DISEÑO E IMPLEMENTACIÓN DE UNA APLICACIÓN PROTOTIPO DE LECTURA Y ESCRITURA PARA DISPOSITIVOS ANDROID COMPATIBLE CON TECNOLOGÍA NFC QUE PERMITA EL MANEJO DEL INVENTARIO DE LIBROS EN LAS BIBLIOTECAS.



UNIVERSIDAD DISTRITAL
FRANCISCO JOSE DE CALDAS

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS
FACULTAD DE INGENIERÍA
PROYECTO CURRICULAR DE INGENIERÍA ELECTRÓNICA
BOGOTÁ
2015

DISEÑO E IMPLEMENTACIÓN DE UNA APLICACIÓN PROTOTIPO DE LECTURA Y ESCRITURA PARA DISPOSITIVOS ANDROID COMPATIBLE CON TECNOLOGÍA NFC QUE PERMITA EL MANEJO DEL INVENTARIO DE LIBROS EN LAS BIBLIOTECAS.

NELSON EDUARDO HUERTAS LOPEZ
DIANA MARCELA BURBANO ERAZO

Trabajo presentado como requisito para optar al título de:

Ingeniero Electrónico

Director:

msc. Roberto Ferro Escobar

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS
FACULTAD DE INGENIERÍA
PROYECTO CURRICULAR DE INGENIERÍA ELECTRÓNICA
BOGOTÁ
2015

Agradecimientos

Gracias a mis padres y a mi hermana por su incondicional apoyo, fueron mi motor y mi fuerza para sacar adelante esta hermosa carrera.

Diana Marcela Burbano Erazo

Gracias a todos por su incondicional apoyo en ese proceso, especialmente a mi hijo Juan David quien con su compresión y cariño me dio la fuerza suficiente par a la culminación de mi carrera.

Nelson Eduardo Huertas López

Resumen

En este trabajo se presenta el diseño e implementación de una aplicación para dispositivos Android que permita la realización de inventarios utilizando la tecnología NFC, tecnología que está presente en gran parte de los dispositivos móviles, lo que facilita la implementación de diversas aplicaciones que conllevan a optimizar tareas de la vida diaria y dar soluciones a otras necesidades. En este caso se utilizara la tecnología NFC para reemplazar los códigos de barras que se usan en la realización de inventarios, para esto se implementara la lectura y escritura de los Tags NFC.

Para llevar a cabo la lectura y escritura de los Tags NFC se implementará una aplicación en Android Studio, el cual permite la visualización de contenido a través de activitys, y utiliza como compilador el lenguaje java (JDK) facilitando la compatibilidad con diferentes sistemas operativos y a su vez permite acceder y controlar la memoria de los tags, y la creación de los mensajes NDEFRecords, que son las unidades de almacenamiento de la información especificados por el NFC Fórum.

La aplicación se relacionara con un sistema de almacenamiento de datos que contenga la información de los libros, de los usuarios y de los estudiantes, para poder articularlos dentro de un sistema que permita la realización de inventarios.

Palabras Clave:

NFC, modulación ASK, codificación, Tags, NDFERecords, Activity, JDK, NFC fórum

ÍNDICE GENERAL

ÍNDICE DE FIGURAS **8**

ÍNDICE DE TABLAS **10**

INTRODUCCION **11**

CAPÍTULO 1 ..... **12**

1.1 IDENTIFICACION DEL PROBLEMA 12
1.2 JUSTIFICACION 13
1.3 ANTECEDENTES 14
1.4 OBJETIVOS 17
1.4.1 Objetivo General *17*
1.4.2 Objetivos Específicos *17*
1.5 METODOLOGIA 18

CAPÍTULO 2 ..... **20**

2 MARCO TEORICO **20**

2.1 NEAR FIELD COMMUNICATION (NFC) 20
2.2 MODULACIÓN ASK 22
2.3 CODIFICACIÓN MANCHESTER 24
2.4 CÓDIGO MANCHESTER 26
2.5 FASES PARA ESTABLECER LA COMUNICACIÓN NFC 27
2.6 ETIQUETAS NFC 28
2.7 ARQUITECTURAS DE FUNCIONAMIENTO 29
2.7.1 Punto a punto *29*
2.7.2 Lectura-escritura *30*
2.7.3 Emulación de tarjeta *30*
2.8 PROTOCOLO DE CONTROL DE ENLACE LÓGICO 31
2.9 ESPECIFICACIONES. 32
2.9.1 NFC Data Exchange Format (NDEF). *32*
2.9.2 NFC Record Type Definition: *32*
2.9.3 NFC Uniform Resource Identifier Service Record Type Definition *33*
2.9.4 NFC Text Record Type Definition *33*
2.9.5 NFC Smart Poster Record Type Specification *34*
2.10 APLICACIONES NFC 34
2.11 VENTAJAS 36

CAPÍTULO 3 ..... **37**

3 ESPECIFICACIONES TÉCNICAS DE LAS TAGS NFC **37**

3.1 ASPECTOS QUE SE DEBEN TENER EN CUENTA 37
3.1.1 Espacio físico disponible para su colocación. *37*
3.1.2 Tamaño de la memoria. *37*
3.1.3 Ambiente donde se colocara la etiqueta. *37*
3.2 TABLA COMPARATIVA 39

3.3 INFORMACIÓN DETALLADA DE LAS ESPECIFICACIONES TÉCNICAS DE LOS TAGS NFC *41*
3.3.1 STANDARD CARD IC MF1 IC S70 ..... *41*
3.3.2 MF3ICD21, ICD41, MF3ICD81[33] *42*
3.4 TAGS NFC DISPONIBLES EN EL GRUPO LÍDER DE LA UNIVERSIDAD DISTRITAL 44
3.4.1 NXP MIFARE NTAG 203 ..... *44*
3.4.2 NXP MIFARE CLASIC 1K *46*
3.4.3 NXP MIFARE CLASIC 4K *48*
3.5 TAG SELECCIONADO PARA LA APLICACIÓN 51

CAPITULO 4 ..... **52**

4 DISEÑO E IMPLEMENTACIÓN DE LA APLICACIÓN **52**

4.1 ANALISIS DE REQUERIMIENTOS 52
4.2 IDENTIFICACION DE ACTORES. 53
4.3 DIGRAMA DE CASOS DE USO 53
4.4 DIAGRAMA DE FLUJO 54
4.5 DIGRAMA DE CLASES 55
4.6 DISEÑO D E INTERFAZ DE LA APLICACIÓN 56
4.6.1 Vista principal *56*
4.6.2 Leer tag *57*
4.6.3 Escribir tag *57*
4.6.4 Opción inventario *58*
4.7 IMPLEMENTACIÓN DE LA APLICACIÓN 58
4.7.1 REQUERIMIENTOS DE HARDWARE SOFTWARE *58*
OTRAS CARACTERÍSTICAS DE ANDROID STUDIO 61
Windows *61*
ARCHIVO STRINGS.XML 68
LAYOUT[44] 70
/APP/BUILD.GRADLE [44] 71
4.8 IMPLEMENTACIÓN DE LA INTERFAZ GRÁFICA 71
4.9 PROGRAMACION DE LA APLICACIÓN 73
4.9.1 Archivo Manifest.xml *73*
4.9.2 Pantalla principal *74*
4.9.3 Lectura de tags NFC *74*
4.9.4 Escritura de tags NFC *77*
4.9.5 Pantalla de login *79*
4.9.6 Realización del inventario *80*
4.10 DISEÑO DE LA BASE DE DATOS 81
4.11 IMPLEMENTACION DE LA BASE DE DATOS 82

CAPÍTULO 5 ..... **83**

5 PRUEBAS Y ANÁLISIS DE RESULTADOS **83**

5.1 PRUEBAS 83
5.1.1 Prueba de escritura *83*
5.1.2 Prueba de lectura *85*
5.1.3 Prueba de conexión *86*
5.2 ANÁLISIS DE RESULTADOS 88

