Escritor NFC ACS ACR122U

CARACTERÍSTICAS	DATOS TÉCNICOS
Dimensiones	98mmx65mmx12.8mm
Peso	70 gr.
Certificaciones	ISO 18092, ISO 14443, PC/SC, CCID, EN60950/ISO 60950, CE, FCC, MIC, KC, VCCI, RoHS 2, USB Full Speed, Microsoft WHQL
Interfaz	USB Full Speed
Distancia de operación	Hasta 50mm
Suministro de Voltaje	5V DC
Suministro de Corriente	200mA(operación); 50mA(standby); 100mA(normal)
Temperatura de Operación	0-50 ºC
Frecuencia de Operación	13,56 MHz
Soporte de Tarjetas Inteligentes	ISO 14443 Tipo A y B MIFARE FeliCa 4 tipos de etiquetas NFC (ISO/IEC18092)
Sistemas Operativos Soportados	Windows Win CE 5.0 y 6.0 Linux Mac OS Android 3.1 y siguientes

Elaborado por: SUAREZ, Jaime, 2018.

2.6 Diseño lógico del prototipo

Para el diseño lógico del prototipo se va a utilizar las aplicaciones a desarrollarse, además del servidor Apache y la base de datos MySQL.

Se eligió como servidor a Apache ya que es versátil porque es de código abierto lo que es primordial en el momento de desarrollar aplicaciones que necesiten de un servidor y los módulos que posee se pueden configurar para que la seguridad de los datos que se manipulen no sea vulnerada.

Y se escogió para la base de datos a MySQL ya que trabaja de muy buena manera con el servidor Apache y de igual forma es de código abierto; NetBeans que es el entorno de desarrollo para la aplicación de escritorio tiene integración con MySQL lo que facilita el desarrollo de esta.

2.6.1 Servidor Apache

Es uno de los servidores más usados en el planeta ya que cuenta con una arquitectura modular, en donde el servidor cuenta con una parte de core y diferentes módulos los cuales son altamente configurables. (Wikipedia, 2012, https://es.wikipedia.org/wiki/Servidor_HTTP_Apache)

El servidor va a permitir que las aplicaciones tanto de la estación como las móviles se comunique con la base de datos, este va a estar a la espera de peticiones de los usuarios o administradores/operadores para acceder a la base de datos. Además, en su configuración se encuentran los scripts escritos en PHP, los cuales van a ser necesarios para que la aplicación móvil pueda realizar las consultas a la base de datos.

Estos scripts tienen una dirección de Localizador Uniforme de Recursos (URL) en el servidor para que puedan ser identificados y utilizados, en su código debe estar específicadas las consultas hacia la base de datos para que la aplicación móvil funcione a través del internet, además de que esta envía objetos en formato de Notación de Objetos de JavaScript (Json) y el script los recibe y traduce en un formato de texto plano para que la consulta a la base de datos se realice de manera correcta.

En este caso se realizará dos scripts, uno para consulta de saldos y el otro para consulta del historial de viajes del usuario.

2.6.1.1 Módulos Apache

Los módulos son una parte importante en el servidor Apache ya que entregan ciertos funcionamientos para este, estos se deben configurar de acuerdo a la forma en la cual se va a emplear el servidor, en la tabla 11-2 se describen algunos de estos.

Tabla 11-2 Módulos más conocidos y usados en Apache Server

MÓDULO	DESCRIPCIÓN
mod_ssl_	Comunicación con TLS
mod_cband_	Limita ancho de banda
mod_security	Filtra información a nivel de capa de aplicación
mod_access	Control de acceso al servidor
mod_auth	Confirma autentificaciones al servidor
mod_deflate	Compresión con el algoritmo deflate
mod_rewrite	Reescribe direcciones

Elaborado por: SUAREZ, Jaime, 2018.

2.6.2 MySQL

MySQL es una base de datos de código abierto desarrollada por Oracle, es usada por sitios web conocidos como Facebook, Twitter, YouTube entre otros sitios, ya que MySQL ofrece robustez, rendimiento y su uso no es complicado.(Oracle, 2017, https://www.oracle.com/ve/mysql/index.html)

En esta base de datos se van a almacenar los datos de los operadores y de los usuarios, así como detalles de saldos, contraseñas y registros de las operaciones realizadas por los operadores de la plataforma.

2.6.2.1 Características de MySQL

MySQL al ser usado por varias de las empresas más grandes del internet, ya que tiene características esenciales como la versatilidad que tiene al ser programada en los lenguajes de programación existentes.

En la tabla 12-2 se describen algunas de las características importantes de MySQL.

Tabla 12-2 Cuadro de características de MySQL

CARACTERÍSTICAS	DESCRIPCIÓN
Portabilidad	Usa Autocnf, GNU, Libtool
API	C, C++, Java, Ruby, Python, PHP, Perl
Conectividad	Socket TCP/IP, named pipes, socket Unix.
Idioma	Todos los idiomas incluyendo el latín
Registros	Longitud fija y variable
Funciones y Sentencias	SQL y ODBC normalmente sin reserva de memoria para consultas
Seguridad	Datos encriptados cuando se conecta con el servidor

Elaborado por: SUAREZ, Jaime, 2018.

2.6.2.2 Sentencias y Funciones MySQL

MySQL tiene soporte para sentencias y funciones con las sintaxis SQL y ODBC, además de comandos propios de MySQL, con esto se controla los datos en nuestras tablas y sus relaciones, los más importantes son descritos en la tabla 13-2.

Tabla 13-2 Sentencias y funciones de MySQL

SENTENCIA O FUNCIÓN	DESCRIPCIÓN
SELECT	Consulta de datos
WHERE	Se usa para incluir condiciones en la consulta a realizar
INSERT	Inserta datos en una tabla

Continua

UPDATE	Actualiza datos en una tabla
DELETE	Borra datos en una tabla
ORDER BY	Ordena resultados de una consulta

Elaborado por: SUAREZ, Jaime, 2018.

2.6.3 Topología de la red

La red va a estar formada por un servidor con una base de datos que va a estar conectada a la aplicación de escritorio por un enlace en el cual se van a intercambiar datos como se muestra en la figura 5-2, este enlace al ser implementado en un escenario real debería estar asegurado a través de una Red Virtual Privada (VPN) para impedir los ataques de Hombre en el Medio (MITM) remotos, como se está realizando en un escenario de pruebas y hay que comprobar la seguridad en el peor de los casos esto no se lo realizará .

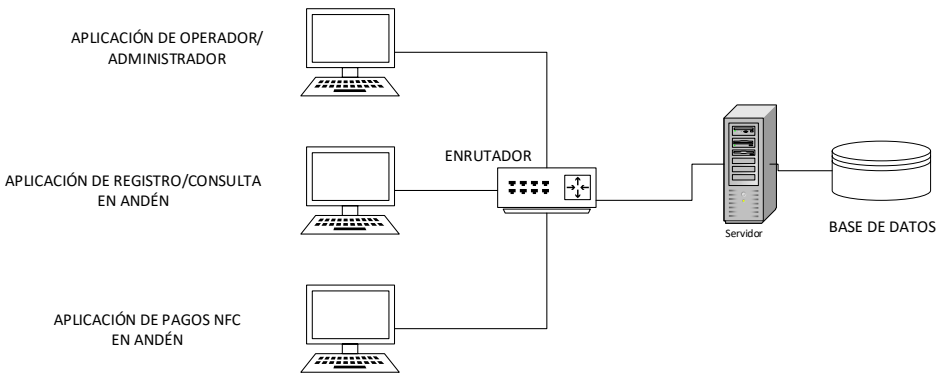


Figura 5-2 Topología de la red del prototipo

Elaborado por: SUÁREZ; Jaime, 2018.

2.7 Diseño e implementación del prototipo

2.7.1 Desarrollo e implementación del servidor y base de datos

Para la implementación del servidor y base de datos se optó el software XAMPP el cual contiene al servidor Apache y MySQL en un solo paquete y con todas las funcionalidades y opciones que ofrecen cada uno, este software se lo puede descargar de la página principal de apache friends.

2.7.1.1 Diseño e Implementación de la base de datos en MySQL

Para realizar de una forma práctica y sencilla la base de datos se dispone del software MySQL Workbench, en donde a través de diagramas Entidad-Relación se crea las variables y las tablas, así como las relaciones entre tablas. En el Anexo D se puede observar las tablas creadas para el prototipo con su respectiva descripción.

2.7.1.2 Implementación del servidor Apache

El servidor Apache en XAMPP ya viene implementado, pero hace falta incluir varias cuestiones de seguridad que robustezcan la comunicación que tiene el servidor con las aplicaciones de escritorio.

Apache por defecto muestra la versión que está corriendo y los módulos que están instalados en el servidor además del sistema operativo en el cual esta desplegado por lo que es importante ocultar estos parámetros a los usuarios maliciosos. Esto se realiza agregando dos directivas en el archivo httpd.conf de apache los cuales son:

- ServerSignature Off
- ServerTokens Prod

Para que el servidor sea más seguro es necesario instalar el mod_security este módulo ayuda en muchos factores de seguridad como: Prevención de ataques Null byte, limita la memoria de subida, la identidad del servidor puede ser enmascarada, valida la codificación Unicode entre otros.

Otro aspecto importante es disminuir el tiempo de espera en Apache es de 300 segundos por defecto por lo que se debe disminuir este valor para prevenir ataques de fuerza bruta, al igual que se debe limitar el número de peticiones máximo ya que en el servidor Apache este número por defecto se encuentra en ilimitado por lo que en este caso al no hacer subidas al servidor de datos de más de 1 MegaByte se lo puede fijar en 1 MB o inclusive más pequeño.

Apache por defecto muestra archivos y directorios por lo que es primordial ocultarlos eso se logra en el archivo de configuración de Apache agregando en la directiva Option -Indexes como se muestra en la Figura 6-2.

```
#
# "C:/xampp/cgi-bin" should be changed to whatever your ScriptA
# CGI directory exists, if you have that configured.
#
<Directory "C:/xampp/cgi-bin">
    AllowOverride All
    Options -Indexes
    Require all granted
</Directory>
```

Figura 6-2 Modificación de fichero conf de Apache para ocultar archivos y directorios

Elaborado por: SUÁREZ; Jaime, 2018.

2.7.2 Estructura de la aplicación de estación de administrador/operador.

2.7.2.1 Control de acceso al operador/administrador

En esta etapa se controlará el acceso a la aplicación a través de inicio de sesión con usuario y contraseña, los que están almacenados en la tabla OPERADOR de la base de datos.

Se definió dos tipos de usuario que son:

- Administrador: el cual tiene potestad para manipular los datos de usuario, registros de saldos y pagos, al igual que crear y modificar operadores y sus contraseñas.
- Operador: el cual tiene la potestad para crear, modificar y borrar usuarios, saldos y pagos.

En la Figura 7-2 se observa que existe una condición en donde se compara el usuario y contraseña según el tipo con el cual se encuentren almacenados en la base de datos, y si no cumple con ninguna de las condiciones saldrá un mensaje de que el usuario no existe, es necesario que la conexión se cifre por lo que también al enviar los datos antes mencionados hacia la base de datos estos se encriptan.

```
    public void acceder(String usuario, String pass) {
        String cap = "";
        String sql = "SELECT * FROM operador WHERE USUARIO='" + usuario + "' && CONTRASEÑAOP='" + pass + "'";
        try {
            Statement st = cn.createStatement();
            ResultSet rs = st.executeQuery(sql);
            while (rs.next()) {
                cap = rs.getString("tipouser");
                nombreOpe = rs.getString("nombop");
                idOperador = rs.getString("ope_id");
            }
            if (cap.equals("ADMINISTRADOR")) {
                this.setVisible(false);
                JOptionPane.showMessageDialog(null, "Bienvenido" );
                ventanaadmin ingresos = new ventanaadmin();
                ingresos.setVisible(true);
                ingresos.pack();
            }
            if (cap.equals("OPERADOR")) {
                this.setVisible(false);
                JOptionPane.showMessageDialog(null, "Bienvenido " + nombreOpe);
                ventanaoper ingreso = new ventanaoper();
                ingreso.setVisible(true);
                ingreso.setUsuarioLog(nombreOpe);
                ingreso.setIdUserLog(idOperador);
                ingreso.pack();

            }
```

Figura 7-2 Parte de la codificación de la pantalla de Ingreso en la aplicación de estación de Operador/Administrador.

Elaborado por: SUÁREZ; Jaime, 2018.

La Figura 8-2 nos muestra cómo se conecta la aplicación al servidor, la persona que se autentica debe estar registrada en la base de datos ya sea como operador o administrador teniendo cada uno de ellos privilegios diferentes, esta conexión se la realiza a través de la librería com.mysql.jdbc.Driver la cual debe estar instalada en el paquete de recursos de NetBeans, en el código se debe indicar el nombre de la base de datos junto con el usuario y contraseña de la mismo además de la dirección ip del servidor en la que se encuentra alojada.

```
 *
 * @author Jaime
 */
public class conectar {
   Connection conectar=null;
public Connection conexion(){
try {
  Class.forName("com.mysql.jdbc.Driver");
  conectar=DriverManager.getConnection("jdbc:mysql://                    ,"");
} catch (Exception e)   {
  System.out.print(e.getMessage());
}
return conectar;
}
}
```

Figura 8-2 Código para hacer la conexión de la aplicación hacia la base de datos

Elaborado por: SUÁREZ; Jaime, 2018.

Para que la autenticación al servidor sea segura se creó una clase en la cual se va a encriptar y desencriptar los datos con una llave simulando una conexión SSL, en donde se tiene una llave definida, esta llave se va a encriptar con SHA-1 y SHA-256 esto para probar el rendimiento en el procesamiento y elegir una de las dos, después este se encripta en 3DES con el texto plano para pasar a codificarlo en base64 y une en un solo vector de bytes la llave con el texto codificado, parte de este código se lo puede observar en la Figura 9-2.

```
MessageDigest md = MessageDigest.getInstance("SHA-1");
byte[] digestOfPassword = md.digest(secretKey.getBytes("utf-8"));
byte[] keyBytes = Arrays.copyOf(digestOfPassword, 24);
```

Figura 9-2 Código que realiza la encriptación de los datos

Elaborado por: SUÁREZ; Jaime, 2018.

Esta clase va a encriptar los datos uno a uno que se envíen en cualquier instancia de la aplicación de la estación de Operador/Administrador.

La ventana a la cual ingresa el operador va a tener 3 botones los cuales son de recarga de saldo, revisar el historial de pagos o viajes y revisar el historial de recargas.

Se definió que el operador solo tenga que ingresar el número de cédula para realizar la recarga, al igual que para realizar las acciones de consulta de recargas y de viajes, con esto se limita el uso de la base de datos por parte del operador, previniendo fugas de información. Igualmente, todos estos datos están encriptados y son desencriptados cuando la base de datos devuelve una respuesta, esto lo hace la aplicación.

El administrador por otro lado va a tener las mismas facultades que el operador, pero agregando las acciones de creación de operadores nuevos y la búsqueda y eliminación de usuarios.

2.7.2.2 Aplicación de pagos en estación a través de NFC

Los pagos serán realizados por una aplicación individual la cual permitirá una mayor seguridad ya que ningún operador puede tener acceso a la aplicación, esta funciona automáticamente poniendo al lector/escritor NFC en modo pasivo cada vez que se realiza un pago por lo que estará a la escucha de un teléfono inteligente que vaya a realizar el pago, los pagos son controlados como un inicio de sesión con usuario y contraseña, en donde el usuario es el número de cédula y la contraseña es definida por el usuario.

La contraseña también va a ser encriptada través de un hash con una función SHA-1 para mayor seguridad. En esta aplicación se consideró el uso de tres botones los cuáles permitirán a los usuarios manejar de mejor manera sus pagos.

