

**DESARROLLO DE UNA APLICACIÓN MÓVIL SOPORTADA EN EL  
PROTOCOLO NFC (ISO/IEC 18092) DE PROPÓSITO GENERAL COMO  
HERRAMIENTA DE CONECTIVIDAD Y SEGURIDAD**

**GUILLERMO ANDRÉS JAIMES BALCUCHO**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FÍSICO MECÁNICAS  
ESCUELA DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA Y DE  
TELECOMUNICACIONES  
BUCARAMANGA**

**2017**

**DESARROLLO DE UNA APLICACIÓN MÓVIL SOPORTADA EN EL  
PROTOCOLO NFC (ISO/IEC 18092) DE PROPÓSITO GENERAL COMO  
HERRAMIENTA DE CONECTIVIDAD Y SEGURIDAD**

**GUILLERMO ANDRÉS JAIMES BALCUCHO**

**Trabajo de Grado para Optar el Título de Ingeniero Electrónico**

**Director**

**EFREN DARIO ACEVEDO CARDENA**

**Mg. Ingeniero Electrónico**

**Codirector**

**OSCAR MAURICIO REYES TORRES**

**Dr. Ingeniero Electrónico**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FÍSICO MECÁNICAS  
ESCUELA DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA Y DE  
TELECOMUNICACIONES  
BUCARAMANGA**

**2017**

## DEDICATORIA

*Principalmente quiero darle las gracias a Dios que es gestor de todos nuestros acontecimientos en la vida, que me dio a mi familia, mis padres y la determinación para salir adelante durante el transcurso de mi carrera y de mi vida.*

*A mi querida madre Cruz Celina que al darme su vida me ha alentado a seguir adelante, que siempre confió en mí y será aquella persona que me impulse a ser cada día mejor.*

*A mi padre Guillermo por ser el alcahuete que me impulsó a estudiar, en la escuela, en el colegio, y en esta maravillosa universidad, que con su sacrificio y apoyo me ha ayudado a superar un largo camino además de afrontar diversas dificultades.*

*A mis hermanas Diana y Clara por ese apoyo incondicional, por soportarme tanto y estar siempre a mi lado, aún cuando cada una tiene su vida sé que nuestro amor será eterno.*

*A mis sobrinos Juan Esteban, Luis Santiago y Ana Sofía, que me han dado una esperanza del futuro que se aproxima y de la vida que aún queda por recorrer, su presencia me hace todos los días ser más fuerte.*

*A mis queridos abuelos, que nunca dejaré de extrañarlos, siempre estarán en mi corazón todo ese tiempo que dedicaron en mí, sé que su presencia continua conmigo aunque nos los pueda ver sé que jamás dejarán de luchar por mí.*

*A mis amigos y familia por ser ese soporte de confianza, de amistad y comprensión, por su compañía y afectos, por sus risas y agradables momentos.*

*Guillermo Andrés Jaimés Balcucho*

## AGRADECIMIENTOS

El autor expresa su agradecimiento a:

A Dios, que ha sido mi guía espiritual en cada fase de mi vida, gracias a su voluntad que logra lo imposible, posible. Cada camino que refleja el éxito se logra cuando culmina una etapa, donde concluir y realizar un proyecto personal es parte de la primera etapa de mi vida profesional, ser un ingeniero. Gracias a Dios he podido sentir y vivir todo esto.

Agradece al profesor Efrén Darío Acevedo, mi director, por su apoyo y confianza, por permitirme desarrollar mis ideas a través de su conocimiento, que además se tomó el arduo trabajo de interpretar y aclarar su conocimiento en mí acerca del tema, por orientarme en cada paso del desarrollo de la tesis.

A la Universidad Industrial de Santander por ser el pilar de mi desarrollo como ingeniero, a la escuela de Física y de Ingeniería Eléctrica, Electrónica y Telecomunicaciones por ser partícipes de mi proceso de formación que me permite tener un progreso tanto intelectual como personal.

Agradecimiento especial y conmemorativo a la selección de baloncesto de la Universidad Industria de Santander y a su comandante el entrenador Fabio Gómez, que en el transcurso de estos 10 años han sido mi segundo hogar y me han llenado de triunfos y alegrías.

A todas aquellas personas que de una u otra manera contribuyeron en la realización de este proyecto.

## CONTENIDO

	Pág.
<b>INTRODUCCIÓN</b>	<b>17</b>
<b>1. GENERALIDADES DEL PROYECTO</b>	<b>19</b>
<b>1.1 PLANTEAMIENTO DEL PROBLEMA</b>	<b>19</b>
<b>1.2 JUSTIFICACIÓN</b>	<b>20</b>
<b>1.3 OBJETIVOS</b>	<b>21</b>
<b>1.3.1 Objetivo General</b>	<b>21</b>
<b>1.3.2 Objetivos Específicos</b>	<b>21</b>
<b>1.4 ALCANCES</b>	<b>22</b>
<b>2. FUNDAMENTO TEÓRICO</b>	<b>23</b>
<b>2.1 NEAR FIELD COMMUNICATION (NFC) Y SUS UTILIDADES EN APLICACIONES MÓVILES</b>	<b>23</b>
<b>2.1.1 ISO/IEC 18092:2013(E)</b>	<b>23</b>
<b>2.1.2 Referencias normativas</b>	<b>23</b>
<b>2.1.3 Términos y definiciones</b>	<b>24</b>
<b>2.1.4 ISO / IEC 14443</b>	<b>24</b>
<b>3. METODOLOGÍA PARA EL DESARROLLO DEL PROYECTO</b>	<b>27</b>
<b>3.1 METODOLOGÍA ÁGILES PARA EL DESARROLLO DE SOFTWARE MÓVIL</b>	<b>27</b>
<b>3.1.1 Metodología guía</b>	<b>27</b>
<b>3.2 METODOLOGÍA PROPUESTA</b>	<b>28</b>

<b>3.2.1 DIAGNÓSTICO</b>	<b>29</b>
<b>3.2.2 EJECUCIÓN</b>	<b>35</b>
<b>3.2.3 PRODUCCIÓN</b>	<b>47</b>
<b>3.2.4 LANZAMIENTO.</b>	<b>85</b>
<b>3.2.5 PRUEBA Y REPARACIÓN DEL SISTEMA (CORRECCIÓN)</b>	<b>89</b>
<b>4. CONCLUSIONES</b>	<b>93</b>
<b>5. OBSERVACIONES Y TRABAJOS FUTUROS</b>	<b>95</b>
<b>BIBLIOGRAFÍA</b>	<b>97</b>
<b>REFERENCIAS BIBLIOGRAFÍA</b>	<b>99</b>
<b>ANEXOS</b>	<b>102</b>

## LISTA DE FIGURAS

	<b>Pág.</b>
Figura 1. Metodología agil Mobile-D	27
Figura 2. Fases de la metodología para el desarrollo de la aplicación	28
Figura 3. Características de la fase Diagnostico	29
Figura 4. Diagrama de Gant para actividades	31
Figura 5. Operación de dispositivos móviles con NFC	33
Figura 6. Ciclos de programación	34
Figura 7. Ciclos de programación con características de desarrollo	35
Figura 8. Características de la fase Ejecución	35
Figura 9. Plataforma Google Play.	37
Figura 10. Video tutoriales	38
Figura 11. Interfaz del programa NetBeans	38
Figura 12. Interfaz del programa CorelDraw X7	39
Figura 13. Interfaz del programa Android Studio	40
Figura 14. Guía de usuario web de Android Studio	41
Figura 15. Mapa conceptual del estado del arte NFC	42
Figura 16. Mapa conceptual programación en java dirigida a objetos	43
Figura 17. Mapa conceptual Material Design	44
Figura 18. Mapa conceptual diseño de logos	44

Figura 19. Ciclos de programación para desarrollar aplicaciones en Android Studio	45
Figura 20. Interacción de procesos para desarrollar aplicaciones en Android Studio	46
Figura 21. Pre-diseño aplicación NFC	46
Figura 22. Pre-diseño aplicación NFC con actividad de contraseña	47
Figura 23. Características de la fase Producción	48
Figura 24. Características principales de la aplicación tomadas por el estado del arte	49
Figura 25. Ciclos de programación java dirigida a objetos	49
Figura 26. Pre-diseño aplicación NFC con asignación de colores y nombres	50
Figura 27. Pre-diseño logo UIS NFC	50
Figura 28. Método de consulta en Android Developers	51
Figura 29. Actividades de la aplicación por medio de NFC	52
Figura 30. Paleta de colores Material Design	53
Figura 31. Elemento Tabs con pestañas fijas	56
Figura 32. Elemento tarjetas con parámetros de diseño	58
Figura 33. Rejillas para la creación de logo	58
Figura 34. Formas Keyline con sombras de línea para logos	59
Figura 35. Logo 1 diseñado con cuadros de escala y sombras de línea	59
Figura 36. Logo 2 diseñado con cuadros de escala y sombras de línea	60
Figura 37. Diagrama de flujo Activity Contraseña	65
Figura 38. Interfaz gráfica Activity Contraseña	66
Figura 39. Diagrama de flujo Activity Información	67

Figura 40. Interfaz gráfica Activity Información	67
Figura 41. Diagrama de flujo Activity Pagina UIS	68
Figura 42. Interfaz gráfica Activity Pagina UIS	68
Figura 43. Diagrama de flujo ajustes NFC	69
Figura 44. Diagrama de flujo menú retroceso	70
Figura 45. Interfaz gráfica Tabhost en Android Studio	71
Figura 46. Interfaz gráfica TabWidget en Android Studio	72
Figura 47. Diagrama de flujo Tabhost	72
Figura 48. Diagrama de flujo para la tarjeta enviar mensaje	74
Figura 49. Interfaz gráfica para la tarjeta enviar mensaje	75
Figura 50. Diagrama de flujo para la tarjeta recibir mensaje	75
Figura 51. Interfaz gráfica para la tarjeta recibir mensaje	76
Figura 52. Diagrama de flujo para la tarjeta enviar foto	77
Figura 53. Interfaz gráfica para la tarjeta enviar foto	77
Figura 54. Diagrama de flujo para la tarjeta recibir foto	78
Figura 55. Interfaz gráfica para la tarjeta recibir foto	78
Figura 56. Diagrama de flujo para la tarjeta programar Tag	80
Figura 57. Interfaz gráfica para la tarjeta programar Tag	80
Figura 58. Diagrama de flujo para la tarjeta registro tipo texto	81
Figura 59. Interfaz gráfica para la tarjeta registro tipo texto	82
Figura 60. Diagrama de flujo para la tarjeta registro tipo URI	83
Figura 61. Interfaz gráfica para la tarjeta registro tipo URI	83
Figura 62. Diagrama de flujo para la tarjeta tomar selfie	84

Figura 63. Interfaz gráfica para la tarjeta tomar selfie	84
Figura 64. Características de la fase Lanzamiento	85
Figura 65. Características de la fase Prueba y Reparación	89

## LISTA DE TABLAS

	<b>Pág.</b>
Tabla 1. Valores de pantalla en diferentes unidades	60
Tabla 2. Escalado de las imágenes	61
Tabla 3. Pruebas para actividad CHATS	87
Tabla 4. Pruebas para actividad FOTOS	87
Tabla 5. Pruebas para actividad TAGS	87
Tabla 6. Prueba de traspaso de datos tipo mensaje de texto	90
Tabla 7. Prueba de traspaso de datos tipo imagen	91
Tabla 8. Prueba de traspaso de datos tipo registro NFC	92
Tabla 9. Plataformas que contienen protocolo NFC.	102
Tabla 10. Términos y definiciones del protocolo NFC	104

## LISTA DE ANEXOS

	<b>Pág.</b>
ANEXO A. PLATAFORMAS DE DISPOSITIVOS MOVILES QUE CONTIENEN EL PROTOCOLO NFC	102
ANEXO B. PROTOCOLO NFC EN LA ISO/IEC 18092	104
ANEXO C. ESTADO DEL ARTE NFC EN LA PLATAFORMA ANDROID	107
ANEXO D. ESPECIFICACIONES DE SMART RING JAKCOM	113
ANEXO E. CODIGO DEL ARCHIVO MANIFESTS UIS NFC	114

## RESUMEN

**TITULO:** DESARROLLO DE UNA APLICACIÓN MÓVIL SOPORTADA EN EL PROTOCOLO NFC (ISO/IEC 18092) DE PROPÓSITO GENERAL COMO HERRAMIENTA DE CONECTIVIDAD Y SEGURIDAD.\*

**AUTORES:** GUILLERMO ANDRÉS JAIMES BALCUCHO\*\*

**PALABRAS CLAVE:** APLICACIÓN, NFC, CONECTIVIDAD, DISPOSITIVO MÓVIL.

### DESCRIPCIÓN:

La tecnología (NFC) Near Field Communication promueve los facilitadores de la computación ubicua, en la asignatura comunicaciones se estudian los medios de comunicación vigentes y a desarrollar elementos que facilitan la conectividad. El desarrollo de software NFC permite a través de la automatización ubicua emplear cualquier dispositivo móvil para el uso diario en la realización de tareas como pasar texto, imágenes o cualquier tipo de bits programables.

Este proyecto de grado tiene como objetivos, entender características del protocolo NFC para la comunicación entre dispositivos móviles o electrónicos, su implementación, los recursos necesarios para desarrollar software NFC y una metodología para el fácil desarrollo de aplicaciones. Se ejecutarán tareas sencillas con un solo instrumento (dispositivo móvil), el cual nos llevará a comprobar la versatilidad de la tecnología NFC.

Se crea una aplicación y documentación que permitirán a futuro desarrollar aplicaciones en la plataforma Android donde los usuarios de dispositivos móviles puedan en forma fácil realizar tareas como: pagar cuentas vía electrónica, ingresar a la habitación de un hotel sin una tarjeta ID, mostrar la localización de la persona portadora del dispositivo y otras funciones donde se utilice la comunicación entre dispositivos electrónicos.

Con el presente trabajo se espera poder contribuir en la descripción del protocolo de comunicación NFC en dispositivos móviles. Se busca generar el incremento de soluciones mediante el uso de un solo dispositivo, lo que conlleva a hacer de los celulares y tabletas la herramienta que permita la comunicación segura entre dispositivos electrónicos.

---

\* Trabajo de Grado

\*\* Facultad de Ingenierías Físico - Mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones (E<sup>3</sup>T), Director Mg. Efrén Darío Acevedo Cardenas, Codirector Dr. Oscar Mauricio Reyes Torres

## ABSTRACT

**TITLE:** DEVELOPMENT OF A MOBILE APPLICATION SUPPORTED IN THE PROTOCOL NFC (ISO / IEC 18092) OF GENERAL PURPOSE AS A TOOL OF CONNECTIVITY AND SAFETY \*

**AUTORES:** GUILLERMO ANDRÉS JAIMES BALCUCHO\*\*

**KEY WORDS:** APPLICATION, NFC, CONNECTIVITY, MOBILE DEVICE.

### DESCRIPTION:

The Near Field Communication technology (NFC) promotes the facilitators of ubiquitous computing. The study of communications area explores the current media and the development of elements that allow easy connectivity. Use of ubiquitous automation in NFC software development allows any electronic device that supports apps to be used in the performance of tasks like sending texts, pictures or any kind of programmable bits in a faster and more secure way than usual.

This graduation project allows to understand features of the NFC Protocol in communication between mobile or electronic devices and its implementation, the necessary resources for the development of NFC software and a simple methodology to create apps. This is demonstrated by executing simple tasks using only one mobile device that allows to check the versatility of NFC technology.

An app is created and its corresponding documentation that will allow future development of apps on the Android platform where mobile device users could access to their credit or debit cards in order to pay bills electronically in seconds just by putting the device close to the receptor or entering in the room of a hotel without a key, show the location of the person carrying the device and other functions where the communication between electronic devices is used.

With this work is expected to contribute in the description of the Protocol of communication NFC on mobile devices. It is pretended to generate the increase of solutions through the use of a single device, which leads to make cell phones and tablets the tool that allows secure communication between electronic devices.

---

\* End pregraduate work

\*\* Physical-Mechanical Engineering Faculty. Electrical, Electronics and Telecommunications (E<sup>3</sup>T) Engineering School, Director MI Efren Dario Acevedo Cardenas, Codirector Ph.D . Oscar Mauricio Reyes Torres.

## INTRODUCCIÓN

Los dispositivos móviles son de tamaño pequeño la mayoría cabe en los bolsillos, con capacidades de procesamiento, con conexión permanente o intermitente a internet o una red, con memoria limitada, diseñado específicamente para una función, pero que pueden llevar a cabo otras funciones más generales. Hay datos que aseguran que, en la Tierra habrá una mayor cantidad de dispositivos móviles que de personas. Según el informe realizado por Cisco, la demanda de nuevas conexiones móviles por parte de los consumidores va en aumento y al parecer no hay signos de que vaya a cambiar esta tendencia. Para el año 2017, Cisco prevé que existirán más de 10.000 millones de dispositivos móviles en el mundo [1].

Esta tendencia hace que por cada habitante en la tierra haya prácticamente un dispositivo móvil, dato que llama la atención al hacer de los dispositivos móviles una herramienta necesaria para los seres humanos, el propósito de realizar distintas operaciones o funciones con dicha herramienta causa un gran impacto en la solución de las necesidades de un usuario.

Hemos visto como el uso de la internet en los celulares conecta a las personas diariamente, por eso se piensa que los dispositivos móviles pueden ser la herramienta necesaria para la solución de múltiples tareas sin necesidad de otro tipo de dispositivos. La implementación del Near Field Communication (NFC) en los dispositivos móviles, pretende hacer de estos un instrumento para la conectividad diaria de los usuarios.

Near Field Communication (NFC) es una tecnología de conectividad inalámbrica de corto alcance basada en estándares que hace la vida más fácil y más conveniente para los consumidores de todo el mundo, haciendo que sea más fácil llevar a cabo operaciones de cambio, contenido digital y conexión de los

dispositivos electrónicos con un toque. NFC es compatible con cientos de millones de tarjetas sin contacto y lectores ya desplegados en todo el mundo [2].

Los dispositivos móviles con NFC necesitan del desarrollo de software en alguna de las plataformas existentes, la más utilizada a nivel global es la plataforma Android de la compañía google, el sistema operativo Android soporta las especificaciones del NFC fórum para desarrollar Apps útiles al servicio de la comunicación por NFC. Estas Apps soportan las utilidades de la tecnología NFC pudiendo así hacer de los dispositivos móviles la herramienta necesaria para la conectividad y soluciones de las demandas de los usuarios.

# 1. GENERALIDADES DEL PROYECTO

## 1.1 PLANTEAMIENTO DEL PROBLEMA

Hoy se presenta un momento de oportunidad con la tecnología NFC, ya que se está haciendo cada vez más popular con todas las nuevas funciones que va adquiriendo, lo cual llama la atención de los usuarios a la hora de obtener un dispositivo móvil. Una de las tendencias de los dispositivos móviles es implementar dentro de su hardware un chip con NFC, para realizar diferentes operaciones como lo son compartir archivos, para pagos, cambiar la configuración del terminal móvil, etc.

En la Universidad Industrial de Santander (UIS) no se tienen indicios de experiencias previas en el desarrollo de aplicaciones basadas en NFC, ya que en el 2004 las compañías Nokia, Philips y Sony, se juntaron para fundar el NFC fórum, es una tecnología por así decir reciente. Se hace necesario comenzar a trabajar sobre esta tendencia a nivel mundial porque cada vez adquiere más campo de acción en la electrónica de consumo.

Debido a que no hay ningún antecedente sobre el NFC en la Universidad Industrial de Santander (UIS) no se cuenta con una aplicación o al menos con un prototipo que permita realizar demostraciones la eficacia, la confiabilidad y en fin la viabilidad de usar las aplicaciones con NFC en problemas cotidianos.

Uno de los campos donde más utilidad puede tener las aplicaciones basadas en NFC es la seguridad. En este campo particular se presenta la siguiente necesidad:

- Uso de tarjetas plásticas con chip para abrir puerta.
- Uso de tarjetas plásticas para retirar dinero de un cajero.

- Uso de tarjetas plásticas para pagar metrolínea.
- Otros tipos de pagos o cupones que se puedan realizar vía electrónica.

Con el uso de NFC este tipo de tarjetas pasan de ser obsoletas por varias razones:

El dispositivo móvil elimina la necesidad de usar un sinnúmero de tarjetas plásticas, se eleva el nivel de seguridad que experimenta el usuario, se eleva también la versatilidad de los servicios que el usuario experimenta pues se mezclan con toda la potencia de cómputo, almacenamiento y conectividad de tal dispositivo.

El anterior problema es un problema que se deja para una investigación a futuro puesto que este proyecto de grado representa un aporte a esas necesidades y se limita a responder las necesidades primordiales para llegar a este fin. Se requiere implementar una aplicación móvil basada en NFC, la primera aplicación de investigación en la Universidad Industrial de Santander (UIS) UIS NFC que permita la conexión por medio de NFC entre dispositivos o Tags, con la finalidad de pasar datos ya sean imágenes o mensajes de texto. Igualmente se requiere documentar esta experiencia en forma de una metodología útil para continuar este tipo de desarrollo en la Universidad Industrial de Santander (UIS).

## **1.2 JUSTIFICACIÓN**

Los teléfonos móviles son cada vez más frecuentes en la población mundial, la mayoría posee un teléfono móvil el cual se puede utilizar con más frecuencia a medida que se desarrollan soluciones específicas enfocadas en los teléfonos o dispositivos móviles. El principal uso que nos interesa es el de Teléfonos con NFC, algunos teléfonos de gama alta y gama media incluyen NFC y cada vez comienza a ser más habitual verlos en el resto de móviles nuevos, en el 2016 se espera un aumento considerable en el uso del NFC [3].

El ANEXO A contiene la Tabla 9, en donde se observan las plataformas creadas para el uso de teléfonos o dispositivos móviles que contienen en su diseño, el protocolo NFC. La plataforma Android ofrece una mayor demanda de usuarios a nivel mundial, cuenta con una gran inversión para seguir con el desarrollo de mejores versiones. Los dispositivos móviles más usados a nivel mundial son fabricados por la compañía SAMSUNG que implementó en sus teléfonos móviles y dispositivos móviles la plataforma Android. Sin contar que otras compañías de dispositivos móviles a nivel mundial integran en sus dispositivos la plataforma Android, es por eso que el desarrollar aplicaciones en Android, genera un impacto global para las necesidades de la gente.

### **1.3 OBJETIVOS**

**1.3.1 Objetivo General** Desarrollar una aplicación móvil soportada en el protocolo NFC (ISO/IEC 18092) de propósito general como herramienta de conectividad y seguridad.

**1.3.2 Objetivos Específicos** El cumplimiento del objetivo general del trabajo de grado comprende:

- Formulación de un grupo de historias sencillas de necesidades de usuario.
- Desarrollar una aplicación móvil que responda a las historias de usuario soportada sobre la plataforma Android que permita establecer comunicación entre dos dispositivos de forma segura.
- Implementación de un módulo de comunicación entre un dispositivo móvil y un Tag NFC.
- Formular una metodología de desarrollo de aplicaciones basado en NFC conforme a la experiencia ganada en el desarrollo del proyecto.
- Validar la aplicabilidad de la tecnología NFC para las necesidades de usuario.

- Documentar la investigación para trabajos futuros.

## 1.4 ALCANCES

Determinar las características que debe poseer una aplicación móvil soportada en el protocolo de comunicación (ISO/IEC 18092) para NFC revisando el estado del arte de las ya creadas para tal fin. Desarrollar un estado del arte para adquirir conocimiento sobre las funcionalidades de las aplicaciones en la plataforma Android con el sistema NFC, basado en esto se construirán ideas sobre cómo se debe transmitir datos y en función de que tarea se desarrollara la comunicación entre dos dispositivos móviles.

Mediante la metodología ágil creada con las necesidades del desarrollo de la aplicación y con guía en la metodología Mobile-D, se producirá la aplicación UIS NFC para dispositivos móviles que contiene seguridad y NFC, documentaremos todas las fases que cumplan con el desarrollo de la aplicación móvil para posibles mejoras de la aplicación y de la metodología creada o de Mobile-D.

Trabajar la aplicación en “Modo desarrollador” en el dispositivo móvil Samsung S6 Edge para efectuar prueba piloto. Presentar informe de los resultados obtenidos donde se describirá en el documento los resultados de la aplicación, si cumple satisfactoriamente y que errores surgen cuando se ejecuta la comunicación entre dos dispositivos móviles mediante la aplicación desarrollada, si es de forma segura y que ventajas deja el uso del NFC. Recomendar la metodología a seguir para crear aplicaciones con NFC.

Desarrollar la aplicación permitirá que la mayoría de usuarios de Android puedan acceder a UIS NFC, para conocer sobre las funcionalidades del protocolo NFC y de sus características, es por eso que la aplicación se creara con una mínima versión de Android que a nivel mundial tenga estadísticas de mayor utilidad.

## 2. FUNDAMENTO TEÓRICO

### 2.1 NEAR FIELD COMMUNICATION (NFC) Y SUS UTILIDADES EN APLICACIONES MÓVILES

(NFC) Es una tecnología creada para la comunicación inalámbrica, como sus siglas lo indican (NFC) o comunicación de campo cercano es de corto alcance, alta frecuencia y permite el intercambio de datos entre dispositivos. La tecnología NFC es compatible con cientos de millones de tarjetas sin contacto y lectores ya desplegados en todo el mundo [2]. NFC es una tecnología que se deriva de los principios de RFID (identificación por radiofrecuencia), los teléfonos o gadgets comienzan a transferir datos entre sí, aunque específicamente en el caso de NFC el campo es muy reducido, por eso se tiene que juntar los equipos todo el tiempo durante la transmisión de datos. Los estándares de NFC utilizan los protocolos de comunicación y formatos de intercambio de datos, y están basados en ISO 14443 (RFID, radio-frequency identificación) e ISO/IEC 18092:2013(E).

**2.1.1 ISO/IEC 18092:2013(E)** Es una Norma Internacional que especifica la interfaz y el protocolo para la comunicación inalámbrica sencilla entre dispositivos móviles de acoplamiento cerrado. La interfaz NFC y Protocolo (1-NFCIP) estándar permite, pero no especifica, aplicaciones en productos de red y los equipos de consumo. Se observan las características en el alcance de la lectura [4].

**2.1.2 Referencias normativas** Los siguientes documentos referenciados son indispensables para la aplicación de la ISO/IEC 18092:2013(E):

UIT-T V.41: 1988, sistema de control de errores independientes del código [4].

- ISO / IEC 13157-1: 2010, Tecnología de la información - Telecomunicaciones e intercambio de información entre los sistemas NFC -

Seguridad - Parte 1: NFC-SEC-1 NFCIP servicios de seguridad y protocolo[4].