CAPÍTULO 6 ..... **89**

6 CONCLUSIONES Y TRABAJOS FUTUROS....89

6.1 CONCLUSIONES....89

6.1 TRABAJOS FUTUROS....90

ANEXO....91

ESTUDIO PARA EL USO DE LA TECNOLOGÍA NFC EN LA BIBLIOTECA DE LA UNIVERSIDAD DISTRITAL....91

BIBLIOGRAFÍA....96

Índice de figuras

Figura 1. Organización encargada de regular los estándares de NFC [7] 20

Figura 2. Logotipos oficiales NFC [9] 21

Figura 3. Modo de funcionamiento: pasivo y Activo [2]. 21

Figura 4. Señal de salida con Modulación ASK[12]. 23

Figura 5. Señal de salida de la demodulación ASK[12] 24

Figura 6.Codificacion Manchester y Manchester diferencial[45]. 25

Figura 7.Presentación de los Tags NFC [7] 28

Figura 8.Arquitectura punto a punto [7] 30

Figura 9. Arquitectura de lectura-escritura [7]. 30

Figura 10. Arquitectura de emulación de tarjeta. [7]. 31

Figura 11. Arquitecturas de funcionamiento NFC [7]. 31

Figura 12. Formato de un mensaje NDEF [7]. 32

Figura 13. Estructura del mensaje NDEF [7]. 33

Figura 14. Aplicaciones móviles [18]. 34

Figura 15. Organización de la memoria para el Tag IC MF1 IC S70[32]. 42

Figura 16. NXP NTAG 203[imagen propia] 44

Figura 17. Interacción entre el Tag y un dispositivo NFC habilitado [34]. 45

Figura 18. MIFARE CLASIC 1K [imagen propia]. 46

Figura 19. Interacción entre el Tag MIFARE CLASIC 1K y un dispositivo NFC habilitado [35]. 46

Figura 20. Organización de la memoria para el Tag MIFARE CLASIC 1K [35]. 47

Figura 21. Interacción entre el Tag MIFARE CLASIC 4K y un dispositivo NFC habilitado [36]. 49

Figura 22. Organización de la memoria para el Tag MIFARE CLASIC 4K [36]. 50

Figura 23. Diagrama de casos de uso 53

Figura 24. Diagrama de Flujo 54
Figura 25. Diagrama de clases 55
Figura 26. Vista principal 56
Figura 27. Vista de leer Tag 57
Figura 28. Vista de escribir Tag 57
Figura 29. Pantalla de login Figura 30. Pantalla de inventario 58
Figura 31. Editor de código inteligente[39] 59
Figura 32. Plantillas de código[39] 60
Figura 33. Soporte para múltiples pantallas[39]. 60
Figura 34. Diferentes dispositivo virtuales[39] 61
Figura 35. Versiones de JDK disponibles para descargar en el sitio oficial de Oracle [40]. 63
Figura 36. Vista de la consola de Windows si el java está configurado 64
Figura 37. Archivo manifest.xml 66
Figura 38. Ciclo de vida de una de una activity [30] 68
Figura 39. Vista del archivo string.xml de la aplicación. 69
Figura 40. Archivo layout de la aplicación. 70
Figura 41. Interfaz principal Figura 42. Interfaz de lectura 72
Figura 43. Interfaz de escritura Figura 44. Interfaz de login 72
Figura 45. Modelo relacional de la base de datos 83
Figura 46. Sistema de información bibliográfico de la UDFJC. 91
Figura 47. Vista Completa de un registro utilizando catalogación estándar. 92
Figura 48.Busquedas implementadas en el sistema de información bibliográfico de la UDFJC. 94

Índice de tablas

Tabla 1. Metodóloga para el desarrollo del proyecto. 19

Tabla 2.Relación de velocidades y codificaciones [12] 22

Tabla 3. Codificación Manchester[15] 26

Tabla 4.Tipos de Tags compatibles según las especificaciones de NFC Fórum [25] 29

Tabla 5.Comparación de los tags disponibles en el mercado y compatibles con el NFC fórum [24][30]–[33]. 39

Tabla 6. Análisis de requerimientos. 52

Tabla 7. Campos de la tabla libros 81

Tabla 8. Campos de la tabla usuarios 81

Tabla 9. Campos de la tabla autor 81

Tabla 10 Campos de la tabla préstamos 82

Tabla 11. Campos de la tabla estudiantes. 82

Tabla 12. Comparación de costos entre la tecnología de código de barras y la tecnología NFC. 95

INTRODUCCION

La tecnología Near Field Communication (NFC) es una tecnología inalámbrica de comunicación que funciona en la banda de 13,56 Mhz y es de corto alcance. Dicha tecnología permite el intercambio de datos entre dispositivos. En la actualidad existen etiquetas y pegatinas que permiten almacenar información y realizar tareas en los dispositivos que se acercan a ellas. Las tareas más básicas permiten abrir una URL (página web), mostrar un texto o una imagen, las más sofisticadas llegan a hacer tareas complejas como la conexión de periféricos, transacciones electrónicas, identificación segura, aspectos de la vida diaria que hasta ahora no se podían realizar con otras tecnologías [1], [2].

Uno de los campos promisorio para el uso de esta tecnología es el manejo y control de inventarios, ya que el crecimiento constante de información y la complejidad de la misma, requiere de aplicaciones tecnológicas eficientes y con gran flexibilidad a la hora de implementar nuevos cambios, con el fin de obtener un mejor desempeño y facilitar la labor de las personas que tienen a cargo dicho manejo.

Una área que se verá beneficiada con esta tecnología es la relacionada con el manejo de los inventarios en las bibliotecas, ya que la gran cantidad de textos que allí se manejan hace que el papel del bibliotecario se vea aumentado, al tener que llevar a cabo tareas de selección y síntesis de contenidos de interés para el usuario. Son muchas las estrategias que los bibliotecarios y expertos en sistemas de control de la información han puesto en práctica para catalogar, administrar y suministrar esa cantidad de información disponible, y en ese proceso ha sido constante la incorporación de los nuevos recursos tecnológicos [3].

Teniendo en cuenta lo expuesto anteriormente, se presentará en este documento una solución a dicho problema utilizando la tecnología NFC y tomando como referencia las bibliotecas, donde la identificación de los libros se realiza a través de código de barras lineales, tecnología que ha prevalecido por mucho tiempo pero que presenta algunas debilidades relacionadas con su durabilidad y flexibilidad ya que su lectura se realiza a través de un equipo específico, el cual les imprime un carácter de rigido a las aplicaciones.

Capítulo 1

1.1 IDENTIFICACION DEL PROBLEMA

Después de llevar a cabo un análisis sobre los procedimientos que emplean las bibliotecas para la realización de inventarios, se determinaron los principales problemas a resolver, iniciando con el deterioro que presentan las etiquetas de código de barras, debido a factores como lo son la manipulación frecuente y la exposición a ambientes con demasiado polvo, lo cual genera errores de lectura impidiendo que el personal a cargo de dicha labor la realice de manera oportuna. Por lo cual se llega a la pregunta. ¿Puede la implementación de un sistema con tecnología NFC resolver este problema?

1.2 JUSTIFICACION

El proceso que se tiene en las bibliotecas para llevar a cabo la realización y actualización del inventario, parte de la lectura a través de un lector óptico de un código de barras impreso en una etiqueta y que previamente se ha adherido al libro, esta etiqueta es colocada en la parte interior de la cubierta del libro con el propósito de que este le sirva de protección mientras se realizan actividades donde el libro está en fricción con otras superficies ya que si la etiqueta estuviera directamente al exterior podría salir afectada. La puesta de la etiqueta en dicho sitio exige que el personal encargado de realizar el inventario tenga que abrir el libro para poder leer el código de barras, lo cual implica un tiempo mayor, dando lugar a que se lleve una o dos veces por año.

El diseño e implementación de una aplicación que utilice una tecnología más robusta que el código de barras se hace necesario para brindar una mejora en la realización de dicho proceso, es por eso que se utilizará la tecnología NFC para resolver estos problemas ya que esta permite ser leída con cualquier dispositivo que tenga un lector NFC y como es una tecnología de campo nos permite ser leída sin necesidad de abrir el libro lo cual se traduce en un ahorro de tiempo.

.