El primer botón INICIO es el encargado de iniciar al lector/escritor NFC en modo pasivo. Al presionar el botón la aplicación inicia el reconocimiento del lector/escritor NFC, en la Figura 10-2 se puede observar como la aplicación inicia el reconocimiento de terminales o en otras palabras de los lectores conectados, después que ya se reconoce al lector/escritor NFC este se pone en modo pasivo esperando que un teléfono con la aplicación y el AID correspondiente se acerque para realizar el pago, después que se realiza el pago no es necesario presionar el botón inicio ya que se activa automáticamente después de realizado el pago.

```
try {
    // Display the list of terminals
    TerminalFactory factory = TerminalFactory.getDefault();
    List<CardTerminal> terminals = factory.terminals().list();
    System.out.println("Terminals: " + terminals);
    System.out.println("Terminals count: " + terminals.size());//
    CardTerminal terminal = terminals.get(0);
    terminal.waitForCardPresent(0);
    Card card = terminal.connect("*");
    System.out.println("Card: " + card);
    CardChannel channel = card.getBasicChannel();
    byte[] AID = hexStringToBytes("              "); //APP ANDROID
```

Figura 10-2 Código de reconocimiento del lector NFC

Elaborado por: SUÁREZ; Jaime, 2018

Después que se realizó la conexión con la aplicación móvil la aplicación de escritorio empieza a enviar el APDU de respuesta con el formato definido en el ISO/IEC 7816-4 con lo que también se pone condiciones por si llega un tráiler mandatorio (SW) con procesos de advertencia o con errores de ejecución como se observa en la Figura 11-2.

```
ResponseAPDU response = channel.transmit(new CommandAPDU(0x00, 0xA4, 0x04, 0x00, AID));
assert response.getSW() == 0x9000;
//  ResponseAPDU response = channel.transmit(new CommandAPDU(hexStringToBytes("FFCA000004")));
System.out.println("Response: " + response.toString());
Thread.sleep(1000);
if (response.getSW1() == 0x63 && response.getSW2() == 0x00) {
    System.out.println("Failed");
}
```

Figura 11-2 Código de APDU de respuesta

Elaborado por: SUÁREZ; Jaime, 2018.

Si el SW es 0X9000 significa que se empezará a realizar el intercambio de datos, entonces la aplicación enviará a la aplicación móvil la respuesta (Response) a través del canal que está utilizando el lector/escritor NFC para la comunicación como se ve en la figura 12-2, después los datos hexadecimales se convierten en letras (String) para que se pueda realizar la comparación de estos datos con los que se encuentran en la base de datos y validar el pago o no.

```
String pp = bytesToHexString(response.getData());
System.out.println("UID: " + pp);
System.out.println("data: " + HexStringToString(pp));
response = channel.transmit(new CommandAPDU(0x00, 0x00, 0x00, 0x00));
//  System.out.println("answer: " + response.toString());
byte r[] = response.getData();
```

Figura 12-2 Conversión de bytes a String y envío de mensaje de respuesta

Elaborado por: SUÁREZ; Jaime, 2018.

2.7.2.3 Aplicación en estación de registro de usuarios y consulta de historial y saldo.

Como tercera parte se tiene la aplicación en la cual el usuario puede registrarse y consultar su saldo e historial de pagos, se la realizó de esta manera y no en la aplicación móvil para que no exista ninguna tercera parte involucrada es decir una operadora de telefonía móvil, además el usuario no necesitará ninguna conexión inalámbrica como wifi o datos móviles para consultar su saldo o registrarse.

Esta aplicación constará de tres botones, el primer botón es el de consulta de saldo, el cual utiliza el lector/escritor NFC para realizar la consulta del saldo del usuario, por lo que para realizar esta acción el usuario debe presionar el botón y acercar el teléfono al lector/escritor NFC y se

desplegará un panel con el saldo del usuario. En este caso la lectura iniciará cuando el teléfono inteligente envía a la aplicación un mensaje de comando con el APDU y el AID respectivo además de los datos que son la cédula y contraseña del usuario para que se validen en la base de datos, al realizarse la validación de la información la aplicación despliega un cuadro con el nombre del usuario, la contraseña y el saldo correspondiente.

El botón historial realiza la misma acción que el botón consulta, pero con la diferencia de que aparecerá una nueva ventana con el historial de pagos del usuario que consulta, este funcionará de la misma manera que los demás botones comprobando el AID de la aplicación y enviado los mensajes de comando y respuesta APDU para iniciar el intercambio de datos entre el teléfono inteligente y el lector/escritor NFC.

El botón de registro abrirá una ventana para que el usuario pueda registrarse previo a la descarga de la aplicación móvil, esta ventana constará de los siguientes botones: lectura de teléfono, limpiar pantalla, guardar datos, atrás y salir. De igual manera utilizará la conexión al lector/escritor NFC para activarse por lo que esta pantalla aparecerá solo si el usuario usa la aplicación y acerca el teléfono con los datos de registro. El usuario podrá comprobar en la ventana si sus datos son correctos y de ser así tendrá que pulsar el botón guardar datos para que estos se almacenen en la base de datos y de no ser correctos puede limpiar la pantalla para realizar el proceso otra vez.

Los datos que serán enviados al guardar el registro serán encriptados por la aplicación lo que dará seguridad a estos sensibles. Igualmente, los datos que serán recibidos por la aplicación móvil a través del lector NFC serán desencriptados para que el usuario pueda ver si estos son correctos en la pantalla de la aplicación de estación.

El proceso de lectura de los datos desde el teléfono hacia la aplicación no tiene demora ya que esta codificado para que la aplicación móvil utilice solo un bloque de los 16 existentes en la arquitectura HCE lo cual hace que esta lectura de datos sea instantánea. Al igual que en la anterior aplicación se usará un constructor de palabras (String Builder) para cambiar los datos de Hexadecimal a texto plano, esta codificación se la puede ver en la Figura 13-2.

```
public static String HexStringToString(String arg) {
    StringBuilder output = new StringBuilder();
    for (int i = 0; i < arg.length(); i += 2) {
        String str = arg.substring(i, i + 2);
        output.append((char) Integer.parseInt(str, 16));
    }

    return output.toString();
}
```

Figura 13-2 Código de Constructor de String

Elaborado por: SUÁREZ; Jaime, 2018.

2.7.3 Estructura de la aplicación móvil

La aplicación móvil se desarrolló con la arquitectura de emulación de tarjeta inteligente basada en host, en este caso se tiene cuatro botones principales los cuales permitirán el pago y registro del usuario a través de la tecnología NFC y por otro lado permitirá igualmente la consulta de saldos e historial a través del internet.

2.7.3.1 Android Manifest

Para que la aplicación funcione correctamente es necesario agregar al Android Manifest el permiso para que la aplicación acceda al servicio de internet y NFC, Android Manifest es un archivo de formato xml en el cual se da los parámetros generales de funcionamiento de la aplicación, en este también se referencian las actividades que posee la aplicación y se puede modificar el nombre que va a llevar esta, su código se puede ver en la Figura 14-2.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.sebastian.hceynfcfinal">

    <uses-permission android:name="android.permission.NFC" />
    <uses-permission android:name="android.permission.INTERNET" />
    <application
        android:allowBackup="true"
        android:icon="@mipmap/log"
        android:label="Pago NFC"
        android:roundIcon="@mipmap/log"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <activity android:name=".vtn_registro"    >

        </activity>
```

Figura 14-2 Parte Inicial del Android Manifest

Elaborado por: SUÁREZ; Jaime, 2018.

Es muy importante mencionar que cada layout (plano de diseño en Android) creado en la aplicación debe ser referenciado en el Android Manifest para que la aplicación tenga su correcto funcionamiento y no bote errores el momento de correrla en el teléfono inteligente, esto debe hacerse referenciando estos layout a través de un activity por cada uno.

Otro parámetro importante es el de declarar los servicios en este caso se debe implementar el servicio de emulación de tarjeta inteligente con su servicio de host de APDU, como se muestra en la figura 15-2.

```
<service
    android:name=".MyHostApdupago"
    android:exported="true"
    android:permission="android.permission.BIND_NFC_SERVICE" >
    <intent-filter>
        <action android:name="android.nfc.cardemulation.action.HOST_APDU_SERVICE" />
    </intent-filter>

    <meta-data
        android:name="android.nfc.cardemulation.host_apdu_service"
        android:resource="@xml/apduservicepago" />
</service>
```

Figura 15-2 Declaración de Servicios en el Android Manifest

Elaborado por: SUÁREZ; Jaime, 2018.

2.7.3.2 Transceiver y Adaptador IsoDep

Primeramente, para que la comunicación entre el terminal móvil inteligente y el lector/escritor NFC se realizó un Adaptador IsoDep y un Transceiver IsoDep, IsoDep es una herramienta que posee Android la cual permite la comunicación a través de ISO-14443-4, este permite el intercambio de los mensajes APDU y de los AID. En la Figura 16-2 se detalla el código con el cual funciona IsoDep también se observa cómo se declara el APDU y el AID para después unirlos en un vector.

El AID es vector de hasta 16 bytes que identifican la aplicación, pueden ser la bien conocidas como son las AID de redes de pagos de Visa o MasterCard que son reconocidas por casi todos los lectores NFC existentes en el mercado y las AID públicas. En este caso se realizó un AID propia siguiendo los lineamientos de la ISO/IEC 7816-4 para evitar colisiones con otras aplicaciones. (Android, 2014, https://developer.android.com/guide/topics/connectivity/nfc/hce)

El uso de AID también permite que solo las aplicaciones de escritorio desarrolladas puedan recibir los datos de la aplicación móvil, evitando robos de información con el uso de lectores o incluso con otros teléfonos inteligentes que tengan instalada algún tipo de aplicación que lea datos de NFC que en la tienda de aplicaciones existen en gran cantidad. Por lo que este vector de bytes también se lo declaro en las aplicaciones de escritorio para que estas se logren comunicar con la aplicación móvil.

```
public IsoDepTransceiver(IsoDep isoDep, OnMessageReceived onMessageReceived) {
    this.isoDep = isoDep;
    this.onMessageReceived = onMessageReceived;
}

private static final byte[] CLA_INS_P1_P2 = { 0x00, (byte)0xA4, 0x04, 0x00 };
private static final byte[] AID_ANDROID = { (byte)                                    };

private byte[] createSelectAidApdu(byte[] aid) {
    byte[] result = new byte[6 + aid.length];
    System.arraycopy(CLA_INS_P1_P2, i: 0, result, i1: 0, CLA_INS_P1_P2.length);
    result[4] = (byte)aid.length;
    System.arraycopy(aid, i: 0, result, i1: 5, aid.length);
    result[result.length - 1] = 0;
    return result;
}
```

Figura 16-2 Transceiver IsoDep formando un vector de datos con el APDU y el AID.

Elaborado por: SUÁREZ; Jaime, 2018.

El transceiver IsoDep crea la trama de datos, los controla como también controla los mensajes APDU para su transmisión al igual que para la recepción, su codificación se la observa en la Figura 17-2.

```
@Override
public void run() {
    int messageCounter = 0;
    try {
        isoDep.connect();
        byte[] response = isoDep.transceive(createSelectAidApdu(AID_ANDROID));
        while (isoDep.isConnected() && !Thread.interrupted()) {
            String message = "Message from IsoDep " + messageCounter++;
            response = isoDep.transceive(message.getBytes());
            onMessageReceived.onMessage(response);
        }
        isoDep.close();
    }
    catch (IOException e) {
        onMessageReceived.onError(e);
    }
}
```

Figura 17-2 Código de Controlador de transmisión y recepción de mensajes APDU

Elaborado por: SUÁREZ; Jaime, 2018.

El adaptador IsoDep en cambio ayudará a que las tramas de datos se adapten ya sea en transmisión leyendo los datos en texto plano para después pasarlos a bytes en el transceiver y de igual manera en la recepción al adaptar las respuestas que recibe el transceiver y que la aplicación entienda que responder, parte de su código se observa en la Figura 18-2.

```
public IsoDepAdapter(LayoutInflater layoutInflater) { this.layoutInflater = layoutInflater; }

public void addMessage(String message) {
    messageCounter++;
    messages.add("Message [" + messageCounter + "]: " + message);
    notifyDataSetChanged();
}

@Override
public int getCount() { return messages == null ? 0 : messages.size(); }

@Override
public Object getItem(int position) { return messages.get(position); }

@Override
public long getItemId(int position) { return 0; }

@Override
public View getView(int position, View convertView, ViewGroup parent) {
    if (convertView == null) {
        convertView = layoutInflater.inflate(android.R.layout.simple_list_item_1, parent, attachToRoot: false);
    }
    TextView view = (TextView)convertView.findViewById(android.R.id.text1);
    view.setText((CharSequence)getItem(position));
```

Platform and Plugin Updates

Figura 18-2 Código inicial del Adaptador IsoDep.

Elaborado por: SUÁREZ; Jaime, 2018.

2.7.3.3 Host APDU

Se realizó un Host el cual va a agregar los datos que se van a enviar al lector/escritor NFC transformándolos de texto plano a bytes, en el caso del pago se transformará la cédula del usuario con su contraseña como se observa en la Figura 19-2, para el otro caso del registro de los datos se realiza el mismo procedimiento.

```
private byte[] getNextMessage1() {

    return ("*"+strcedula1+"*"+strpassword+"*  ").getBytes();

}

private byte[] getNextMessage0() {

    return ("*No hay info* ").getBytes();

}
```

Figura 19-2 Código que convierte el texto plano en bytes.

Elaborado por: SUÁREZ; Jaime, 2018.

La clase Host APDU también agrega la información de cabecera a los datos que van a ser enviados, es decir el AID y el APDU de comando como se nota en la Figura 20-2, estos datos de la cabecera son tomados del transceiver IsoDep.

```
private boolean selectAidApdu(byte[] apdu) {
    return apdu.length >= 2 && apdu[0] == (byte)0 && apdu[1] == (byte)0xa4;
}
```

Figura 20-2 Código que retorna el APDU con su AID.

Elaborado por: SUÁREZ; Jaime, 2018.

2.7.3.4 Actividades (Activities)

Las actividades se dividen en dos en Android Studio la primera parte son llamadas layout que son la parte en donde se indica cómo se va a ver la aplicación es decir la parte gráfica y cada layout tiene su clase en la cual se va a programar como va a funcionar cada elemento agregado a cada

layout. La aplicación planteada tiene cinco actividades las cuales permitirán el funcionamiento adecuado de la misma y facilitarán al usuario el uso de la misma.

2.7.3.4.1 Layout activity_main, Clase MainActivity

Aquí se encuentra la pantalla principal la cual permitirá elegir al usuario a través de cuatro botones que acción desea.

La clase MainActivity se codifico que realizará cada botón, es decir la pantalla principal será el enlace para layout elegido, por ejemplo, si el usuario desea realizar una consulta de saldo deberá seleccionar saldo para que se inicie la actividad correspondiente a saldo y así con los demás botones. Para que la aplicación se inicie se puso una condición en la cual la aplicación iniciará solo si esta encendido el NFC del teléfono inteligente, una parte de esta clase se observa en la Figura 21-2.