- ISO / IEC 14443-2: 2010, Tarjetas de identificación - Tarjetas de circuito integrado sin contacto - Tarjetas de proximidad - Parte 2: potencia de frecuencia y señal de interfaz de radio[4].
- ISO / IEC 14443-3: 2011, Tarjetas de identificación - Tarjetas de circuito integrado sin contacto - Tarjetas de proximidad - Parte 3: Inicialización y anticolisión[4].
- ISO / IEC 14443-4: 2008, Tarjetas de identificación - Tarjetas de circuito integrado sin contacto - Tarjetas de proximidad - Parte 4: Protocolo de transmisión[4].

**2.1.3 Términos y definiciones** En el ANEXO B se crea la Tabla 10 con las características que posee la ISO/IEC 18092 para el entendimiento del protocolo NFC.

**2.1.4 ISO / IEC 14443** El ISO / IEC 14443 trata sobre el protocolo con Tarjetas de identificación, Tarjetas de circuito integrado sin contacto, Tarjetas de proximidad. Es un estándar internacional que define y especifica las tarjetas de proximidad que se desean utilizar para la identificación u otras necesidades aplicables con tarjetas dentro del protocolo de transmisión en la comunicación entre dichos dispositivos [5].

El estándar ISO 14443 está constituido por partes donde se especifican los parámetros con los que debe cumplir el estándar, las partes están constituidas de la siguiente forma:

- ISO / IEC 14443-1: 2008 Parte 1: Características físicas
- ISO / IEC 14443-2: 2010 Parte 2: Interfaz de potencia de frecuencia y la señal de Radio

- ISO / IEC 14443-3: 2011 Parte 3: Inicialización y anticolisión
- ISO / IEC 14443-4: 2008 Parte 4: Protocolo de transmisión.

Según lo establecido por la ISO 14443 las tarjetas pueden ser de tipo A y tipo B, donde las tarjetas se comunican a través de radio frecuencia a 13,56 MHz. Una de las principales diferencias de estos dos tipos de tarjetas se refiere a métodos de modulación, esquemas diferentes en señal de radio e interfaz de potencia de frecuencia y procedimientos de inicialización e anticolisión del protocolo en codificación. Ambas tarjetas Tipo A y Tipo B utilizan el mismo protocolo de transmisión [5].

El protocolo de transmisión especifica el intercambio en un bloque de datos y mecanismos relacionados:

- Bloque de datos de encadenamiento
- esperando prórroga multi-activación

ISO / IEC 14443 usos siguientes términos para los componentes:

- PCD: dispositivo de acoplamiento de proximidad (el lector de tarjetas )
- PICC: proximidad tarjeta de circuito integrado

La parte física de las tarjetas cumplirá de manera específica que la tarjeta debe ser conforme a la norma ISO / IEC 7810 o ISO / IEC 15457-1, o "un objeto de cualquier otra dimensión"[6].

Implementaciones notables:

- MIFARE tarjetas.
- Calypso (sistema de billeteaje electrónico).
- Los pasaportes biométricos.
- EMV de tarjetas de pago (PayPass, payWave, ExpressPay).
- Documento de identidad alemán.

Realiza dos modos de funcionamiento, todos los dispositivos del estándar NFCIP-1 deben soportar ambos modos:

- Activo: donde el emisor como el receptor son dispositivos que generan su propio campo electromagnético, que utilizarán para transmitir sus datos, entonces hablaríamos de que ambos dispositivos son emisor-receptor y receptor-emisor lo que da a entender que son dispositivos que transmiten información entre sí.
- Pasivo: solo un dispositivo genera el campo electromagnético y el otro se aprovecha de la modulación de la carga para poder transferir los datos. El iniciador de la comunicación es el encargado de generar el campo electromagnético.

El protocolo NFCIP-1 implementado hasta ahora puede funcionar a diversas velocidades como 106, 212, 424 o 848 Kbit/s. Según el entorno en el que se trabaje, las dos partes pueden ponerse de acuerdo de a qué velocidad trabajar y reajustar el parámetro en cualquier instante de la comunicación [6].

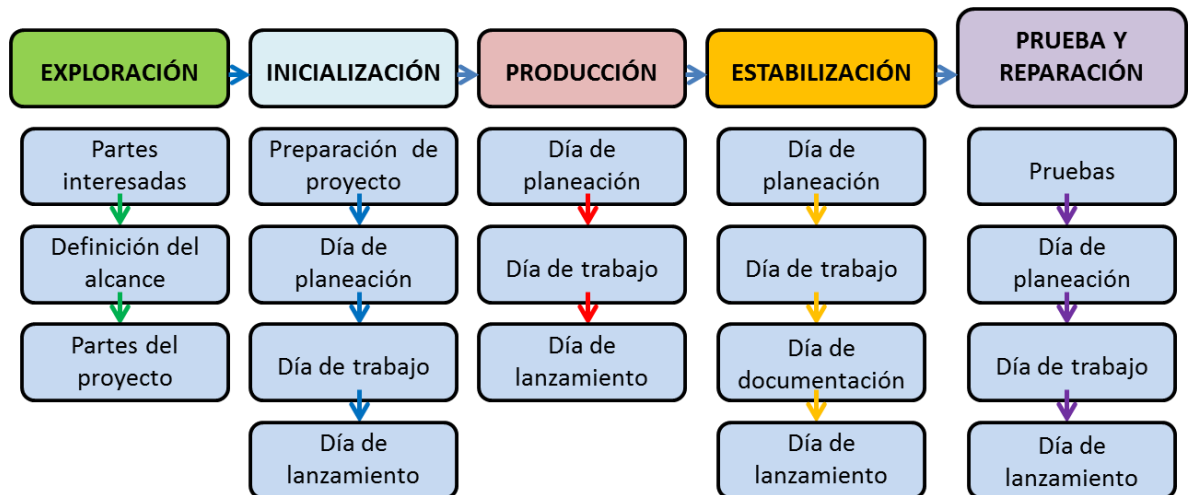
### 3. METODOLOGÍA PARA EL DESARROLLO DEL PROYECTO

#### 3.1 METODOLOGÍA ÁGILES PARA EL DESARROLLO DE SOFTWARE MÓVIL

Las metodologías ágiles contienen propiedades que las hacen totalmente aplicables al dominio del software en los dispositivos móviles. Por eso se identifican los métodos ágiles como la solución potencial para el desarrollo de software en móviles. La metodología se apoya en las bases (home ground) haciendo un análisis comparativo para probar la idoneidad de los métodos ágiles sobre el desarrollo de software para móviles[7].

**3.1.1 Metodología guía** Para el desarrollo de aplicaciones basado en NFC utilizaremos de referencia la metodología ágil Mobile-D.

**Figura 1. Metodología ágil Mobile-D**



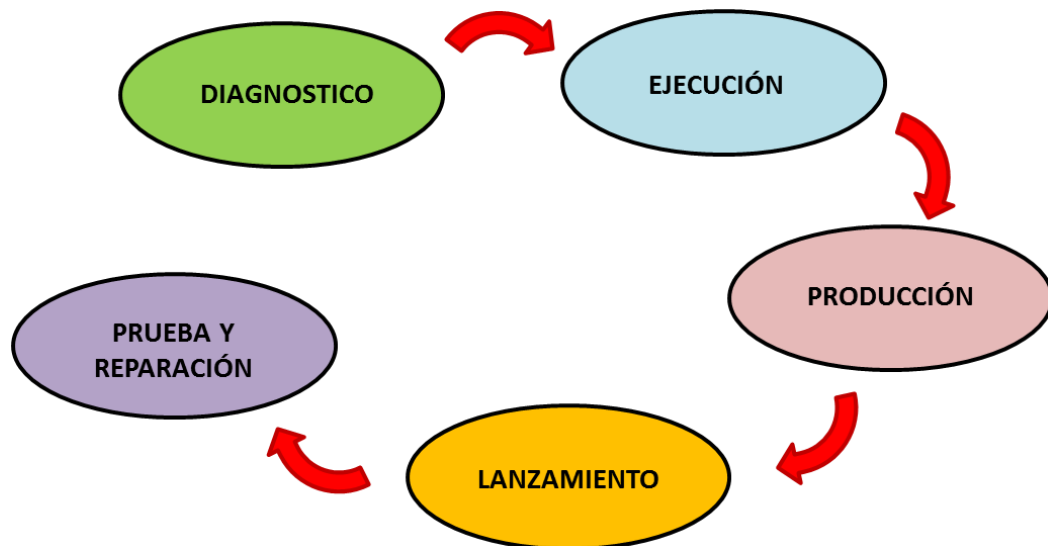
*Adaptada de Fumero, A. (2016). Metodología de desarrollo ágil para sistemas móviles. [Figura].*

Mobile-D apunta a la necesidad de disponer de un ciclo de desarrollo muy rápido para equipos muy pequeños. La metodología se ilustra perfectamente en la Figura 1, Mobile-D está pensado para grupos de no más de 10 desarrolladores colaborando en un mismo espacio físico. Si se trabaja con los ciclos de desarrollo de la metodología, los proyectos deberían finalizar con el lanzamiento de productos completamente funcionales en menos de diez semanas.

### 3.2 METODOLOGÍA PROPUESTA

El desarrollo del presente trabajo de grado toma como referencia la metodología MOBILE-D que forma parte de las metodologías ágiles para el desarrollo de software móvil y formular una metodología de desarrollo para aplicaciones en NFC conforme a la experiencia ganada en la ejecución del proyecto. La siguiente Figura 2. Fases de la metodología para el desarrollo de la aplicación se muestra las etapas de la metodología propuesta para el desarrollo de la aplicación soportada en el protocolo NFC.

**Figura 2. Fases de la metodología para el desarrollo de la aplicación**



**3.2.1 DIAGNÓSTICO** La fase de diagnóstico realiza un reconocimiento del problema a solucionar y los medios con los que se ejecutara la solución se muestra en la Figura 3.

**Figura 3. Características de la fase Diagnostico**



**3.2.1.1 Participación de los clientes** Se observa que a nivel mundial 2 de las aplicaciones más grandes y conocidas por sus servicios son WhatsApp e Instagram. WhatsApp es una aplicación de mensajería instantánea para teléfonos móviles, que envía y recibe mensajes mediante Internet e Instagram es una aplicación para subir fotos y videos mediante internet, donde se pueden aplicar efectos fotográficos mediante filtros. Las aplicaciones descritas en esencia cumplen con estos requisitos pero además tienen otras funciones no mencionadas que se obvian por razones de importancia.

La utilidad de una aplicación se denota por el número de usuarios que la utilizan, la red social de fotografía Instagram anunció que cuenta con 500 millones de usuarios activos cada mes, 300 millones de los cuales se conectan a diario al servicio [8]. La compañía WhatsApp, informaba que la aplicación (App) había

alcanzado la increíble cifra de 1.000 millones de usuarios [9]. Los clientes están interesados en transmitir información de forma rápida y segura dependiendo del tipo de información los usuarios utilizan ciertas aplicaciones, por eso aplicaciones que transmitan mensajes de texto e imágenes son de necesidad más frecuente para los clientes.

**Historias de usuarios** Para crear una aplicación que tenga gran demanda de usuarios se hace alusión a historias sencillas de usuarios, con las cuales se genera las especificaciones que debe cubrir la aplicación a desarrollar. Las siguientes son las historias de usuarios en donde se involucra a los posibles clientes para construir la aplicación:

- Hay lugares en los que no puedo transmitir mensajes de texto con información privada que solo quiero que vea un compañero asignado, pues no me entra el internet en el lugar o zona especificada.
- A veces quiero compartir fotos que solo deseo posea una persona veo que la privacidad es fuente fundamental de mi seguridad y deseo compartir esas fotos solo con alguien específico.
- A veces cuando no tengo internet deseo compartir alguna información de mi teléfono móvil, casi no se usar el bluetooth y creo que si lo activo todos los demás pueden ver mi información privada que deseo enviar.
- Pienso que mi teléfono móvil aparte de servir en las llamadas puede utilizarse para cosas de la vida más cotidiana, que sirva como herramienta de conectividad y de versatilidad para solucionar problemas.
- Si el teléfono móvil es de uso personal, podríamos utilizarlo como una tarjeta de identificación o como una tarjeta sin contacto para evitar cargar multiplex tarjetas de plástico.

**3.2.1.2 Plan de proyecto** Se realiza un plan de proyecto mediante el diagrama de gant de la Figura 4. Con el objetivo de sectorizar las funciones y las etapas que se desarrollan durante la creación de la aplicación móvil.

**Figura 4. Diagrama de Gant para actividades**

ACTIVIDAD	INICIO	FINAL	SEM 1	SEM 2	SEM 3	SEM 4	SEM 5	SEM 6	SEM 7	SEM 8	SEM 9	SEM 10	SEM 11	SEM 12	SEM 13	SEM 14	SEM 15	SEM 16	SEM 17	SEM 18	SEM 19	SEM 20	SEM 21
ACTIVIDAD A	07-mar-16	02-may-16	■	■	■	■	■	■	■	■													
ACTIVIDAD B	04-abr-16	30-may-16					■	■	■	■	■	■	■	■									
ACTIVIDAD C	02-may-16	27-jun-16									■	■	■	■	■	■	■	■	■				
ACTIVIDAD D	30-may-16	01-ago-16													■	■	■	■	■	■	■	■	■

Se proporciona el tiempo establecido en que cada actividad debe ser desarrollada, dichas actividades se definen de la siguiente manera:

**ACTIVIDAD A: ESTADO DEL ARTE NFC**, la función de esta actividad es crear un anexo de investigación en donde compilamos los resultados de otras aplicaciones ya creadas en Play Store. Se extraen las características más relevantes que implementaremos en la aplicación.

**ACTIVIDAD B: PROGRAMACION EN JAVA DIRIGIDA A OBJETOS**, la actividad consiste en aprender la programación en java dirigida a objetos y sus conceptos básicos, para el diseño y programación en Android Studio plataforma que desarrolla las aplicaciones para el sistema operativo Android.

**ACTIVIDAD C: MATERIAL DESING Y DISEÑO DE LOGOS**, la función de esta actividad es la de guiar la estructura interna, el diseño estético y las partes específicas que componen una aplicación, por medio de Material Desing. La creación de los logos en Corel y diseños gráficos que requiera la aplicación.

**ACTIVIDAD D: PROGRAMACION EN ANDROID STUDIO**, la actividad consiste en aprender la programación en Android Studio para desarrollar la aplicación móvil, manejo de conceptos del programa y ejecución de sus funciones, Android Studio es el IDE oficial para Android [10].

**3.2.1.3 Requisitos iniciales** Las historias de usuarios hacen referencia a necesidades que un usuario tiene al manejar un dispositivo móvil (Smartphone), se observa que los usuarios quieren transmitir información por medios diferentes al internet y utilizar el dispositivo móvil como herramienta de conectividad. Los siguientes son conceptos que dan a entender y justificar el plan de proyecto que soluciona la mayoría de peticiones en las historias de usuarios.

**Smartphone** El teléfono inteligente (Smartphone en inglés) es un tipo de teléfono móvil construido sobre una plataforma informática móvil, con mayor capacidad de almacenar datos y realizar actividades, semejante a la de una minicomputadora, y con una mayor conectividad que un teléfono móvil convencional [6].

**¿Qué es NFC?** NFC (Near Field Communication) se crea para conseguir un protocolo compatible con las tecnologías sin contactos existentes en ese momento, Mifare de Philips, FeliCa de Sony y otras más [11]. La tecnología NFC es compatible con el estándar ISO/IEC-14443 para tarjetas de proximidad sin contactos, se utiliza el NFC-forum [12]. Para establecer los alcances de la tecnología NFC.

**¿Qué hace?** Los dispositivos habilitados con NFC son los únicos que pueden soportar tres modos de funcionamiento: el modo lector / escritor, modo P2P (peer-to-peer) y el modo emulación de tarjeta como lo muestra la Figura 5.

**Figura 5. Operación de dispositivos móviles con NFC**



*Adaptada de NFC Forum. (2017). Operación de dispositivos NFC en 3 modos [Figura].*

**Modo lector / escritor** Modo lector / escritor permite a los dispositivos NFC para leer la información almacenada en las etiquetas NFC de bajo costo incrustados en carteles inteligentes y pantallas.

**Modo P2P (Peer-to-Peer)** Modo P2P permite que dos dispositivos habilitados con NFC se comuniquen entre sí e intercambiar información y compartir archivos, por lo que dispositivos habilitados con NFC pueden compartir rápidamente la información de contacto y otros archivos con un toque.

**Modo emulación de tarjeta** En el modo emulación de la tarjeta, el dispositivo con NFC se comunica con un lector externo muy similar a una tarjeta inteligente sin contacto tradicional.

**3.2.1.4 Establecimiento del proyecto** En la etapa de establecimiento del proyecto se definen los recursos (técnicos) que se necesitan para que el proyecto de desarrollo de software inicie.

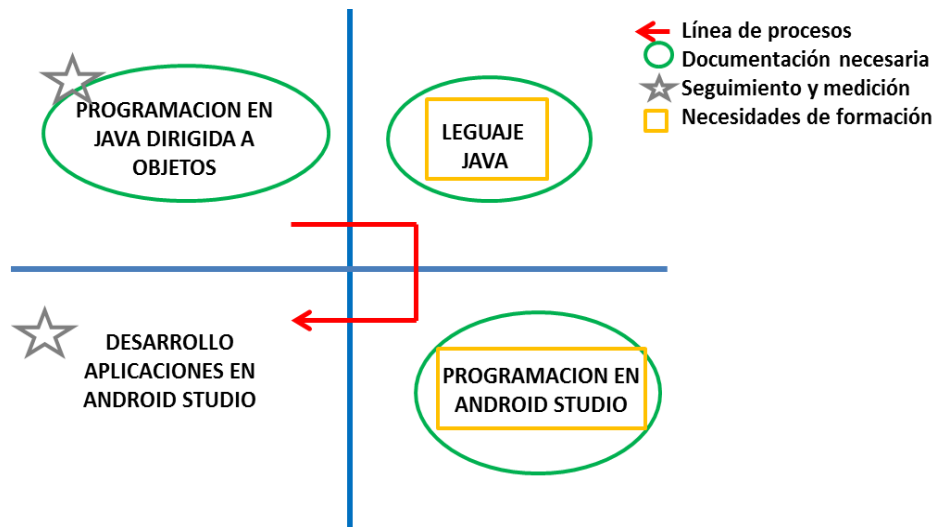
**Definición de la arquitectura** La etapa consiste en definir suficientemente los problemas de arquitectura para que el proyecto se realice con éxito. Observamos los problemas de arquitectura en forma de ciclos con los que se desarrolla una aplicación básica en Android en la Figura 6.

**Figura 6. Ciclos de programación**



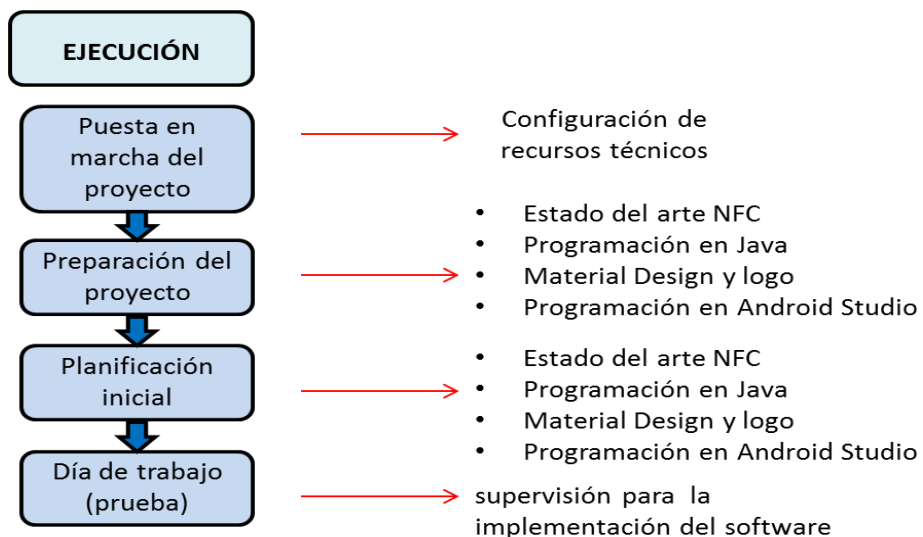
**Proceso de establecimiento** Se construye el proceso de establecimiento desarrollando una serie de pasos que se observan en la Figura 7, donde se agregan características de desarrollo dentro de los ciclos de programación de la Figura 6. Para entender las necesidades de cada ciclo.

**Figura 7. Ciclos de programación con características de desarrollo**



**3.2.2 EJECUCIÓN** En esta etapa se reúne toda la información obtenida sobre el proyecto y se planifican las tareas a realizar en las fases posteriores. Se preparan los planes para las siguientes fases y se establece los recursos tecnológicos. Esta fase se muestra en la Figura 8. Características de la fase Ejecución

**Figura 8. Características de la fase Ejecución**



**3.2.2.1 Puesta en marcha del proyecto** El propósito de esta etapa es la configuración de los recursos físicos y técnicos para el proyecto, así como el medio para el seguimiento del proyecto, capacitar al equipo de proyecto como es necesario, y establecer las formas específicas de proyectos para comunicarse con el grupo de clientes.

**3.2.2.2 Preparación proyecto** Se utiliza el NFC como medio de solución para satisfacer las historias de usuarios, utilizando los modos Peer-to-Peer y lector/escritor. Se hace una investigación para observar los resultados de la tecnología NFC en aplicaciones móviles, generando un estado del arte que recopilar información de las necesidades del usuario y toma las características principales que posee la aplicación.

**Estado del arte aplicaciones NFC** Se realiza un estado del arte en la plataforma de Google Play, o Play Store donde encontraremos por medio de la palabra clave “NFC” en el buscador de aplicaciones una amplia gama de proyectos con el fin de compilar resultados de otros proyectos o aplicaciones que manejan NFC. Se observa en la Figura 9. Algunos proyectos gratuitos y de fácil acceso para los usuarios que manejan sistema operativo compatible con Google, en nuestro caso Android, donde se desarrollara la aplicación. Compilaremos el estado del arte como un documento para revisar que contiene aplicaciones gratuitas ya desarrolladas y las características que ofrece el desarrollador en el ANEXO A.

Del estado del arte tomaremos las características principales con las cuales será implementada la aplicación, se tiene en cuenta las historias de usuarios para extraer las funciones necesarias, dichas características son:

- Mensajería instantánea para dispositivos móviles, mediante NFC.
- Enlace entre dispositivos móviles para enviar o recibir imágenes, mediante NFC.

- Uso de una tarjeta sin contacto para configurar una actividad en el dispositivo móvil, mediante NFC.

**Figura 9. Plataforma Google Play.**



*Google Play (2017). NFC [Figura]. Recuperado de [13].*

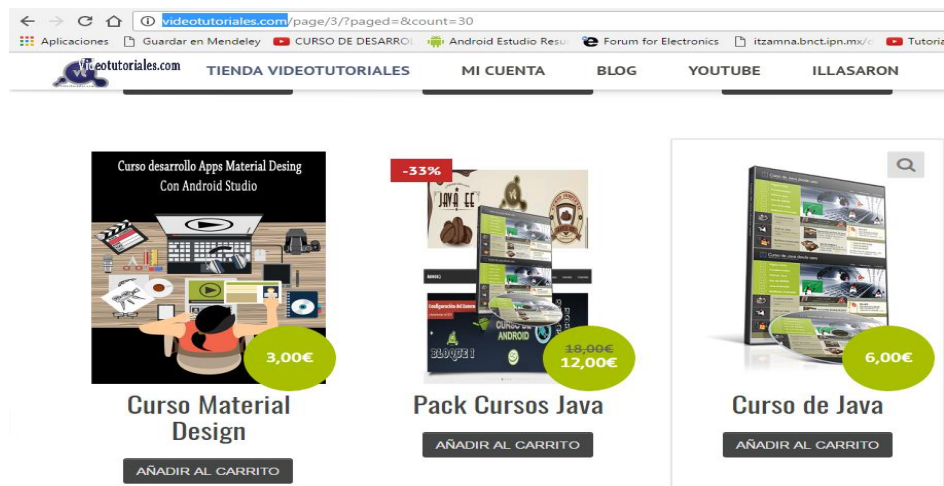
**Programación en java dirigida a objetos** El lenguaje de programación que maneja el entorno de desarrollo Android Studio para la plataforma Android es Java. Para aprender los conceptos del lenguaje de programación java se utiliza la introducción a la programación java, parte 1[14] y parte 2 [15]. Por medio del curso de programación en la web y videos tutoriales de <http://videotutoriales.com> configuraremos los recursos necesarios en el aprendizaje de Java.

En la Figura 10 hacemos referencia a la página y al tutor que guían la capacitación en el desarrollo de programación en Java, para desarrollar las aplicaciones en Java se utilizaron los siguientes recursos:

- Portátil con sistema operativo Windows 7 Home Premium de 64 bits.
- Programa NetBeans IDE 8.0.2 que se descarga directamente de Oracle.

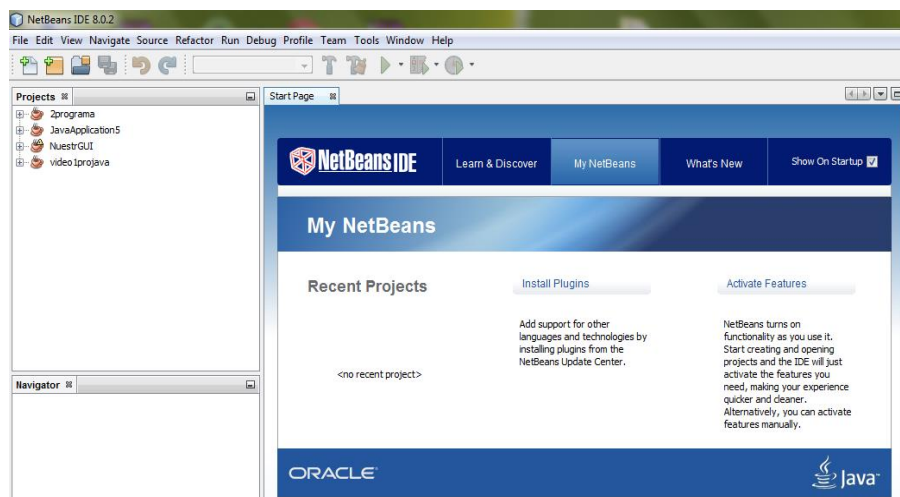
Los videos tutoriales de la Figura 10. Video tutoriales sobre programación en Java dirigida a objetos muestran de forma explícita la instalación y configuración de NetBeans IDE 8.0.2, en la Figura 11. Interfaz del programa NetBeans observamos el programa en ejecución con su interfaz de entrada y características de aplicación. Con la los tutoriales de programación en Java dirigida a objetos se pretende entender el lenguaje de máquina que maneja la plataforma de desarrollo Android Studio y la lógica del código en su estructura.