Cabe destacar, que una razón por la cual se busca la realización de este proyecto, es la de estar un paso más cerca de las tecnologías utilizadas por los diferentes países desarrollados, y que pueden ser aplicadas para solucionar problemas cotidianos.

1.3 ANTECEDENTES

Desde hace algunos años, la tecnología NFC ha abierto el camino de nuevas posibilidades en el ámbito de las aplicaciones móviles. Si bien, los tipos de uso que pueden realizarse están bien definidos, aún hay lugar para las nuevas ideas que puedan convertir el NFC en una tecnología que cumpla con unas necesidades que hasta ahora no podían cubrirse de esta manera tan sencilla, segura y a la vez interoperable con otros datos personales que existen dentro de un teléfono móvil [4].

La tecnología NFC permite una conectividad cercana, de unos pocos centímetros, para el intercambio de datos entre dispositivos. Su funcionamiento requiere dos elementos: una etiqueta NFC previamente programada con la instrucción de la tarea a realizar y un dispositivo que incorpore dicha tecnología, y que tenga instalada una aplicación lectora, posteriormente éste debe situarse cerca de la etiqueta para que se ejecute la instrucción [1].

Uno de los escenarios aplicados con la tecnología NFC orientado a las bibliotecas es el proyecto desarrollado en Klagenfurt, en el Sur de Austria, ciudad con unos 90.000 habitantes y que carece de biblioteca pública, donde se han situado tags NFC en lugares públicos y debidamente identificados para que los usuarios con sólo acercar su teléfono móvil puedan descargar títulos que no estén protegidos por las leyes de propiedad intelectual o están exentos de derechos de autor, la mayoría de los títulos están englobados en otro proyecto denominado Proyecto Gutenberg 3 [3] relacionado con este, y con el fin de promocionar autores de la propia región. En este caso NFC actúa como medio para abrir el correspondiente enlace de comunicaciones vía Wi-Fi o 3G y posibilite la descarga del contenido del título escogido. Es habitual ver lugares identificados con etiquetas amarillas invitando a los usuarios a descargar contenidos, los lugares están relacionados con el autor de la obra o con el contenido de la misma, así puede descargarse por ejemplo el título El asesino, del autor austriaco Arthur Schnitzler, desde el tag situado en la comisaría de la ciudad, o bien Sueño de una noche de verano de Shakespeare desde la playa. El proyecto, denominado Projekt Ingeborg4 en honor a la poeta y escritora,

nacida en la ciudad de Klagenfurt, Ingeborg Bachmann, dispone de aplicaciones específicas para las plataformas más extendidas de smartphones con el fin de ofrecer información adicional relacionada con este servicio, como mapas con los lugares donde poder encontrar otras etiquetas, librerías donde adquirir el título descargado, información del autor, formato de descarga, etc [3], [5].

Otra interesante experiencia creada en torno a NFC y la biblioteca se ubica en New York, el nombre del proyecto Underground Library 5 [3] deja entrever la iniciativa. Underground Library5 es el resultado de una iniciativa que trata de fomentar y acercar la lectura a los usuarios del metro cuando las ventas en libros comenzaron a descender. El proyecto fue desarrollado por alumnos de la Miami Ad School. En este caso NFC no es el medio para conectarse a ninguna web de descargas, debido a que en el metro de New York, y prácticamente en el de todas las ciudades, la cobertura con las redes públicas de datos es intermitente y se generan cortes en las comunicaciones, lo que originará que las descargas se interrumpieran[6].

En el metro se dispone de posters que simulan una librería con unos cuantos ejemplares, el usuario pasa su móvil por el mismo y se ofrece un catálogo de los best seller actuales, ofreciendo una imagen de los libros disponibles, se escoge el título deseado y mediante NFC se descargan las diez primeras páginas del libro seleccionado, amenizando, con su lectura, el trayecto a los usuarios del metro. Una vez que el usuario llega a su parada y sale del metro, el usuario podrá ver en el Smartphone, en un plano de la ciudad, las librerías más cercanas donde poder adquirir un ejemplar del libro escogido[3].

Por otro lado La Biblioteca Pública de Hanno, en la prefectura de Saitama (Japón) ha implantado la tecnología NFC (Near Field Communication) en sus instalaciones. En la web de la empresa Fujitsu, que ha colaborado en este proyecto, se explica cómo la proximidad del *smartphone* con las etiquetas NFC, que en este caso se ha colocado en varias estanterías, fuentes de información, como la Wikipedia o varias bases de datos. De esta manera se puede ampliar información sobre los libros de la colección, lo que ayuda al lector a tomar una decisión a la hora de escoger sus lecturas en el momento de la elección, en la biblioteca. Este portal web, cuyo diseño está optimizado para dispositivos móviles, también permite efectuar reservas de libros y publicar reseñas [1], [7].

El proyecto Karirutatchi así se denomina se ha implementado por primera vez en esta biblioteca con unas cien etiquetas NFC iniciales que pretenden aumentarse más adelante. En un futuro está previsto que se extienda este proyecto a quinientas bibliotecas de todo el país [3].

Otros posibles usos de la tecnología NFC en bibliotecas son la descarga de libros, como proponen el proyecto Ingebord y la propuesta Library Underground una idea de un grupo de estudiantes de llevar la biblioteca al metro y poder descargar desde él libros electrónicos a modo de muestra, el auto préstamo o sustituyendo al carnet de la biblioteca. Estos dos últimos sí podrían justificar el uso de la tecnología NFC, que agilizaría los procesos [4].

Gracias a los dispositivos móviles y la conexión a internet cada vez tenemos más medios a nuestro alcance para facilitar a los usuarios el acceso a la información desde la biblioteca. Es el momento de encontrar la forma de explotar esos medios y buscar soluciones óptimas, al menor coste posible, que realmente funcionen y sean útiles y eficaces para los usuarios. Pero antes que nada debemos conocer cada tecnología y saber aplicarla a cada caso concreto. De ello dependerá su éxito, y también el de nuestras bibliotecas en la era digital [3].

1.4 OBJETIVOS

1.4.1 Objetivo General

Diseñar e implementar una aplicación prototipo de lectura y escritura para dispositivos Android compatible con tecnología NFC que permita el manejo del inventario de libros en las bibliotecas.

1.4.2 Objetivos Específicos

1. Realizar un estudio de las especificaciones técnicas de marcas de tags NFC disponibles en el mercado actualmente.

2. Diseñar e implementar la interfaz gráfica de la aplicación utilizando la plataforma Android Studio.

3. Realizar un código que nos permita la lectura y escritura de los Tags NFC.

4. Diseñar e implementar una base de datos que permita el almacenamiento de información de los libros.

5. Implementar un algoritmo que permita la conexión de la aplicación a la base de datos.

6. Realizar una prueba piloto de la aplicación.

1.5 METODOLOGIA

En el diseño e implementación de una aplicación prototipo de lectura y escritura para dispositivos Android compatible con tecnología NFC que permita el manejo del inventario de libros en las bibliotecas, es necesario tomar cada uno de los objetivos y dividirlo en fases de trabajo.

Fase 1: Investigación, análisis y selección del tag
Fase 2: Diseño de la aplicación
Fase 3: Diseño de la interfaz grafica
Fase 4: Diseño computacional del sistema de almacenamiento
Fase 5: Prueba de la aplicación
Fase 6: Análisis de resultados

En la siguiente tabla se puede apreciar las fases, junto con los objetivos de cada una de ellas y las actividades a realizar.