```
        if(adapter!=null)
        {
            if(adapter.isEnabled())
            {
//Nfc settings are enabled

                btn_con.setOnClickListener((view) → {

                        startActivity(new Intent( packageContext: MainActivity.this,saldo.class));
                        finish();
                });

                btn_his.setOnClickListener((view) → {

                        startActivity(new Intent( packageContext: MainActivity.this,hist.class));
                        finish();
                });


                btn_registro.setOnClickListener((view) → {

                    startActivity(new Intent( packageContext: MainActivity.this,vtn_registro.class));
                    finish();
            });


                btn_pagar.setOnClickListener((view) → {
                        startActivity(new Intent( packageContext: MainActivity.this,vtn_pago.class));
```

Figura 21-2 Parte de la Clase MainActivity.

Elaborado por: SUÁREZ; Jaime, 2018.

2.7.3.4.2 Layout activity_vtn_pago, Clase vtn_pago

Esta actividad es la que controlará los pagos desde el teléfono inteligente hacia el lector, su layout presenta dos editores de texto en los cuales el usuario podrá poner su cedula y contraseña, estos datos también pueden ser tomados si el usuario se registró previamente en el teléfono inteligente

Para controlar esta actividad se utilizó la clase llamada vtn_pago, en donde primero se comprobará que estos campos no se encuentren vacíos, si no lo están el texto plano de estos campos serán enviados al host APDU y al Transceiver IsoDep para que sean convertidos en tramas de datos que el lector/escritor NFC pueda recibirlos al igual que la aplicación de escritorio, estos datos son enviados al presionar en el botón Pago, pero solo la primera vez ya que HCE (Host Card Emulation) tiene la capacidad de seguir funcionando en segundo plano mientras el NFC del teléfono inteligente se encuentre prendido por lo que para realizar el pago basta con acercar el teléfono desbloqueado al lector/escritor NFC.

2.7.3.4.3 Layout activity_vtn_registro, Clase vtn_registro

Esta actividad permitirá realizar el registro de los usuarios a través de NFC, el layout consta de campos que deben ser llenados por el usuario, estos son Nombres, Apellidos, Cédula, Dirección, Teléfono y contraseña, estos datos son encriptados en 3DES a excepción de la contraseña en donde se utiliza un hash SHA-1. El proceso de encriptación inicia cuando el usuario presiona el botón registrar. La aplicación envía los campos encriptados uno a uno sumado una llave la cual esta encriptada en SHA-1 a esta sirve para desencriptar los datos en la aplicación de escritorio la cual también tiene esta llave sin la cual no podría desencriptar estos datos, el único campo que no es enviado con llave es el de la contraseña ya que no es necesario desencriptar por eso se la encripto con SHA-1 el cual no se puede desencriptar.

2.7.3.4.4 Layout activity_saldo, Clase saldo

Esta actividad permitirá realizar la consulta de saldo a través de internet, el layout de esta actividad utiliza los datos que se escribieron anteriormente en la actividad de pagos, el usuario solo tiene que presionar el botón de consulta y este le devolverá los datos del saldo de viajes restantes.

La clase saldo es la que controla el comportamiento del layout activity_saldo, en este se codifica el acceso a internet, a través de la compilación de Volley, este complemento es muy indispensable al realizar consultas a una base de datos remota ya que permite la manipulación de objetos Json el cual es un formato de intercambio de datos muy ligero, a través de este formato se van a realizar las consultas desde el teléfono inteligente hasta la base de datos en el servidor ya creado a través de internet, para ello también es necesario la creación de un script en php en donde se detalla la consulta a la base de datos y como recibirá el servidor estos objetos Json. Los scripts usados para esta comunicación se encuentran en el Anexo E

Como se ve en la Figura anterior se debe especificar el nombre de la base de datos, el usuario, y contraseña, de la misma manera hay que especificar la consulta hacia la base de datos, de acuerdo a las condiciones puestas el script devolverá un objeto Json, el cual va a ser interpretado por la aplicación móvil devolviendo el resultado del saldo.

En la codificación de la aplicación móvil es necesario crear una instancia para que la conexión hacia el internet funcione, en este caso se la llamó cargarWebService, aquí la aplicación coge los datos de la cédula del usuario, la encripta y hace la conexión con el URL en donde se encuentra el Script PHP, para después enviar esto en formato Json al script y posteriormente a la base de datos realizando la consulta, esta codificación se la puede observar en la Figura 22-2.