**Figura 10. Video tutoriales**



Videotutoriales.com (2017). Video tutoriales [Figura]. Recuperado de [16].

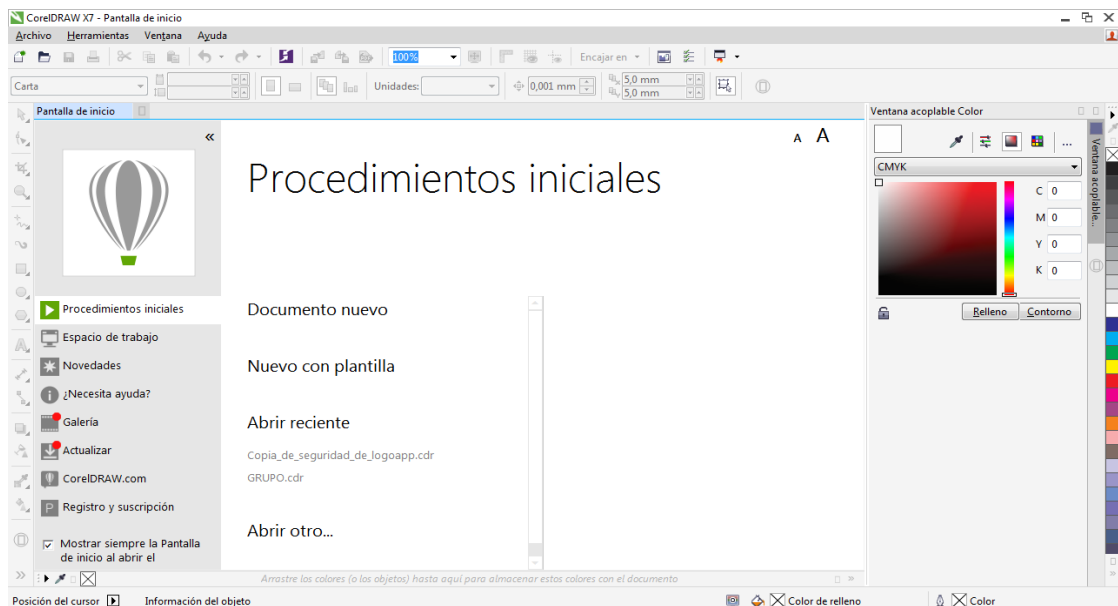
**Figura 11. Interfaz del programa NetBeans**



**Material Design y diseño de logos** Material Design crea un lenguaje visual para los usuarios que utilizan las aplicaciones móviles, sintetiza los principios clásicos de un buen diseño con la innovación. Es un documento que se encuentra en la web y se actualiza a medida que se desarrollan más aplicaciones, son un conjunto de normas para construir aplicaciones sin necesidad de ser reglamentos pues el diseño de una aplicación lo impone cada desarrollador. Es de utilidad para mantener diseños visibles y de gusto a los usuarios y da ejemplos para desarrollar cada interfaz gráfica.

El diseño de los logos es de carácter personal y le da imagen única a la aplicación, para ello se aprovechara el conocimiento en CorelDraw X7 y la aplicación previamente instalada en el computador portátil con sistema operativo Windows 7 Home Premium de 64 bits. La Figura 12 muestra la interfaz del programa donde se desarrollaran los diseños de los logos y cualquier otra imagen que se plantee para la interfaz gráfica de la aplicación.

**Figura 12. Interfaz del programa CorelDraw X7**



**Programación en Android Studio** Android Studio facilita las herramientas más rápidas para la creación de aplicaciones la plataforma Android, para todos los dispositivos móviles y portátiles. Permite la edición de códigos de primer nivel y concentrarse en la creación de aplicaciones únicas y de alta calidad. Los requisitos para la instalación son los siguientes:

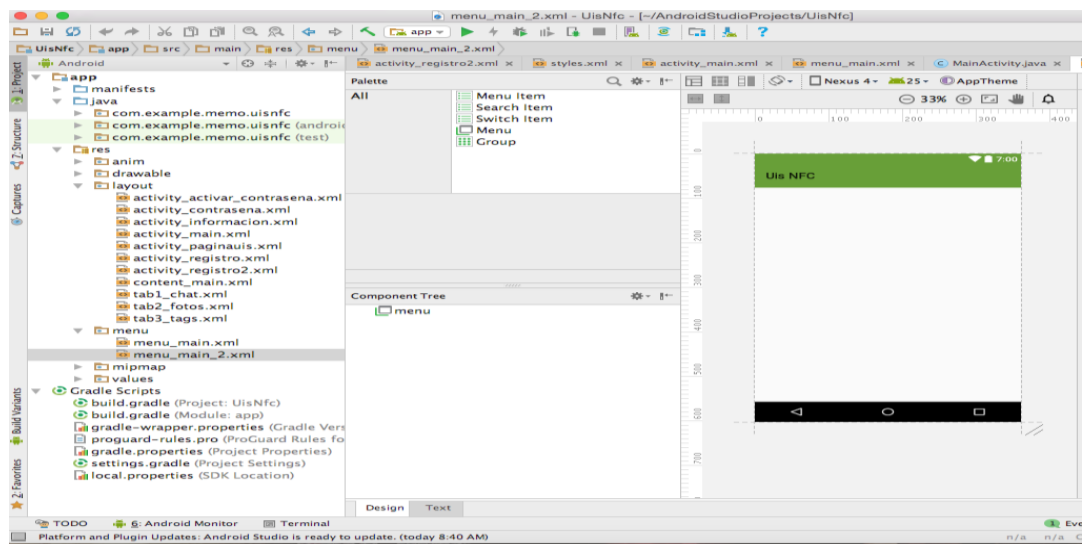
- Portátil con sistema operativo Windows 7 Home Premium de 64 bits.
- Programa Android Studio que se descarga directamente de la plataforma Android en la web.

Para instalar Android Studio en Windows, procede de la siguiente manera:

- Ejecuta el archivo.exe que descargaste.
- Sigue las indicaciones del asistente de configuración para instalar Android Studio y las herramientas de SDK necesarias.

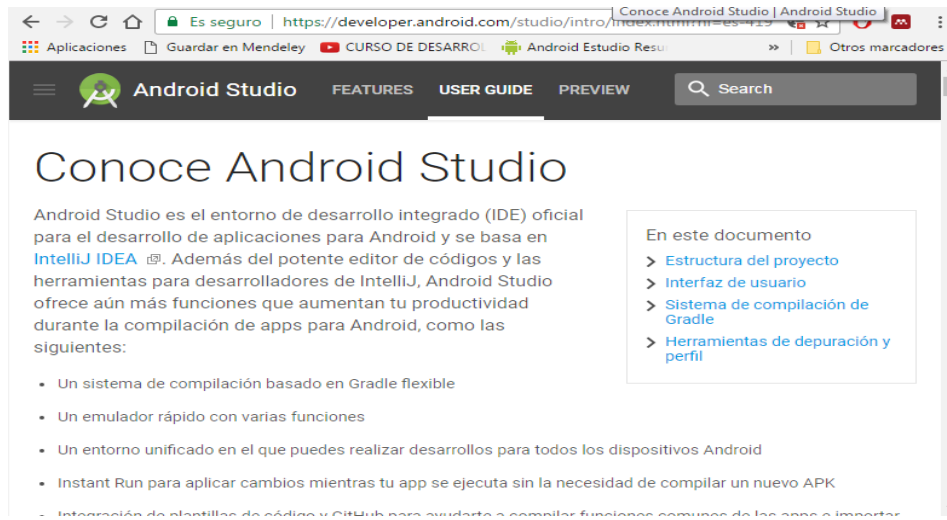
En la Figura 13 se muestra la interfaz del programa Android Studio previamente instalado en el portátil con sistema operativo MAC.

**Figura 13. Interfaz del programa Android Studio**



Android Studio es la plataforma de desarrollo que más se utiliza en la creación de aplicaciones para Android, por eso en la Figura 14 mostraremos la página web de la guía del usuario para el entendimiento y la capacitación en la plataforma de desarrollo Android Studio.

**Figura 14. Guía de usuario web de Android Studio**



*Android Studio (2017). User guide [Figura]. Recuperado de [17].*

Android Studio maneja dentro de su plataforma los conceptos : fundamentos de la NFC, avanzada tecnología NFC y emulación de tarjeta basada en host que permiten desarrollar las funciones de la NFC en dispositivos móviles por medio de la aplicaciones desarrolladas dentro de la plataforma.

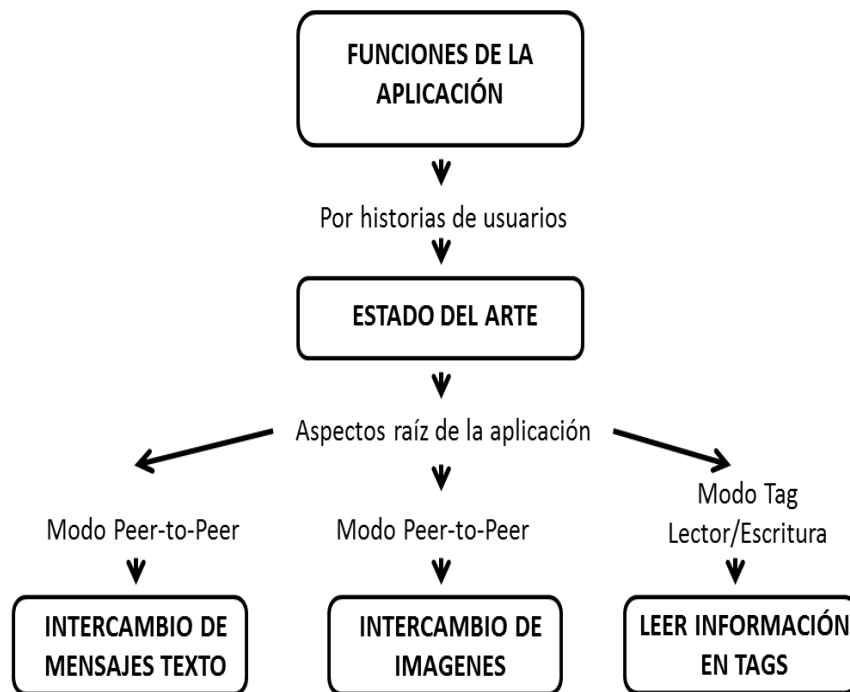
**3.2.2.3 Planificación inicial** El propósito del modelo de planificación inicial es obtener una buena comprensión general del producto a desarrollar, para preparar y perfeccionar los planes para las próximas fases del proyecto y preparar los planes de comprobación y resolución de todas las cuestiones fundamentales del desarrollo final de la fase actual.

**Estado del arte NFC** De la investigación del estado del arte se sacan las características principales de funcionalidad para los usuarios con las que la aplicación es desarrollada, se tiene en cuenta las historias de usuarios para extraer las funciones necesarias, dichas características son:

- Mensajería instantánea para dispositivos móviles, mediante NFC.
- Enlace entre dispositivos móviles para enviar o recibir imágenes, mediante NFC.
- Uso de una tarjeta sin contacto (Tag) para configurar una actividad en el dispositivo móvil, mediante NFC.

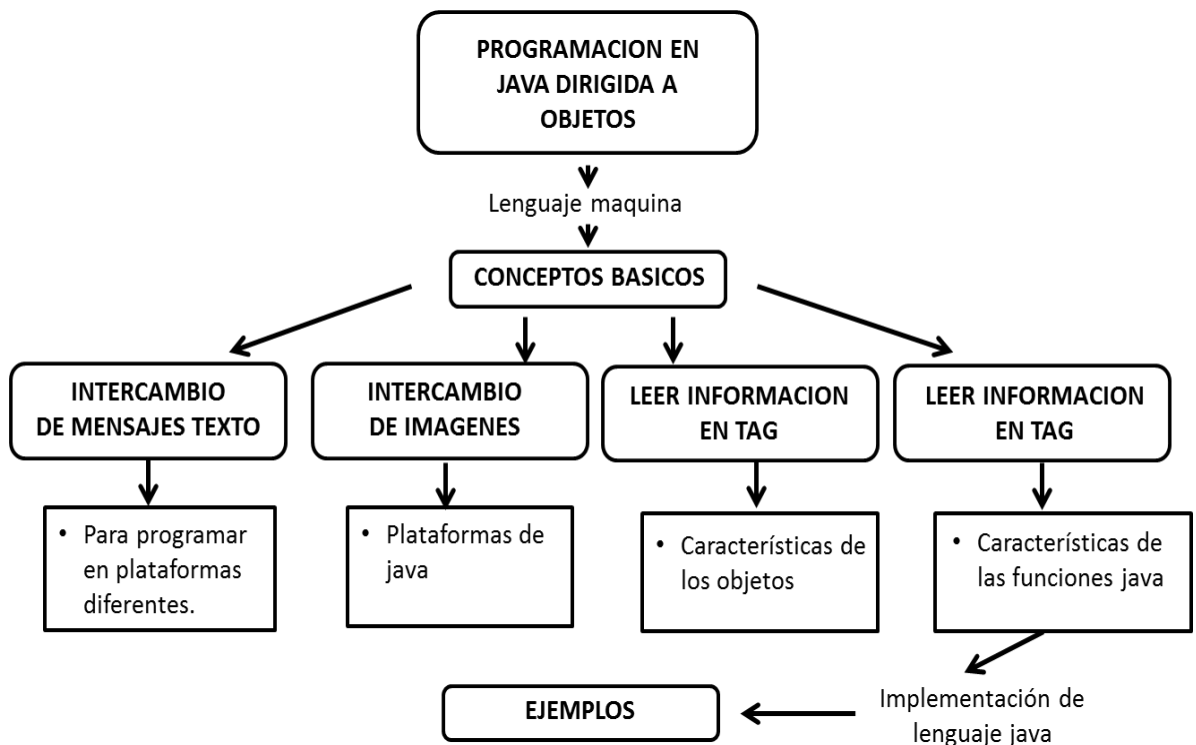
La Figura 15 plantea un mapa conceptual de las funciones raíz que posee la aplicación y da por finalizada la acción del estado del arte como investigación en las etapas siguientes.

**Figura 15. Mapa conceptual del estado del arte NFC**



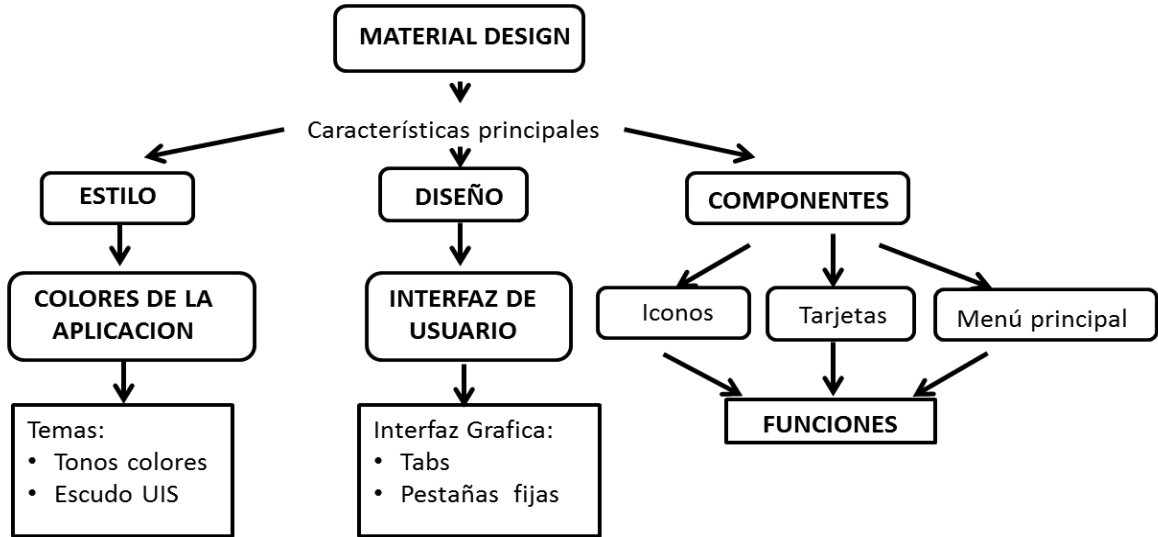
**Programación en java dirigida a objetos** Con la intención de crear una aplicación en Android se estudió el lenguaje de máquina que maneja el programa para la construcción de aplicativos, el lenguaje de máquina que maneja la plataforma es Java y está dirigida a objetos para un mayor entendimiento. El objetivo de la programación en java se muestra en la Figura 16 mediante un mapa conceptual, se aprecian las características fundamentales de la programación en Java y sus objetivos para guiar el desarrollo de aplicativos.

**Figura 16. Mapa conceptual programación en java dirigida a objetos**



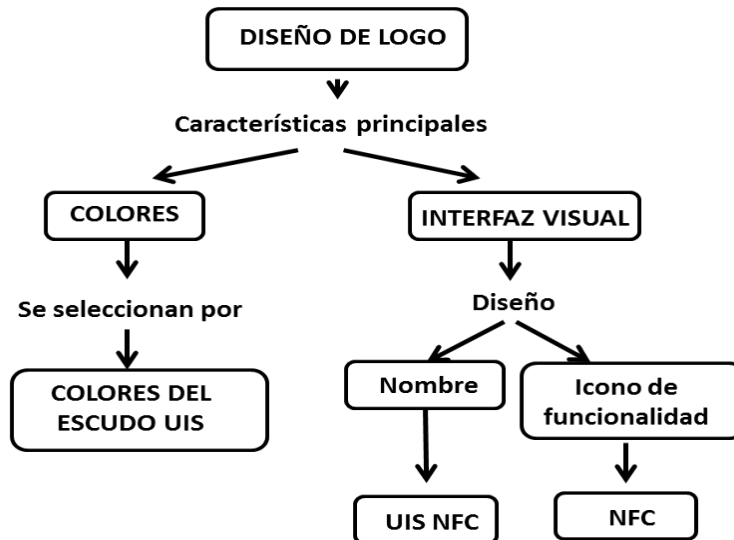
**Material Design y diseño de logos** La prioridad es mejorar el lenguaje visual con el que el usuario tendrá una mejor experiencia al momento de usar la aplicación, en la Figura 17 se observa un mapa conceptual con las características de nuestra aplicación y la definición de su interfaz visual.

**Figura 17. Mapa conceptual Material Design**



El logo es de vital importancia pues identifica en el sistema operativo Android la aplicación, hace que la imagen sea una característica única para diferenciar y reconocer la aplicación además del nombre, en la Figura 18 se observan las características con las q se construye el logo mediante un mapa conceptual.

**Figura 18. Mapa conceptual diseño de logos**



**Programación en Android Studio** En la Figura 19 se observa un cuadro de procesos que guían a la creación de la aplicación NFC, Se agregan características de desarrollo para empalmar cada fase del proceso similar al de ciclos de programación de la Figura 6.

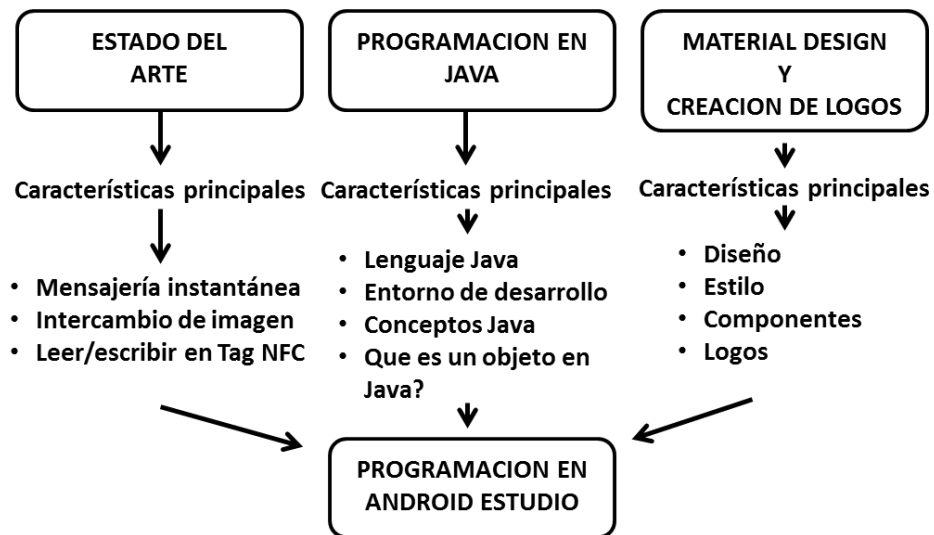
**Figura 19. Ciclos de programación para desarrollar aplicaciones en Android Studio**



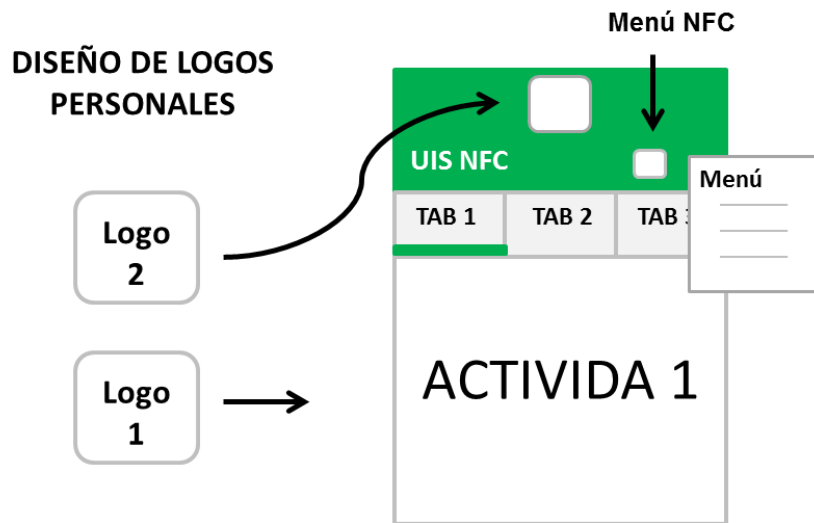
La programación en Android Studio se desarrolla a partir de la unión de los procesos anteriores, dado que los procesos son bases para poder generar aplicativos, están atados en un mismo programa y se implementan a la hora de compilar una aplicación. En la Figura 20 observamos los requisitos necesarios para empezar a programar en Android Studio y crear aplicativos que permitan una solución a usuarios interesados.

Se realiza un primer diseño (esqueleto) con el cual se enfoca la programación en Android Studio para el desarrollo de la aplicación.

**Figura 20. Interacción de procesos para desarrollar aplicaciones en Android Studio**



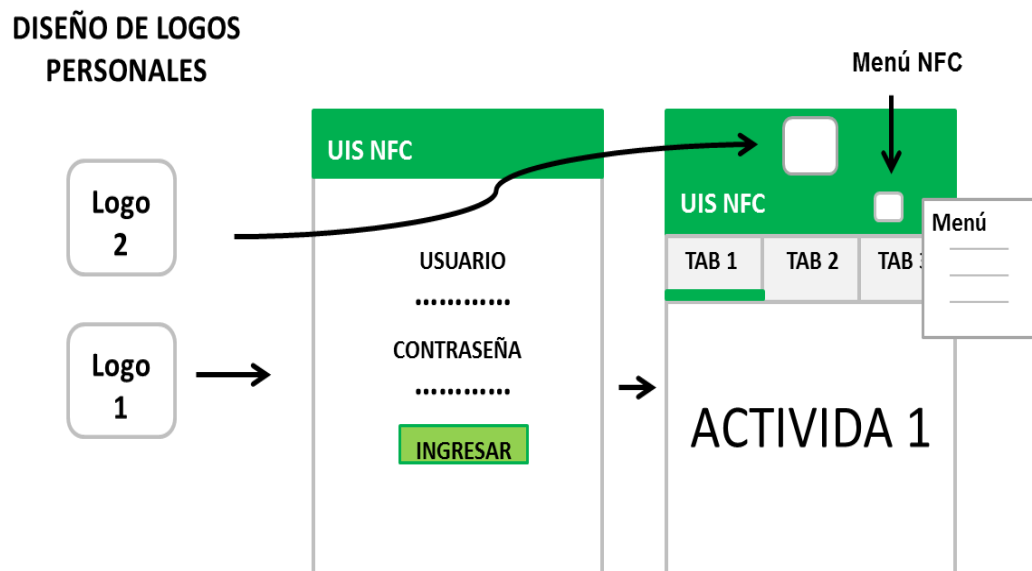
**Figura 21. Pre-diseño aplicación NFC**



En la Figura 21. Pre-diseño aplicación NFC se muestra como debe ser la interfaz visual de la aplicación en la plataforma, y las características con las que la programación Android Studio creara la aplicación.

**3.2.2.4 Día de trabajo (prueba)** La función de esta etapa es de prueba y sirve de supervisión para iniciar la implementación del software. Además, el propósito es implementar algunas funciones al núcleo del sistema (por ejemplo, la comunicación código de trabajo. Puede agregar nuevas investigaciones tecnológicas en esta etapa. En la Figura 22 se agrega una función de usuario y contraseña para entrar a la aplicación lo que garantiza seguridad en la información que maneja la aplicación.

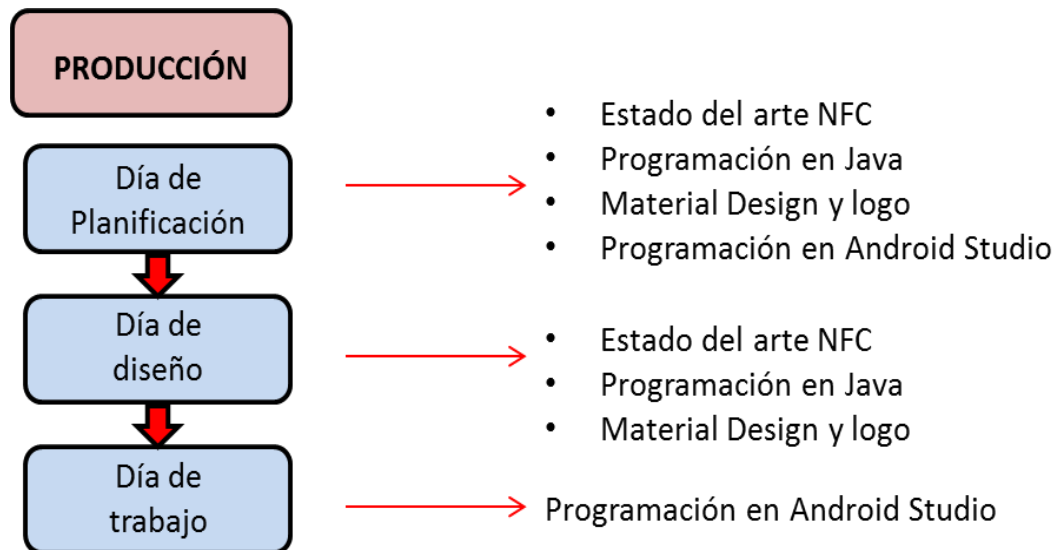
**Figura 22. Pre-diseño aplicación NFC con actividad de contraseña**



**3.2.3 PRODUCCIÓN** En esta fase se diseña, desarrollan e implementan todas las funciones que el software necesita para crear una aplicación. Para cada función se realiza un único proceso que se emplea en la construcción de UIS NFC. En la Figura 23 observamos las etapas de la fase producción y el desarrollo de cada actividad generada por las fases anteriores.