Fase	Objetivo por fase	Actividades	Resultado
Investigación, análisis y decisión	Realizar un estudio de las especificaciones técnicas de los Tags disponibles en el mercado actualmente	Recopilación y comparación de los diferentes Tags NFC	Selección del Tag NFC con el que se va a desarrollar el proyecto
Diseño de la aplicación	Diseñar una aplicación que permita la lectura y escritura de los Tags NFC para dispositivos Android	Implementación de una aplicación prototipo de lectura y escritura para dispositivos Android	Aplicación prototipo de lectura y escritura para dispositivos Android compatible con

		Adaptar la aplicación de lectura y escritura de Tags para realizar inventario en bibliotecas	tecnología NFC que permita el manejo del inventario de libros en las bibliotecas.
Diseño computacional del sistema de almacenamiento	Diseñar e implementar una base de datos que permita el almacenamiento de información de los libros para realizar una prueba piloto en la biblioteca de la Universidad Distrital Francisco José De Caldas	Análisis de requerimientos para el sistema de información. Implementación de la Base de Datos	Sistema de almacenamiento con los campos necesarios para realizar inventarios.
Prueba de la aplicación	Verificar el correcto funcionamiento de los modos de lectura y escritura	Verificación del funcionamiento del dispositivo lector-escritor en conjunto con la etiqueta o tarjeta Realizar diferentes pruebas de lectura y escritura	Documentar las pruebas
Realización de la prueba piloto	Verificar la lectura y escritura de un Tag NFC con los datos de identificación del libro	Realizar la prueba de lectura y escritura de 15 libros en la Biblioteca de la Universidad Distrital	Documentar los resultados del desempeño de la aplicación

Tabla 1. Metodóloga para el desarrollo del proyecto.

Capítulo 2

2 MARCO TEORICO

2.1 NEAR FIELD COMMUNICATION (NFC)

Near field communication (NFC) es una tecnología de comunicación inalámbrica, de corto alcance y alta frecuencia que permite el intercambio de datos entre dispositivos. Esta tecnología inalámbrica se basa en una plataforma abierta pensada desde el inicio para teléfonos y dispositivos móviles.

La tecnología NFC comenzó a desarrollarse en el año 2002 en una acción conjunta de Philips y Sony, con el fin de conseguir un protocolo compatible con las tecnologías sin contactos propietarias existentes en el mercado: Mifare de Philips y FeliCa de Sony. Finalmente, NFC fue aprobado como el estándar ISO 18092 en diciembre de 2003 y posteriormente, en marzo de 2004, Philips, Sony y Nokia formaron el NFC Forum para avanzar en el desarrollo de las especificaciones NFC y velar por su interoperabilidad [2],[8].
La organización encargada de regular y determinar los estándares de NFC nace en el 2004 cuando se crea NFC-Forum, pero es hasta 2006 cuando sale a la luz la primera especificación técnica de dicha tecnología [3].



Figura 1. Organización encargada de regular los estándares de NFC [7]

Se reconocen dos logotipos oficiales:



Figura 2. Logotipos oficiales NFC [9]

NFC es un estándar de comunicación compatible con ISO 18000-3, estándar internacional para todos los dispositivos que se comunican de forma inalámbrica en la frecuencia de 13,56 MHz, que explica cómo un dispositivo (Interrogador) y la etiqueta NFC que se está leyendo debe comunicarse unos con otros. Además NFC está gobernado bajo IEC y la especificación ISO 13157 [4],[6].

El NFC está pensado para transferencia instantánea de información sin emparejamiento previo entre dispositivos, pero sin embargo esta comunicación debe realizarse desde una distancia reducida ya que el alcance de esta tecnología es limitado a un máximo de 10 cm [9][8][10].
Esta tecnología soporta dos modos de funcionamiento:

- **Activo:** Ambos dispositivos transmiten y reciben bidireccionalmente.
- **Pasivo:** Un dispositivo genera campos electromagnéticos y el otro se aprovecha de la carga para transmitir.



Figura 3. Modo de funcionamiento: pasivo y Activo [2].

La tecnología NFC deriva de las etiquetas RFID, soporta transferencia de datos a una tasa relativamente baja de 424 kb/s, aunque 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, además toma menos de 0.2 segundos al momento de establecer la conexión, su consumo de energía durante la lectura de datos está por debajo de los 15 mA aunque puede aumentar al momento de la escritura [2], [4], [11]. La comunicación se realiza utilizando Modulación ASK y codificación Manchester.

Velocidad	Modo Activo	Modo Pasivo
424 kbps	Codificación: Manchester Modulación. ASK Índice modulación: 0,1	Codificación: Manchester Modulación. ASK Índice modulación: 0,1
212 kbps	Codificación: Manchester Modulación. ASK Índice modulación: 0,1	Codificación: Manchester Modulación. ASK Índice modulación: 0,1
106 kbps	Codificación: Manchester Modulación. ASK Índice modulación: 0,1	Codificación: Manchester Modulación. ASK Índice modulación: 0,1

Tabla 2.Relación de velocidades y codificaciones [12]

2.2 MODULACIÓN ASK

Para la transmisión de datos digitales, existen principalmente tres métodos de modulación que permiten alterar el ancho de banda sobre el cual será enviada la información. Estos tres métodos son muy empleados debido a su relativa sencillez y a que son ideales para la transmisión de datos digitales, ellos son, el ASK (Amplitude Shift Keying), FSK (Frequency Shift Keying) y PSK (Phase Shift Keying)[12].

En la modulación ASK que es el método que nos interesa, la amplitud de la señal de salida se rige por la siguiente ecuación.

$$v_{AM}(t) = [1 + v_m(t)] * \left[\frac{A}{2} * cos(2\pi f_C t)\right]$$

Dónde:

$v_{AM}(t)$: Salida del modulador.

$v_m(t)$: Señal digital moduladora.

A: Amplitud de la señal portadora.

f_C: Frecuencia de la señal portadora.

Tomando en cuenta esta ecuación, si $v_m(t)$=+1, la salida es $A*cos(2\pi f_C t)$y si $v_m(t)$=-1, la salida es cero, de modo que oscila entre estos valores[12].

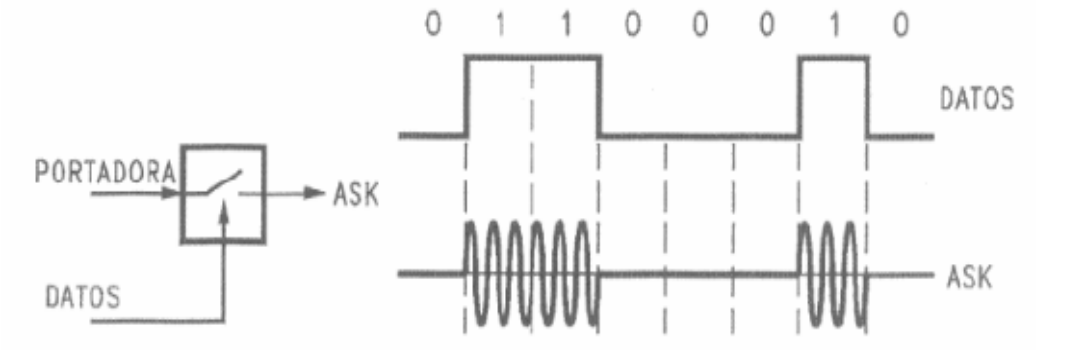


Figura 4. Señal de salida con Modulación ASK[12].

2.2.1 Los principales factores de la modulación ASK son:

- Requiere circuitos poco complejos.
- Muy sensibles a las interferencias (probabilidad de error elevada).
- Siendo Fb la velocidad de transmisión de los bits, el espectro mínimo Bw de la señal modulada resulta mayor que Fb.
- La eficiencia de transmisión, definida como la relación entre Fb y Bw resulta menor que 1.

- El Baudio, definido como la velocidad de modulación o velocidad de símbolo, es igual a la velocidad de transmisión Fb.
- La demodulación de esta señal puede hacerse de manera coherente, mediante la mezcla con la señal de un oscilador local sincronizado con la portadora o de manera incoherente, mediante la detección de envolvente. Aunque es de bajo costo su implementación, es de baja calidad para la transmisión de grandes cantidades de información[12].