```
private void cargarWebService() {
    progreso=new ProgressDialog( context: this);
    progreso.setMessage("Consultando...");
    progreso.show();
    strusuario= e_usu.getText().toString();
    usucifrado=hash.sha1(strusuario);
    String url="http://192.168.0.100/bdmovil/consultasaldo.php?cedula="+usucifrado;
    jsonObjectRequest=new JsonObjectRequest(Request.Method.GET,url, jsonRequest: null, listener: this, errorListener: this);
    request.add(jsonObjectRequest);
}
```

Figura 22-2 Codificación de la conexión al Script PHP

Elaborado por: SUÁREZ; Jaime, 2018.

Es muy importante codificar de manera correcta la dirección ip que tiene el servidor ya que si esta se encuentra mal no podrá tener conexión el cliente con la base de datos, también es importante que el script php se encuentre en el servidor para que cada vez que exista una petición desde el usuario se realice la consulta en la base de datos sin ningún problema. En este caso la aplicación comparte el número de cédula de la persona que realiza los pagos con ese teléfono por lo que el usuario solo podrá consultar sus datos.

Para la consulta del historial se lo realizó de la misma manera, pero cambiando el uso de Json, en este caso se usó el vector Json ya que ahora se van a mostrar datos agrupados y no un solo dato por lo que también debe cambiar la consulta que se programa en el script php para que el funcionamiento sea el adecuado.

CAPÍTULO 3

3. RESULTADOS

3.1 Funcionamiento de la aplicación para teléfono inteligente.

Para el uso de la aplicación móvil es necesario que se realice la instalación en el teléfono inteligente con sistema operativo Android el cual debe ser de la versión 4.4 para arriba, ya que como se está utilizando el modo HCE (Emulación de Tarjeta con Host) y este modo se implementó en la versión 4.4 KitKat de Android.

En la Figura 1-3 se puede observar el icono de acceso directo a la aplicación de escritorio.

Figura 1-3 Acceso directo de la aplicación para realizar pagos de transporte público en Android

Elaborado por: SUAREZ, Jaime, 2018

Al ingresar a la app se visualizará 4 botones los cuáles están diferencias por un icono como se observa en la Figura 2-3, la aplicación tiene una restricción especial en donde esta solo se lanzará

si el NFC del dispositivo está encendido caso contrario la aplicación no podrá lanzarse por lo que es necesario encender el NFC y dejarlo encendido para que el momento de pagar el usuario no tenga que ingresar a la aplicación, solo tendrá que pasar el teléfono inteligente por el lector.

Figura **2-3** Pantalla Principal de la aplicación móvil para pagos de transporte Público a través de NFC.

Elaborado por: SUAREZ, Jaime, 2018

El primer paso que debe realizar un usuario es Registrarse por lo que en la pantalla principal elegirá el botón Registrarse, se abrirá una instancia en donde el usuario podrá ingresar los datos que se le pide, es importante que el usuario ingrese sus dos nombres y dos apellidos, además de una contraseña de ocho caracteres, esto se lo puede observar en la Figura 3-3, después el usuario tiene que presionar el botón Registrar en ese momento el teléfono inteligente debe estar hasta cuatro centímetros sobre el lector NFC para que este reciba los datos de registro, el uso de la aplicación de la estación para Registro se la explicará más adelante.

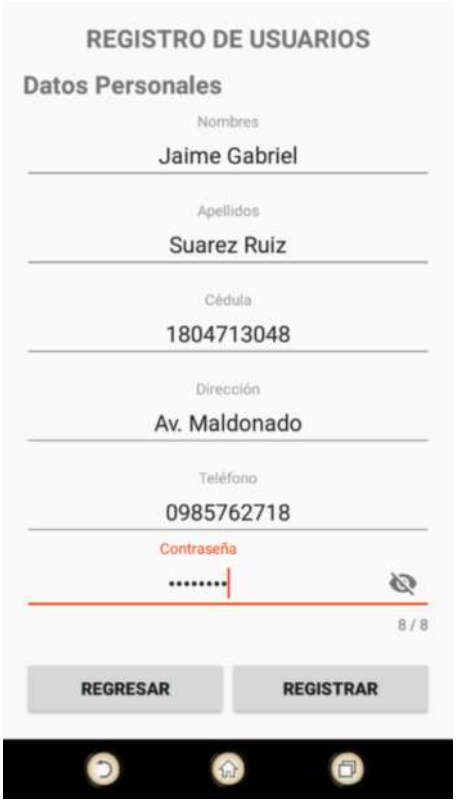


Figura 3-3 Pantalla de Registro de la aplicación móvil para pagos de transporte Público a través de NFC.

Elaborado por: SUAREZ, Jaime, 2018

El momento de que el usuario se registró la aplicación toma los datos de la cédula y su contraseña y son puestos automáticamente en la pantalla de Pagos, por lo que después de registrarse debe presionar el botón Regresar y volverá a la pantalla de inicio.

El siguiente botón es el de Pagos, al presionar este se encuentra una pantalla de inicio de sesión, pero ya con los datos escritos ya que estos son tomados del Registro como se mencionó anteriormente, en el sólo se debe presionar el botón pago para realizar los pagos de los viajes en transporte público esto solo la primera vez, después de esto es automático y solo necesita el usuario tener el teléfono inteligente prendido y desbloqueado para realizar el pago.

En la Figura 4-3 se puede observar la pantalla de Pago.

Figura 4-3 Pantalla de Pago de la aplicación móvil para pagos de transporte Público a través de NFC.

Elaborado por: SUAREZ, Jaime, 2018

El botón que sigue es el de Saldo en el cual el usuario puede consultar su saldo de viajes a través del internet, al ingresar se verá una pantalla como en la Figura 5-3 en donde el usuario tiene que presionar el botón Consultar Saldo y este se verá en la parte inferior.

Figura 5-3 Pantalla de Consulta de Saldo de la aplicación móvil para pagos de transporte Público a través de NFC.

Elaborado por: SUAREZ, Jaime, 2018

El último botón es el de Historial en el cual se puede observar el historial de los últimos 5 viajes, el funcionamiento es similar al de Saldo ya que sólo hay que presionar el botón de Consultar Historial para que se puede visualizar los viajes, esta pantalla se la puede ver en la Figura 6-3.

Figura 6-3 Pantalla de consulta de historial de viajes de la aplicación móvil para pagos de transporte Público a través de NFC.

Elaborado por: SUAREZ, Jaime, 2018

3.2 Funcionamiento de la aplicación de registro y pago en estaciones

La aplicación de Registro y pago en la estación está dividida en dos, la primera es la de Registro de Usuarios que además posee capacidad para realizar consultas de saldo a través de NFC y también consulta de historial de viajes de igual manera con la tecnología NFC, está aplicación está separada de la aplicación de pago en la cual solo se realiza los pagos automáticamente.

La pantalla principal de la aplicación de Registro se la puede observar en la Figura 7-3, esta consta de 3 botones, el más importante es el de Registro de usuarios, este botón interactúa con la aplicación móvil antes descrita, el registro funciona de la siguiente manera, primero el usuario debe llenar todos los campos de registro antes visto, después el usuario va a presionar el botón de Registro en la aplicación de la estación, aparecerá un ventana emergente pidiendo al usuario que acerque el teléfono al lector NFC.

Figura 7-3 Pantalla Principal de aplicación de registros y consultas en la estación.

Elaborado por: SUAREZ, Jaime, 2018

El usuario al acercar el teléfono enviará todos los datos que serán mostrados en la pantalla para que el mismo pueda verificar si están correctos, de ser correctos el usuario debe presionar el botón guardar datos para que estos sean insertados en la base de datos. En la Figura 8-3 se puede ver la pantalla de Registro de datos.

Figura 8-3 Pantalla de registro de datos de usuario en la aplicación de estación.

Elaborado por: SUAREZ, Jaime, 2018

El botón Consulta de Saldo permite que el usuario consulte el saldo en la estación, el usuario tiene que acercar el teléfono inteligente al lector NFC como si se tratará de realizar un pago, pero primero debe presionar el botón de Consulta de Saldo en la aplicación de estación para que el

lector NFC este en modo pasivo esperando por el teléfono, una vez realizado esto aparecerá el nombre del usuario y su saldo correspondiente en número de viajes como se puede ver en la Figura 9-3.

Figura 9-3 Pantalla de consulta de saldo de viajes de la aplicación de estación.

Elaborado por: SUAREZ, Jaime, 2018

El botón Consulta de Historial permite ver al usuario el historial completo de viajes que ha realizado el usuario, para su uso el usuario al igual que en la consulta de saldo debe acercar el teléfono al lector NFC, pero antes el usuario debe presionar el botón de consulta de historial en la aplicación de estación, luego de esto en la pantalla de la estación se desplegará el historial completo de los viajes que ha realizado el usuario desde el último hasta el primero como se observa en la Figura 10-3.

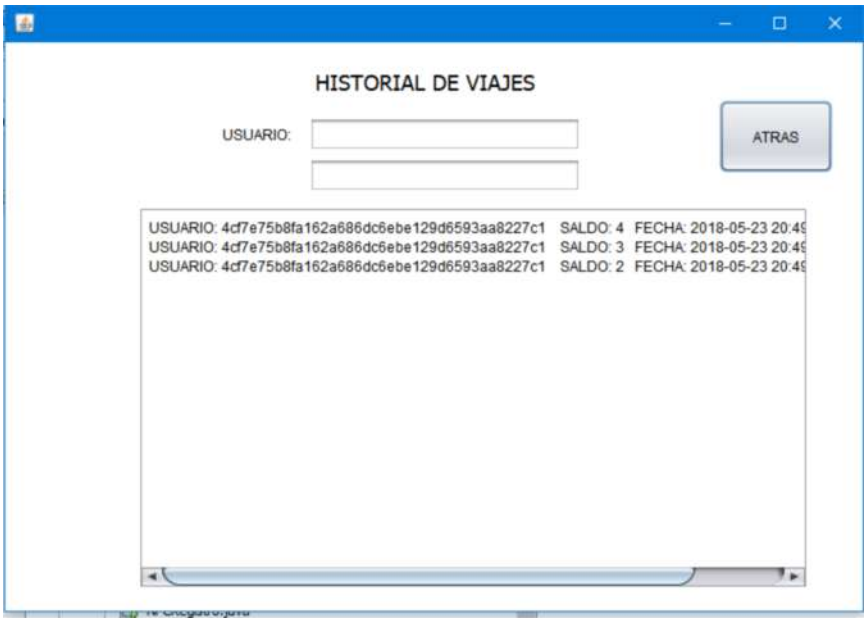


Figura 10-3 Pantalla de consulta de historial de viajes de la aplicación en la estación.

Elaborado por: SUAREZ, Jaime. 2018

Por otro lado los pagos se realizan de igual manera en la estación con la ayuda de la aplicación de Pagos, el usuario tiene solamente que acercar el teléfono inteligente con el NFC encendido al lector NFC, con esto el pago esta realizado y en la pantalla saldrá un mensaje de bienvenida, en el caso de que el usuario tenga saldo vigente por el contrario no podrá pagar y el mensaje cambia diciendo que el cliente no tiene saldo o los datos no son los verdaderos, esto se lo puede observar en la Figura 11-3.

Figura 11-3 Pantalla de aplicación de pagos en estación.

Elaborado por: SUAREZ, Jaime, 2018

3.3 Funcionamiento de la aplicación de escritorio para operadores y administradores.

La aplicación diseñada para el operario y el administrador es una sola, pero cada uno de ellos tiene sus privilegios. El administrador puede crear operarios, borrar operarios, buscar usuarios y borrarlos. La pantalla principal de la aplicación es igual para ambos como se observa en la Figura 12-3, sus privilegios son definidos por el tipo de ente que va a ingresar.

Figura 12-3 Pantalla de ingreso aplicación de operadores y administradores.

Elaborado por: SUAREZ, Jaime. 2018

En el caso de los administradores al ingresar tendrán una pantalla como se muestra en la Figura 13-3, en la cual en la parte superior tienen un menú cada uno con su función específica.

Figura 13-3 Ventana de Administrador.

Elaborado por: SUAREZ, Jaime. 2018

El menú sesión sirve para cerrar la sesión, también para salir de la ventana de administrador como se observa en la Figura 14-3.

Figura 14-3 Menú Sesión en ventana de Administrador.

Elaborado por: SUAREZ, Jaime. 2018

El Menú Operadores despliega una ventana en la cual se puede agregar, buscar o eliminar operadores y administradores como se observa en la Figura 15-3.

Figura 15-3 Ventana de Registro de Operadores.

Elaborado por: SUAREZ, Jaime. 2018

Al presionar el botón Nuevo las celdas de texto se liberan para poder llenarlas de la información correspondiente, en la celda Tipo se escribe el tipo en mayúsculas ya se OPERADOR o ADMINISTRADOR, cuando se finalizó el llenado de datos se presiona el botón Guardar, al agregar un nuevo usuario estos se mostrarán en la tabla que se encuentra en la parte inferior, para modificar algún dato se selecciona en la tabla de la parte inferior con el botón izquierdo del mouse,

se despliega un menú en donde se puede eliminar o modificar los datos como se ve en la Figura 16-3.

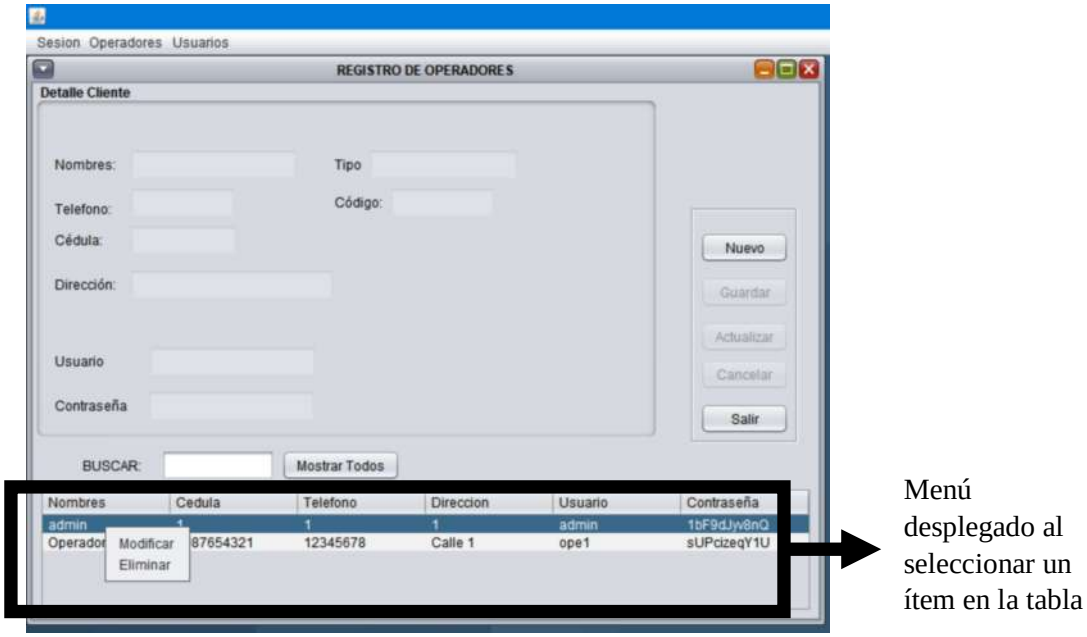


Figura 16-3 Menú para eliminar o modificar datos de operadores y administradores.

Elaborado por: SUAREZ, Jaime. 2018

El menú de Usuarios permite buscar el usuario a través de un número de cédula, el administrador debe ingresar el número de cédula en la casilla correspondiente para que la aplicación busque esos datos en la base de datos, el administrador puede borrar al usuario si este lo desea o para modificar sus datos y hacer un nuevo registro de este usuario. El ejemplo de esta pantalla se lo puede mirar en la Figura 17-3.

Figura 17-3 Ventana de Búsqueda de usuario

Elaborado por: SUAREZ, Jaime. 2018

Por otro lado, el operador al ingresar a la plataforma tendrá una ventana igual a la que se observa en la Figura 18-3.

Figura 18-3 Ventana de Operador

Elaborado por: SUAREZ, Jaime. 2018

En la ventana se encuentran tres botones y un menú, el primer botón es el de Recarga de Saldos en donde el operador va a realizar las recargas que el cliente solicite, al presionar el botón Recargar Saldo se desplegará una ventana como se observa en la Figura 19-3, el operador debe ingresar el número de cédula correctamente del usuario en la celda correspondiente y escribir el valor de recarga de viajes en la celda correspondiente, con esos datos correctos se presiona en actualizar saldo para que la recarga se realice, si es correcto se despliega un mensaje con el nombre del usuario y su nuevo saldo como se observa en la Figura 20-3.

Figura 19-3 Ventana de Recarga de saldo

Elaborado por: SUAREZ, Jaime. 2018

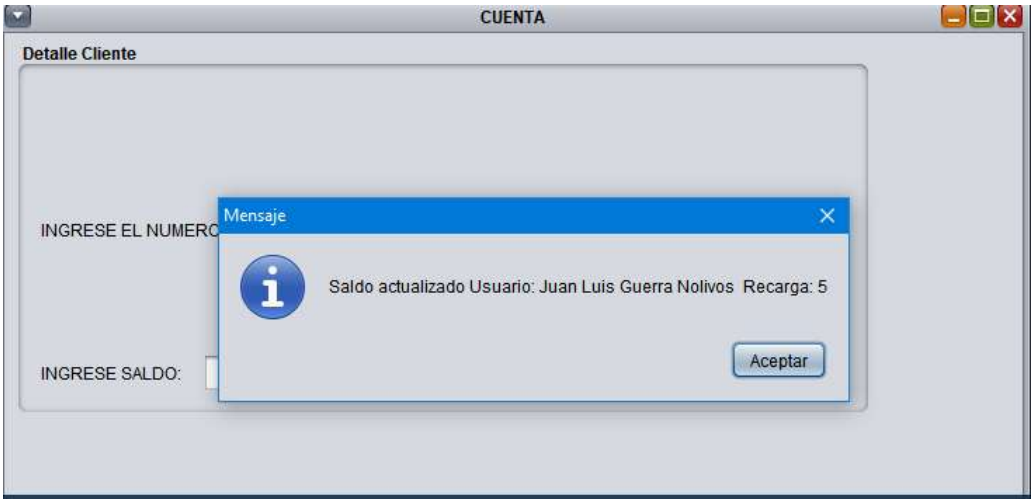


Figura 20-3 Mensaje de saldo actualizado.

Elaborado por: SUAREZ, Jaime. 2018

El siguiente botón es el de historial de pagos, al presionarlo se abre una ventana con un cuadro de texto en el cual se debe ingresar el número de cédula del usuario a consultar y se desplegará la lista de todos los viajes que ha realizado ese usuario como se puede ver en la Figura 21-3.