**3.2.3.1 Planificación** El propósito es seleccionar y planificar el contenido del trabajo, dejando en claro etapas que pueden culminar y verifica que las características de la aplicación son correctamente abarcadas.

**Figura 23. Características de la fase Producción**



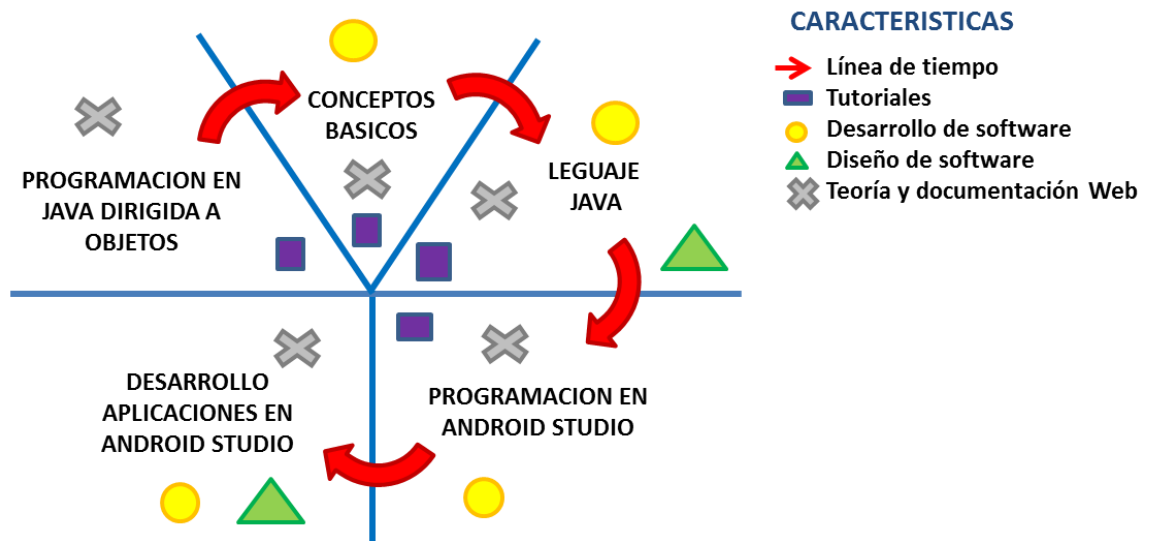
**Estado del arte NFC** Es la última vez que se emplea esta etapa pues su funcionalidad es más de investigación, pero mantiene y simplifica el plan de la fase inicialización, en la Figura 24 se observa el esquema de las características principales que entrega el estado del arte para la aplicación.

**Programación en java dirigida a objetos** Utilizando los Ciclos de programación para desarrollo de aplicaciones en Android Studio de la Figura 19 se crea un plan para aprender la programación en Java dirigida a objetos, colocando las características y conceptos básicos que se deben abarcar el aprendizaje de la programación en Java en un ciclo nuevo como se ve en la Figura 25.

**Figura 24. Características principales de la aplicación tomadas por el estado del arte**



**Figura 25. Ciclos de programación java dirigida a objetos**

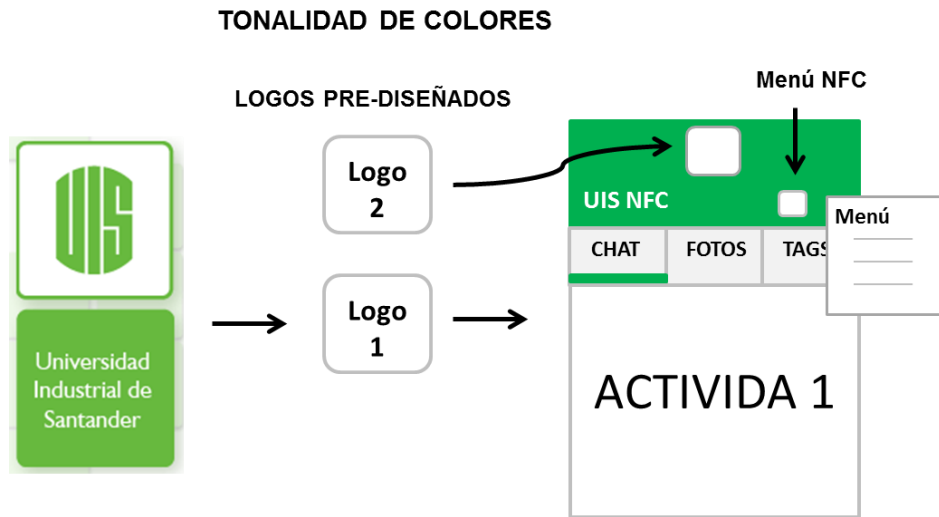


*Elaboración propia.*

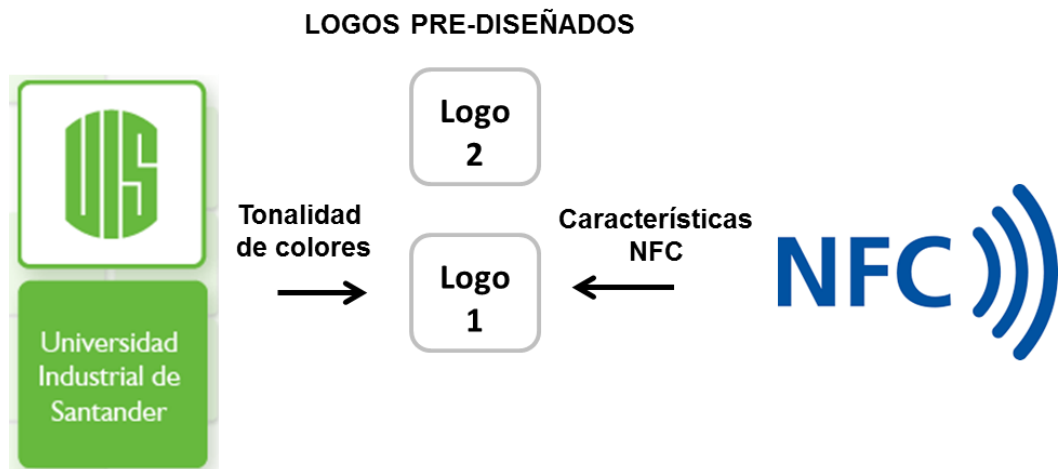
### Material Design y diseño de logos

En la Figura 26 se muestra el pre diseño de la aplicación y la tonalidad de colores que se planean para la aplicación, además de los nombres y las características generales que debe tener la aplicación.

**Figura 26. Pre-diseño aplicación NFC con asignación de colores y nombres**



**Figura 27. Pre-diseño logo UIS NFC**

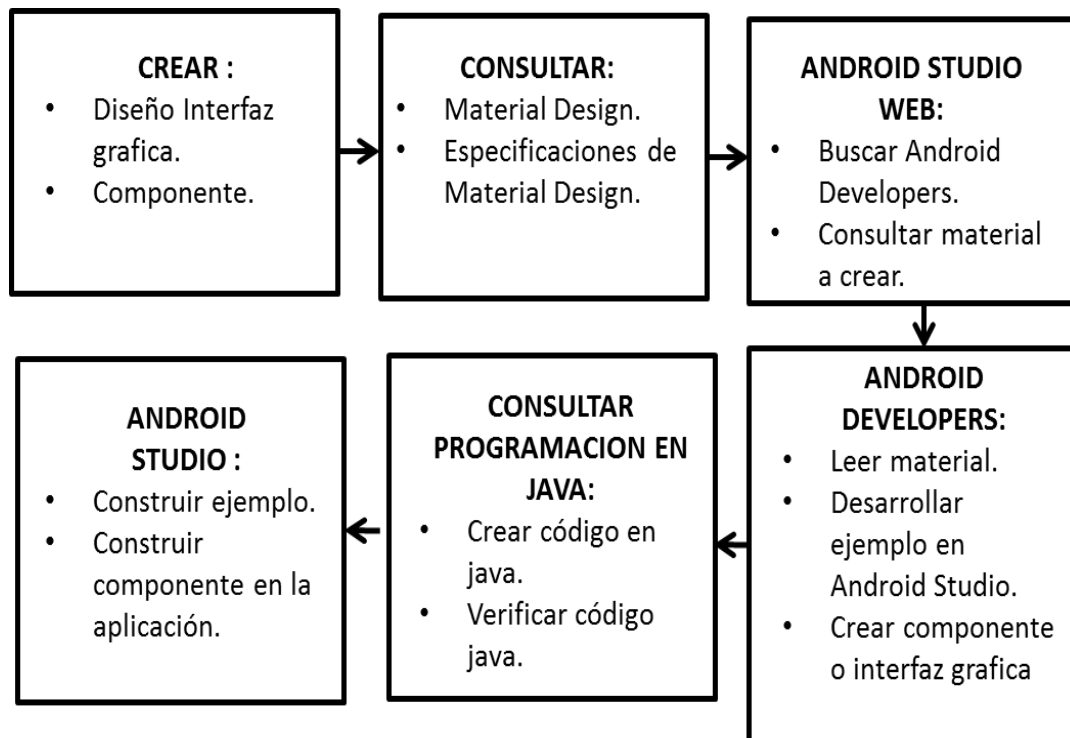


El logo que se planifica es un logo sencillo alusivo al escudo de la UIS pero que explica la funcionalidad y característica de la aplicación, en la Figura 27. Pre-diseño logo UIS NFC se observa el nombre que se le da a la aplicación (UIS NFC) y su pre diseño de logo.

**Programación en Android Studio** Se mantiene el concepto de la Figura 20. Interacción de procesos para desarrollar aplicaciones en Android Studio de la fase inicialización, donde los conceptos de procesos anteriores generan un entendimiento de la plataforma Android Studio.

Se construye un diagrama de flujo que contiene el método para desarrollar actividades en la plataforma Android Studio, cada vez que se necesita consultar un desarrollo se planifica su creación por el método de la Figura 28.

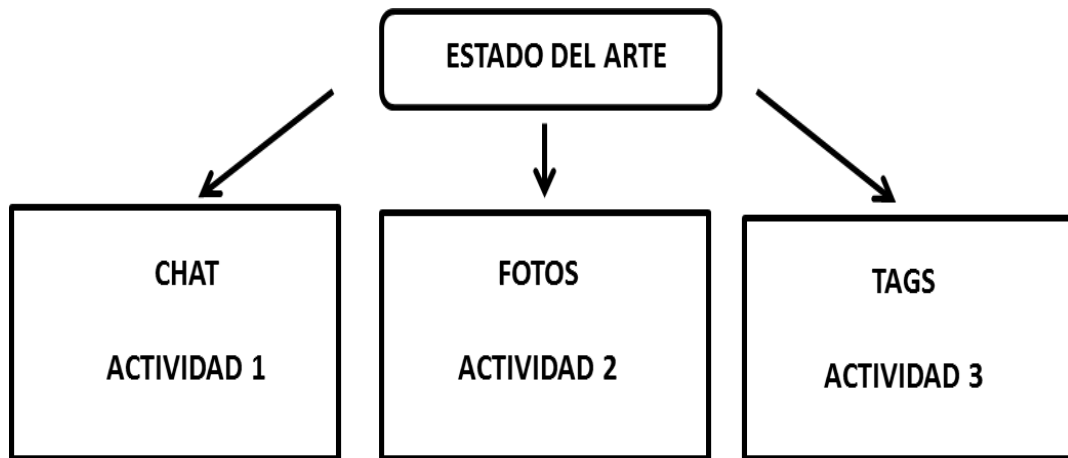
**Figura 28. Método de consulta en Android Developers**



**3.2.3.2 Día de Diseño** El propósito de esta etapa es la de diseñar todo lo planificado o previsto en el día de planificación, se centraliza en la funcionalidad y el diseño de cada etapa planificada.

**Estado del arte NFC** Las características que aporta el estado del arte son las que se observan en la Figura 29, consisten en mensajería instantánea, intercambio de imagen, leer la información de un Tag. Con estas características principales se basa la creación de la aplicación y busca satisfacer los problemas generados por las historias de usuarios.

**Figura 29. Actividades de la aplicación por medio de NFC**



**Programación en java dirigida a objetos** La programación dirigida a objetos muestra cómo se comporta el lenguaje java y como crea aplicaciones del mundo real usando la estructura del lenguaje. En el curso de programación java parte 1[14], parte 2[15] se observa un resumen con los aspectos más incidentes de la programación en java dirigida a objetos para comprender y relacionar lo aprendido en el curso virtual.

**Material Design y diseño de logos** Con material Design implementaremos el bosquejo de la Figura 26 en la aplicación según el método de la Figura 28. Construiremos la interfaz visual final de la aplicación con los principios de Material Design y se elaborara la interfaz gráfica que contiene la aplicación.

**Esquemas de color** Se desarrolla un estilo mediante los colores principales de la aplicación en la Figura 30 se muestran los colores verdes tomados de Material Design con los que se va a trabajar y su código de referencia para Android Studio.

**Figura 30. Paleta de colores Material Design**

Green		Light Green		Lime	
500	#4CAF50	500	#8BC34A	500	#CDDC39
50	#E8F5E9	50	#F1F8E9	50	#F9FBE7
100	#C8E6C9	100	#DCEDC8	100	#F0F4C3
200	#A5D6A7	200	#C5E1A5	200	#E6EE9C
300	#81C784	300	#AED581	300	#DCE775
400	#66BB6A	400	#9CCC65	400	#D4E157
500	#4CAF50	500	#8BC34A	500	#CDDC39
600	#43A047	600	#7CB342	600	#C0CA33
700	#388E3C	700	#689F38	700	#AFB42B
800	#2E7D32	800	#558B2F	800	#9E9D24
900	#1B5E20	900	#33691E	900	#827717

*Material Design (2017). Paleta de colores [Figura]. Recuperado de [18].*

**Color primario** Se elige el verde de referencia #689F38, este color debe ser el más ampliamente utilizado en todas las pantallas y componentes.

**Color secundario** Se elige el verde de referencia #689F38, el color secundario se pueden utilizar para indicar una acción o información relacionada.

**Colores de acento alternativo** El color de acento alternativa es la versión 500 de su color primario en los fondos blancos, el acento se debe utilizar para el botón de acción flotante y elementos interactivos, tales como:

- Los campos de texto y cursores

- La selección de texto
- Los controles de selección, botones y controles deslizantes

**Colores de fondo y de texto** Para texto oscuro sobre fondo claro, se aplican los siguientes niveles de opacidad:

- El texto más importante tiene una opacidad del 87% se escoge el verde de referencia #689F38.
- texto secundario, que es más bajo en la jerarquía visual, tiene una opacidad de 54% se escoge negro de referencia #000000.
- consejos de texto (como los de los campos de texto y etiquetas) y el texto discapacitados tienen aún menor relevancia visual con una opacidad del 38% se escoge entre negro de referencia #000000, blanco de referencia #FFFFFF y verde de referencia #99CC00.

**Tipografía** Se utiliza Roboto porque es el tipo de letra estándar en Android, se toman las especificaciones de Material Design y se escoge el script de tipo:

- English e English-like (latino, griego y cirílico)

Las características para la aplicación en su tipografía para diferentes componentes se dan por las siguientes medidas:

- barra de la aplicación: estilo de título, 20SP Medio
- Botones: English, 14sp mediano, todas las tapas
- Subtítulo: English, 16sp Regular(dispositivo), 15sp (escritorio)
- Cuerpo 1: Ingles, 14sp Regular(dispositivo), 13sp (escritorio)
- Relaciones de contraste de texto: Mínimo 4.5:1, preferidos 7:1

**Longitud de la línea** Se utiliza el consejo sobre la legibilidad y la longitud de línea del Instituto Baymard:

"Debe tener alrededor de 60 caracteres por línea, si quiere una buena experiencia de lectura. Tener la cantidad correcta de caracteres en cada línea es clave para la legibilidad del texto" [19].

**Pestañas** Se escoge la vista en pestañas pues hace que sea fácil explorar y cambiar entre diferentes puntos de vista. Permite organizar contenidos en un alto nivel, como conjuntos de datos o aspectos funcionales de una aplicación. Su estructura se conforma de la siguiente manera:

- Tipos: fijo y desplazable, se escoge fijo por comodidad.
- Etiquetas de las pestañas: las etiquetas de las pestañas pueden ser tanto todos los iconos o todo el texto.

**Color** Se aplica el color de su aplicación a las actividades de acento, en este caso a la barra indicadora, para los campos de texto un color blanco referencia #FFFFFF y el negro referencia #000000 para el texto de cada pestaña.

**Pestañas móviles** Se escoge el modelo de pestañas móviles de la Figura 31. Elemento Tabs con pestañas fijas que contiene pestañas móviles para su interfaz de usuario con las especificaciones estipuladas por Material Design.

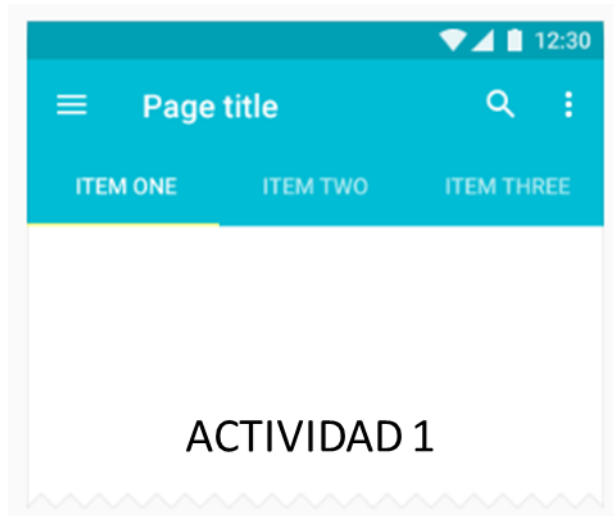
**Características de la ficha** Se utilizan fichas de control para la visualización del contenido en una ubicación fija.

**Contenido** Del estado del arte colocamos los títulos de las pestañas que son:

- Chat

- Fotos
- Tags

**Figura 31. Elemento Tabs con pestañas fijas**



*Material Design (2017). Pestañas [Figura]. Recuperado de [20].*

**Pestañas fijas** Las pestañas fijas muestran todas las pestañas al mismo, las pestañas fijas tienen igual anchura. Para navegar entre las pestañas fijas se toca el área de la ficha y sus especificaciones de tamaños se toman de Material Design. Para Las fichas con sólo texto:

- Texto: 14sp Roboto Medio, 12SP cuando envuelto a través de un máximo de dos líneas, todas las tapas, horizontalmente y verticalmente centrada, color activo #000000 o color de acento, fuera de foco color de pestaña #FFFFFF 70%.

**Tarjetas** Se utilizara el componente tarjeta pues una tarjeta es una lámina de material que sirve como punto de acceso a la información más detallada [21]. Se utilizan las especificaciones de Material Design para el elemento tarjetas.

**Uso y contenido** La visualización de contenido se compone de diferentes elementos y se realiza por medio de tarjetas. Tarjetas proporcionan contexto y un punto de acceso a la información más sólida y puntos de vista [21].

**Jerarquía de contenido** Se utiliza jerarquía dentro de la tarjeta para dirigir la atención de los usuarios a la información más importante [21].

**Comportamiento y Acciones suplementarias** Tarjetas tienen una anchura constante y altura variable. La altura máxima se limita a la altura del espacio disponible en una plataforma [21]. Acciones suplementarias dentro de la tarjeta se llaman explícitamente en el uso de iconos, texto y controles de interfaz de usuario, se colocan en la parte inferior de la tarjeta [21].

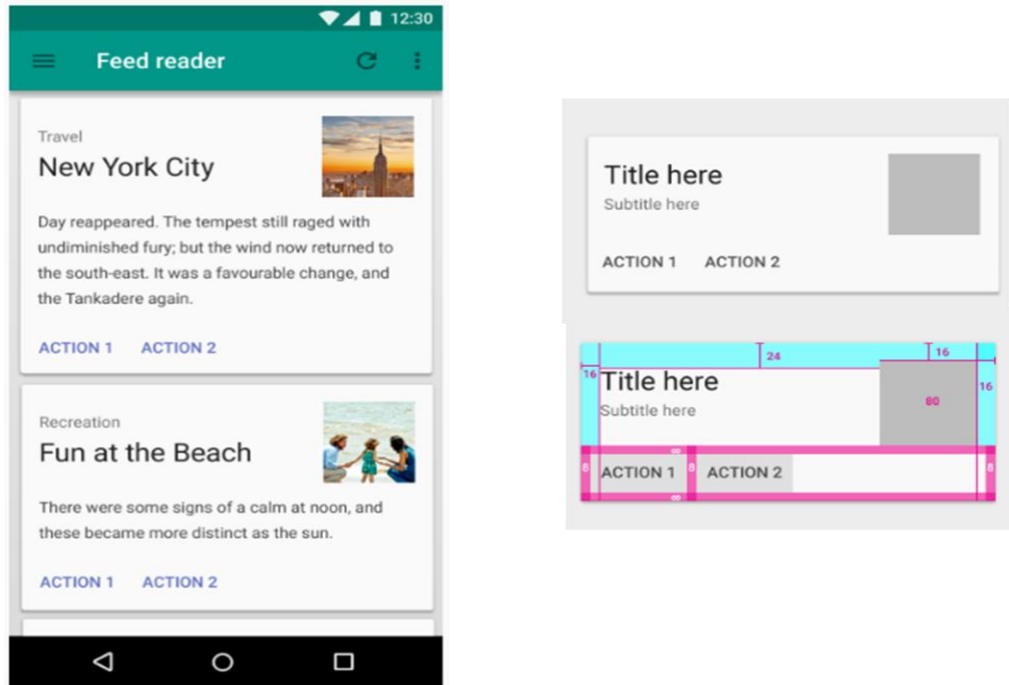
**Bloques de contenido** Las tarjetas se construyen con las siguientes características:

- Un título primario
- texto de apoyo
- Comportamiento

**Tipos de bloques de contenido** En la Figura 32 se observa los parámetros que cumple los bloques de contenido para las tarjetas de la aplicación.

**Logo del producto** Se utilizan estas directrices de Material Design como punto de partida para asegurar que los colores en el icono del producto y los elementos clave reflejan su identidad de marca [22]. Los colores son verdes referencia #689F38 y #99CC00 alusivos al escudo de la UIS (Universidad Industrial de Santander), elemento de señal alusivo a la funcionalidad de conexión inalámbrica y por último texto alusivo al producto.

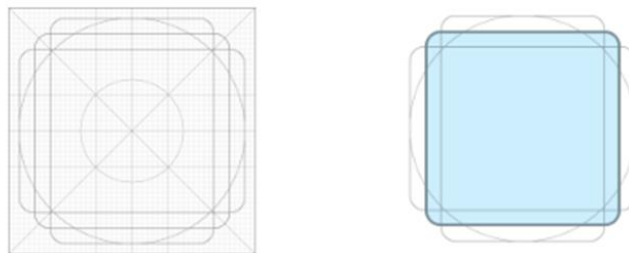
**Figura 32. Elemento tarjetas con parámetros de diseño**



*Material Design (2017). Tarjetas [Figura]. Recuperado de [21].*

**Icono del producto rejilla** Se utiliza la rejilla de icono de Material Design para desarrollar el icono del producto. Esta estandarización se traduce en un sistema flexible pero coherente y se observa en la Figura 33.

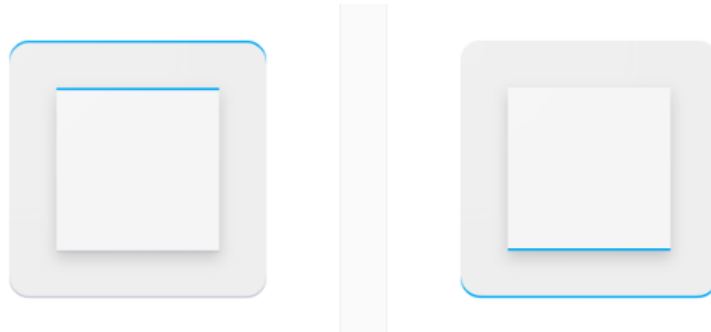
**Figura 33. Rejillas para la creación de logo**



*Material Design (2017). Rejillas [Figura]. Recuperado de [22].*

**Formas Keyline** Se observan en la Figura 34 las Formas Keylinen aplicadas para la creación de los logos.

**Figura 34. Formas Keyline con sombras de línea para logos**



*Material Design (2017). Forma Keyline [Figura]. Recuperado de [22].*

**Enfoque de diseño** El diseño del icono para el producto se inspira en la funcionalidad de la aplicación, en los colores del escudo de la UIS y la comunicación inalámbrica. Se observa en la Figura 35 el logo diseñado con los cuadros de escala, vemos el diseño del icono del producto (logo), finalizado y creado con las funcionalidades propuestas.

**Figura 35. Logo 1 diseñado con cuadros de escala y sombras de línea**



En la Figura 36 se observa un icono secundario creado para la interfaz principal de la aplicación y con los mismos estándares de Material Design.

**Figura 36. Logo 2 diseñado con cuadros de escala y sombras de línea**



**Densidad de píxeles** Se utiliza la densidad de píxeles para trabajar en toda la creación de la aplicación. Densidad de la pantalla = ancho de la pantalla (o altura) en píxeles/ancho de la pantalla (o altura) en pulgadas. Un dp es igual a un píxel físico en una pantalla con una densidad de 160. Para calcular dp [23]:

- $dp = (\text{anchura en píxeles} * 160) / \text{densidad de pantalla}$
- Al escribir CSS, utilice px donde se afirma DP o sp. Dp sólo necesita ser utilizado en el desarrollo de Android.

**Píxeles escalables (sp)** Al desarrollar para Android, los píxeles escalables (sp) cumplen la misma función que dp, pero para las fuentes. El valor por defecto de un sp es el mismo que el valor predeterminado para dp [23].