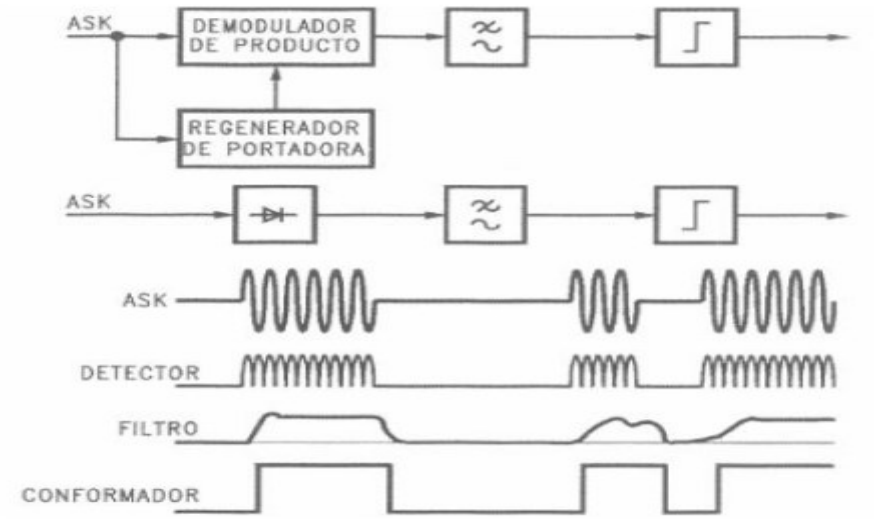


Figura 5. Señal de salida de la demodulación ASK[12]

2.3 CODIFICACIÓN MANCHESTER

En la codificación Manchester, cada período de un bit se divide en dos intervalos iguales. Un bit binario de valor 1 se transmite con valor de tensión alto en el primer intervalo y un valor bajo en el segundo. Un bit 0 se envía al contrario, es decir, una tensión baja seguida de un nivel de tensión alto.

Este esquema asegura que todos los bits presentan una transición en la parte media, proporcionando así un excelente sincronismo entre el receptor y el transmisor. Una desventaja de este tipo de transmisión es que se necesita el doble del ancho de banda para la misma información que el método convencional.

La codificación diferencial Manchester es una variación puesto que en ella, un bit de valor 1 se indica por la ausencia de transición al inicio del intervalo, mientras que un bit 0 se indica por la presencia de una transición en el inicio,

existiendo siempre una transición en el centro del intervalo. El esquema diferencial requiere un equipo más sofisticado, pero ofrece una mayor inmunidad al ruido. El Manchester Diferencial tiene como ventajas adicionales las derivadas de la utilización de una aproximación diferencia[13]l.

Todas las técnicas bifase fuerzan al menos una transición por cada bit pudiendo tener hasta dos en ese mismos periodo. Por tanto, la máxima velocidad de modulación es el doble que en los NRZ, esto significa que el ancho de banda necesario es mayor. No obstante, los esquemas bifase tienen varias ventajas:

» Sincronización: debido a la transición que siempre ocurre durante el intervalo de duración correspondiente a un bit, el receptor puede sincronizarse usando dicha transición. Debido a esta característica, los códigos bifase se denominan auto-sincronizados.

» No tienen componente en continua.

» Detección de errores: se pueden detectar errores si se detecta una ausencia de la transición esperada en la mitad del intervalo. Para que el ruido produjera un error no detectado tendría que intervenir la señal antes y después de la transición[13].

Los códigos bifase se usan con frecuencia en los esquemas de transmisión de datos. Unos de los más conocidos es el código Manchester que se ha elegido como parte de la especificación de la normalización IEEE 802.3 para la transmisión en redes LAN con un bus CSMA/CD usando cable coaxial en banda base o par trenzado. El Manchester Diferencial se ha elegido en la normalización IEEE 802.5 para redes LAN en anillo con paso de testigo, en las que se usan pares trenzados apantallados.

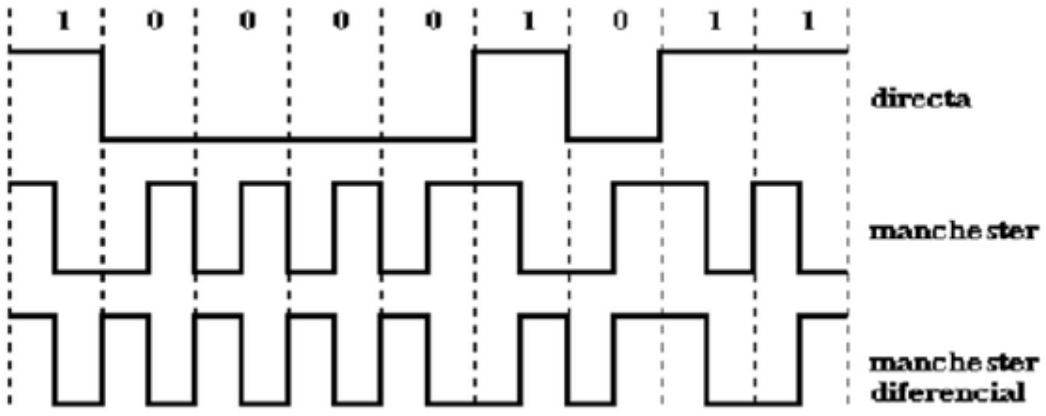


Figura 6.Codificacion Manchester y Manchester diferencial[45].

2.4 CÓDIGO MANCHESTER

El primero de estos fue publicado porr Thomas G.E. en 1949 y es seguido por numerosos autores (por ejemplo, Tanenbaum). Especifica que para representar un 0 lógico la señal debe cambiar del nivel Bajo a Alto (suponiendo una amplitud física de codificación de los datos), manteniendo el nivel bajo en la primera mitad del período del bit, y un nivel alto en la segunda mitad. Para un 1 lógico los niveles de la señal deberán cambiar de Alto a Bajo[14].

La segunda convención es también seguida por numerosos autores (por ejemplo, Stallings), así como por las normas IEEE 802.4 (token bus) y IEEE 802.3 (Ethernet). Afirma que un 0 lógico es representado por una secuencia de señales Alto-Bajo y un 1 lógico está representado por una secuencia de señal Bajo-Alto[13].

Como puede apreciarse si a una señal codificada en Manchester la invertimos, hemos pasado de una convención a la otra.

Entonces, las reglas de codificación según la primera convención son:
Si el dato original es un 0, el *código Manchester* es: 01
Si el dato original es un 1, el *código Manchester* es: 10
Cada bit se transmite en un tiempo fijo (el "periodo").
Las transiciones que significan 0 o 1 se producen en el punto medio de un período.

El proceso para realizar la codificación es muy simple y solo es necesario efectuar un X-OR entre los datos y el clock para obtener la señal codificada.

Datos	Clock	Manchester
0	0	0
0	1	1
1	0	1
1	1	0

Tabla 3. Codificación Manchester[15]

Desventajas

Se puede observar que hay dos bits de datos codificados en *Manchester* por cada bit de datos originales. Por lo tanto, al usar una codificación *Manchester* es que los datos codificados requieren el doble de ancho de banda.
Ventajas
1. Transmisión serial de bits con componente nula de DC.
2. Detección de errores fácil de aplicar.
Usos
El **código Manchester** es utilizado en la capa física de la comunicación LAN ETHERNET donde el ancho de banda adicional no es un impedimento para el cable coaxial. También suele utilizarse cuando se realiza una comunicación por medio de RF para evitar el ruido.
Existen unos integrados denominados RF800 que son codificador/decodificador para transmisiones por RF, codificando los datos en *Manchester*, y agregando un chequeo de error RCR[15].

2.5 FASES PARA ESTABLECER LA COMUNICACIÓN NFC

Esta tecnología parece destinada a desbancar a otras que están funcionando en la actualidad como los códigos QR y similares. Lo primero que hay que hacer es acercar el teléfono y automáticamente se empezaran a realizar las acciones determinadas en la etiqueta que se acaba de leer.

Estas acciones pueden ser desde abrir el navegador para ir a una página web, hasta cambiar configuraciones internas como poner en silencio o encender y apagar el Bluetooh, entre otras [16], [17].

Como se mencionó previamente, la tecnología NFC tiene dos modos de funcionamiento, **"Modo Pasivo"** (un dispositivo genera un campo mientras que el otro se aprovecha de la modulación para trasferir los datos), y el **"Modo Activo"** (Ambos dispositivos generan campos electromagnéticos).

Cuando el lector se aproxima a una etiqueta RFID o a otro lector, emite una señal de radio de corto alcance que excita el microchip de la etiqueta, con lo que se puede acceder a leer la pequeña cantidad de datos que se encuentran almacenados en ésta [9], [18].