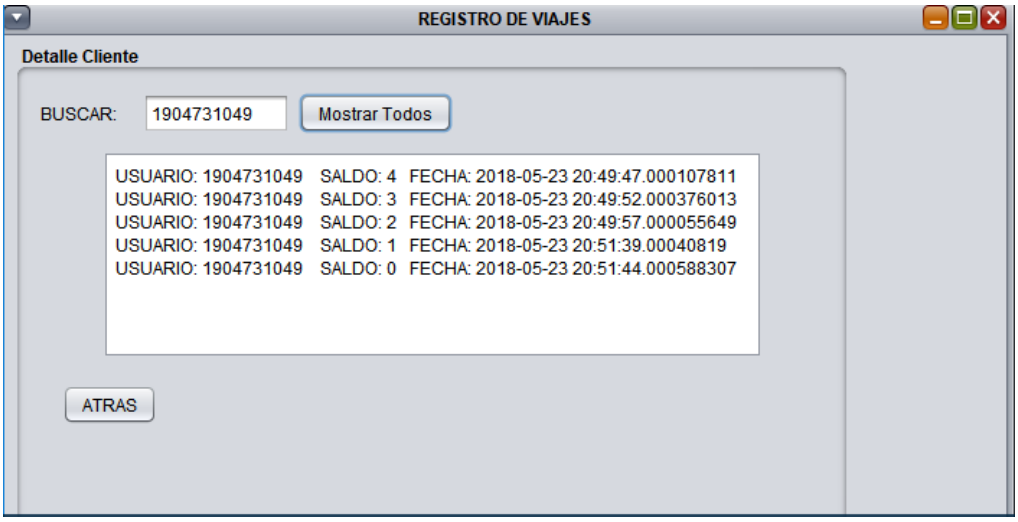


Figura 21-3 Ventana de Registro de Viajes en la aplicación del operador

Elaborado por: SUAREZ, Jaime. 2018

El último botón llamado Registro de Saldo muestra todas las recargas que ha realizado el usuario, igualmente en una ventana como se observa en la Figura 22-3.

Figura 22-3 Ventana de Registro de Recargas en la aplicación del operador

Elaborado por: SUAREZ, Jaime. 2018

El menú consta de 3 opciones, la primera es la de búsqueda de usuarios, la segunda de cerrar sesión y la tercera para salir del programa como se observa en la Figura 23-2.

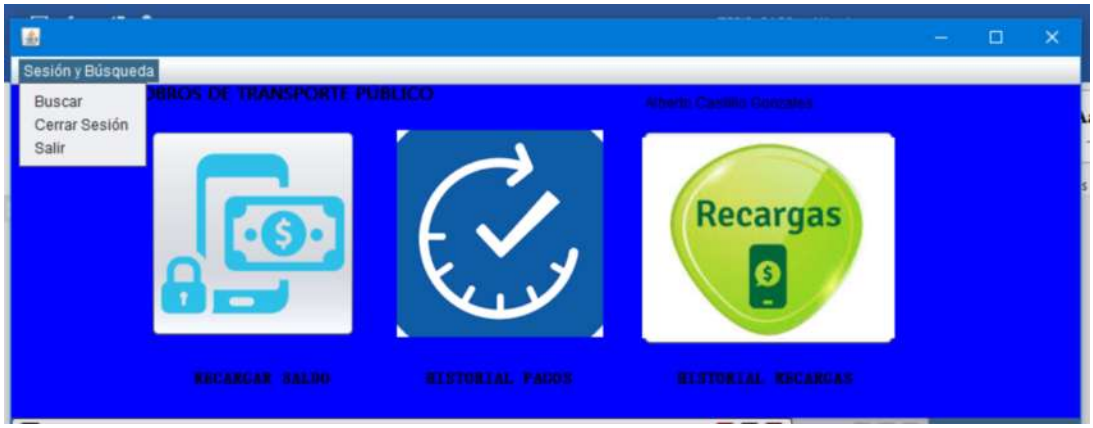


Figura 23-3 Menú desplegado en la ventana del operador.

Elaborado por: SUAREZ, Jaime. 2018

La búsqueda de usuarios es la misma que la del administrador por lo que no es necesario mostrarla nuevamente.

3.3 Comprobación de las seguridades informáticas implementadas en las aplicaciones.

Para comprobar que la plataforma de pagos es segura y que los datos que en esta se manejan son seguros, se realizó un ataque de hombre en el medio (man in the middle) para analizar qué datos son vulnerables si un ente no autorizado realiza este tipo de ataque.

Se analizó este tipo de ataque, aunque sería casi imposible de realizarlo entre la comunicación entre el teléfono inteligente con el lector NFC por la proximidad en la que se realiza esta comunicación, sin embargo, un atacante podría realizar este ataque conectando en el lector un software analizador de protocolos y obtener los datos fácilmente o ingresar a la plataforma y recopilar datos entre los nodos de comunicación por lo que resulta una amenaza para una transacción segura. (Lee, Kim and Jung, 2013)

Para el proceso de este ataque se utilizó una distribución del sistema operativo Linux llamado Wifislax, en esta distribución se encuentran herramientas para realizar auditorias de seguridad en redes, para lo cual se utilizaron dos de estas herramientas que recopilan datos en nodos específicos, estas herramientas son Ettercap y Wireshark, la primera realiza un ataque de hombre en el medio recopilando datos a través de envenenamiento de ARP (ARP spoofing) entre los nodos de comunicación seleccionados vinculando la dirección MAC del equipo atacante con las direcciones ip seleccionadas por lo que va a recibir todos los paquetes que lleguen a las direcciones ip seleccionadas por el atacante.

Por otra parte, Wireshark va a ser utilizada al mismo tiempo de Ettercap, con el objetivo de visualizar de una forma más técnica las tramas de datos y protocolos que son recopilados por Ettercap y analizarlos de mejor manera.

3.3.1 Ataque de Hombre en el medio a la comunicación entre la aplicación de operador/administrador con el servidor.

La aplicación de escritorio de operador/administrador realiza conexiones al servidor cada vez que estos realizan una acción en la plataforma de pagos como por ejemplo recargas de saldo, si un atacante se encuentra recopilando datos tendría acceso a información como datos importantes de ingreso en la base de datos y datos de usuarios.

En la captura de Ettercap de la Figura 24-3. Se puede observar que la aplicación al conectarse a la base de datos, se observa el nombre de usuario, esto se lo hizo deliberadamente para observar

en el análisis que elementos de conexión podría observar un atacante, por lo que no fue encriptado este campo, además se observa que la contraseña se encuentra encriptada asegurando la comunicación con el servidor de la aplicación de operador/administrador.

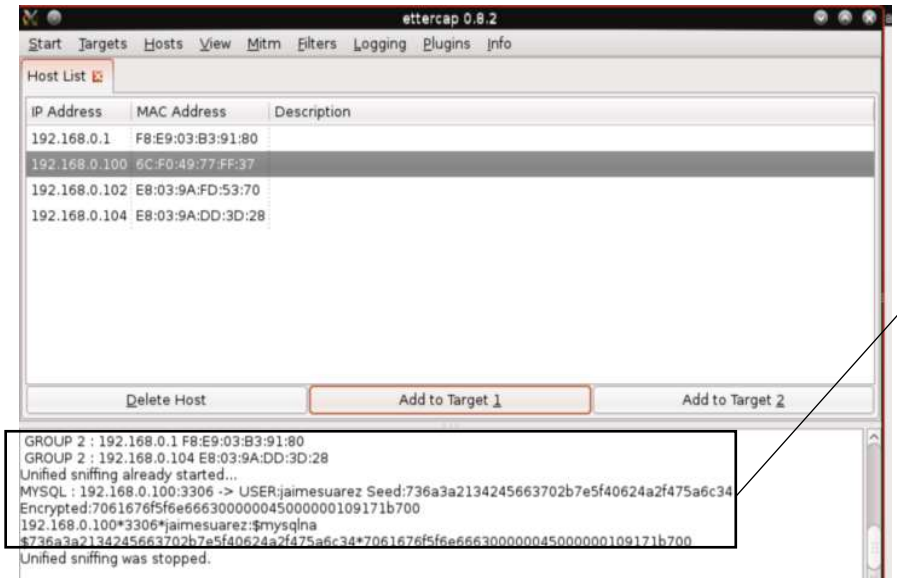


Figura 24-3 Ataque de hombre en el medio con Ettercap en comunicación entre aplicación de operador/administrador con servidor.

Elaborado por: SUAREZ, Jaime. 2018

Los datos recopilados por Wireshark se observan en la Figura 25-3, se puede ver que estos datos se encuentran encriptados de igual haciendo segura la comunicación.

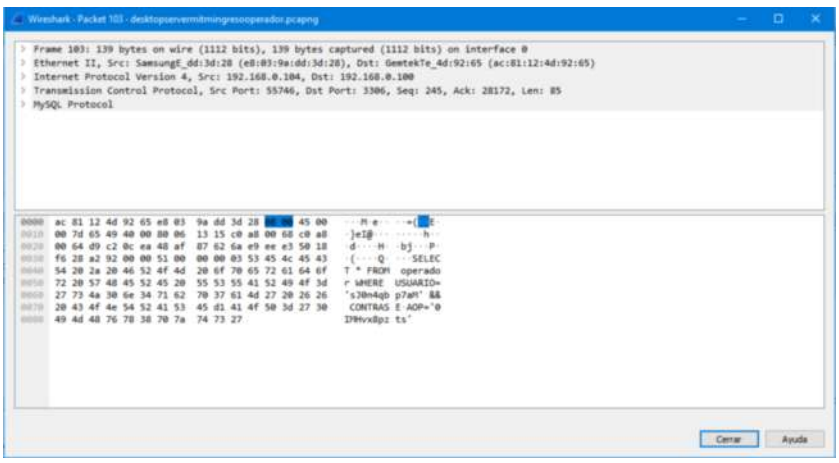


Figura 25-3 Trama conseguida con ataque de hombre en el medio en Wireshark al ingresar un operador en la plataforma.

Elaborado por: SUAREZ, Jaime. 2018

Cuando la aplicación realiza la consulta para la recarga, el atacante verá los datos del usuario encriptados como se observa en la Figura 26-3.

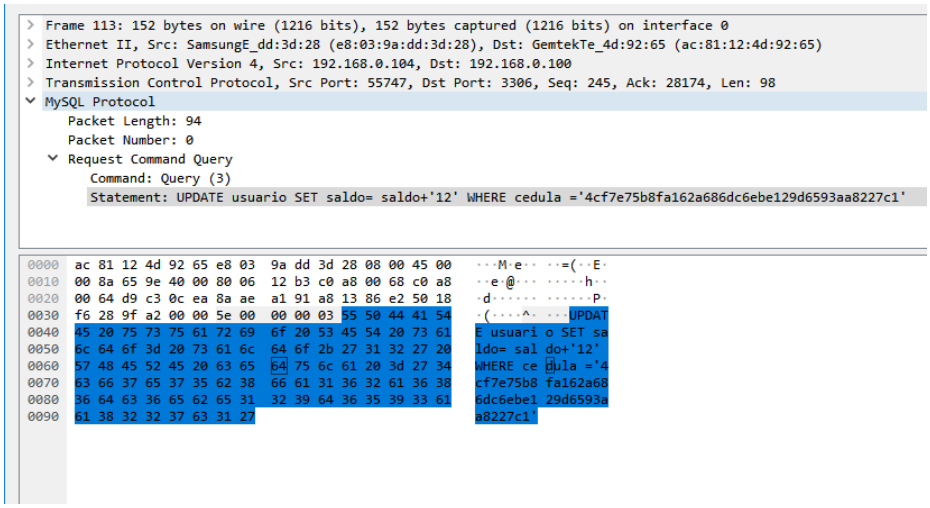


Figura 26-3 Trama conseguida con ataque de hombre en el medio en Wireshark al realizar recarga de saldo.

Elaborado por: SUAREZ, Jaime. 2018

3.3.2 Ataque de Hombre en el medio a la comunicación entre la aplicación de pagos en la estación y registro en la estación con el servidor.

Al realizar los pagos se envían datos de usuario que son la cédula y la contraseña, por lo que es necesario que estos datos sean irreconocibles para el o los atacantes. Al realizar el ataque de hombre en el medio a la comunicación entre las aplicaciones de pago y registro con el servidor se puede observar que los datos críticos están encriptados por lo que se encuentran seguros, esto se lo ve en la Figura 27-3.

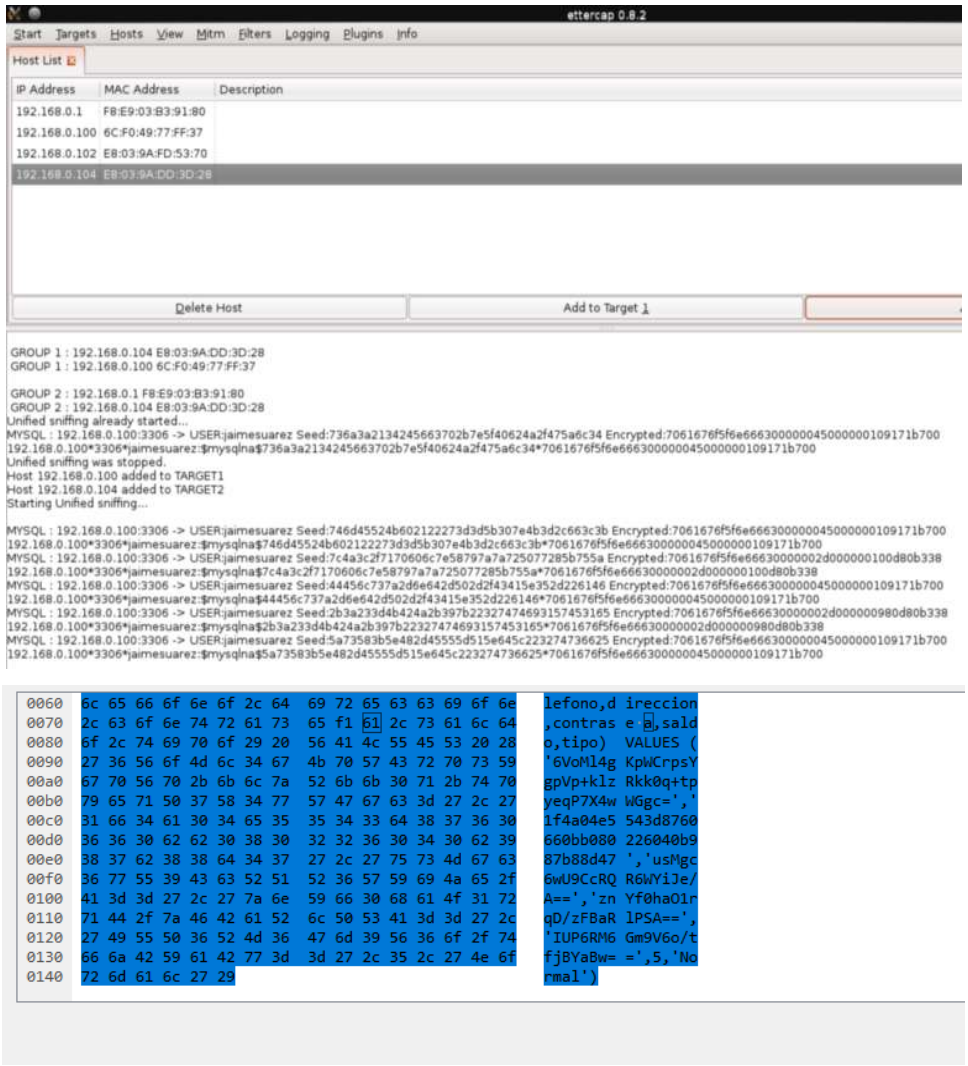


Figura 27-3 Capturas de tramas en Ettercap y Wireshark al momento de realizar registros y pagos desde las aplicaciones de la estación hacia el servidor.

Elaborado por: SUAREZ, Jaime. 2018

3.3.3 Ataque de hombre en el medio a la comunicación entre el lector/escritor NFC y las aplicaciones de Pago en la estación y Registro/Consultas en estación.

Este ataque se lo realizó con un analizador de protocolos USB que posee Wireshark deliberadamente implementado en el host de las aplicaciones de escritorio, estos datos son los que pasan a través del lector/escritor NFC para ser tomados por las aplicaciones de escritorio.

En el análisis se comprobó que estos datos al realizar registros de usuario y pagos comprobándose que estos datos están asegurados ya que el atacante podría ver datos encriptados solamente como se observa en la Figura 28-3.

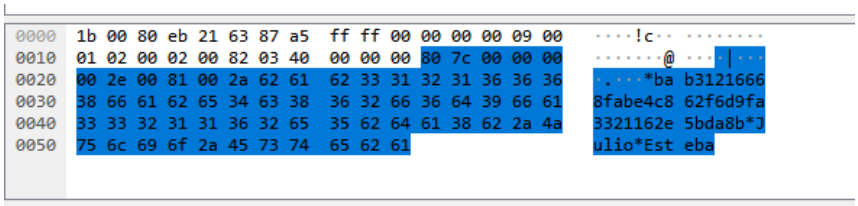


Figura 28-3 Capturas de tramas en Wireshark al momento de realizar registros en la comunicación del lector/escritor NFC con la aplicación de registro en la estación.

Elaborado por: SUAREZ, Jaime. 2018

3.4 Análisis de procesamiento, integridad y confidencialidad

Es necesario analizar el procesamiento para tener en cuenta el poder de procesamiento que necesita cada operación que se realiza en las aplicaciones de la plataforma de pagos al momento de encriptar el texto plano y desencriptarlo, para esto se realizó un análisis de protocolos con Wireshark para observar el número de bits de entrada o iniciales que son los que no presentan ningún tipo de encriptación y compararlos con el número de bits finales es decir los que se encuentran encriptados, esto se lo hizo analizando en los tramos importantes de la comunicación, es decir en la comunicación entre el servidor con la aplicación de pagos y registro/consultas en ***estación***, además de la comunicación entre las aplicaciones antes mencionadas con el lector/escritor NFC.

Primero se realizó el análisis de la comunicación de entre el lector/escritor NFC con la aplicación de estación de Registro, en la Figura 29-3 se puede observar la longitud en bits y bytes de una trama de datos sin encriptación, además que se observa al detalle todos los datos del usuario que se está registrando para ello se quitó la encriptación de las aplicaciones.

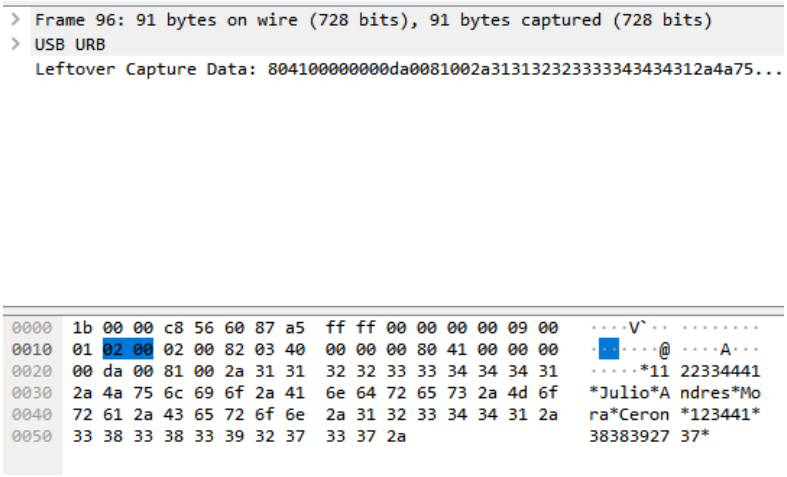


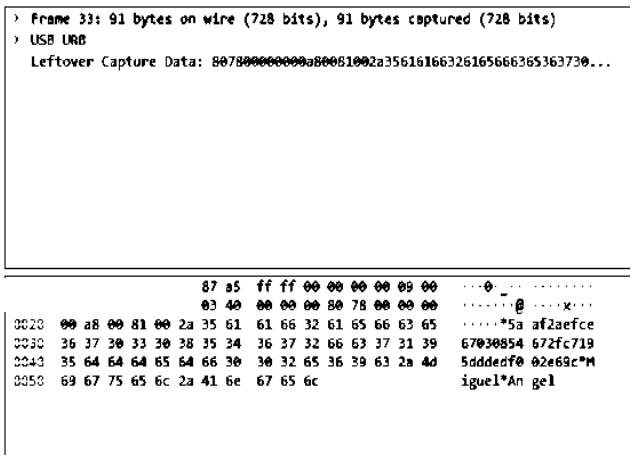
Figura 29-3 Captura de datos sin encriptar en la aplicación de Registro en la estación

Elaborado por: SUAREZ, Jaime, 2018

Los datos son enviados en dos bloques por lo que podemos observar que el primer bloque es de 91 bytes y el segundo de 38 bytes sumando en total 129 bytes para este usuario.

Para los bytes finales o encriptados se realizó el mismo procedimiento, aumentando en el segundo bloque a 91 bytes como se observa en la Figura 30-3, Los nombres y apellidos se los dejo sin encriptar deliberadamente para reconocer que en que trama de datos se encuentra el registro de usuario NFC, como se observa son dos registros cada uno de 91 bytes sumando en total 182 bytes.

Como se explicó en el capítulo anterior el registro va a encriptar los datos de cada campo con 3DES sumado una llave para su desencriptación a excepción de la contraseña que esta encriptada en SHA-1 ya que no es necesario desencriptar las contraseñas.



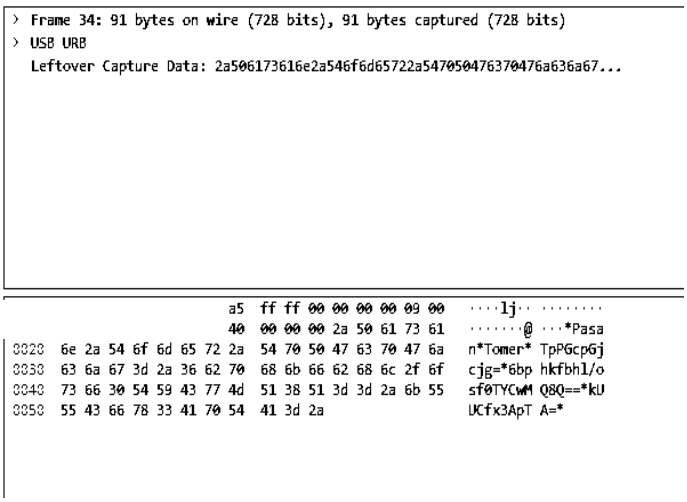


Figura 30-3 Captura de datos encriptados en la aplicación de Registro en la estación

Elaborado por: SUAREZ, Jaime. 2018

Se realizaron 30 operaciones de registro de usuarios en la plataforma. En la Tabla 1-3 se observa los resultados de la comunicación entre la aplicación de registro con el servidor tomados de 30 operaciones de Registro realizadas en la plataforma.

Tabla 1-3 Datos de Bytes recogidos por el Analizador de Protocolos en la comunicación entre el lector/escritor NFC con la aplicación de la estación de registro.

NÚMERO	bytes iniciales registro	bytes finales registro con SHA-1	bytes finales registro con SHA-256	Diferencia SHA-1	Diferencia SHA-2
1	129	182	194	53	65
2	130	182	194	52	64
3	129	182	194	53	65
4	130	182	194	52	64
5	129	182	194	53	65
6	130	182	194	52	64
7	129	182	194	53	65
8	129	182	194	53	65

Continúa

9	129	182	194	53	65
10	129	182	194	53	65
11	129	182	194	53	65
12	129	182	194	53	65
13	129	182	194	53	65
14	130	182	194	52	64
15	129	182	194	53	65
16	129	182	194	53	65
17	129	182	194	53	65
18	129	182	194	53	65
19	129	182	194	53	65
20	129	182	194	53	65
21	129	182	194	53	65
22	129	182	194	53	65
23	129	182	194	53	65
24	129	182	194	53	65
25	129	182	194	53	65
26	129	182	194	53	65
27	129	182	194	53	65
28	129	182	194	53	65
29	129	182	194	53	65
30	130	182	194	52	64
PROMEDIO	129,1	182	194	52,86	64,83

Elaborado por: SUÁREZ, Jaime, 2018

En la tabla anterior se puede observar el promedio de incremento de bytes es de 52.86 bytes, un nombre puede ocupar hasta 60 bytes, la cédula ocupa 12 bytes, la dirección hasta 30 bytes y el teléfono ocupa 12 bytes, mientras que la contraseña en SHA-1 ocupa 20 bytes

El atacante al realizar el ataque de hombre en el medio podría disponer de los datos encriptados por lo que es necesario analizar el número de operaciones que se necesitarían si el atacante realiza ataques de fuerza bruta para obtener los datos desencriptados.

3DES realiza 2.16 x 10^{17} operaciones por cada octeto de bits, según la Tabla 1-3 se tiene en promedio 182 bytes, pero este número toma en cuenta cabeceras de comunicación y la contraseña que se encuentra en SHA-1, quitando los bytes antes mencionados se tiene que los bytes restantes encriptados en 3DES son 114 bytes. Si un atacante quisiera descifrar estos datos debe realizar (2.16 x 10^{17}operaciones x 114 octetos) que resulta en 2.46 x 10^{19} operaciones de fuerza bruta, y para descifrar la contraseña deberá realizar 2^{160} operaciones es decir 1.46 x 10^{48} operaciones.

Si la contraseña está en SHA-2 tendría una longitud de 32 bytes por lo que para descubrir este cifrado se necesitaría 2^{256} operaciones es decir 1.15 x 10^{77} operaciones de fuerza bruta.

Al analizar la comunicación de la aplicación de Registro con el servidor se obtuvo de bytes iniciales 204 bytes de datos los cuales están sin encriptar, como se observa en la Figura 31-3, esta longitud va a variar de acuerdo a la longitud de caracteres que el usuario haya ingresado.

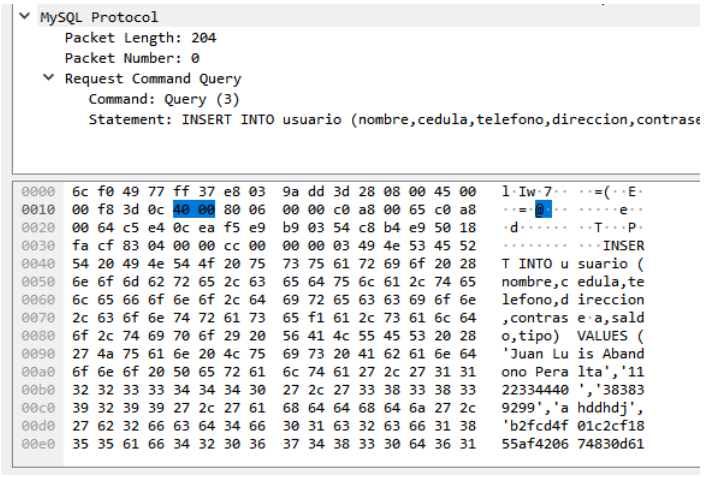


Figura 31-3 Captura de datos sin encriptar enviados desde la aplicación de Registro en la estación hacia el servidor.

Elaborado por: SUAREZ, Jaime. 2018

Mientras que la longitud de trama de datos encriptados para este usuario tendrá 244 bytes como se observa en la Figura 32-3.

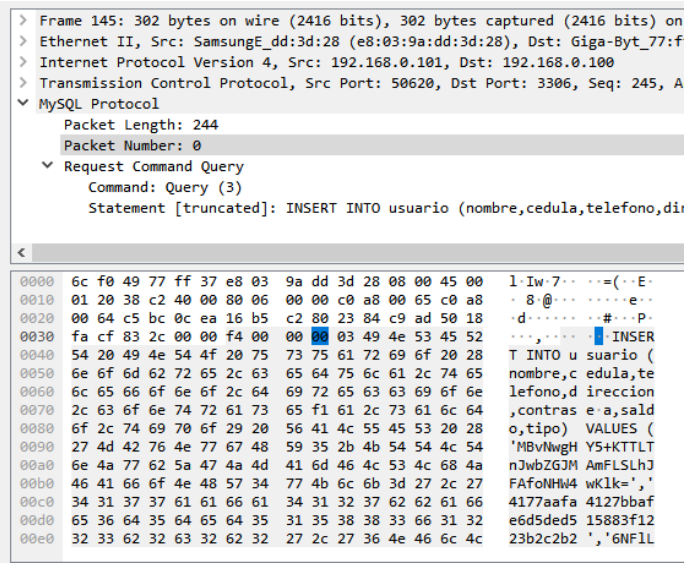


Figura 32-3 Captura de datos sin encriptar enviados desde la aplicación de Registro en la estación hacia el servidor.

Elaborado por: SUAREZ, Jaime. 2018

Los datos de las capturas del Analizador de Protocolos se los puede observar en la Tabla 2-3, en donde en promedio existen 50 bytes de diferencia entre texto plano y texto encriptado.

Tabla 2-3 Datos de Bytes recogidos por el Analizador de Protocolos en la comunicación entre la aplicación de la estación de registro con el servidor.

NÚMERO	bytes iniciales registro	bytes finales registro con SHA-1	bytes finales registro con SHA-256	Diferencia SHA-1	Diferencia SHA-2
1	209	268	280	59	71
2	203	244	256	41	53
3	204	249	261	45	57
4	207	259	271	52	64