**Tabla 1. Valores de pantalla en diferentes unidades**

densidad de la pantalla	Anchura de la pantalla en píxeles	Anchura de la pantalla en píxeles independientes de la densidad
120	180 px	240 dp
160	240 px	
240	360 px	

*Material Design (2017). Valores de pantalla en diferentes unidades [Tabla]. Recuperado de [23].*

Las imágenes pueden ser escaladas y tener el mismo aspecto a través de diferentes resoluciones de pantalla mediante el uso de estas relaciones [23]:

**Tabla 2. Escalado de las imágenes**

Resolución de la pantalla	ppp	proporción de píxeles	Tamaño de imagen (píxeles)
xxxhdpi	640	4.0	400 x 400
xxhdpi	480	3.0	300 x 300
xhdpi	320	2.0	200 x 200
IPAP	240	1.5	150 x 150
mdpi	160	1.0	100 x 100

*Material Design (2017). Escalado de las imágenes [Tabla]. Recuperado de [23].*

**3.2.3.3 Día de trabajo** El propósito de esta etapa es la implementación de todo lo planificado o previsto en el día de planificación y el día de diseño, se centraliza en la funcionalidad y el desarrollo de cada etapa planificada.

**Programación en Android Studio** La plataforma Android Studio se emplea para crear aplicaciones en Android, sus recursos se programan a través de lenguaje Java y toma los conceptos de Material Design en la construcción de todas las Activiys que maneja un aplicativo. Se construye la aplicación a través de los requisitos de las fases anteriores y mediante la fase de planificación se instalan los programas necesarios para desarrollar en Android Studio, se realiza la construcción de cada Activity y partes de la aplicación para cumplir con el diseño de la aplicación.

**Creación de la aplicación en Android Studio** En la fase de planificación se verifico que el software de desarrollo Android Studio estuviera correctamente instalado, se abre la aplicación y se procede con los siguientes pasos para la creación de un nuevo proyecto o aplicación:

- Se escoge la opción Start a new Project
- Le damos nombre a la aplicación UisNfc y en dominio memo.example.com
- Se escoge el SDK mínimo API 15: Android 4.0.3 (IceCream Sandwich), esta es la plataforma de Android que soporta la aplicación a crear.
- Se escoge la opción Empty Activity
- Por defecto se deja el nombre a la primera Activity como MainActivity

***Inserción del logo*** La aplicación que se genera es una aplicación predeterminada que el programa Android Studio da, su logo es predeterminado con el diseño de Android. Para hacer que la aplicación tenga una identidad única se diseña y crea el logo estándar con las características que sugiere Material Design y se implementa en la interfaz gráfica de la siguiente manera:

- Se busca la carpeta app/res/mipmap que contiene el archivo de imagen predeterminada creada por Android Studio.
- En la carpeta mipmap se guarda el logo diseñado para la aplicación en los tamaños existentes y recomendados por la plataforma (hdp, mdp, xhd, xxhdp).
- Se elimina la imagen predeterminada después de insertar la diseñada.

***Nombre de la aplicación*** La aplicación se identifica en plataforma de la Google Play y en cualquier otro ámbito por su nombre que previamente es escogido como UIS NFC, al crear de cero la aplicación Android Studio le coloca el nombre con el que creo todo el paquete.

***Colores de la interfaz gráfica y estilos*** Los colores predeterminados al crear la aplicación en blanco se cambian en la carpeta values de la siguiente manera:

- Nos dirigimos a la carpeta app/res/values en el menú de Android

- En la carpeta values se escoge el archivo colors.xml
- Se escogen los 3 tipos de colores para la aplicación: colorPrimary #99CC00 verde, colorPrimaryDark #669900 verde y colorAccen #669900 verde

Los colores tiene un encadenamiento con los estilos desarrollados para las aplicaciones en Android, al crear una aplicación desde cero el estilo predeterminado es “AppTheme” que esta encadenado al estilo predeterminado “Theme.AppCompat.Light.DarkActionBar”, esta carpeta se modifica de la siguiente forma:

- El archivo xml para modificar los estilos está ubicado en la carpeta app/values/styles.xml
- En el archivo styles.xml encontraremos encadenados los colores colorPrimary, colorPrimaryDark y colorAccen.

**Creación del menú inicial de la aplicación** Android Studio maneja una interfaz gráfica .xml y archivos java en este caso un paquete com.example.memo.uisnfc con el cual se entrelaza las actividades para generar la aplicación, en la interfaz gráfica el archivo inicial que se crea predeterminado es ativity\_main.xml y se encuentra en la dirección app/res/layout, en el archivo .xml encontraremos:

- CoordinatorLayout que es el contenedor para la organización de la interfaz gráfica (todo archivo .xml de interfaz gráfica debe tener un contenedor principal como regla).
- AppBarLayout que es el contenedor para albergar nuestro menú inicial.
- Include sirve para llamar otra Activity en el espacio contenido.

**Creación de la interfaz gráfica del menú inicial** Para crear nuestra interfaz de menú vamos al archivo `activity_main.xml` y dentro del `AppBarLayout` agregaremos un `ImageView` con el segundo logo creado para la aplicación, que nos dará una característica única de presentación al menú principal de la aplicación. Se deja el toolbar predeterminado que alberga el nombre del menú y el menú desplegable. Se deja el `Include` en blanco para agregar más adelante la Activity correspondiente.

**Creación del menú desplegable** El menú desplegable viene predeterminado por Android Studio al crear un nuevo proyecto se genera este tipo de menú, en este menú insertaremos las características que sean en general para la aplicación y se muestra una línea de opciones con una opción por línea.

El menú desplegable se encuentra en la carpeta `app/res/menú` y esta contenido en un archivo `menu_main.xml`, dentro del contenedor `menu` se crearan las siguientes opciones generales de nuestra aplicación:

- Contraseña: para la seguridad de nuestra aplicación se necesita tener opciones de contraseña y generalidades de seguridad.
- Información: para la identificación de nuestra aplicación se desea dar información acerca de la aplicación y su propósito.
- Página UIS: para la conectividad y enlace nos direcciona a la página web de la UIS (Universidad Industrial de Santander).

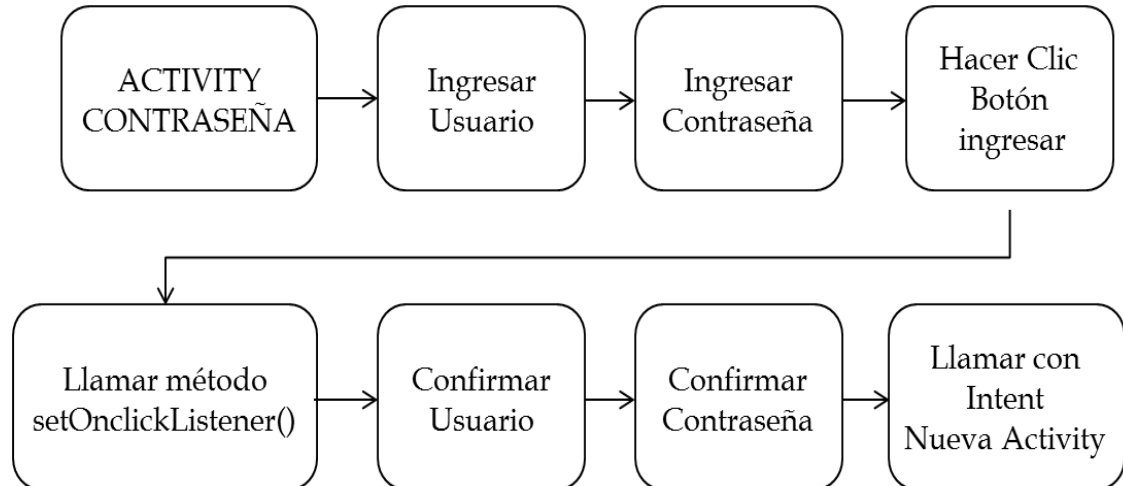
Al crear las líneas anteriores del menú desplegable necesitaremos crear cada Activity de manera independiente, cada Activity consta de un código en java para ejecutar la programación de objetos y un código en xml para su interfaz gráfica, crearemos las tres opciones del menú desplegable de la siguiente forma:

- Nos ubicamos en la carpeta `app/java/com.example.memo.uisnfc`.

- Creamos una nueva Activity con Basic Activity le damos el nombre al archivo ActivarContraseñaActivity y al layout activity\_activar\_contraseña para la opción contraseña.
- Creamos una nueva Activity con Basic Activity le damos el nombre al archivo InformacionActivity y al layout activity\_informacion para la opción información.
- Creamos una nueva Activity con Basic Activity le damos el nombre al archivo PaginauisActivity y al layout activity\_paginauis para la opción página UIS.

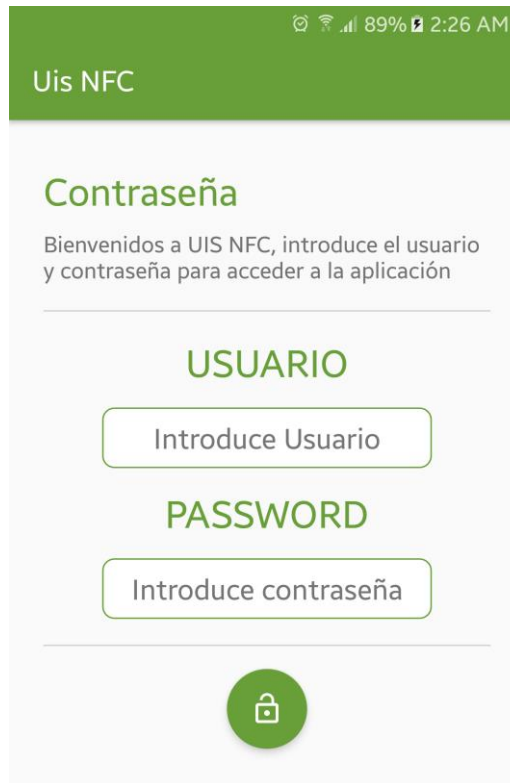
Activity Contraseña: Para mostrar la función de la actividad primero se crea el diagrama de flujo de la Activity en la Figura 37.

**Figura 37. Diagrama de flujo Activity Contraseña**



Se desarrolla la interfaz gráfica a gusto propio mediante la plataforma Android Studio siguiendo las guías de desarrollo estipuladas por Material Design y se observa la nueva interfaz gráfica de la Activity Contraseña en la Figura 38.

**Figura 38. Interfaz gráfica Activity Contraseña**

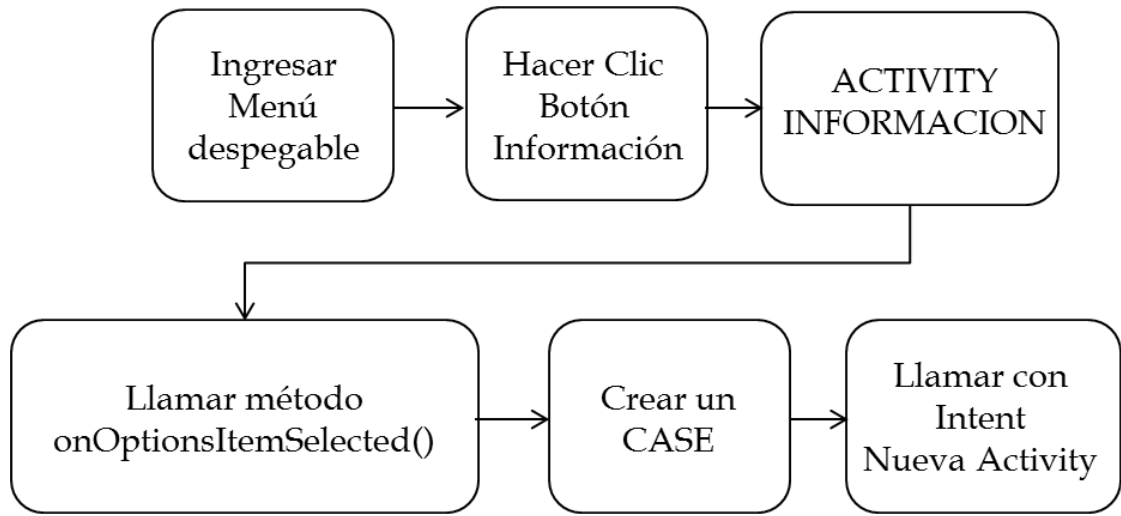


Se enlazan las actividades del archivo `activity_activar_contraseña.xml` con las de la programación de objetos en java y se desarrolla el diagrama de flujo de actividades con el código del programa anexo (Interfaz gráfica Activity Contraseña).

Activity Información: Para mostrar la función de la actividad primero se crea el diagrama de flujo de la Activity en la Figura 39.

Se desarrolla la interfaz gráfica a gusto propio y se observa en la Figura 40 y se enlazan las actividades del archivo `activity_informacion.xml` con las de la programación de objetos en java y se desarrolla el diagrama de flujo de actividades con el código del programa anexo (Interfaz gráfica Activity Información).

**Figura 39. Diagrama de flujo Activity Información**

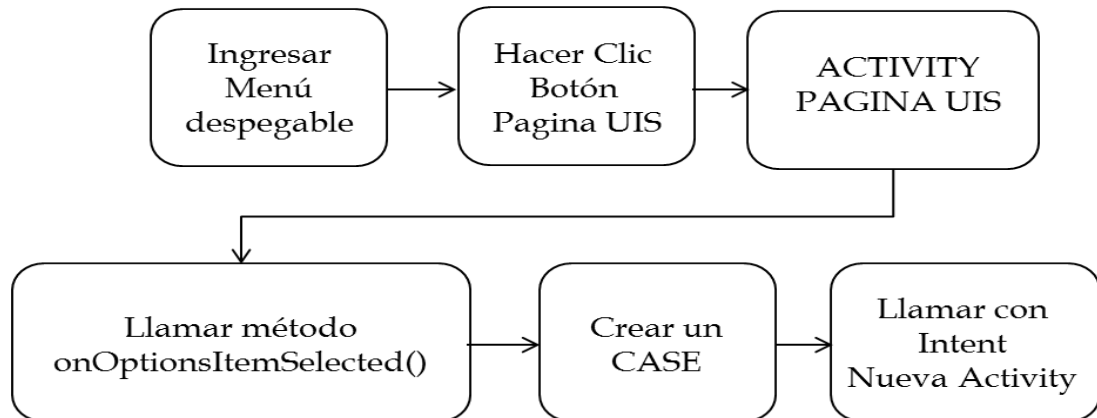


**Figura 40. Interfaz gráfica Activity Información**



Activity Página UIS: Para mostrar la función de la actividad primero se crea el diagrama de flujo de la Activity en la Figura 41.

**Figura 41. Diagrama de flujo Activity Pagina UIS**



Se desarrolla la interfaz gráfica a gusto propio y se observa en la Figura 42.

**Figura 42. Interfaz gráfica Activity Pagina UIS**

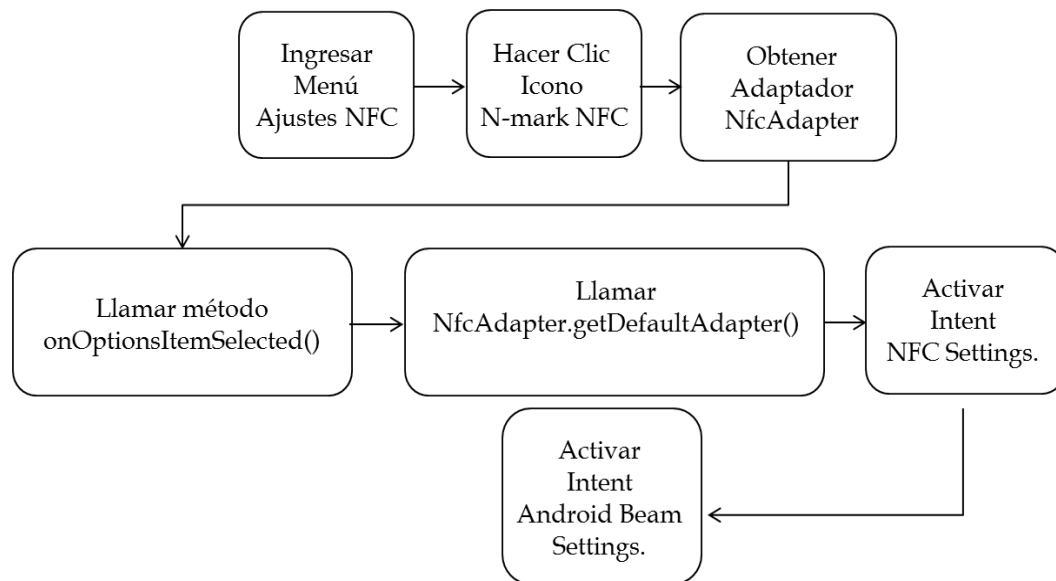


Se enlazan las actividades del archivo `activity_paginauis.xml` con las de la programación de objetos en java y se desarrolla el diagrama de flujo de actividades con el código del programa anexo (Interfaz gráfica Activity Pagina UIS).

Para enlazar cada actividad del menú desplegable al menú principal en este caso el MainActivity código en java que se encuentra en la carpeta app/java/com.example.memo.uisnfc, con el código del programa anexo (Interfaz gráfica Menú Desplegable).

**Creación del menú NFC** Se crea un menú para ajustar la configuración NFC del dispositivo móvil desde la aplicación directamente, para ello en el archivo menu\_main.xml se agrega un nuevo ítem con el logo de la n-mark. La n-mark es el logo oficial que el NFC-forum creo para estandarizar sus productos y se descarga directamente de la página web. Se crea un diagrama de flujo en la Figura 43 para desarrollar las acciones del menú ajustes NFC.

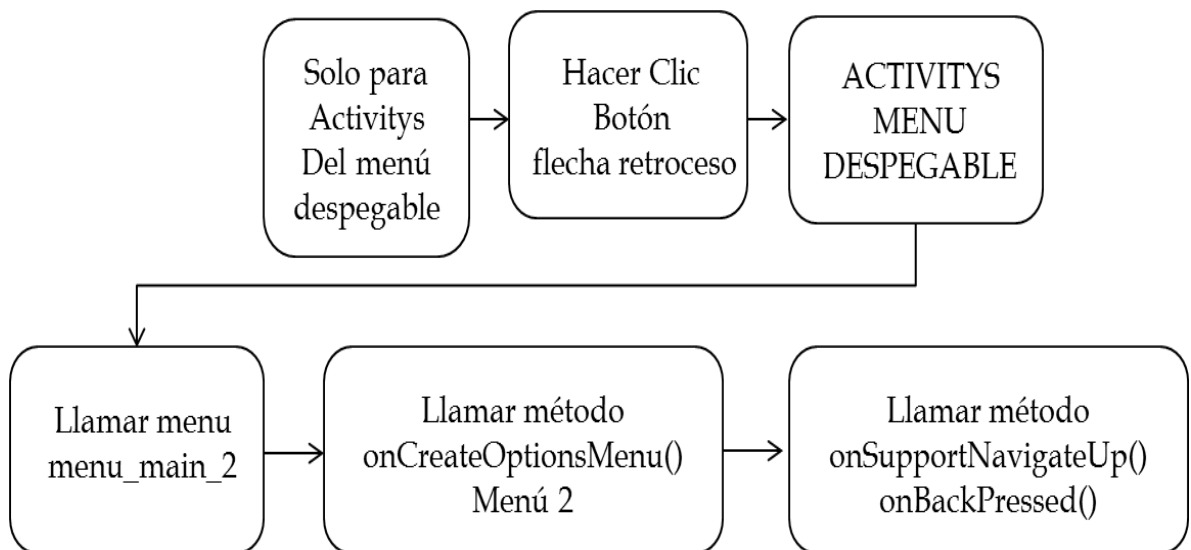
**Figura 43. Diagrama de flujo ajustes NFC**



Se utiliza el método nfcAdapter que posee Android Studio en el archivo MainActivity y se agrega un nuevo case en la parte del código del menú desplazable, con el código del programa anexo (Ajustes NFC) se cumple con el diagrama.

**Creación del menú retroceso** Este menú es una flecha de retroceso que se agrega antes del nombre de cualquier actividad como se ve en la Figura 44. La flecha de menú retroceso se crea en la carpeta app/res/menu con nuevo archivo xml llamado menu\_main\_2.xml en este caso solo con un contenedor menu vacío y en la parte de java se enlaza con el código del programa anexo (Menú Retroceso) para ActivarContraseñaActivity, InformacionActivity y PaginauisActivity.

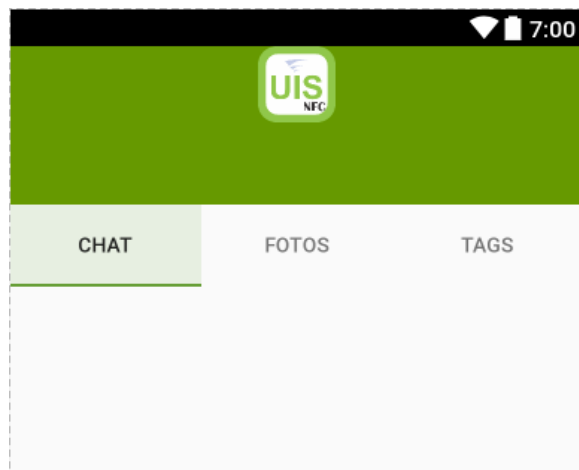
**Figura 44. Diagrama de flujo menú retroceso**



**Creación del componente Tabs (pestañas)** El componente Tabs se utiliza para organizar el contenido de la aplicación ya sea como un conjunto de datos o funcionalidades que se muestran mediante pestañas, el componente se agrega luego de utilizar el Include del archivo activity\_main.xml ubicado en la carpeta app/res/layout se genera un nuevo archivo que enlaza el Include con el nuevo archivo llamado content\_main.xml, el nuevo archivo se ubica en la carpeta layout. Al crear el content\_main.xml se utiliza como contenedor un RelativeLayout que ocupa todo el ancho de la aplicación para desarrollar las actividades.

**Creación del Tabhost** Dentro del RelativeLayout se agrega el elemento Tabhost que contiene los elementos que se muestran en la Figura 45. Las tres pestañas que parcialmente arroja el Tabhost se modifican a las tres características que va a trabajar la aplicación que son CHAT, FOTOS y TAGS. Dentro de la Figura 45 se observa la interfaz gráfica Tabhost en Android Studio. Cada pestaña contiene un Include, este se utilizará más adelante para llamar el archivo de interfaz gráfica correspondiente a cada pestaña del Tabhost.

**Figura 45. Interfaz gráfica Tabhost en Android Studio**

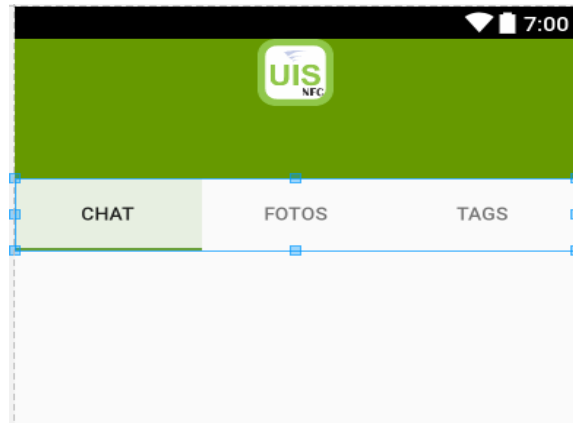


**Creación de las pestañas** El componente Tabhost maneja de manera predeterminada tres pestañas las cuales modificamos con los nombres CHAT, FOTOS y TAGS.

**Creación del TabWidget** El componente TabWidget viene predeterminado dentro del Tabhost, en el modificaremos su interfaz gráfica para hacer de nuestro diseño único y diferente.

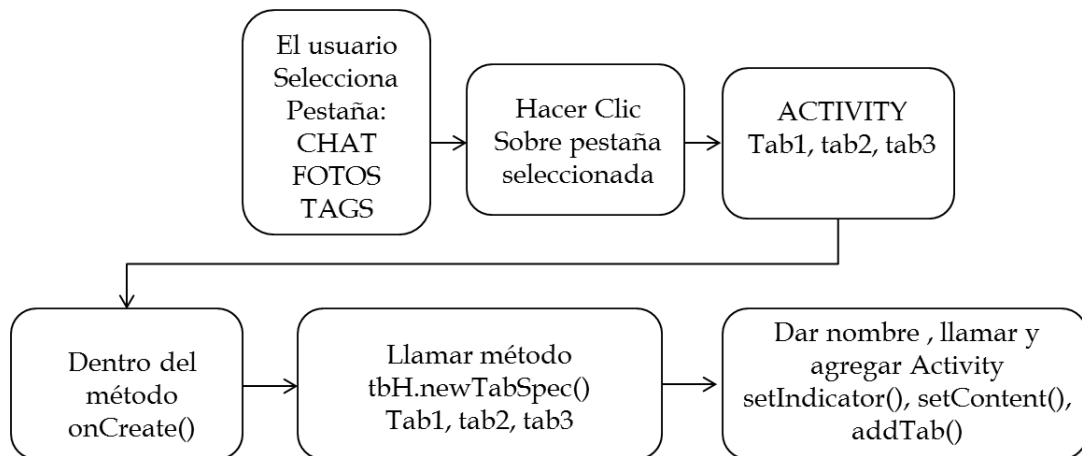
En la siguiente Figura 46 se observa la interfaz gráfica que maneja la aplicación para las pestañas fijas en el componente Tabhost.

**Figura 46. Interfaz gráfica TabWidget en Android Studio**



**Llamado del Tabhost** Para hacer que el elemento Tabhost se ejecute en la aplicación es necesario ver cómo funcionan las pestañas fijas, en la Figura 47 se muestra el diagrama de flujo de los Tabs para cada pestaña.

**Figura 47. Diagrama de flujo Tabhost**



Se enlaza con el código java dentro del método onCreate() del archivo principal MainActivity con el código del programa anexo (Elemento TabHost).

**Creación de la actividad CHAT** Se crea una nueva Activity en la carpeta app/res/layout con el nombre de tab1\_chat.xml que se enlaza con content\_main.xml mediante el Include del TabWidget tab1 y realizara sus procesos en el MainActivity directamente a través de código java. La Activity tab1\_chat.xml tiene como contenedor principal un ScrollView de color gris colorButtonNormal predeterminado de Android Studio, y un contenedor secundario LinearLayout para albergar las dos CardView (tarjetas) que ejecutaran las acciones de la aplicación en la pestaña CHAT. Estas acciones son enviar un mensaje y recibir un mensaje para la modalidad de chat compartido por NFC.

Para la actividad de chat se utiliza la solicitud de registro de Android (AAR) que proporciona una forma más directa de que la aplicación se inicie al escanear el registro NDEF. Al hacer que el registro creado tenga un NDEF único, solo el filtro de intención en el archivo manifests ejecutara la Activity señalada, dando unicidad y seguridad a los archivos que traspasan información, en este caso mensajes de texto.