Toda comunicación NFC consta de 5 partes:

- **Descubrimiento**: En esta fase los dispositivos inician la etapa de rastrearse el uno al otro y posteriormente su reconocimiento.

- **Autenticación:** En esta parte los dispositivos verifican si el otro dispositivo está autorizado o si deben establecer algún tipo de cifrado para la comunicación.

- **Negociación:** En esta parte del establecimiento, los dispositivos definen parámetros como la velocidad de transmisión, la identificación del dispositivo, el tipo de aplicación, su tamaño, y si es el caso también definen la acción a ser solicitada.

- **Transferencia:** Una vez negociados los parámetros para la comunicación, se puede decir que ya está realizada exitosamente la comunicación y ya se puede realizar el intercambio de datos.

- **Confirmación:** El dispositivo receptor confirma el establecimiento de la comunicación y la transferencia de datos.

2.6 ETIQUETAS NFC

Son unos dispositivos físicos donde se guarda información, pueden ser llamados Etiquetas inteligentes o SmartTags NFC. Vienen en forma de pegatinas o llaveros, posters y pulseras, donde se almacenan URLs, Cards, se pueden hacer check-ins, y una infinidad de posibilidades que permiten realizar diversas acciones pre-configuradas [19].



Figura 7.Presentación de los Tags NFC [7]

Estas etiquetas inteligentes están especificadas en 4 grupos o tipos:

Tipo 1: (ISO14443-A). Las etiquetas son de lectura y escritura[20].

Tipo 2: (ISO 14443-A estándar). Las etiquetas son de lectura y escritura que además pueden venir en varios formatos (etiquetas de interior y etiquetas sin contacto). Se ha utilizado en varias RFID y soluciones móviles RFID/NFC[20].

Tipo 3: (JIS x 6319-4). Incluye un chip sin contacto IC. Configuradas de fábrica para ser de escritura o solo lectura con disponibilidad de memoria variable[20].

Tipo 4: (ISO / IEC 7816-4 - ISO 14443-A - ISO 14443-B). Pre configuradas de fábrica para lectura y re-escritura, o de solo lectura [20].

Plataforma NFC Fórum	Productos Compatibles
Tipo 1	Broadcon Topaz
Tipo 2	NXP Mifare Ultralight, NXP Mifare Ultralight C,NXP NTAG203
Tipo 3	Sony FeliCa
Tipo4	NXP DESFire, NXP SmartFX with JCOP y otros productos sin compatibles sin contacto.

Tabla 4.Tipos de Tags compatibles según las especificaciones de NFC Fórum [25]

2.7 ARQUITECTURAS DE FUNCIONAMIENTO

2.7.1 Punto a punto

Utilizado para el intercambio de datos o establecimiento de las comunicaciones entre dispositivos NFC [21].

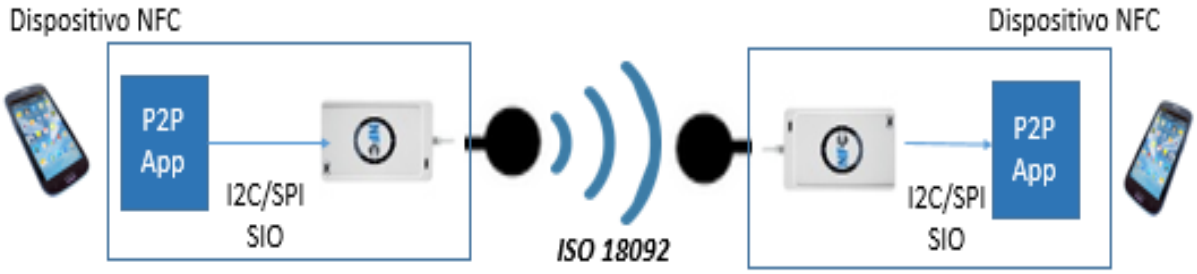


Figura 8.Arquitectura punto a punto [7]

2.7.2 Lectura-escritura

Tiene la capacidad de leer o escribir etiquetas. Suele utilizarse para los denominados posters inteligentes, ya que al leer la etiqueta incluida en el póster, ésta trasmite al teléfono la dirección de una página web, abriendo automáticamente el navegador [9].

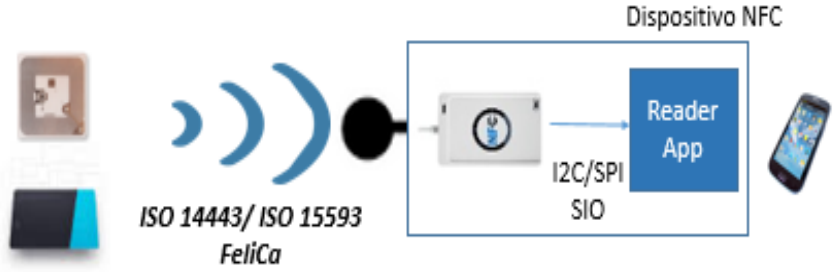


Figura 9. Arquitectura de lectura-escritura [7].

2.7.3 Emulación de tarjeta

En este modo, el dispositivo NFC se comporta como una tarjeta inteligente, apareciendo ante el lector como una tarjeta sin contacto. Con esta configuración se pueden utilizar las características del elemento de seguridad incorporado como medio de pago, así como para el almacenamiento y gestión de todo tipo de entradas y recibos [9].

La Emulación de Tarjeta de Host (HCE) permite a los dispositivos NFC realizar transacciones sin contacto en el modo de emulación de tarjeta cuando el pago, otras credenciales y solicitudes de tarjetas relacionados son

almacenados en algún lugar que no sea la SE: por ejemplo, en la nube, en un entorno de ejecución de confianza en el dispositivo móvil, o en una infraestructura virtual, basada en software del dispositivo móvil. NFC se definió por el Foro NFC y se incluye en las especificaciones de NFC Fórum, incluyendo el NFC interfaz de controlador (NCI) de especificación , que en combinación con normas como la ISO / IEC 14443 y JIS X 6319-4 , activa las HCE implementación[20].

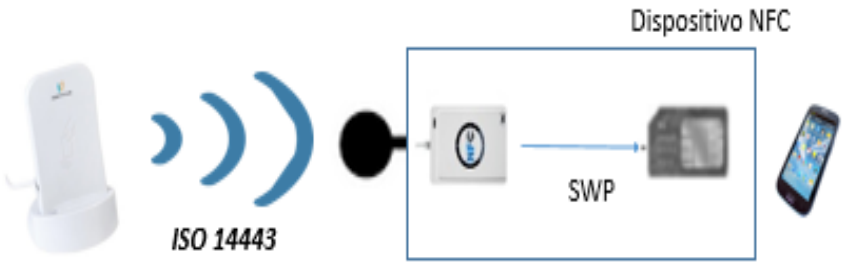


Figura 10. Arquitectura de emulación de tarjeta. [7].

A continuación se muestra una ilustración proporcionada por NFC Fórum que resume las tres arquitecturas correspondientes a la tecnología NFC.

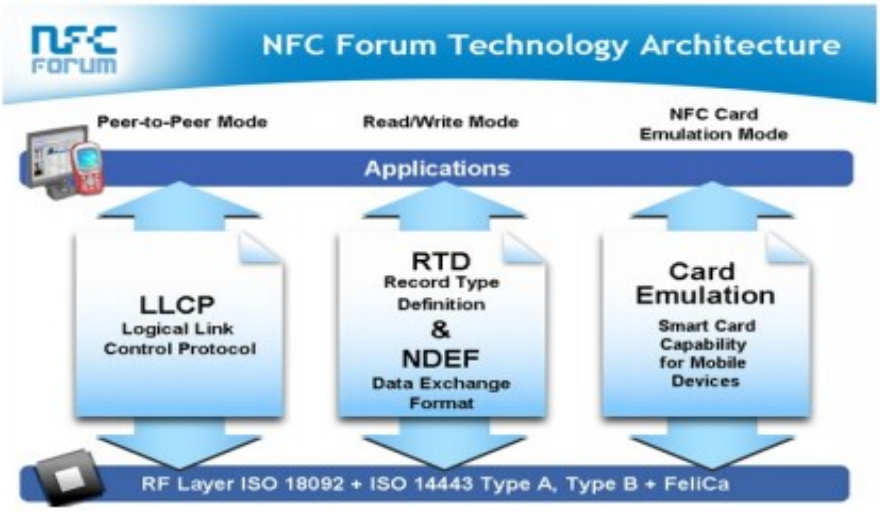


Figura 11. Arquitecturas de funcionamiento NFC [7].