5	209	268	280	59	71
6	203	244	256	41	53

Continúa

7	209	268	280	59	71
8	209	268	280	59	71
9	203	244	256	41	53
10	204	248	260	44	56
11	207	259	271	52	64
12	207	259	271	52	64
13	205	253	265	48	60
14	203	244	256	41	53
15	203	244	256	41	53
16	207	259	271	52	64
17	207	259	271	52	64
18	209	268	280	59	71
19	205	253	265	48	60
20	203	244	256	41	53
21	203	244	256	41	53
22	203	244	256	41	53
23	209	268	280	59	71
24	207	259	271	52	64
25	207	259	271	52	64
26	209	268	280	59	71
27	209	268	280	59	71
28	209	268	280	59	71
29	207	259	271	52	64
30	209	268	280	59	71
PROMEDIO	206,17	256,52	268,93	50,34	62,34

Elaborado por: SUÁREZ, Jaime, 2018

De igual forma se quitan los bytes de la contraseña y las cabeceras quedando 114 bytes en promedio dependiendo de la longitud de los nombres y direcciones, es decir se tiene que realizar (2.16×10^{17}) x 114 octetos lo que es igual a 2.46×10^{19} operaciones si el atacante quiere encontrar los datos por fuerza bruta.

En la comunicación entre la aplicación de registro en la estación con el lector/escritor NFC se encontró que los bytes iniciales son de 27 bytes como se observa en la Figura 33-3.

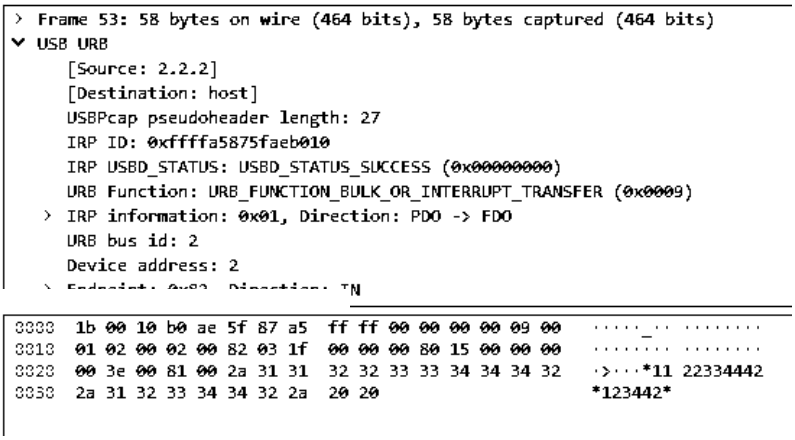


Figura 33-3 Captura de Wireshark de datos de pago sin encriptar

Elaborado por: SUAREZ, Jaime. 2018

Como se ve en la figura anterior el atacante podría ver claramente los datos de la cédula que es 1122334442 y la contraseña que es 123442 además estos datos son enviados en un solo bloque, pero al encriptar estos datos no serán reconocibles como se ilustra en la Figura 34-3 y son enviados en dos bloques.

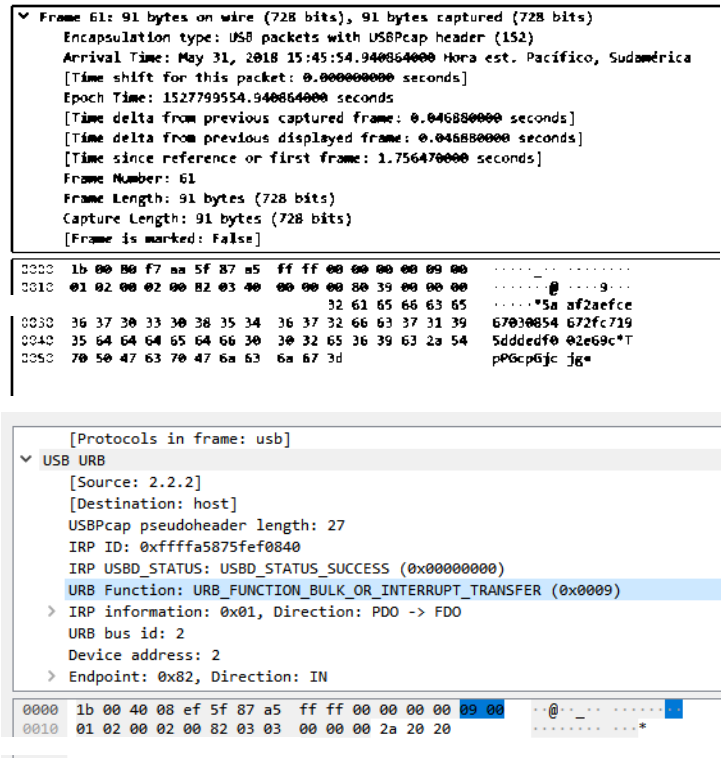


Figura 34-3 Captura de Wireshark de datos de pago encriptados

Elaborado por: SUAREZ, Jaime.2018

Como se observa al sumar los dos bloques tenemos 54 bytes, en la Tabla 3-3 se puede observar la comparación realizada.

Tabla 3-3 Datos de Bytes recogidos por el Analizador de Protocolos en la comunicación entre la aplicación de la estación de pago con el lector/escritor NFC.

NÚMERO	bytes iniciales pago-nfc	bytes finales pago-nfc con SHA-1	bytes finales pago-nfc con SHA-256	Diferencia SHA-1	Diferencia SHA-2
1	27	54	66	27	39
2	27	54	66	27	39
3	27	54	66	27	39
4	27	54	66	27	39
5	27	54	66	27	39
6	27	54	66	27	39
7	27	54	66	27	39
8	27	54	66	27	39
9	27	54	66	27	39
10	27	54	66	27	39
11	27	54	66	27	39
12	27	54	66	27	39
13	27	54	66	27	39
14	27	54	66	27	39
15	27	54	66	27	39
16	27	54	66	27	39
17	27	54	66	27	39
18	27	54	66	27	39
19	27	54	66	27	39
20	27	54	66	27	39
21	27	54	66	27	39
22	27	54	66	27	39
23	27	54	66	27	39
24	27	54	66	27	39
25	27	54	66	27	39

Continúa

26	27	54	66	27	39
27	27	54	66	27	39
28	27	54	66	27	39
29	27	54	66	27	39
30	27	54	66	27	39
PROMEDIO	27	54	66	27	39

Elaborado por: SUÁREZ, Jaime, 2018

Como se observa en los 30 usuarios de los cuales se tomó la muestra de los pagos la longitud de bytes no cambia ya que la longitud de la cédula es de 10 caracteres y la contraseña es de 8 caracteres, por lo que estos valores van a ser iguales en todos los usuarios.

En este caso son 54 bytes de salida, pero de estos 20 bytes son de la contraseña en SHA-1 12 bytes de la cédula encriptada en 3DES y lo que sobra es las cabeceras de comunicación, por lo tanto, el atacante tendría que realizar ($2.16 \times 10^{17} * 12$ octetos) es decir 2.59×10^{18} operaciones para encontrar la cédula.

Para la comunicación entre la aplicación de pagos en la estación con el servidor, se obtuvieron de bytes iniciales 275 bytes, esta cifra incrementó de la cifra anterior ya que en esta parte de la comunicación se realiza tres operaciones en la base de datos, la primera es consultar si el usuario y la contraseña son quien dicen ser, esto tiene una longitud de 95 bytes como se observa en la Figura 35-3.

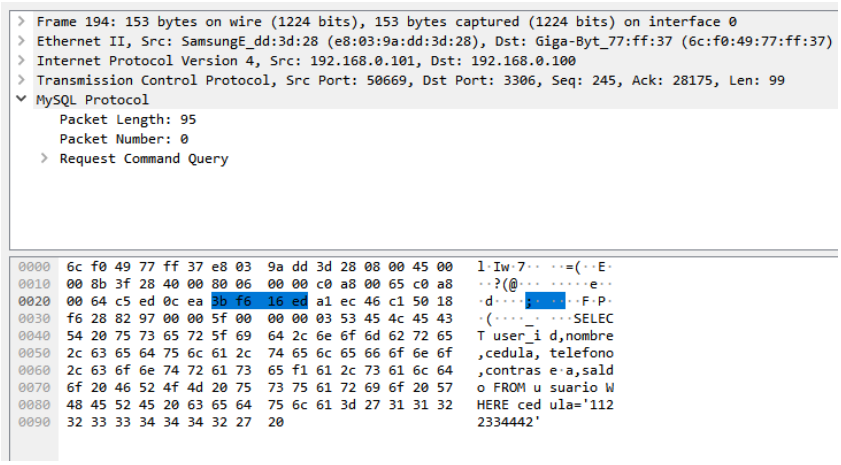


Figura 35-3 Captura de Wireshark de la primera operación en la base de datos al realizar un pago sin encriptar

Elaborado por: SUÁREZ, Jaime, 2018

La segunda operación es la de actualizar el saldo en la base de datos, esta operación tiene una longitud de 60 bytes como se observa en la Figura 36-3.

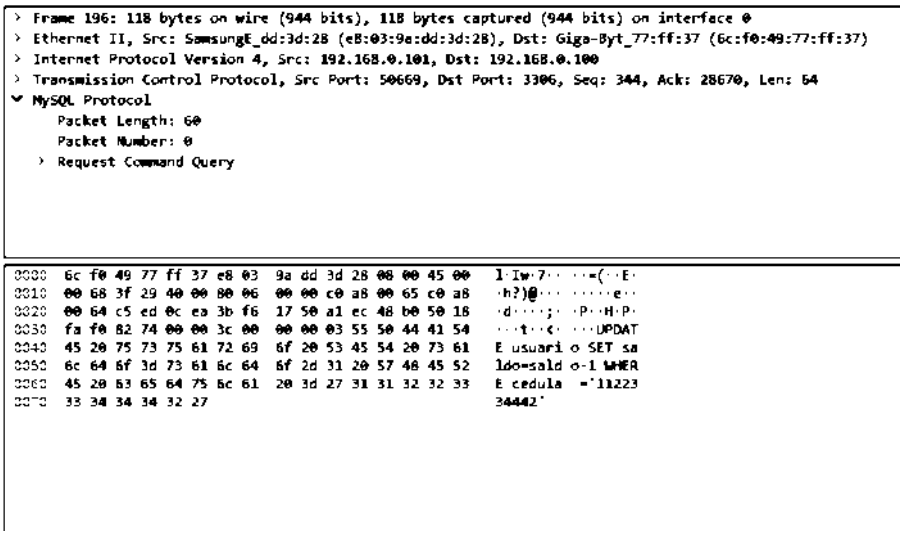


Figura 36-3 Captura de Wireshark de la segunda operación en la base de datos al realizar un pago sin encriptar

Elaborado por: SUÁREZ, Jaime. 2018

Para la tercera operación que es insertar un nuevo registro de viaje se obtuvo una longitud de 120 bytes como se ve en la Figura 37-3.

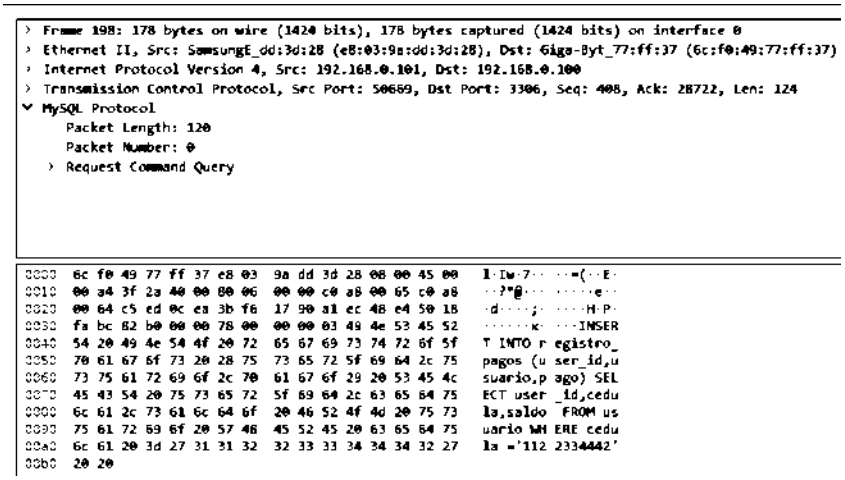


Figura 37-3 Captura de Wireshark de la tercera operación en la base de datos al realizar un pago sin encriptar

Elaborado por: SUÁREZ, Jaime, 2018

Para los bytes finales las longitudes aumentan en cada operación siendo de 125 bytes, 90 bytes y 150 bytes cada uno como se ve en la Figura 38-3, sumando 365 bytes la operación completa.

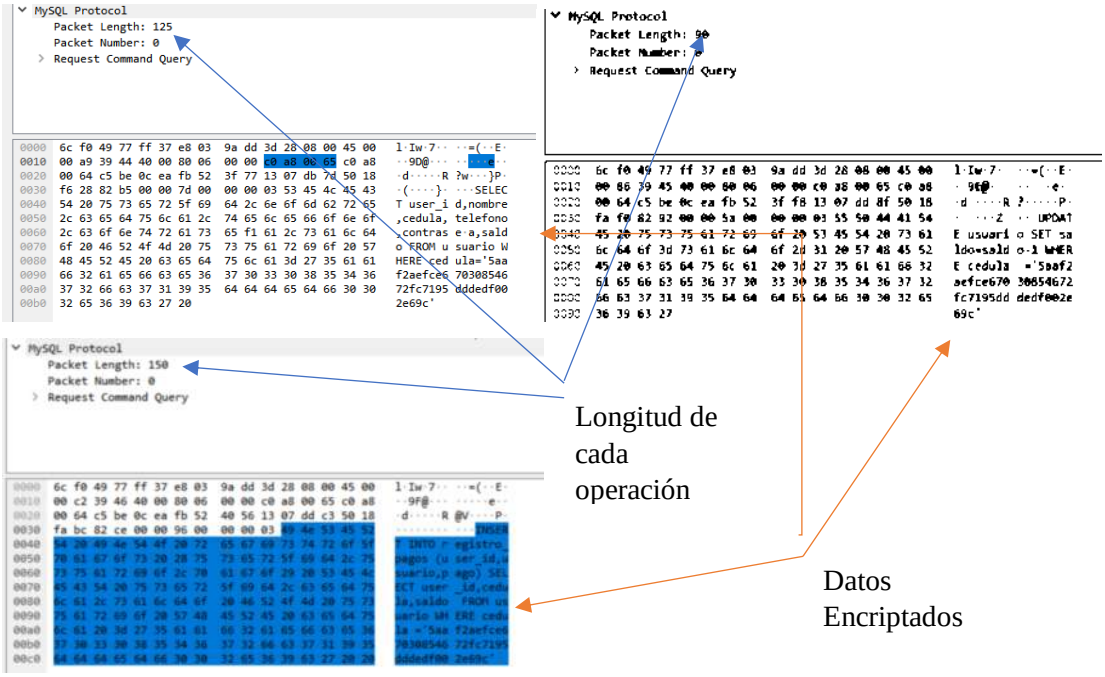


Figura 38-3 Capturas de Wireshark de las tres operaciones en la base de datos al realizar un pago con encriptación.

Elaborado por: SUÁREZ, Jaime, 2018

Con estos datos se realizó la Tabla 4-3 que se muestra a continuación.

Tabla 4-3 Datos de Bytes recogidos por el Analizador de Protocolos en la comunicación entre la aplicación de la estación de pago con el servidor.

NÚMERO	bytes iniciales pago -bd	bytes finales pago-bd con SHA-1	bytes finales pago-bd con SHA-256	Diferencia SHA-1	Diferencia SHA-2
1	275	365	377	90	102
2	275	365	377	90	102

Continúa

3	275	365	377	90	102
4	275	365	377	90	102
5	275	365	377	90	102
6	275	365	377	90	102
7	275	365	377	90	102
8	275	365	377	90	102
9	275	365	377	90	102
10	275	365	377	90	102
11	275	365	377	90	102
12	275	365	377	90	102
13	275	365	377	90	102
14	275	365	377	90	102
15	275	365	377	90	102
16	275	365	377	90	102
17	275	365	377	90	102
18	275	365	377	90	102
19	275	365	377	90	102
20	275	365	377	90	102
21	275	365	377	90	102
22	275	365	377	90	102
23	275	365	377	90	102
24	275	365	377	90	102
25	275	365	377	90	102
26	275	365	377	90	102
27	275	365	377	90	102
28	275	365	377	90	102
29	275	365	377	90	102
30	275	365	377	90	102
PROMEDIO	275	365	377	90	102

Elaborado por: SUÁREZ, Jaime, 2018

En la tabla anterior se aprecia que los valores no varían esto es porque como se dijo anteriormente la cédula está compuesta por 10 caracteres y la contraseña por 8 caracteres, al restar los 20 bytes de la contraseña que se encuentra en SHA-1 para calcular las operaciones que debe realizarse por fuerza bruta tenemos 345 bytes es decir 345 octetos que debe operar, pero cada uno es una operación diferente por lo que se realiza esto separado.

La primera operación tiene 125 bytes menos los bytes de la contraseña y de las cabeceras, quedan 12 bytes los cuales se encuentran encriptados, por lo tanto, se tiene (2.16×10^{17}) x 12 octetos que resulta 2.59×10^{18} operaciones que debe realizarse por fuerza bruta.

Para la segunda y tercera operación se tiene los mismos bytes de la operación anterior por lo tanto sumados las 3 operaciones se tiene un total de 7.77×10^{18} operaciones para descifrar la cédula en el pago.

Después de los análisis realizados se puede observar que los datos se encuentran encriptados en todas las etapas propuestas por lo que sería complicado para un atacante descifrar estos datos, por lo que se cumple la confidencialidad de los datos.

Con estos resultados de las operaciones que debería hacer un atacante con fuerza bruta se puede estimar el tiempo, que le tomaría obtener los datos con ese ataque, considerando que la capacidad promedio de un ordenador actual es de 4 mil millones de cálculos por segundo.(Labaca, 2014) De igual forma se consideró un clúster que puede tener 348.000 millones de cálculos por segundo además de un supercomputador que hasta la fecha el más poderoso puede realizar 1000×10^{15} cálculos por segundo.(BBC Mundo, 2015)
Entonces al saber los cálculos por segundo de los procesadores se puede calcular el tiempo que se demoraría en descifrar el texto por ataque de fuerza bruta, haciendo:

$$\boldsymbol{tiempo} = {}^{operaciones\ a\ realizarse}/_{calculos\ por\ segundo\ de\ procesador}$$

En la tabla 5-3 están recogidos los datos que determinan cuanto demoraría en descifrar un dato por fuerza bruta de cada segmento de la comunicación en el prototipo en 3DES mientras que la tabla 6-3 los datos de las contraseñas en SHA-1.

Tabla 5-3 Tiempo que tomaría descifrar los datos por ataque de fuerza bruta a 3DES

	Ordenador	**Clúster**	**Supercomputadora**
Lector NFC – Aplicación Registro	6.15×10^9 segundos	7.06×10^7 segundos	24.6 segundos
Aplicación – Servidor Registro	6.15×10^9 segundos	7.06×10^7 segundos	24.6 segundos
Lector NFC – Aplicación Pago	6.47×10^8 segundos	7.44×10^6 segundos	2.59 segundos
Aplicación – Servidor Pago	1.92×10^9 segundos	2.21×10^7 segundos	7.7 segundos

Elaborado por: SUÁREZ, Jaime, 2018

Traduciendo estos tiempos en mejores términos se tiene que en la comunicación entre el Lector NFC con la aplicación de Registro y en la comunicación Aplicación Registro con el servidor que el ataque podría durar 1708333.33 horas, en días serían 71180.55 días, en años serían 195.01 años si se realizan con un ordenador. Si el ataque se lo realiza con un clúster el resultado sería en horas de 19611.11 horas, 817.12 días y en años serían 2.23 años.

Mientras que en la comunicación entre el Lector NFC con la aplicación de pago el ataque con ordenador podría durar 179722.22 horas, en días serían 7488.42 días, y en años son 20.51 años. Si se lo realiza con un clúster en horas sería de 2066.66 horas, en días sería de 86.11 días.

En la comunicación de la aplicación de pago con el servidor con un ordenador el ataque podría durar 533333.33 horas, en días serían 22222.22 días y en años 60.88 años. Si se realiza el ataque con un clúster serían 6138.88 horas, lo que en días serían 255.78 días.

De igual forma se hizo el cálculo para la contraseña que se encuentra en SHA-1 o SHA-2, estos datos se pueden observar en la tabla 6-3.

Tabla 6-3 Tiempo que demoraría descifrar la contraseña por ataque de fuerza bruta

	Ordenador	**Clúster**	**Supercomputadora**
Contraseña en SHA-1	3.65×10^{38} segundos	4.19×10^{36} segundos	1.46×10^{30} segundos
Contraseña en SHA-2	2.87×10^{67} segundos	3.30×10^{65} segundos	1.15×10^{59} Segundos

Elaborado por: SUÁREZ, Jaime, 2018

Al usar un ordenador para el ataque de fuerza bruta a la contraseña en SHA-1 el valor en horas sería de 1.01×10^{35} horas, este valor en días es de 4.22×10^{33} días y en años sería de 1.15×10^{31} años. Si el ataque se lo realiza con un clúster serán 1.16×10^{33} horas, o en días serían 4.84×10^{31} días y en años 1.32×10^{29} años. Si se utiliza una supercomputadora en horas serían 4.05×10^{26} horas, en días son 1.68×10^{23} días y en años 4.62×10^{22} años.

Al usar un ordenador para el ataque de fuerza bruta a la contraseña en SHA-2 el valor en horas sería de 7.97×10^{63} horas, este valor en días es de 3.33×10^{62} días y en años sería de 9.10×10^{59} años. Si el ataque se lo realiza con un clúster serán 9.16×10^{61} horas, o en días serían 3.81×10^{60} días y en años 1.04×10^{58} años. Si se utiliza una supercomputadora en horas serían 3.19×10^{55} horas, en días son 1.33×10^{54} días y en años 3.64×10^{51} años.

Como se puede observar el tiempo es bastante alto, pero no imposible ya que los atacantes siempre buscan las vulnerabilidades en los sistemas para atacarlos ahí por lo que hay que hacer revisiones de seguridad cada cierto tiempo para realizar parches y que se dificulte la tarea de robar los datos. De igual forma se observa que la encriptación de datos en SHA-1 es mucho más complicada de descifrar que 3DES ya que es encriptación asimétrica de una sola vía por lo mismo que solo fue realizada para la contraseña ya que no es necesario desencriptar este tipo de dato, por lo que para los demás datos se utilizó 3DES ya que siendo encriptación simétrica se tiene la posibilidad de desencriptar los datos de forma segura, y 3DES al realizar menor número de operaciones para encriptar un dato en relación con SHA-1 mejora el tiempo de procesamiento de datos de la plataforma de pagos ahorrando recursos de hardware y siendo seguro de igual forma.

3.5 Análisis de Costos

En la tabla 7-3 se detalla los costos de proyecto.

Tabla 7-3 Tabla de costos de proyecto

CANTIDAD	DESCRIPCIÓN	COSTO UNITARIO	COSTO TOTAL
1	Lector/ Escritor NFC ACR122U	$ 159.99	$ 159.99
1	Lector/ Escritor NFC ACR122U stick	$ 79.00	$ 79.00
1	Ordenador Portátil	$ 700.00	$ 700
1	Ordenador Escritorio	$ 400.00	$ 400
1	Enrutador	$ 25.00	$ 25.00
1	Software en General	$ 0.00	$ 0.00