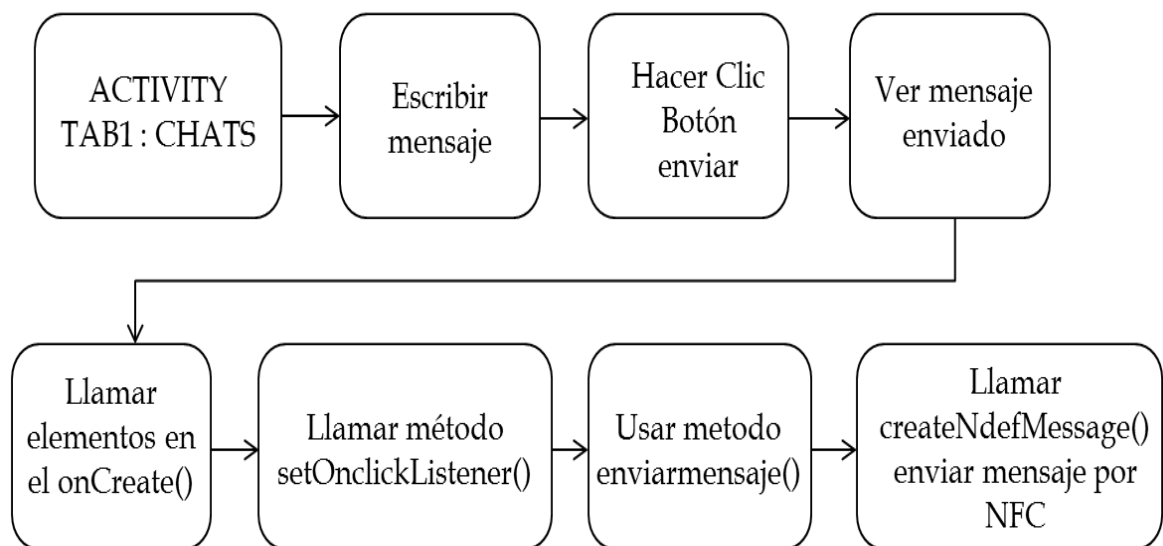
Si una etiqueta contiene un AAR, el sistema de etiqueta de envío distribuye de la manera siguiente [24]:

- Trate de iniciar una actividad utilizando un filtro de intención con normalidad. Si la actividad que coincide con la intención también coincide con la AAR, iniciar la actividad.
- Si la actividad que se filtra para obtener el propósito no coincide con la AAR, si varias actividades pueden manejar la intención, o si no hay actividad maneja la intención, se inicia la aplicación especificada por la AAR.
- Si ninguna aplicación puede comenzar con el AAR, ir a Google Play para descargar la aplicación en base a la AAR.

**Creación de tarjeta enviar mensaje** La tarjeta enviar mensaje crea la opción de enviar un mensaje a través de NFC por el dispositivo móvil, su función se ve en la Figura 48 mediante el diagrama de flujo se explica el proceso para enviar el mensaje.

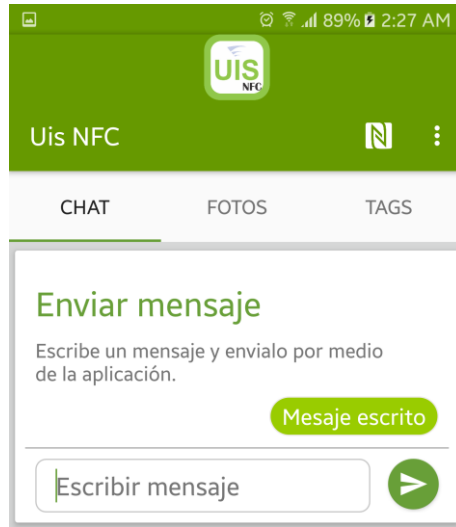
En el MainActivity se crea la interacción entre la interfaz gráfica y el código java para ejecutar los procesos planteados, la interacción se crea a través del llamado de los elementos y la ejecución de métodos que enlazan los elementos creados en el archivo tab1\_chat.xml. Los métodos están descritos con el código del programa anexo (Llamado de elementos).

**Figura 48. Diagrama de flujo para la tarjeta enviar mensaje**



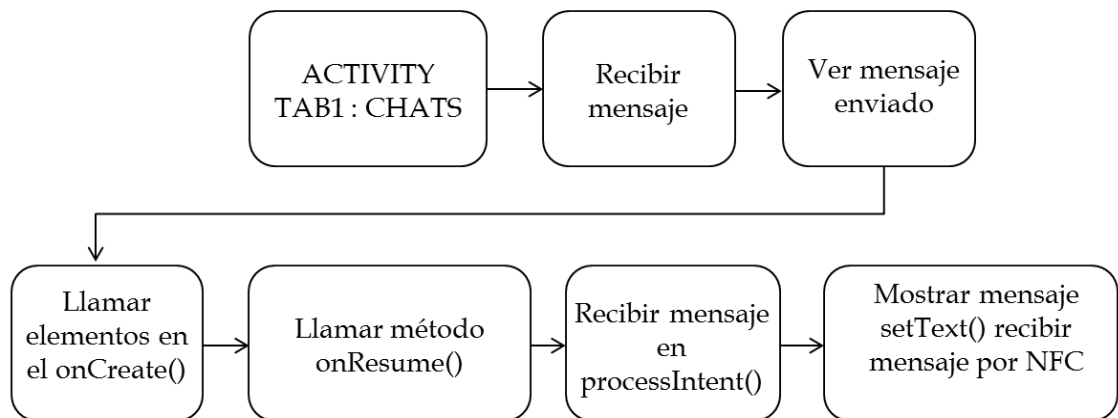
Se crea una tarjeta con el componente CardView como lo muestra la Figura 49 los elementos que componen la interfaz de usuario son textView, View, editText y floatingActionButton.

**Figura 49. Interfaz gráfica para la tarjeta enviar mensaje**



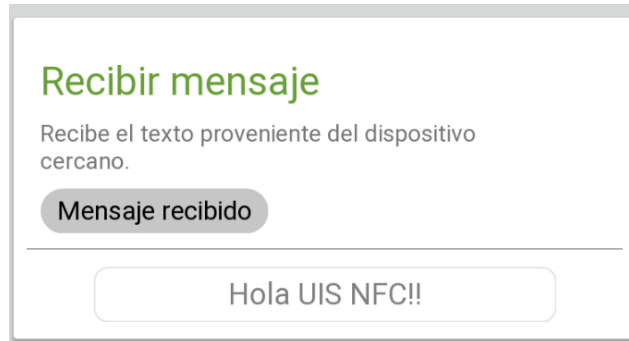
**Creación de tarjeta recibir mensaje** La tarjeta recibir mensaje crea la opción de recibir un mensaje a través de NFC por el dispositivo móvil, su función se ve en la Figura 50 mediante el diagrama de flujo se explica el proceso para enviar el mensaje.

**Figura 50. Diagrama de flujo para la tarjeta recibir mensaje**



Se crea una tarjeta con el componente CardView de la muestra la Figura 51 se agregan los elementos textView, View y editText.

**Figura 51. Interfaz gráfica para la tarjeta recibir mensaje**



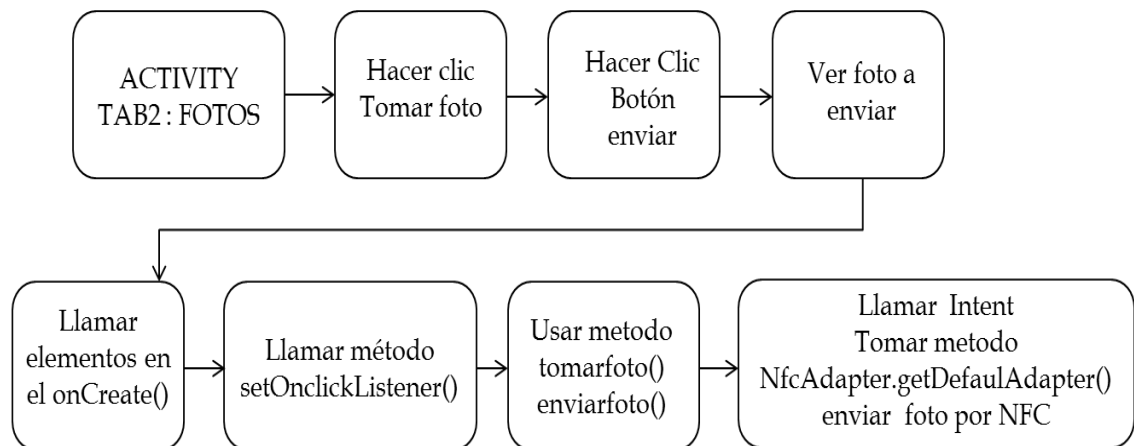
En el MainActivity se crea la interacción entre la interfaz gráfica y el código java para ejecutar los procesos planteados, la interacción se crea a través del llamado de los elementos y la ejecución de métodos que enlazan los elementos creados en el archivo tab1\_chat.xml. Los métodos están descritos con el código del programa anexo (Recibir mensaje).

**Creación de la actividad FOTOS** Se crea una nueva Activity en la carpeta app/res/layout con el nombre de tab2\_chat.xml que se enlaza con content\_main.xml mediante el Include del TabWidget tab2 y realizara sus procesos en el MainActivity directamente a través de código java. La Activity tab2\_fotos.xml tiene como contenedor principal un ScrollView de color gris colorButtonNormal predeterminado de Android Studio, y un contenedor secundario LinearLayout para albergar las dos CardView (tarjetas) que ejecutaran las acciones de la aplicación en la pestaña FOTOS. Estas acciones son enviar una foto y recibir una foto para la modalidad de fotos compartidas por NFC.

**Creación de tarjeta enviar foto** La tarjeta enviar foto crea la opción de enviar un mensaje a través de NFC por el dispositivo móvil, la primera función es tomar una foto y la segunda función es enviar una foto por NFC, en la Figura 52 mediante el diagrama de flujo se explica el proceso para tomar y enviar foto (imagen).

En el MainActivity se crea la interacción entre la interfaz gráfica y el código java para ejecutar los procesos planteados, la interacción se crea a través del llamado de los elementos y la ejecución de métodos que enlazan los elementos creados en el archivo tab2\_fotos.xml. Los métodos están descritos con el código del programa anexo (Enviar foto).

**Figura 52. Diagrama de flujo para la tarjeta enviar foto**



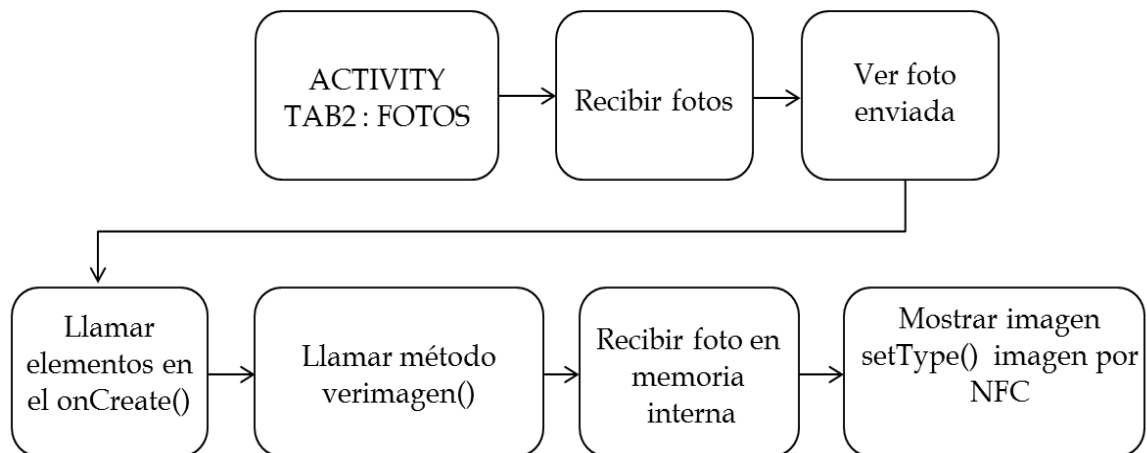
Se crea una tarjeta con el componente CardView, agregando los elementos textView, View, imageView y floatingActionButton, para crear la interfaz visual del Tab enviar foto como lo muestra la Figura 53.

**Figura 53. Interfaz gráfica para la tarjeta enviar foto**



**Creación de tarjeta recibir foto** La tarjeta recibir foto crea la opción de recibir una foto a través de NFC por el dispositivo móvil, su función se ve en la Figura 54 mediante el diagrama de flujo se explica el proceso para enviar el mensaje. Se crea una tarjeta con el componente CardView agregando los elementos textView, View, imageView y floatingActionButton, como lo muestra la Figura 55.

**Figura 54. Diagrama de flujo para la tarjeta recibir foto**



**Figura 55. Interfaz gráfica para la tarjeta recibir foto**



En el MainActivity se crea la interacción entre la interfaz gráfica y el código java para ejecutar los procesos planteados, la interacción se crea a través del llamado de los elementos y la ejecución de métodos que enlazan los elementos creados en

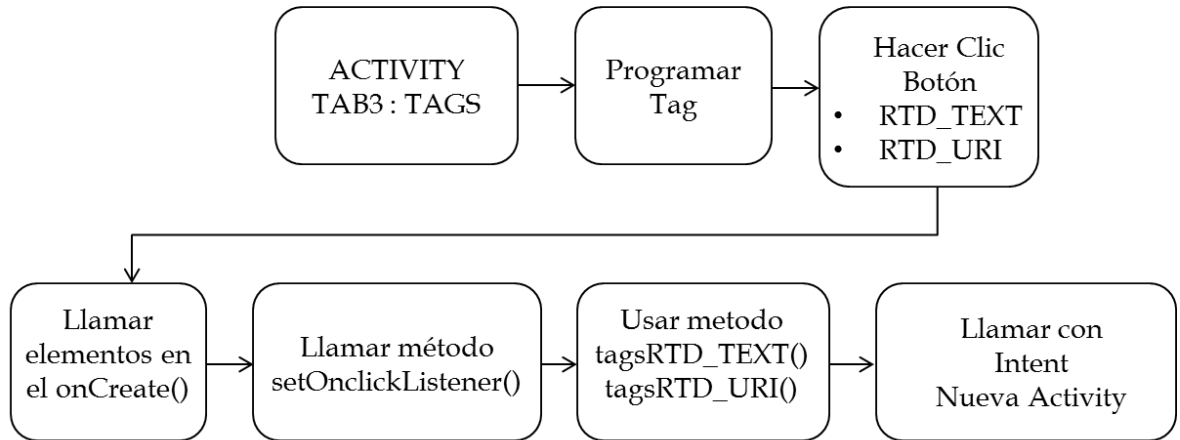
el archivo tab2\_fotos.xml. Los métodos están descritos con el código del programa anexo (Recibir foto).

**Creación de la Activity TAGS** Se crea una nueva Activity en la carpeta app/res/layout con el nombre de tab3\_tags.xml que se enlaza con content\_main.xml mediante el Include del TabWidget tab3 y realizara sus procesos en el MainActivity directamente a través de código java. La Activity tab3\_tags.xml tiene como contenedor principal un ScrollView de color gris colorButtonNormal predeterminado de Android Studio, y un contenedor secundario LinearLayout para albergar las dos CardView (tarjetas) que ejecutaran las acciones de la aplicación en la pestaña TAGS. Estas acciones son escribir registros tipo TEXT y URI en Tags NFC, y abrir la intención de tomar una foto al escanear el Tag NFC (configurado con el tipo de registros) almacenándola en el dispositivo móvil.

En esta Activity no se especifica la forma de lectura para la información de datos dentro del Tag NFC pues los filtros de intención se colocan en la carpeta app archivo manifests ANEXO E , los filtros de intención reacción al escanear un Tag NFC que contienen la información escrita mediante la Activity TAGS.

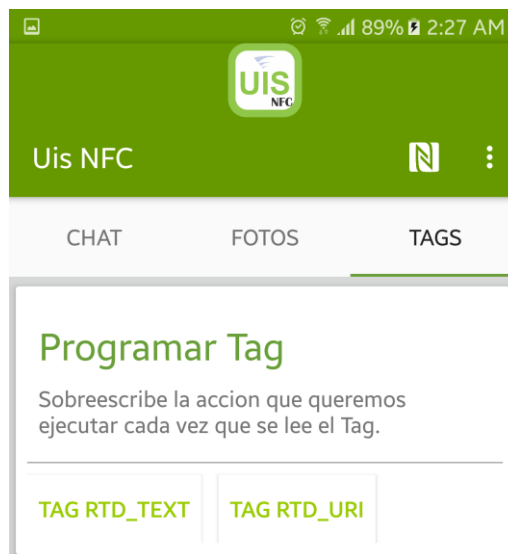
**Creación de tarjeta programar Tag** La tarjeta programar Tag crea la opción de escribir un registro través de NFC por el dispositivo móvil a un Tag como se ven en la Figura 56, la primera función es escribir un registro tipo RTD\_TEXT a un Tag y la segunda función es escribir un registro tipo RTD\_URI a un Tag por NFC, estas funciones se realizaran a través de dos botones uno que se llama TAG RTD\_TEXT y el otro TAG RTD\_URI. Se crea una tarjeta con el componente CardView agregando los elementos textView, View, imageView y los dos botton con los que se llamaran las Activitys, como lo muestra la imagen Figura 57.

**Figura 56. Diagrama de flujo para la tarjeta programar Tag**



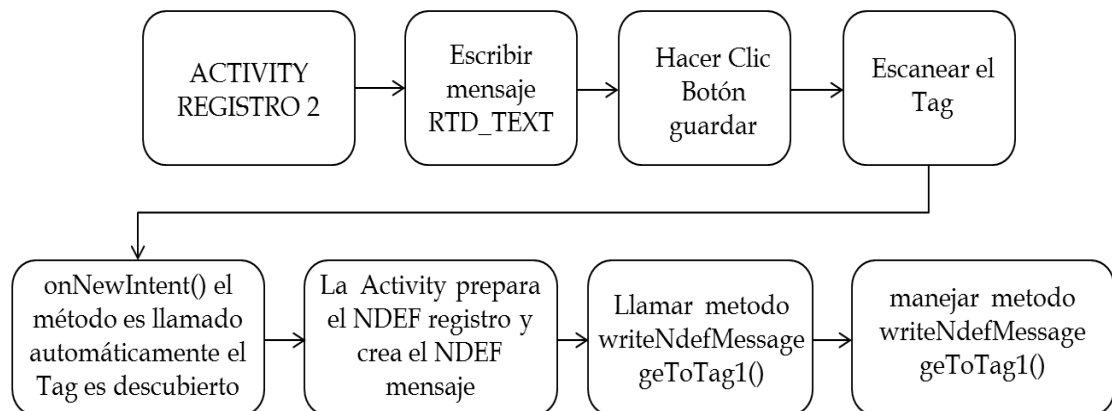
En el MainActivity se crea la interacción entre la interfaz gráfica y el código java, para ejecutar los procesos planteados por los dos botones se crea una interacción a través del llamado de cada botón y la ejecución de métodos que enlazan los elementos creados en el archivo tab3\_tags.xml. Los métodos están descritos con el código del programa anexo (Programar Tag). Cada botón llama una Activity que contiene dependiendo del botón seleccionado el tipo de registro que se va a escribir en un Tag.

**Figura 57. Interfaz gráfica para la tarjeta programar Tag**



**Creación de tarjeta registro texto** Al seleccionar el botón TAG RTD\_TEXT se abre una intención que llama a la nueva Activity, en la carpeta `app/java/com.example.memo.uisnfc` se crea una Activity con Basic Activity de nombre `Registro2Activity` y con archivo xml `activity_registro2.xml`. Como contenedor principal para el archivo xml se usa un `ScrollView` de color gris `colorButtonNormal` predeterminado de Android Studio, y un contenedor secundario `LinearLayout` para albergar la `CardView` (tarjeta) que ejecutara las acciones de la aplicación para escribir registros tipo `RTD_TEXT`. La tarjeta registro texto crea la opción de escribir un registro tipo `RTD_TEXT` a través de NFC por el dispositivo móvil, su función se ve en la Figura 58. Diagrama de flujo para la tarjeta registro tipo texto

**Figura 58. Diagrama de flujo para la tarjeta registro tipo texto**

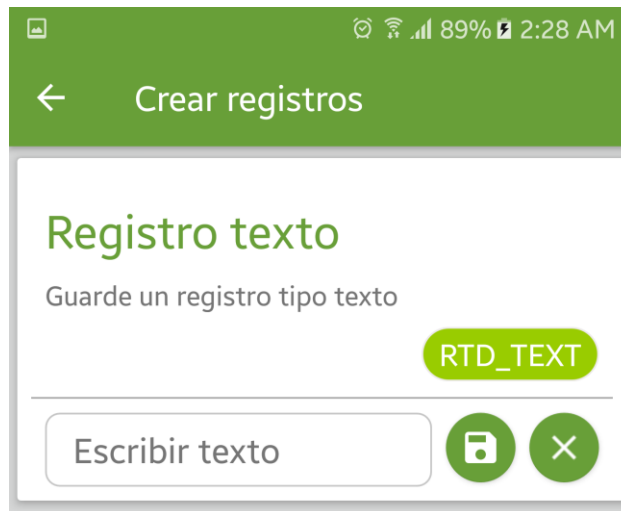


Se crea una tarjeta con el componente `CardView` agregando los elementos `textView`, `View`, `editText` y `floatingActionButton`, como lo muestra la Figura 59.

En el `Registro2Activity` se crea la interacción entre la interfaz gráfica y el código java para ejecutar los procesos planteados, la interacción se crea a través del llamado de los elementos y la ejecución de métodos que enlazan los elementos

creados en el `activity_registro2.xml`. Los métodos están descritos con el código del programa anexo (Registro tipo texto).

**Figura 59. Interfaz gráfica para la tarjeta registro tipo texto**

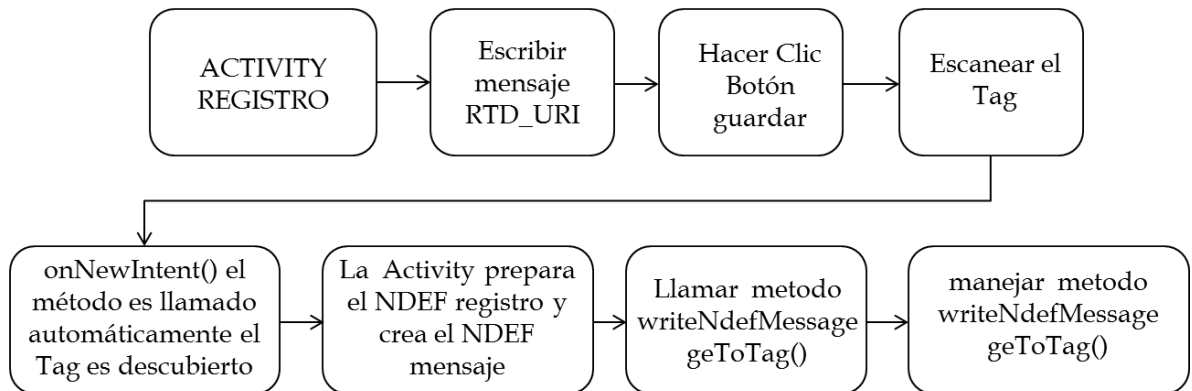


**Creación de tarjeta registro URI** Al seleccionar el botón TAG RTD\_URI se abre una intención que llama a la nueva Activity, en la carpeta `app/java/com.example.memo.uisnfc` se crea una Activity con Basic Activity de nombre `RegistroActivity` y con archivo xml `activity_registro.xml`. Como contenedor principal para el archivo xml se usa un `ScrollView` de color gris `colorButtonNormal` predeterminado de Android Studio, y un contenedor secundario `LinearLayout` para albergar la `CardView` (tarjeta) que ejecutara las acciones de la aplicación para escribir registros tipo RTD\_URI.

La tarjeta registro URI crea la opción de escribir un registro tipo RTD\_URI a través de NFC por el dispositivo móvil, su función se ve en la Figura 60 mediante el diagrama de flujo se explica el proceso para escribir el registro.

Se crea una tarjeta con el componente `CardView` agregando los elementos `textView`, `View`, `editText` y `floatingActionButton`, como lo muestra la Figura 61.

**Figura 60. Diagrama de flujo para la tarjeta registro tipo URI**



**Figura 61. Interfaz gráfica para la tarjeta registro tipo URI**

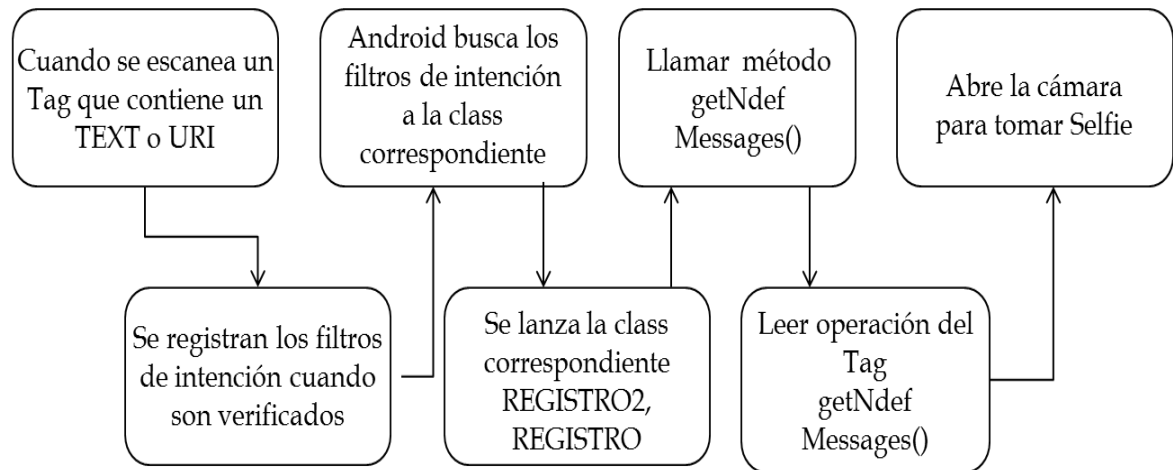


En el RegistroActivity se crea la interacción entre la interfaz gráfica y el código java para ejecutar los procesos planteados, la interacción se crea a través del llamado de los elementos y la ejecución de métodos que enlazan los elementos creados en el activity\_registro2.xml. Los métodos están descritos con el código del programa anexo (Registro tipo URI).