2.8 PROTOCOLO DE CONTROL DE ENLACE LÓGICO

El protocolo de Control de Enlace Lógico (LLC) permite la comunicación entre dos dispositivos multiplexados Peer NFC Fórum, donde cualquiera de los actores involucrados puede enviar unidades de datos de protocolo en cualquier momento (modo equilibrado asíncrono). Los criterios de valoración de comunicación se denominan puntos de acceso de servicio (SAP) y están

dirigidas por un identificador numérico de 6 bits. Las unidades de datos de protocolo se intercambian entre puntos de acceso, a partir de una fuente de SAP (SSAP) a un SAP de destino (DSAP). El punto del espacio de direcciones de acceso al servicio se divide en 3 partes: una dirección entre 0 y 15 identifica un servicio muy conocido, una dirección de entre 16 y 31 identifica un servicio que se ha registrado en el entorno de servicio local y direcciones entre 32 y 63 son no registrado y que normalmente se utiliza como dirección de origen por las aplicaciones cliente que envían o se conectan a pares servicios[22].

2.9 ESPECIFICACIONES.

Las principales especificaciones para NFC son propuestas por NFC Fórum y su principal finalidad es la de lograr la compatibilidad entre los diferentes dispositivos. Entre especificaciones son:

2.9.1 NFC Data Exchange Format (NDEF).

Especifica el formato y reglas para la construcción de tipos de registros estándar utilizados por las definiciones de aplicaciones NFC Fórum y terceros que se basan en el formato de datos NDEF. La especificación de IDT ofrece una manera de definir de manera eficiente formatos de registro para nuevas aplicaciones y ofrece a los usuarios la oportunidad de crear sus propias aplicaciones basadas en las especificaciones de NFC Fórum[20].

Define el formato de intercambio de información pudiendo estar compuesto por uno más registros, formando de esta manera un mensaje NDEF.

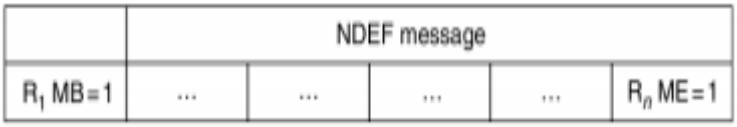


Figura 12. Formato de un mensaje NDEF [7].

2.9.2 NFC Record Type Definition:

Es un formato de mensaje binario para el intercambio de cargas definidos por el Foro NFC para almacenar cargas útiles en una etiqueta NFC. Una carga útil

se describe por un tipo, una longitud y un identificador opcional codificado en una estructura de registro NDEF. Un mensaje NDEF es una secuencia de registros NDEF con un marcador que comienza en la primera y un marcador en el registro final [23].

La especificación NDEF define el formato común de la información pero no especifica ninguno en detalle. Cada NDEF contiene una cadena con el nombre del tipo registro, cada RTN (Records Type Name) se puede especificar en diferentes formatos.
Los tipos conocidos por NFC Fórum contienen un identificador (NID) “NFC.” Nadie puede utilizar este identificador para especificar otro espacio de este tipo.

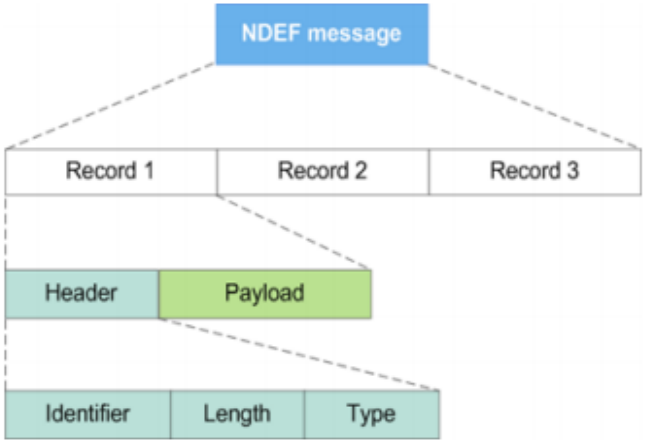


Figura 13. Estructura del mensaje NDEF [7].

2.9.3 NFC Uniform Resource Identifier Service Record Type Definition

NFC URI es el tipo de registro que es usado por NDEF para recibir la URI almacenada en una Etiqueta de un elemento como pudiera ser un poster inteligente. RTD puede ser considerado como una extensión de un tipo reconocido por NFC [3].

2.9.4 NFC Text Record Type Definition

Es de texto plano y puede ser utilizado en combinación con otros campos para proporcionar una información extra de contenido de la etiqueta [3].

2.9.5 NFC Smart Poster Record Type Specification

El tipo Smart Poster define la capacidad de añadir metadatos a la. Una etiqueta Smart poster al acercar el dispositivo sobre ella puede proporcionar acciones a realizar embebidas como parte del registro, que desencadenen acciones a realizar por parte del dispositivo como iniciar, por ejemplo, el navegador en una determina URL [19].

2.10 APLICACIONES NFC

Los primeros usos de la tecnología NFC están estrechamente ligados a los teléfonos móviles debido a su ubicuidad y al hecho de que sea el único dispositivo que llevamos a todas partes [9], [24], [25].

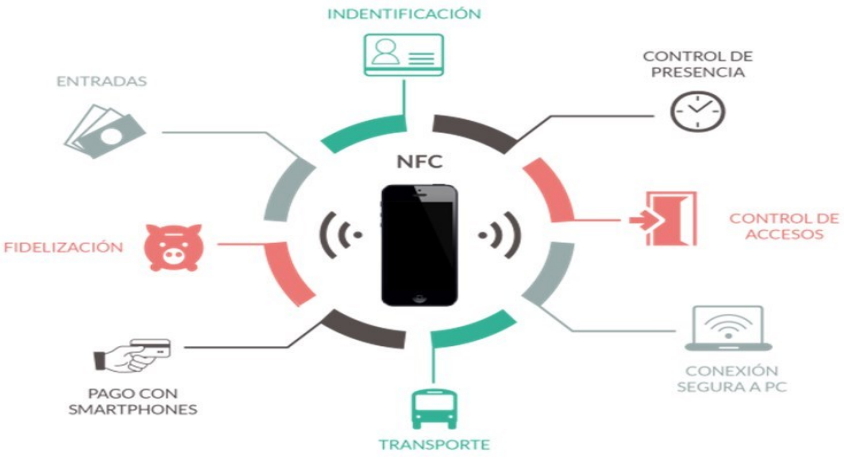


Figura 14. Aplicaciones móviles [18].

La premisa básica a la que se acoge el uso de la tecnología NFC es aquella situación en la que es necesario un intercambio de datos de forma inalámbrica. Lo usos que más futuro tienen son la identificación, la recogida e intercambio de información y sobre todo, el pago [24].

Pagos móviles:
Entre las aplicaciones más comunes, la más conocida de la tecnología NFC es la de pagos mediante móviles. Pasando el terminal móvil por un sensor se puede producir una transacción de manera cómoda y ágil. Para esta transmisión se requiere un terminal NFC y además disponer de una SIM NFC insertada en el terminal que contiene los datos bancarios de forma segura [9].

Tarjetas de acceso:

En algunas universidades en el extranjero, los estudiantes utilizan el NFC para acceder a las instalaciones del campus y además esta acreditación permite realizar un control de asistencia automático [9].

Identificación:
El acceso a lugares donde es precisa una identificación podría hacerse simplemente acercando nuestro teléfono móvil o tarjeta con chip NFC a un dispositivo de lectura. Los abonos de autobús son un ejemplo muy válido [24].

Libros con contenido Multimedia:
La tecnología NFC puede incluirse en los libros a través de etiquetas que se encuentran en las hojas del interior. Gracias a acercar el terminal móvil a estas etiquetas existe la posibilidad de descargarse archivos multimedia que complementan las lecturas [9].