1	Smartphone con tecnología NFC	$ 216.99	$ 216.99
TOTAL		$ 1580.98	$ 1580.98

Elaborado por: SUÁREZ, Jaime. 2018

CONCLUSIONES

- Existen varios métodos por los cuales se pueden robar información y datos en una plataforma de pagos usando tecnologías de campo cercano NFC, por lo que la inherencia en la seguridad de estas tecnologías debe ser reforzada y mejorada al realizar las aplicaciones tanto móviles como de escritorio.

- Se desarrollo la aplicación móvil para el sistema operativo Android ya que es el sistema operativo presente en la mayoría de teléfonos inteligentes a nivel mundial, además de que la arquitectura de emulación de tarjeta inteligente basada en host fue adoptada por este sistema operativo.

- El uso de tecnologías inalámbrica de corto alcance como NFC para realizar pagos electrónicos es uno de los caminos que se quiere llegar a futuro con el denominado Internet de las cosas.

- La compatibilidad de NFC con otras tecnologías permite que el mercado de aplicaciones prácticas de esta tecnología gane terreno a nivel mundial y sin la necesidad del cambio total de infraestructura.

- El corto alcance de trabajo de NFC brinda una seguridad ante un posible ataque de hombre en el medio, por lo que hay que asegurar los datos en el lector y en la comunicación de las aplicaciones de escritorio con el servidor, para asegurar la confidencialidad de los datos.

- El mayor obstáculo al realizar este proyecto fue desarrollar las aplicaciones tanto de escritorio como móvil ya se necesita un conocimiento medio-avanzado en los lenguajes de programación para entender las instrucciones de comunicación y su recepción en el lector NFC.

- Java es un lenguaje de programación muy versátil y su orientación al software libre hace que se desarrollen cada día más aplicaciones para NFC, pero los nuevos lenguajes de programación como Python, kotlin entre otros hacen que se diversifique las plataformas de desarrollo.

- Una mejora que se puede realizar al sistema es la implementación de una comunicación cifrada con claves públicas y privadas mejorando su seguridad en entornos no seguros.

RECOMENDACIONES

- Al implementar la plataforma en un lugar fuera del desarrollo se debe realizar la implementación de una comunicación a través de una red privada virtual entre las aplicaciones de estación y el servidor.

- Para evitar o dificultar los ataques de hombre en el medio, se necesita encriptar las comunicaciones con el servidor o conectarlas con una red privada virtual que sería la mejor opción para evitar este tipo de ataques.

- El usuario al registrarse debe ingresar los datos correctamente para que al comunicarse con la aplicación a través del lector/escritor NFC reconozca de forma correcta los datos.

- Las aplicaciones de estación no van a ser publicadas porque podrían hacerse ingeniería inversa de forma que una persona con amplios conocimientos podría desencriptar los datos sensibles de manera muy fácil.

- Se debe realizar pruebas de penetración periódicamente en el momento de ser implementado para desarrollar nuevos parches que aseguren los datos de mejor manera ante nuevos ataques.

- Se recomienda que los operadores tengan acceso a las aplicaciones de pago ya que si existiese algún tipo de falla simplemente se remediaría con un reseteo de la aplicación y no sería necesario llamar a la persona que implemento la plataforma.

BIBLIOGRAFÍA

ALBIÑANA, A. Desarrollo de una guía para la implementación de aplicaciones basadas en NFC (Trabajo de Titulación) (Ingeniería Informática) [pdf]. Universitat Politécnica de Valencia, Escola Tecnica Superior d'Enginyería Informática. Valencia-España. 2016. Pp 15-28. https://riunet.upv.es/handle/10251/74909

ALBIÑANA, A., CARDONA, E. y PILES, D.F. NFC. [pdf]. Universitat Politécnica de Valencia. Valencia-España. 2012. Pp 2-4.
[consulta: 28 de octubre 2017]
http://histinf.blogs.upv.es/files/2012/11/HDI-Trabajo_NFC.pdf

ALLIANCE, SMART CARDS. A smart card alliance mobile & nfc council white paper host card emulation (HCE) 101. [En línea]. New Jersey- Estados Unidos. 2014. Pp 19-21. http://www.smartcardalliance.org/wp-content/uploads/HCE-101-WP-FINAL-081114-clean.pdf.

ANDROID, Developer. Host-Based Card Emulation. [en línea]. California-Estados Unidos.2014
[Consulta: 10 diciembre 2017].
https://developer.android.com/guide/topics/connectivity/nfc/hce.html.

BBC MUNDO. *La supercomputadora más poderosa del mundo con la que EE.UU. quiere superar a China*. [En línea]. Londres-Reino Unido. 2015
[Consulta: 5 junio 2018].
http://www.bbc.com/mundo/noticias/2015/07/150731_tecnologia_eeuu_supercomputador_mas_poderoso_autorizo_obama_lv.

CERÓN, J.L., NARVAEZ, D.L. y RAMÍREZ, G. *Mobile payments system employing NFC technology under the Android operating system.Universidad del Cauca*. [pdf]. Departamento de Telemática. Popayán-Colombia. Pp. 77-87. 2015
[consulta: 13 de febrero 2018]
https://www.icesi.edu.co/revistas/index.php/sistemas_telematica/article/view/2082.

CORLETTI ESTRADA, A. *Seguridad por niveles*. Madrid-España: DarFE, 2011.
[consulta: 9 de marzo del 2018]
https://books.google.com.co/books?id=PmcOKclsKQC&dq=seguridad+por+niveles&hl=es&sa=X&ved=0ahUKEwiR9JrEwtPZAhUBw1kKHf3Db9Wq6AEIJjAA.

DIARLU. Los 5 Mejores IDE para programar en JAVA. [en línea]. Illinois-Estados Unidos. 2016
[consulta:14 enero 2018].
https://www.diarlu.com/mejores-ide-programar-java/

FORUM, Nfc. NFC Forum Technical Specifications. [en línea]. Massachusetts-Estados Unidos. 2017
https://nfc-forum.org/our-work/specifications-and-application-documents/specifications/nfc-forum-technical-specifications/#protocol.

INTECO. La tecnología NFC: Aplicaciones y gestión de seguridad. [pdf] Instituto Nacional deTecnologías de la Información. Madrid-España. 2013. Pp 12-13.
[consulta: 12 de febrero 2018].
http://www.egov.ufsc.br/portal/sites/default/files/cdn_nfc_final.pdf.

JACQUINOT CONSULTING, I. Smart Card ISO 7816-4. [en línea]. Georgia- Estados Unidos. 2018.
[Consulta: 12 diciembre 2017].
http://cardwerk.com/smart-card-standard-iso7816-4-section-5-basic-organizations/#chap5_3.

LABACA, R. La matemática de las claves: ¿numérica o alfanumérica? [en línea]. Bratislava - República Eslovaca. 2014
[Consulta: 5 junio 2018].
https://www.welivesecurity.com/la-es/2014/06/13/matematica-claves-numerica-alfanumerica/.

LEE, Y., KIM, E. y JUNG, M. A NFC based Authentication method for defence of the Man in the Middle Attack. [pdf] 3rd International Conference on Computer Science and Information Technology. Bali-Indonesia. 2013. Pp. 5.
[Consulta: 28 diciembre 2017].
http://psrcentre.org/images/extraimages/113113.pdf.

LESAS, A.-M. y MIRANDA, S. *The art and science of NFC programming*. Londres-Reino Unido: ISTE Ltd and John Wiley & Sons Inc., 2017. ISBN 9780073385037, pp 21-25.

MEDINA VARGAS, YURI TATIANA; MIRANDA MENDÉZ, H.A. Comparison of Algorithms Based Cryptography Symmetric DES, AES and 3DES. *Revista Mundo Fesc*, vol. 9, 2015, Colombia pp. 14-21.

MINIHOLD, R. Near Field Communication (NFC) Technology and Measurements White Paper. [pdf] Rohde and Schwarz. Munich-Alemania.2011. pp. 10.
http://cdn.rohdeschwarz.com/dl_downloads/dl_application/application_notes/1ma182/1MA182_4e.pdf.

ORACLE. Oracle MySQL la base de datos mas popular del mundo. [en línea]. Washington-Estados Unidos.2017
[Consulta: 16 abril 2018].
https://www.oracle.com/ve/mysql/index.html

TIEDEMANN, S. Logical Link Control Protocol. [en línea]. Stuttgart-Alemania.2009 [Consulta: 15 octubre 2017]. http://nfcpy.readthedocs.io/en/latest/topics/llcp.html.

VALENCIA, U.P. Historia de la Informática-NFC. [en línea]. Valencia-España. 2012 [Consulta: 12 octubre 2017]. http://histinf.blogs.upv.es/2012/11/21/nfc/.

VELOZ, D.F. Diseño e implementación de un prototipo para control de acceso de personas aplicando la tecnología nfc por medio del uso de teléfonos celulares compatibles con esta tecnología (Tesis)(Ingeniería Electrónica y Telecomunicaciones) [pdf]. Escuela Politécnica Nacional, Facultad de Ingeniería Eléctrica y Electrónica. Quito-Ecuador-2010. Pp 13-20 [Consulta: 22 octubre 2017]. http://bibdigital.epn.edu.ec/handle/15000/2227

WYKIPEDIA. Servidor HTTP Apache. [en línea]. San Francisco-Estados Unidos.2012 [Consulta: 28 mayo 2018]. https://es.wikipedia.org/wiki/Servidor_HTTP_Apache

WYSEUR, B. WBC: protecting cryptographic keys in software applications. [en línea]. Cheseaux-sur-Lausanne -Suiza.2012 [Consulta: 13 enero 2018]. http://www.whiteboxcrypto.com/

ANEXOS

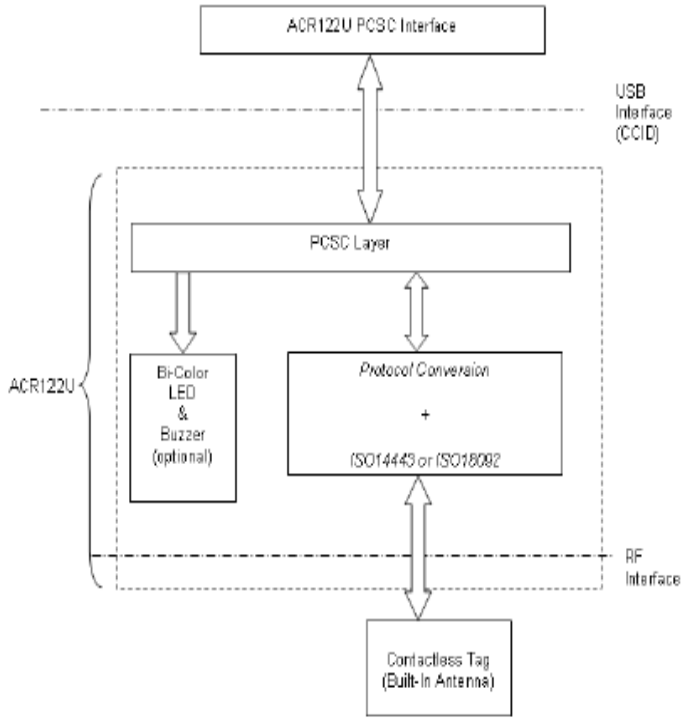
ANEXO A: TABLA DE FLUJO DE COMUNICACIÓN EN EL LECTOR/ESCRITOR NFC ACR122U

Advanced Card Systems Ltd.
Card & Reader Technologies

2.0. Implementation

2.1. Communication Flow Chart of ACR122U

The Standard Microsoft CCID and PCSC drivers are used. Therefore, no ACS drivers are required because the drivers are already built inside the windows operating system. You also have to modify your computer's registry settings to be able to use the full capabilities of the ACR122U NFC Reader. See ACR122U PCSC Escape Command for more details.



ANEXO B: FORMATO DE ATR PARA ISO 14443-4

Advanced Card Systems Ltd.
Card & Reader Technologies

3.1.2. ATR format for ISO 14443 Part 4 PICCs

Byte	Value (Hex)	Designation	Description
0	3B	Initial Header	
1	8N	T0	Higher nibble 8 means: no TA1, TB1, TC1 only TD1 is following. Lower nibble N is the number of historical bytes (HistByte 0 to HistByte N-1)
2	80	TD1	Higher nibble 8 means: no TA2, TB2, TC2 only TD2 is following. Lower nibble 0 means T = 0
3	01	TD2	Higher nibble 0 means no TA3, TB3, TC3, TD3 following. Lower nibble 1 means T = 1
4 to 3 + N	XX XX XX XX	T1 Tk	Historical Bytes: ISO14443A: The historical bytes from ATS response. Refer to the ISO14443-4 specification. ISO14443B: The higher layer response from the ATTRIB response (ATQB). Refer to the ISO14443-3 specification.
4+N	UU	TCK	Exclusive-oring of all the bytes T0 to Tk

Table 3: ATR format for ISO 14443 Part 4 PICCs

We take for example, an ATR for DESFire which is:

DESFire (ATR) = 3B 86 80 01 06 75 77 81 02 80 00

ATR						
Initial Header	T0	TD1	TD2	ATS		
				T1	Tk	TCK
3B	86	80	01	06	75 77 81 02 80	00

This ATR has 6 bytes of ATS which is: [06 75 77 81 02 80]

ANEXO C: DIAGRAMA DE FLUJO BÁSICO PARA APLICACIONES

Advanced Card Systems Ltd.
Card & Reader Technologies

7.0. Basic Program Flow for Contactless Applications

Step 0. Start the application. The reader will do the PICC Polling and scan for tags continuously. Once the tag is found and detected, the corresponding ATR will be sent to the PC. You must make sure that the PCSC Escape Command has been set. See **ACR122U PCSC Escape Command** for more details.

Step 1. The first thing is to connect the "ACR122U PICC Interface".

Step 2. Access the PICC by sending APDU commands.

:
:
:

Step N. Disconnect the "ACR122U PICC Interface". Shut down the application.

NOTE:

1. The antenna can be switched off in order to save the power.
   - Turn off the antenna power: FF 00 00 00 04 D4 32 01 00
   - Turn on the antenna power: FF 00 00 00 04 D4 32 01 01
2. Standard and Non-Standard APDUs Handling.
   - PICCs that use Standard APDUs: ISO14443-4 Type A and B, MIFARE .. etc
   - PICCs that use Non-Standard APDUs: FeliCa, Topaz .. etc.

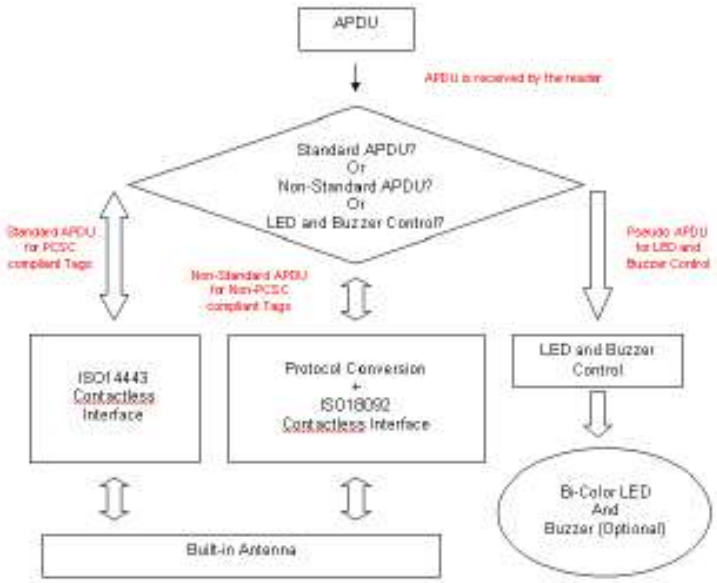
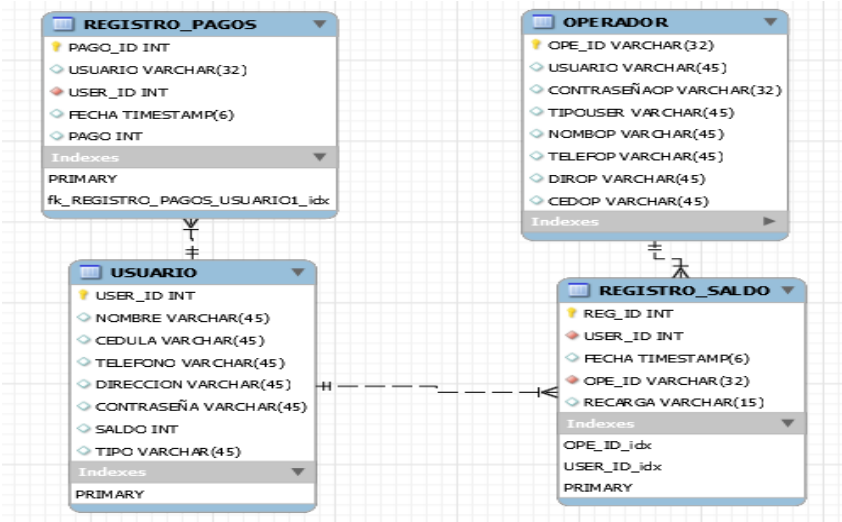


Figure 2: Basic Program Flow for Contactless Applications

ANEXO D: TABLAS DE LA BASE DE DATOS DEL PROTOTIPO



La tabla REGISTRO_PAGOS tiene los atributos pago_id como clave principal, usuario, user_id como clave foránea, fecha y pago, estos atributos servirán para tener el registro de los pagos que realizará por los viajes en el servicio de transporte público, esta tabla se relaciona de N:1 con la tabla USUARIO.

USUARIO tiene los siguientes atributos user_id como clave primaria, cédula, teléfono, dirección, contraseña, saldo, tipo; esta tabla tiene una relación de 1:N con la tabla REGISTRO_PAGOS y con la tabla REGISTRO_SALDO, los atributos de esta tabla al ser llenados van a estar encriptados para mayor seguridad.

La tabla REGISTRO_SALDO tiene los atributos reg_id como clave primaria, user_id como clave foránea, fecha, ope_id como clave foránea y recarga, la relación que tiene REGISTRO_SALDO con USUARIO es de N:1 al igual que con la tabla OPERADOR.

La tabla OPERADOR contiene los atributos ope_id como clave primaria, usuario, contraseña, tipo_user, nombop, telefop, dirop, cedop.

ANEXO E: SCRIPTS PHP

Script para la consulta de historial desde el teléfono inteligente hacia el servidor