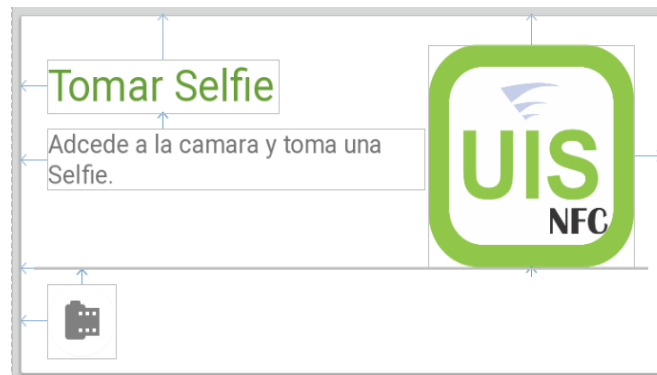
**Creación de tarjeta tomar selfie** La tarjeta tomar selfie crea la opción de tomarse una foto a través de NFC por el dispositivo móvil cuando responde a los registros escritos en un Tag, su función se ve en la Figura 62 mediante el diagrama de flujo se explica el proceso para tomarse la selfie.

Se crea una tarjeta con el componente CardView agregando los elementos textView, imageView y floatingActionButton, como lo muestra la interfaz gráfica de la Figura 63.

**Figura 62. Diagrama de flujo para la tarjeta tomar selfie**



**Figura 63. Interfaz gráfica para la tarjeta tomar selfie**

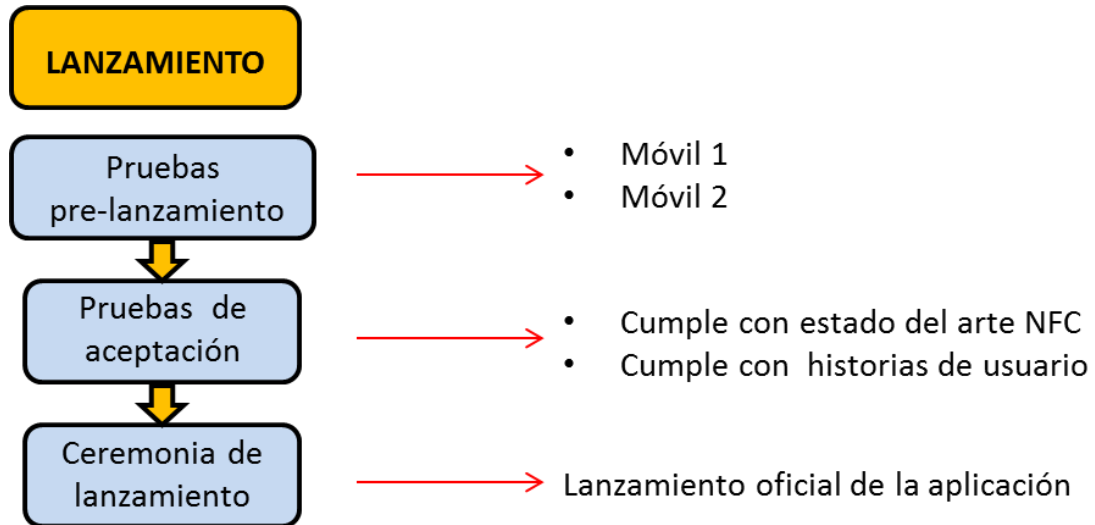


En el MainActivity se crea la interacción entre la interfaz gráfica y el código java para ejecutar los procesos planteados, la interacción se crea a través del llamado de los elementos y la ejecución de métodos que enlazan los elementos creados en el archivo tab3\_tags.xml. Los métodos están descritos con el código del programa anexo (Tomar selfie).

### 3.2.4 LANZAMIENTO.

En esta fase se realiza el lanzamiento de la aplicación. En la Figura 64. Características de la fase Lanzamiento observamos las etapas de la fase y el desarrollo de cada actividad.

**Figura 64. Características de la fase Lanzamiento**



La aplicación transmite datos mediante NFC y los escribe en etiquetas Tag (NFC) lo cual permite satisfacer las necesidades de usuarios establecidas además se recopila toda la información con las pruebas pre-lanzamiento dándole a la aplicación una estabilidad para trabajar en la transmisión de datos mediante NFC.

**3.2.4.1 Pruebas pre-lanzamiento** Se implementa la aplicación en dos tipos diferentes de dispositivos móviles un Samsung Galaxy S6 Edge (Android 6.0.1 Marshmallow) y un Samsung Galaxy S4 mini (Android 4.2.2 Jelly Bean), el terminal de Android Studio verifica que no hay problemas en la ejecución del código y procede a abrir la aplicación en modo desarrollador dentro de cada dispositivo móvil.

**Samsung Galaxy S6 Edge (Adroid 6.0.1 Marshmallow)** Al ejecutar la instalación de la aplicación en modo desarrollador dentro del dispositivo móvil, la aplicación es lanzada una vez termina el proceso de instalación abriendo la Activity principal y su interfaz de usuario. Como primera actividad tenemos introducir usuario y contraseña, si alguno de los 2 es incorrecto obtenemos un mensaje de cual está en error, si están correctos seleccionamos nuevamente el botón ingresar y entramos a la interfaz principal de la aplicación.

Dentro de la interfaz de usuario principal primero revisamos si los dos menús de la barra principal están funcionando correctamente. El primer menú es el logo de la N-mark donde se comprueba si Android NFC y Android Beam están activados, para ello se desactivas las dos funciones del dispositivo móvil y se verifica por medio del logo N-mark la activación de ambas funciones.

El menú desplegable muestras las actividades que construimos en un listado al seleccionarlo, se verifica que al hacer clic sobre cada ítem de la lista se ingresa a las Activitys Contraseña, Información y Pagina UIS. Cada Activity funciona correctamente y su botón de retroceso en la barra menú responde correctamente.

En la interfaz principal se observa correctamente el funcionamiento de cada pestaña fija y muestra cada actividad en cada pestaña seleccionada, se verifica el funcionamiento de cada actividad mediante 3 tablas:

- Tabla CHATS de enviar y recibir mensajes de texto
- Tabla FOTOS de enviar y recibir imagenes
- Tabla TAGS de escribir/leer registros en Tags NFC

### Actividad CHATS

Las pruebas se realizan entre dos dispositivos móviles Samsung Galaxy S6 Edge (Adroid 6.0.1 Marshmallow) móvil 1, Samsung Galaxy S4 mini (Adroid 4.2.2 Jelly Bean) móvil 2 y arrojan los siguientes resultados.

**Tabla 3. Pruebas para actividad CHATS**

DISPOSITIVO	CHATS		MENSAJE	
	ENVIA	RECIBE	ENVIADOS	RECIBIDOS
MOVIL 1	Si	Si	10	10
MOVIL 2	Si	Si	10	10

### Actividad FOTOS

Las pruebas se desarrollan entre los dos dispositivos móviles, emparejando la información por Android Beam

**Tabla 4. Pruebas para actividad FOTOS**

DISPOSITIVO	FOTOS		IMAGENES	
	ENVIA	RECIBE	ENVIADOS	RECIBIDOS
MOVIL 1	Si	Si	10	10
MOVIL 2	Si	Si	10	10

### Actividad TAGS

Las pruebas se realizan con los dos dispositivos móviles, cada uno escanea previamente el mismo Tag NFC para obtener diferentes registros.

**Tabla 5. Pruebas para actividad TAGS**

DISPOSITIVO	TAGS		REGISTROS	
	ESCRIBIR	LEER	RTD_TEXT	RTD_URI
MOVIL 1	Si	Si	10	10
MOVIL 2	Si	Si	10	10

**Samsung Galaxy S3 (Adroid 4.2.2 Jelly Bean)** Las tablas 8, 9 y 10 contienen el resultado de la aplicación dentro del móvil 2, se realizó el mismo proceso de instalación y revisión de las funcionalidades contenidas dentro de la aplicación, obteniendo los mismos resultados del Samsung Galaxy S6 Edge (Adroid 6.0.1 Marshmallow)

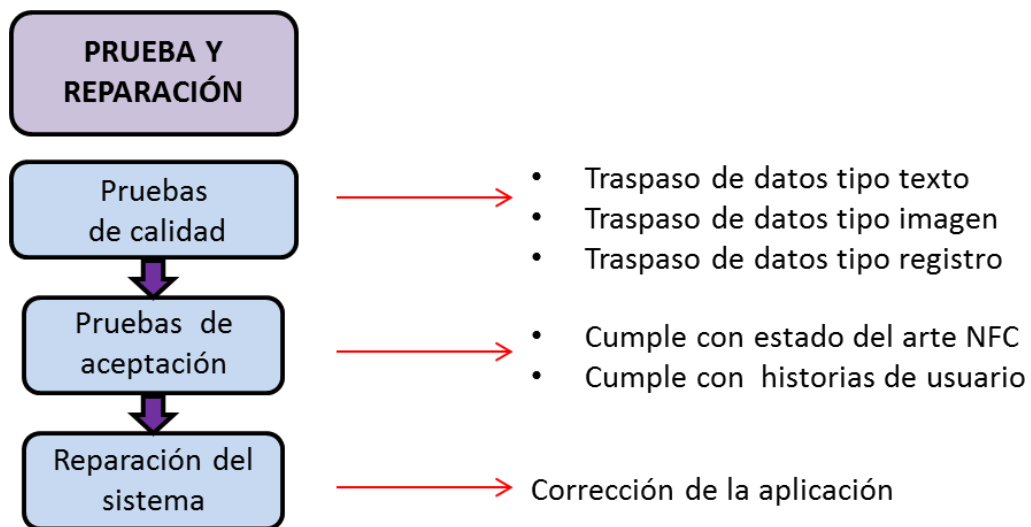
**3.2.4.2 Prueba de aceptación** En el estado del arte se fijaron los objetivos de la aplicación para satisfacer a los usuarios, al realizar las pruebas pre-lanzamiento se verifico lo siguiente:

- La aplicación contiene un nombre de usuario y contraseña, si son incorrectas no se puede ejecutar las actividades siguientes.
- En la pestaña CHAT realiza el intercambio de mensajes de texto mediante NFC (por Android Beam) entre dos dispositivos móviles con la aplicación instalada y los muestra en las burbujas de texto correspondientes a cada tarjeta.
- En la pestaña FOTOS realiza el intercambio de imágenes mediante NFC (por Android Beam) entre dos dispositivos móviles con la aplicación instalada y los muestra en pequeñas imágenes correspondientes a cada tarjeta.
- Escribe registros en etiquetas NFC (Tags) tipo TEXT y URI.
- Lee registros tipo TEXT y URI de etiquetas NFC, abre la intención de tomarse una selfie cada vez que el dispositivo móvil este desbloqueado y que escanee un Tag con este tipo de registro en su interior.

**3.2.4.3 Ceremonia de lanzamiento** Al verificar que la pruebas de aceptación se satisfacen se procede a dejar el software desarrollado en una carpeta exclusiva como proyecto finalizado y dejarlo previamente instalado en los dos dispositivos de prueba.

**3.2.5 PRUEBA Y REPARACIÓN DEL SISTEMA (CORRECCIÓN)** En esta fase se realiza las pruebas de la aplicación después de lanzada en una muestra de la población, se verifica su eficacia y si es necesario se corrigen errores de utilidad. En la Figura 65. Características de la fase Prueba y Reparación observamos las etapas de la fase y el desarrollo de cada actividad.

**Figura 65. Características de la fase Prueba y Reparación**



Esta última fase se prueba con la aplicación en los dos dispositivos de prueba Samsung Galaxy S6 Edge (Adroid 6.0.1 Marshmallow) y un Samsung Galaxy S4 mini (Adroid 4.2.2 Jelly Bean) en búsqueda de errores inesperados a fin de solucionarlos y entregar un producto final lo más completo y estable posible.

Se realizaran cuatro tipos de pruebas con resultados escritos en tablas, las pruebas son las siguientes:

- Prueba de traspaso de datos tipo mensaje texto entre los dos dispositivos móviles.
- Prueba de traspaso de datos tipo imagen entre los dos dispositivos móviles.
- Prueba escritura de registro tipo URI y texto entre dispositivo móvil y Tag (NFC).
- Prueba lectura de registro tipo URI y texto entre Tag (NFC) y dispositivo móvil.

**3.2.5.1 Traspaso de datos tipo mensaje texto** Las pruebas se realizan entre dos dispositivos móviles Samsung Galaxy S6 Edge (Adroid 6.0.1 Marshmallow) móvil 1, Samsung Galaxy S4 mini (Adroid 4.2.2 Jelly Bean) móvil 2 y arrojan los siguientes resultados.

**Tabla 6. Prueba de traspaso de datos tipo mensaje de texto**

DISPOSITIVO	TRANSMISION DATOS TEXTO			
	B	$\bar{B}$	s	$\bar{s}$
MOVIL 1	22	20	0.5	0.53
	20		0.5	
	18		0.6	
MOVIL 2	48	34.7	0.6	0.53
	34		0.5	
	22		0.5	

Las especificaciones en la tabla son las siguientes:

- B: bytes
- $\bar{B}$ : bytes promedio
- s: segundos
- $\bar{s}$ : segundos promedio

**3.2.5.2 Traspaso de datos tipo imagen** Las pruebas se realizan entre dos dispositivos móviles Samsung Galaxy S6 Edge (Adroid 6.0.1 Marshmallow) móvil 1, Samsung Galaxy S4 mini (Adroid 4.2.2 Jelly Bean) móvil 2 y arrojan los siguientes resultados.

**Tabla 7. Prueba de traspaso de datos tipo imagen**

DISPOSITIVO	TRANSMISION DATOS IMAGEN			
	MB	$\overline{MB}$	s	$\bar{s}$
MOVIL 1	1.79	1.79	25	24.7
	1.79		24	
	1.79		25	
MOVIL 2	1.67	1.67	18	18.3
	1.67		19	
	1.67		18	

Las especificaciones en la tabla son las siguientes:

- MB: megabytes
- $\overline{MB}$ : megabytes promedio
- s: segundos
- $\bar{s}$ : segundos promedio

**3.2.5.3 Traspaso de datos tipo registros NFC** Las pruebas se realizan entre dos dispositivos móviles Samsung Galaxy S6 Edge (Adroid 6.0.1 Marshmallow) móvil 1, Samsung Galaxy S4 mini (Adroid 4.2.2 Jelly Bean) móvil 2 y un Smart Ring marca Jakcom que contiene las especificaciones del Anexo B. La tabla 13 contiene los siguientes resultados.

Las especificaciones en la tabla son las siguientes:

- B: bytes
- $\overline{B}$ : bytes promedio
- s: segundos

- $\bar{s}$ : segundos promedio

**Tabla 8. Prueba de traspaso de datos tipo registro NFC**

Móvil	ESCRITURA								LECTURA							
	RTD_TEXT				RTD_URI				RTD_TEXT				RTD_URI			
	B	$\bar{B}$	s	$\bar{s}$	B	$\bar{B}$	s	$\bar{s}$	B	$\bar{B}$	s	$\bar{s}$	B	$\bar{B}$	s	$\bar{s}$
<b>Móvil 1 TAG</b>	18		0.5		7		0.5		18		0.6		7		0.5	
	20	20	0.6	0.53	8	8.7	0.8	0.6	20	20	0.7	0.7	8	8.7	0.7	0.6
	22		0.5		11		0.5		22		0.8		11		0.7	
<b>Móvil 2 TAG</b>	24		0.5		9		0.5		24		0.6		9		0.7	
	34	35.3	0.6	0.57	19	17.3	0.5	0.5	34	35.3	0.8	0.7	19	17.3	0.7	0.7
	48		0.6		24		0.5		48		0.7		24		0.7	

#### 4. CONCLUSIONES

- En el presente trabajo de grado, se construyó la aplicación UIS NFC para la plataforma Android realizando el estudio del estado del arte en aplicaciones NFC dentro de la Play Store, metodologías ágiles para el desarrollo de software y programación en Android Studio. De esta manera se creó la Aplicación UIS NFC como pionera en el traspaso de datos mediante la tecnología NFC.
- Se realizó un estado del arte del NFC en la plataforma Android, para solucionar historias de usuarios que permitió generar actividades principales para la aplicación desarrollada, además el estado del arte nos muestra que tan confiable, concurrida y avanzada esta la tecnología NFC. A partir del estado del arte se genera una aplicación más actualizada y a la vanguardia.
- Se eligió una metodología ágil para la creación de software de dispositivos móviles, teniendo en cuenta el estudio previo del estado del arte, el personal para el desarrollo y los conocimientos relacionados al desarrollo de aplicaciones móviles. De esta manera, fue posible seccionar por etapas y fases las actividades correspondientes para el desarrollo de la aplicación NFC. Solucionando problemas y capacitando al desarrollador para cumplir los objetivos de cada fase.
- Desarrollar una aplicación en la plataforma Android genera un conocimiento en programación java, pues Google que es el desarrollador del sistema operativo Android ha fusionado a través del tiempo el software para el desarrollo de aplicativos móvil con el lenguaje java como plataforma de desarrollo global. Para entender la plataforma java se estudió mediante tutoriales de fácil acceso en la web la programación dirigida a objetos que es

la base del lenguaje máquina para el desarrollo de aplicaciones en Android Studio.

- La programación dirigida a objetos en java desarrolla el conocimiento de cómo funciona el código dentro de la aplicación creada, esto se logra a través del estudio del lenguaje máquina y el entendimiento de los principios en programación java. La programación dirigida a objetos en java hace que el código se pueda seccionar en diferentes partes y ayuda a unir características que poseen otros objetos, para desarrollar sin necesidad de crear nuevas propiedades en cada programa de java.
- Material Design mediante estudios, desarrolla guías de construcción para aplicaciones, los cuales contiene todos los elementos que deben generarse en la interfaz gráfica de la aplicación. Seguir estas guías de interfaz gráfica garantizan una mayor probabilidad que la aplicación está diseñada correctamente en Android Studio y que tenga un mayor éxito de calidad para los usuarios que la adquieran.
- Utilizando la iteración de procesos para desarrollar aplicaciones móviles en Android Studio de la fase planificación en la etapa producción, logramos desarrollar aplicaciones en la plataforma de desarrollo Android Studio que justifican las necesidades planteadas en el estado del arte, la programación dirigida a objetos en java y el desarrollo de interfaz gráfica guiado por Material Design. Android Studio posee una plataforma de desarrollo web que sirven de guía para la ejecución de actividades, conformados por ejemplos y conceptos que aclaran la programación en Android Studio, además cuenta con una gran base de desarrollo que permiten crear todos los componentes de una aplicación móvil.

## 5. OBSERVACIONES Y TRABAJOS FUTUROS

- Las pruebas realizadas en la fase día de lanzamiento, certifican el funcionamiento de las actividades desarrolladas por la aplicación y corrobora que el desarrollo del proyecto se cumple a totalidad.
- Se verifica la calidad de la aplicación mediante la fase de prueba y reparación del sistema, realizando pruebas que recolectan información sobre la calidad y eficiencia de la aplicación en la transmisión de datos NFC, las pruebas realizadas en la fase de prueba demuestran que los datos se transmiten a la tasa estandarizada por el protocolo NFC.
- Las pruebas que se realizaron con el dispositivo móvil Samsung Galaxy S6 Edge (Android 6.0.1 Marshmallow) móvil 1 y el Smart Ring marca Jakcom se hicieron repetitivamente escogiendo las acertadas, pues muchas de las pruebas realizadas resultaron fallidas debido a que el Smart Ring contiene dos chips de alta velocidad NFC de doble núcleo. El Chip 1 del Anexo B es el que contiene la tecnología que soporta nuestra aplicación, mientras que el Chip 2 del Anexo B no contiene las tecnologías soportada por nuestra aplicación, lo que hace que al leer el Smart Ring con la aplicación si lee el Chip 2 se cierra el ciclo y no se ejecuta ninguna intención, mientras si lee el Chip 1 la intención de la aplicación UIS NFC se ejecuta correctamente.
- Las pruebas que se realizaron con el dispositivo móvil Samsung Galaxy S4 mini (Android 4.2.2 Jelly Bean) móvil 2 y el Smart Ring marca Jakcom se hicieron repetitivamente escogiéndose las acertadas, pues muchas de las pruebas realizadas resultaron fallidas debido al problema mencionada en la conclusión anterior. Al leer el Chip 1 la intención de la aplicación UIS NFC se ejecuta correctamente y certificando la función de la aplicación.

## Trabajos Futuros

- Desarrollar aplicaciones basadas en NFC con seguridad. El dispositivo móvil elimina la necesidad de usar un sinnúmero de tarjetas plásticas, se eleva el nivel de seguridad que experimenta el usuario, se eleva también la versatilidad de los servicios que el usuario experimenta pues se mezclan con toda la potencia de cómputo, almacenamiento y conectividad de tal dispositivo. En este campo particular se presenta la siguientes necesidades:
  - Uso de tarjetas plásticas con chip para abrir puerta.
  - Uso de tarjetas plásticas para retirar dinero de un cajero.
  - Uso de tarjetas plásticas para pagar metrolínea.
  - Otros tipos de pagos o cupones que se puedan realizar vía electrónica.
- Desarrollar aplicaciones NFC mediante otras metodologías ágiles para el desarrollo de software y encontrar cual es la más eficiente en diversos desarrollos. Igualmente se requiere documentar esta experiencia en forma de una metodología útil para continuar este tipo de desarrollos en Universidad Industrial de Santander (UIS).

## BIBLIOGRAFÍA

ANDROID STUDIO, Conoce Android Studio | Android Studio. [En línea]. Disponible en: <https://developer.android.com/studio/intro/index.html?hl=es-419>. [Accedido: 18-may-2017].

BLANCO P., CAMARERO J., FUMERO A., WERTERSKI A. y RODRÍGUEZ P. Metodología de desarrollo ágil para sistemas móviles Introducción al desarrollo con Android y el iPhone, pp. 1-30.

CANCINO M. y SANDOVAL Nicolas. Presentación de mobile-D | Portafolio personal en WordPress.com. [En línea]. Disponible en: <https://nicolasacancino.wordpress.com/2014/04/25/presentacion-de-mobile-d/>. [Accedido: 11-may-2016].

COSKUN V., OK K. y OZDENIZCI B. NFC Application Development for Android. 2013. 316 p.

GOOGLE PLAY. NFC - Aplicaciones de Android en Google Play. [En línea]. Disponible en: <https://play.google.com/store/search?q=nfc&c=apps&hl=es>. [Accedido: 26-abr-2017].

HYVONEN L., PINTO A., y TROELSEN J. Near field communication. 2012. 20 p.

ISO/IEC 14443-2. INTERNATIONAL STANDARD ISO / IEC Contents, vol. 2001, 2001.

ISO/IEC 18092. INTERNATIONAL STANDARD ISO / IEC Telecommunications and information Communication — Interface and Protocol , Int. Stand. Ref. number ISO/IEC 180922013(E), vol. Second Edi, 2013.

MATERIA DESIGN. Color - Style - Material design guidelines. [En línea]. Disponible en: <https://material.io/guidelines/style/color.html#color-color-palette>. [Accedido: 24-abr-2017].

NFC FORUM. Bluetooth Secure Simple Pairing Using NFC Application Document, 2014.

NFC FORUM. What Is NFC? - NFC Forum | NFC Forum. [En línea]. Disponible en: <http://nfc-forum.org/what-is-nfc/>.

## REFERENCIAS BIBLIOGRAFÍA

- [1] M. A. PEREZ, «Más dispositivos móviles que habitantes en la Tierra a finales de 2013». [En línea]. Disponible en: <http://blogthinkbig.com/dispositivos-moviles-habitantes-tierra/>.
- [2] NFC FORUM, «What Is NFC? - NFC Forum | NFC Forum». [En línea]. Disponible en: <http://nfc-forum.org/what-is-nfc/>.
- [3] L. HYVONEN, A. PINTO, y J. TROELSEN, *Near field communication*. 2012.
- [4] ISO/IEC 18092, «INTERNATIONAL STANDARD ISO / IEC Telecommunications and information Communication — Interface and Protocol», *Int. Stand. Ref. number ISO/IEC 180922013(E)*, vol. Second Edi, 2013.
- [5] FQ INGENIERIA ELECTRONICA, «Estándares y regularizaciones para RFID | FQ Ingeniería Electrónica». [En línea]. Disponible en: <http://www.fqingenieria.com/es/conocimiento/estandares-y-regularizaciones-para-rfid-36>. [Accedido: 28-mar-2016].
- [6] V. COSKUN, K. OK, y B. OZDENIZCI, «NFC Application Development for Android».
- [7] P. BLANCO, J. CAMARERO, A. FUMERO, A. WERTERSKI, y P. RODRÍGUEZ, «Metodología de desarrollo ágil para sistemas móviles Introducción al desarrollo con Android y el iPhone», pp. 1-30.
- [8] AGENCIA EFE, «¿Cuántos usuarios tiene Instagram?» [En línea]. Disponible en: <http://www.primerahora.com/tecnologia/nota/cuantosusuariostieneinstagram-1160278/>. [Accedido: 07-feb-2017].
- [9] Crear Comunidad, «¿Cuántos usuarios tiene Whatsapp? – Crear Comunidad». [En línea]. Disponible en: <https://crearcomunidad.com/2016/02/08/cuantos-usuarios-tiene-whatsapp/>. [Accedido: 07-feb-2017].

- [10] ANDROID DEVELOPERS, «Conoce Android Studio | Android Studio». [En línea]. Disponible en: <https://developer.android.com/studio/intro/index.html>. [Accedido: 01-feb-2017].
- [11] BLOG HISTORIA DE LA INFORMATICA, «NFC – Historia de la Informática». [En línea]. Disponible en: <http://histinf.blogs.upv.es/2012/11/21/nfc/>. [Accedido: 09-feb-2017].
- [12] NFC FORUM, «Sobre Nosotros - NFC Forum | NFC Forum». [En línea]. Disponible en: <http://nfc-forum.org/about-us/>. [Accedido: 01-feb-2017].
- [13] GOOGLE PLAY, «Aplicaciones de Android en Google Play». [En línea]. Disponible en: <https://play.google.com/store/search?q=nfc&c=apps&hl=es>. [Accedido: 18-may-2017].
- [14] J. STEVEN PERRY, «Introducción a la programación Java, parte 1: Conceptos básicos del lenguaje Java». [En línea]. Disponible en: <https://www.ibm.com/developerworks/ssa/java/tutorials/j-introtojava1/>. [Accedido: 06-feb-2017].
- [15] J. STEVEN PERRY, «Introducción a la programación Java, parte 2: Construcciones para aplicaciones del mundo real». [En línea]. Disponible en: <http://www.ibm.com/developerworks/ssa/java/tutorials/j-introtojava2/index.html>. [Accedido: 06-feb-2017].
- [16] ILLASARON, «videotutoriales.com | Muchas cosas que aprender.» [En línea]. Disponible en: <https://videotutoriales.com/>. [Accedido: 18-may-2017].
- [17] ANDROID STUDIO, «Conoce Android Studio | Android Studio». [En línea]. Disponible en: <https://developer.android.com/studio/intro/index.html?hl=es-419>. [Accedido: 18-may-2017].
- [18] MATERIA DESIGN, «Color - Style - Material design guidelines». [En línea]. Disponible en: <https://material.io/guidelines/style/color.html#color-color-palette>. [Accedido: 24-abr-2017].
- [19] BAYMARD INSTITUTE, «Legibilidad: la duración óptima de línea - Artículos - Instituto Baymard». [En línea]. Disponible en: <https://baymard.com/blog/line-length-readability>. [Accedido: 24-abr-2017].