```
<?PHP
$hostname_localhost ="localhost";
$database_localhost ="pago_nfc";
$username_localhost ="jaimesuarez";
$password_localhost ="";
$json=array();
        if(isset($_GET["usuario"])){
                $usuario=$_GET["usuario"];

                $conexion =
mysqli_connect($hostname_localhost,$username_localhost,$password_localhost,$database_loc
alhost);
                $consulta="select usuario,fecha,pago from registro_pagos where usuario=
'{$usuario}' limit 3";
                $resultado=mysqli_query($conexion,$consulta);

                while($registro=mysqli_fetch_array($resultado)){
                $json['historial'][]=$registro;

                }
                mysqli_close($conexion);
                echo json_encode($json);
        }
        else{
                $resultar["success"]=0;
                $resultar["message"]='Ws no Retorna';
                $json['historial'][]=$resultar;
                echo json_encode($json);
        }
?>
```

Script para la consulta del saldo desde el teléfono inteligente hacia el servidor a través del internet.

```
<?PHP
$hostname_localhost ="localhost";
$database_localhost ="pago_nfc";
$username_localhost ="jaimesuarez";
$password_localhost ="";
$json=array();
	if(isset($_GET["cedula"])){
		$cedula=$_GET["cedula"];
		$conexion =
mysqli_connect($hostname_localhost,$username_localhost,$password_localhost,$database_loc
alhost);
		$consulta="select saldo from usuario where cedula='{$cedula}'  ";
		$resultado=mysqli_query($conexion,$consulta);
		if($registro=mysqli_fetch_array($resultado)){
			$json['usuario'][]=$registro;

		}else{
			$resultar["cedula"]=0;
			$resultar["saldo"]='no registra';
			$json['usuario'][]=$resultar;
		}
		mysqli_close($conexion);
		echo json_encode($json);
	}
	else{
		$resultar["success"]=0;
		$resultar["message"]='Ws no Retorna';
		$json['usuario'][]=$resultar;
		echo json_encode($json);
	}
?>
```