- [20] MATERIA DESIGN, «Tabs - Componentes - directrices de diseño de materiales». [En línea]. Disponible en: <https://material.io/guidelines/components/tabs.html>. [Accedido: 24-abr-2017].
- [21] MATERIA DESIGN, «Tarjetas - Componentes - directrices de diseño de materiales». [En línea]. Disponible en: <https://material.io/guidelines/components/cards.html>. [Accedido: 24-abr-2017].
- [22] MATERIA DESIGN, «Iconos - estilo - directrices de diseño de materiales». [En línea]. Disponible en: <https://material.io/guidelines/style/icons.html>. [Accedido: 24-abr-2017].
- [23] MATERIA DESIGN, «Unidades y medidas - Presentación - directrices de diseño de materiales». [En línea]. Disponible en: <https://material.io/guidelines/layout/units-measurements.html#units-measurements-density-independent-pixels-dp>. [Accedido: 24-abr-2017].
- [24] ANDROID DEVELOPERS, «NFC Básico | Desarrolladores de Android». [En línea]. Disponible en: <https://developer.android.com/guide/topics/connectivity/nfc/nfc.html>. [Accedido: 18-may-2017].
- [25] GOOGLE PLAY, «nfc - Aplicaciones de Android en Google Play». [En línea]. Disponible en: <https://play.google.com/store/search?q=nfc&c=apps&hl=es>. [Accedido: 26-abr-2017].

## ANEXOS

### ANEXO A. PLATAFORMAS DE DISPOSITIVOS MOVILES QUE CONTIENEN EL PROTOCOLO NFC

**Tabla 9. Plataformas que contienen protocolo NFC.**

<p><b>1. IOS</b></p> <ul style="list-style-type: none"> <li>• Apple Iphone 6</li> <li>• Apple Iphone 6 Plus</li> </ul> <p><b>2. ANDROID</b></p> <ul style="list-style-type: none"> <li>• Blu Life Pure XL</li> <li>• Motorola moto X XT1058</li> <li>• Motorola Droid Ultra X1080</li> <li>• Motorola Razr i XT8903</li> <li>• Motorola Razr D3 (XT919-XT920)</li> <li>• Motorola Razr HD (XT925-XT926)</li> <li>• Samsung Galaxy Note 24</li> <li>• LG Spectrum 2 Model # VS930 4G</li>   <li>• LG Optimus L5 E612/E615</li> <li>• LG Optimus L7 P700/P705</li> <li>• LG Optimus F6 D500</li> <li>• LG Optimus L9 P760/778g</li> <li>• LG Optimus G E970/E973</li> <li>• LG Optimus 4X HD P8805</li> <li>• LG G2 y LG G2 MINI</li> <li>• Samsung Galaxy Ace 2</li> </ul>	<ul style="list-style-type: none"> <li>• Huawei Ascend G630</li> <li>• Huawei Ascend G300 (no todas las versiones)</li> <li>• Huawei Ascend G510 (no todas las versiones)</li> <li>• Huawei U8650-1 (Integrado, pero no activado)</li> <li>• Turkcell T20.11</li> <li>• Sony Xperia S</li> <li>• Sony Xperia Acro S</li> <li>• Sony Xperia sola</li> <li>• Sony Xperia L</li> <li>• Sony Xperia M</li> <li>• Sony Xperia P</li> <li>• Sony Xperia ZL</li> <li>• Sony Xperia Z</li> <li>• Sony Xperia ion</li> <li>• Sony Xperia GX</li> <li>• Sony Xperia SX (no oficial)</li> <li>• Sony Xperia T</li> <li>• Sony Xperia V</li> <li>• Sony Xperia SP</li> <li>• Sony Xperia Z</li> <li>• Sony Xperia ZR</li> <li>• Sony Xperia Z1</li> <li>• Sony Xperia M2</li> <li>• Sony Xperia Z2</li> <li>• Sony Xperia Z3</li> <li>• HTC One X</li> <li>• HTC One</li> <li>• Panasonic Eluga</li> </ul>	<p><b>5. J2ME</b></p> <ul style="list-style-type: none"> <li>• Samsung S5230 Tocco Lite/Star/Player One/Avila21</li> <li>• Samsung SGH-X700 NFC22</li> <li>• Samsung D500E17</li> </ul> <p><b>6. BADA</b></p> <ul style="list-style-type: none"> <li>• Samsung Wave 578</li> </ul> <p><b>7. MEEGO</b></p> <ul style="list-style-type: none"> <li>• Nokia N923 24 25</li> </ul> <p><b>8. BLACKBERRY</b></p> <ul style="list-style-type: none"> <li>• BlackBerry Bold 9790 (Codename Bellagio)</li> <li>• BlackBerry Bold 9900/9930 (Codename Dakota/Montana)</li> <li>• BlackBerry Torch 9810/9850/986026 27</li> <li>• BlackBerry Curve 9350/9360/9370/938028</li> <li>• BlackBerry Porsche Design P'9981</li> <li>• BlackBerry Z10</li> <li>• BlackBerry Q10</li> <li>• BlackBerry Q5</li> <li>• BlackBerry Z30</li> <li>• Blackberry Passport</li> </ul> <p><b>9. WINDOWS MOBILE 6.0</b></p>
---	--	--

<ul style="list-style-type: none"> <li>• Samsung Galaxy Mini II (no todas las versiones)</li> <li>• Samsung Galaxy Mega</li> <li>• Samsung Galaxy S3,S4,S5</li> <li>• Nexus S6</li> <li>• Google Nexus S 4G7</li> <li>• Nexus 7</li> <li>• Nexus 4</li> <li>• Nexus 5</li> <li>• Galaxy Nexus by Samsung8</li> <li>• Samsung Galaxy S II (no todas las versiones)9</li> <li>• Samsung Galaxy Fame.</li> <li>• Samsung Galaxy S II Mini (no todas las versiones)</li> <li>• Samsung Galaxy S II plus</li> <li>• Samsung Galaxy S III</li> <li>• Samsung Galaxy S III Mini</li> <li>• Samsung Galaxy S IV Mini</li> <li>• Samsung Galaxy S V</li> <li>• Samsung Galaxy S V Mini</li> <li>• Samsung Galaxy Note )</li> <li>• Samsung Galaxy Note II</li> <li>• Samsung Galaxy Note III</li> <li>• Galaxy Nexus10</li> <li>• HTC Amaze 4G</li> <li>• HTC One (M8)</li> <li>• Huawei Ascend G615 (HONOR 2)</li> </ul>	<ul style="list-style-type: none"> <li>• Zopo ZP998</li> <li>• XIAOMI MI3</li> <li>• XIAOMI MI2A</li> <li>• Simply PAD 720</li> <li>• THL T100s</li> <li>• iNew V3</li> <li>• iNew V8</li> <li>• Kingzone K1 Turbo</li> <li>• Oneplus One12</li> </ul> <p><b>3. PLATAFORMA S40 (NOKIA)</b></p> <ul style="list-style-type: none"> <li>• Nokia 6212 Classic13</li> <li>• Nokia 6131 NFC14</li> <li>• Nokia 6216 Classic15 (Nokia canceló el desarrollo del teléfono en febrero de 2010)16</li> <li>• Nokia 3220 + NFC Shell17</li> <li>• Nokia 5140(i) + NFC Shell18</li> </ul> <p><b>4. SYMBIAN OS (ANNA Y BELLE)</b></p> <ul style="list-style-type: none"> <li>• Nokia 600 (oficialmente cancelado19 )</li> <li>• Nokia 603</li> <li>• Nokia vp</li> <li>• Nokia 700</li> <li>• Nokia 701</li> <li>• Nokia 808 PureView</li> <li>• Nokia C7</li> <li>• Nokia C7, 20 Nokia Astound y las variantes Nokia Oro, con la característica NFC habilitada para comenzar a funcionar con el lanzamiento de Symbian Anna.</li> </ul>	<ul style="list-style-type: none"> <li>• Benq T8017</li> </ul> <p><b>10. WINDOWS PHONE 8</b></p> <ul style="list-style-type: none"> <li>• Nokia Lumia 1020</li> <li>• Nokia Lumia 1520</li> <li>• Nokia Lumia 930</li> <li>• Nokia Lumia 925</li> <li>• Nokia Lumia 920</li> <li>• Nokia Lumia 820</li> <li>• Nokia Lumia 720</li> <li>• Nokia Lumia 620</li> <li>• HTC Windows Phone 8X</li> <li>• Samsung ATIV S</li> </ul> <p><b>11. OTROS</b></p> <ul style="list-style-type: none"> <li>• SAGEM my700X Contactless14</li> <li>• LG 600V contactless14</li> <li>• Motorola L7 (SLVR)17</li> <li>• Sagem Cosyphone29</li> <li>• Sonim XP1301 CORE NFC30</li> <li>• Nintendo Wii U</li> <li>• New Nintendo 3DS/New Nintendo 3DS XL</li> <li>• Play Station 3: La Play Station 3 de serie no lo incorpora, hay que añadir dispositivo RC-S380 de Sony</li> <li>• Zopo ZP998</li> </ul>
--	--	---

## ANEXO B. PROTOCOLO NFC EN LA ISO/IEC 18092

**Tabla 10. Términos y definiciones del protocolo NFC**

<b>MODO DE COMUNICACIÓN ACTIVO</b>
El modo en el que tanto el iniciador y el destino utilizan su propio campo de RF para permitir la comunicación[4].
<b>MODULACIÓN ASK</b>
Modulación por desplazamiento de amplitud, en el que la amplitud de la frecuencia de la portadora se modula de acuerdo con la lógica de los datos que deben transmitirse[4].
<b>DECIMAL CODIFICADO EN BINARIO</b>
(BCD) Sistema de representación de cada uno de los números decimales de 0 a 9 por un código binario de cuatro bits[4]. NOTA Los bits, de izquierda a derecha, son dignos de 8, 4, 2 y 1, respectivamente, en decimal, así que por ejemplo el número 6 en BCD es 0110[4].
<b>COLISIÓN</b>
La transmisión por dos o más objetivos o iniciadores durante el mismo período de tiempo, de tal manera que el iniciador o el destino es incapaz de distinguir de qué objetivo se origina los datos[4].
<b>MARCO</b>
Secuencia de bits de datos y bits de detección de errores opcionales, con delimitadores de marco al inicio y al final[4].
<b>HTHRESHOLD</b>
El valor de umbral para detectar un campo de RF externo[4].
<b>INICIADOR</b>
Generador del campo de RF y el arranque de la comunicación NFCIP-1[4].
<b>MODULACIÓN DE CARGA</b>
Proceso de la amplitud de la modulación de un campo de frecuencia de radio mediante la variación de las propiedades de un circuito resonante colocado dentro

del campo de radiofrecuencia[4].
<b>LSB PRIMERO</b>
El bit menos significativo, lo que indica un sistema de transmisión de datos en serie que envía LSB antes de todos los demás bits[4].
<b>LSB PRIMERO</b>
Byte menos significativo en primer lugar, lo que indica un sistema de transmisión de datos en serie que envía LSB antes de todas las demás bytes[4].
<b>CODIFICACIÓN MANCHESTER</b>
Método de codificación de bits por el que un nivel lógico durante una duración de bit se representa mediante una secuencia de dos estados físicos definidos de un medio de comunicación[4].
<b>ÍNDICE DE MODULACIÓN</b>
Señal de relación de amplitud de [máximo - mínimo] / [+ pico mínimo][4].
<b>MSB PRIMERO</b>
El bit más significativo que indica un sistema de transmisión de datos en serie que envía el bit más significativo antes de todos los demás bits[4].
<b>MSB PRIMERO</b>
Byte de mayor valor que indica un sistema de transmisión de datos en serie que envía el MSB al de los demás bytes[4].
<b>NFCIP-1 DISPOSITIVO</b>
Entidad [ISO / IEC 18092][4].
<b>IDENTIFICADOR DE LA NFC</b>
(NFCIDn) Un número generado al azar utilizado por la secuencia de dispositivos de detección y prevención de colisiones de RF y el único tanto para los modos de comunicación activa y pasiva[4].
<b>NFC-SEC</b>
NFCIP-1 Servicios de Seguridad y Protocolo como se especifica en la norma ISO / IEC 13157-1[4].
<b>MODO DE COMUNICACIÓN PASIVA</b>

Cuando el iniciador está generando el campo de RF y el objetivo responde a un comando iniciador en un esquema de modulación de carga[4].
<b>PREVENCIÓN DE COLISIONES DE RF</b>
(ACRF) Método para detectar la presencia de un campo de RF basado en la frecuencia portadora y el método para detectar y resolver colisiones en nivel de protocolo[4].
<b>SAK</b>
Seleccione reconocer [ISO / IEC 14443-3]. NOTA SAK sustituye a los SEL_RES [ISO / IEC 18092: 2004][4].
<b>DETECCIÓN</b>
Dispositivo NFCIP-1 en el modo de comunicación activo esperando una respuesta a una solicitud que ha enviado en el campo de RF para detectar el inicio de la comunicación para recibir la Solicitud[4].
<b>DETECCIÓN SOLO DISPOSITIVO</b>
(SDD) Algoritmo utilizado por el iniciador para detectar uno de los diversos objetivos fijados en sus campos de radiofrecuencia (anticolisión [ISO / IEC 14443-3])[4].
<b>OBJETIVO</b>
Responde a iniciador de comandos, ya sea usando el esquema de modulación de carga (campo de RF generado por el iniciador) o el uso de la modulación de la auto generada campo de RF[4].
<b>PERÍODO DE TIEMPO</b>
Define el número de ranuras utilizado para prevención de colisiones de RF[4].
<b>EL INTERVALO DE TIEMPO</b>
Método de preparación de una ventana de tiempo cuando un objetivo respuestas, y la asignación e identificación de dos o más canales lógicos[4].
<b>TRANSACCIÓN</b>
Inicialización, el intercambio de datos y el dispositivo de selección[4].

*Adaptada de International Standard. (2013). ISO/IEC 18092. [Tabla].*

## ANEXO C. ESTADO DEL ARTE NFC EN LA PLATAFORMA ANDROID

### APLICACIONES SENCILLAS CON NFC EN GOOGLE PLAY GRATUITAS

**Conexión fácil NFC** La aplicación Conexión fácil NFC permite conectar fácilmente el smartphone con los dispositivos Sony compatibles con NFC (altavoces, auriculares, etc.)[25].

**NFC Tools** NFC Tools es una aplicación que permite leer, escribir o programar tareas sobre sus etiquetas NFC y otros chips RFID compatibles[25].

**NFC Tasks** NFC Tasks es una aplicación adicional que permite la ejecución de las tareas NFC guardadas desde la aplicación NFC Tools [25].

**Trigger** Esta aplicación interactúa con el entorno para configurar automáticamente ajustes del teléfono. Combina funciones de activación y acciones para crear tareas [25].

**NFC Widget** Convierta rápidamente encendido / apagado de la interfaz NFC y ver el estado en su pantalla de inicio, ahorra energía y evita la NFC-lecturas no intencionales [25].

**NFC TagWriter by NXP** NFC TAGwriter por NXP almacena contactos, favoritos, ubicación geográfica, Bluetooth Entrega, SMS, correo, mensajes de texto y muchos más a las etiquetas NFC, así como a los elementos como carteles, tarjetas de visita, relojes y muchos más contiene habilitado con NFC la electrónica[25].

**NFC Smart Tags** Permite a los usuarios que lean y escriban la RFID, NDEF, Mifare Etiquetas y otras etiquetas compatibles NFC almacenando URL, SMS,

números de teléfono para una petición de llamada inmediata, carteles inteligentes o incluso crear tarjetas de visita [25].

**NFC TagInfo by NXP** TagInfo es la segunda aplicación proporcionado a usted por NXP Semiconductors, que utiliza la tecnología Near Field Communication. NXP Semiconductors es el co-inventor de la NFC [25].

**NFC Developer** Crear y escribir mensajes NFC el uso de su móvil y el ordenador juntos en un flujo de trabajo eficaz. Soporta todo tipo de NDEF Registros y tipos de datos (NDEF es el formato nativo de datos NFC en Android) [25].

**NFC ReTag FREE** Reutilización / reciclar protegido contra escritura NFC Tags tales como hoteles clave-tarjetas, tarjetas de acceso, etiquetas de precios, tarjetas de elevación, llaveros, etc[25].

**Share Tether NFC** Share Tether NFC es una aplicación con la que podrás compartir tethering entre dos dispositivos mediante NFC o escaneando un código QR. Simplemente usando Android Beam y conectando tu Nexus 4 y Nexus 7 [25].

**Metro tickets of Moscow** Esta aplicación es un plug-in para aplicaciones llamadas "METRO 24". Esto permite leer entradas del metro de Moscú con NFC en el teléfono. Lectura de datos[25].

**Clonificador de etiquetas NFC** Clonifica etiquetas de NFC y hazlo en una manera divertida, Los modos de clonificación que solo soportan[25]:

- Lee uno, escribe una etiqueta
- Lee uno, escribe varias etiquetas

**NFC Swipe Settings** Al iniciar esta aplicación se obtiene un interruptor NFC que

puede utilizar cada vez que se necesita para cambiar NFC encendido y apagado. Arrastre el botón hacia arriba para encontrar el botón de NFC[25].

**T-money Balance Check (NFC)** Esta es una aplicación no oficial para el control de saldo de la tarjeta T-Money. Ahora también puede utilizar esta aplicación para gestionar sus tarjetas T-Money y los registros de saldo de tarjetas[25].

**NFC TagInfo** Lea la información detallada de etiquetas NFC y tarjetas inteligentes sin contacto.

La aplicación NFC TagInfo lee la información de metadatos y datos de contacto RFID y transponders NFC[25].

**NFC Passport Reader** La NFC Pasaporte Reader Showcase aplicación lee y verifica el chip RFID incrustado en los pasaportes electrónicos y otros documentos de identidad que cumplen con la OACI (pasaporte electrónico, o, en la OACI Doc 9303 terminología, Documentos de viaje de lectura: MRTD)[25].

**Octopus** Se trata de una aplicación oficial desarrollada por Octopus Cards Limited y es una aplicación gratuita para NFC (Near Field Communication) habilitado Android dispositivos móviles y aplicable a Pulpo con número Pulpo 9 dígitos (incluyendo ocho dígitos más un dígito dentro del soporte) [25].

**NFC Reader** NFC Reader puede leer sin contacto de tarjeta IC utilizar hardware NFC. Soporte ISO7816-4, FeliCa compatible monedero electrónico / e-efectivo, Se puede leer en cuenta la validez, el equilibrio fuera de línea y los registros de transacciones de estas tarjetas si hay restricción de lectura [25].

**NFC Actions** NFC Actions es una aplicación android que permite escribir Acciones Quick y Cloud (y Flomio Zapps) a tags NFC. Después que un tag es

escrito puede ser escaneado para que rápidamente ejecute acciones como enviar un email, abrir una url, hacer un checkin con Foursquare, etc[25].

**NFC File Transfer** Enviar fotos, música, videos, archivos a través de Android Beam. Sólo necesita NFC habilitado devices. Super fácil y sencilla aplicación para administrar archivos y compartirlos con tus amigos y familiares a través de NFC[25].

**AnyTAG NFC Launcher** Automatice su teléfono mediante NFC con AnyTAG NFC. AnyTAG NFC Launcher es un lanzador de aplicaciones de tareas que permite iniciar las tareas mediante el escaneo de etiquetas NFC[25].

**NFC tag Matic** Puede enviar un correo electrónico o hacer una llamada telefónica. Cualquier número de cosas suceden con sólo un toque, una nueva experiencia de usuario sentido [25].

**NFC On** La aplicación NFC On de Orange, compatible con cualquier operador móvil, te permite empezar a disfrutar del mundo NFC. Con esta aplicación podrás leer y escribir etiquetas NFC para configurar tu teléfono o realizar acciones cuando la etiqueta es leída, conectarte a otros dispositivos NFC como altavoces o auriculares y compartir archivos con otros móviles NFC [25].

**SmartPassLock NFC** "SmartPassLock NFC" es una aplicación de la pantalla de bloqueo de seguridad utilizando la tecnología NFC (Near Field Communication). Usted puede proteger el dispositivo que está utilizando a un tercero malicioso.

**SuperBeam | WiFi Direct Share** Superbeam 4.0 es la más fácil y rápida forma de compartir archivos de gran tamaño entre sus dispositivos utilizando WiFi directa. Los dispositivos pueden ser emparejados con códigos QR (con el escáner de código QR incluido), NFC o manual de reparto (PRO) [25].

**Tapkey** Tapkey convierte tu smartphone en una clave de super-seguro para cualquier Tapkey habilitado bloqueo fácilmente. Utilizar el teléfono para desbloquear casas habilitadas para Tapkey, habitaciones de hotel, coches, coches de alquiler, oficinas, y más. Tapkey convierte tu teléfono en un llavero digital inteligente[25].

**ABA NFC** ABA NFC es una aplicación en fase beta de lectura y escritura de etiquetas NFC desarrollada por ABAMobile. Seleccionada por AndroidCentral.com como aplicación de la semana[25].

**PassWallet - Passbook + NFC** "Recopila todas tus entradas y cupones de Passbook, desde tarjetas de embarque hasta las de fidelización de cafeterías en una sola aplicación fácil de usar [25].

**NFC EXPRESS** NFC expreso Desktops PC proporciona acceso conveniente y copia de seguridad de archivos multimedia a través de la tecnología NFC [25].

**Microsoft Tag, QR & NFC Reader** La aplicación Microsoft Tag le permite conectarse instantáneamente a un nuevo mundo de información y entretenimiento. Esta última versión ahora lee los códigos de barras de la etiqueta de Microsoft y los códigos QR, así como puntos de contacto NFC para aquellos con dispositivos habilitados para NFC[25].

**WifiTap WiFi NFC** Con esta aplicación instalada que sólo puede mantener su dispositivo móvil cerca de un NFC Etiqueta tu amigo colocó en su entrada hogares o en cualquier otro lugar en el interior[25].

**NFC Profile** Con NFC perfil Puede cambiar la configuración del sistema, como la conectividad WiFi y se apaga con un simple toque de una etiqueta NFC [25].

**Send! | File Transfer** La manera más fácil de enviar fotos, música, videos o cualquier archivo fuera de línea para otros dispositivos Android[25].

**Samsung NFC Connection** Apoyo a los dispositivos móviles con tecnología NFC (Tag): Galaxy S2, S3, Nota Galaxy Galaxy (no compatible dispositivo: Galaxy Nota 2 y así sucesivamente)  
Apoyo operativo Android: Gingerbread, Icecream Sandwich[25].

**Easy NFC Writer** Fácil NFC Writer es una aplicación muy sencilla e intuitiva que le permite escribir y codificar NFC Etiquetas de ningún tipo. Con Easy NFC escritor, puede codificar una etiqueta NFC con una dirección web, un contacto V-Card, o texto simple[25].

**TK-FNS040BK** Se trata de un servicio para conectarse a (TK-FNS040BK) teclado NFC.

Además, se mostrará el icono de esta aplicación en la lista de aplicaciones en la pantalla de inicio

**eCLOWN: an NFC passport tool** Es una utilidad de Android pasaportes electrónicos para los teléfonos con NFC que permite leer y clonar el contenido de chip de su pasaporte electrónico[25].

**Badge NFC** La aplicación que le permitirá administrar su propio sistema de tarjetas (reloj de entrada / salida), utilizando el chip NFC del teléfono y cada etiqueta NFC se puede encontrar en el mercado [25].

**NFC Launcher** NFC Launcher es una solución basada en la nube para la gestión de campañas de marketing NFC, lo que permite Brands para manejar sus etiquetas, carteles inteligentes y todas las comunicaciones realizadas a través de NFC [25].

## ANEXO D. ESPECIFICACIONES DE SMART RING JAKCOM

### Descripción:

- Tipo: Anillo inteligente NFC
- Material: aleación de titanio, oro rosa 18k, de epoxy de cristal cerámico
- Color: negro
- Para Android y Windows Phones con función NFC
- Posee dos chips de alta velocidad NFC de doble núcleo

### Descripción de cada chip NFC:

La descripción de anillo se hace mediante la lectura de sus componentes internos a través de la aplicación NFC Tools.

#### Chip 1:

- Tipo de etiqueta: ISO 14443-3A, NXP MIFARE Ultralight (Ultralight C) – NTAG213
- Tecnologías posibles: NfcA, MifareUltralight, Ndef
- Número de serie: 04:48:38:9A:2C:49:81
- Formato de los datos: NFC Forum Type 2
- ATQA: 0x0044
- SAK: 0x00
- Tamaño: 142 Bytes

#### Chip 2:

- Tipo de etiqueta: ISO 14443-3A, NXP MIFARE Classic 1k Not Supported
- Tecnologías posibles: NfcA
- Número de serie: B9:58:2E:02
- ATQA: 0x0004
- SAK: 0x08
- Tamaño: 1 KBytes

## ANEXO E. CODIGO DEL ARCHIVO MANIFESTS UIS NFC

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.memo.uisnfc">
    <!-- agrega permiso de internet -->
    <uses-permission android:name="android.permission.INTERNET" />
    <!-- agrega permiso de camara -->
    <uses-feature
        android:name="android.hardware.camera"
        android:required="true" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <!-- agrega permiso de nfc -->
    <uses-permission android:name="android.permission.NFC" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
    <uses-feature
        android:name="android.hardware.nfc"
        android:required="true" />
    <!-- MAIN ACTIVITY -->
    <activity
        android:name=".MainActivity"
        android:label="@string/app_name_menu"
        android:theme="@style/AppTheme.NoActionBar"
        android:windowSoftInputMode="adjustPan"
        android:configChanges="screenSize|orientation">
        <!-- filtro menu -->
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.DEFAULT" />
        </intent-filter>
        <!-- filtros NFC -->
        <intent-filter>
            <action android:name="android.nfc.action.NDEF_DISCOVERED"/>
            <category android:name="android.intent.category.DEFAULT"/>
            <data android:scheme="tel" />
        </intent-filter>
        <intent-filter>
            <action android:name="android.nfc.action.NDEF_DISCOVERED" />
            <category android:name="android.intent.category.DEFAULT" />
            <data android:mimeType="text/plain"/>
        </intent-filter>
        <!-- filtro filtro nfc actividad mensaje -->
    </activity>
</application>
</manifest>
```