Intercambio de datos:
Google es el principal protagonista de este uso, pues en combinación con las etiquetas RFID, utilidades como marcar dónde estamos, recibir información de un evento o establecimiento son inmediatas [9].

Accesorios conectados:
Al igual que con Bluetooth, NFC se puede utilizar para la conexión de distintos dispositivos, desde altavoces inalámbricos, manos libres, entre otros. Con NFC solo es necesario acercar dos dispositivos para automáticamente iniciar el emparejamiento Bluetooth con un tag en cada pantalla lo que es aprovechado para estabilizar una conexión Wi-Fi segura también [9], [25].

Tarjetas de visita:
Han empezado a aparecer empresas que proporcionan tarjetas de visita con NFC y que permiten con tan sólo acercar el teléfono móvil a la tarjeta, descubrir toda la información relativa a la persona dueña de la tarjeta: teléfono de contacto, dirección postal, correo electrónico, logos o galería fotográfica [9][25].

Lector de etiquetas RFID:
Un dispositivo NFC puede actuar como un lector de etiquetas RFID, leyendo información embebida en forma de etiquetas RFID en medios tales como posters, vallas publicitarias folletos, menús y otro material informativo o promocional similar [25].

Identificación Nacional, Internacional y Corporativa:
Se tiene pensado para un futuro que los dispositivos NFC almacenen y reemplacen todos los documentos personales: seguridad social, tarjeta de

identidad nacional, licencia de conducir, carnet de empleado y pasaporte, con medidas de seguridad agregadas en forma de escaneo biométrico [25].

2.11 VENTAJAS

Las ventajas que ofrece utilizar aplicaciones NFC son las siguientes:

- **Proximidad:** Esta característica hace más seguro e intuitivo su uso, pues solo se realiza la correspondiente operación (pagos, identificación,...), si el usuario es consciente y tiene verdadera intención de ello. Además dificulta que exista un tercero que observe la información transmitida.

- **Seguridad:** La mayor parte de las aplicaciones se ejecutan dentro de un entorno seguro (Elemento Seguro o Secure Element) que se encuentra dentro del teléfono o la tarjeta SIM. Para hacer uso de éste, se necesitan aplicaciones firmadas, lo que dificulta y imposibilita el uso de las aplicaciones con fines fraudulentos.

- **Paso al modo electrónico**: Funciones como el pago, el acceso a lugares con llaves convencionales pasa a ser parte de esta tecnología, que a partir de ahora evoluciona para poder integrarse e interactuar con otras utilidades.

- **Todo en uno:** La tendencia actual es poder realizar todo tipo de gestiones desde un Smartphone. La inclusión del NFC dentro de los terminales incrementa el número de situaciones en las que puede resultar útil tener todas las aplicaciones en un mismo dispositivo y que interactúan entre ellas(por ejemplo, realizar check-in de un evento o lugar y compartirlo en una red social)

Capítulo 3

3 ESPECIFICACIONES TÉCNICAS DE LAS TAGS NFC

3.1 ASPECTOS QUE SE DEBEN TENER EN CUENTA

Para seleccionar la etiqueta NFC se debe conocer de antemano algunos aspectos relacionados con la cantidad de información que se desea almacenar, el espacio físico donde se colocar, el ambiente de trabajo y algunos requerimientos de seguridad.

3.1.1 Espacio físico disponible para su colocación.

El tamaño físico de la incrustación NFC podría afectar significativamente su rendimiento en el uso final. Es importante seleccionar el producto adecuado de tamaño para cada aplicación específica. Se pueden probar diferentes tamaños de etiquetas en el entorno real de uso final, esto dará una mejor referencia para la planificación de proyectos[26][27].

3.1.2 Tamaño de la memoria.

Tamaño de la memoria también debe ser comparable a los requisitos de la solución, ni demasiado grande ni demasiado pequeño. Si el objetivo es codificar simplemente un enlace web, un ID, una llamada necesitaras un tag con poca memoria. Si por el contrario se desea generar tarjetas de visitas, promociones específicas, instrucciones, se necesitaran tags con más prestaciones y mayor capacidad memoria. Sin olvidar que esta característica influye en el precio de las etiquetas[28].

3.1.3 Ambiente donde se colocara la etiqueta.

- Materiales impresos (carteles, postales): Los usuarios finales no saben dónde se encuentra la etiqueta a menos que se indique de alguna manera. Se

recomienda marcar la ubicación de la etiqueta NFC con una imagen o con el logotipo de NFC Fórum, que es una marca estandarizada globalmente[27].

- A través del uso de ventanas/cartelera: Se debe tener en cuenta espesor de ventanas o cristales, estas pueden variar mucho y afectar el rendimiento de lectura del Chip. Asegúrese de suministrar buenas instrucciones para localizar la etiqueta NFC en el entorno del usuario final y que los materiales de referencia disponibles al público estén en ubicaciones estratégicas.
- Soluciones de etiquetas NFC robustos para industria textil (etiquetas tejidas, etiquetas colgantes, lavables)[27].

- Incorporación de etiquetas dentro de un producto (madera, plástico, textil, electrónica).

Uno de los grandes beneficios de etiquetas NFC es que pueden ser incrustados dentro de objetos como cubiertas de plástico, tejidos o elementos de madera, y que no requieren una línea de visión externa. Normalmente, en la fase de diseño del producto, es bueno tener en cuenta donde se va a colocar la etiqueta en el interior del producto para que la experiencia de usuario sea la más conveniente y así asegurar el rendimiento óptimo del producto[28].

Por ejemplo, cuando hay un objeto de metal cerca de la antena de la etiqueta NFC, se requiere una hoja magnética de alto rendimiento en la etiqueta NFC para evitar un fallo de comunicación[29][27].

3.2 TABLA COMPARATIVA

A continuación se presenta una tabla comparativa de los Tags NFC disponibles en el mercado donde se pueden ver las características técnicas de estos con el fin de facilitar la elección del tag que se ajuste a las necesidades del proyecto.

	Ultralight	**Ultralight C**	**Standard 1k**	**NTAG203**	**Felica Lite-S**	**Felica-RC 4KB**
Memoria Total	64 Bytes	192 Bytes	1024 Bytes	168 Bytes	240 Bytes	4 KBytes
Memoria disponi-ble	46 Bytes	137 Bytes	716 Bytes	137 Bytes	224 Bytes	2 464 Bytes
Longitud de URL	46 caracteres	132 caracteres	256 caracteres	132 caracteres	-------	-------
Longitud de texto	39 caracteres	130 caracteres	709 caracteres	130 caracteres	-------	------
Compati-bilidad Móvil	SI	SI	NO	SI	SI	SI
Tipo de tag	Tipo 2	Tipo 2	Tipo A	Tipo 2	Tipo 3	Tipo 3
Numero de serie	SI	SI	SI	SI	SI	SI
Criptogra-fía	NO	3DES	Cryto-1	NO	NO	3DES
Fortaleza de escaneo	Hasta 5 cm	Hasta 3 cm	Hasta 5 cm	Hasta 10 cm	Hasta 5 cm	Hasta 5 cm
Acceso de datos	Lectura/escri-tura o solo lectura	Lectura/es-critura o solo lectura	Lectura/es-critura o solo lectura	Lectura/es-critura o solo lectura	Lectura/es-critura o solo lectura	Lectura/es-critura o solo lectura
Utilidades	Smart Pósters y aplicacione s NFC en general.	Exclusivam ente para aplicaciones que requieren tecnología de encriptación	Recomenda do para Vcards y utilización de alta capacidad.	Última tecnología, gran rendimiento. Ideal para todo tipo de aplicaciones	Ideal para todo tipo de aplicaciones	Se puede usar en múltiples a aplicacio-nes
Precio por Unidad	bajo	bajo	medio	bajo	medio	alto

Tabla 5.Comparación de los tags disponibles en el mercado y compatibles con el NFC fórum [24][30]–[33].

Memoria Total: Cantidad total de memoria disponible en el chip. Esta memoria puede ser bloqueada para impedir que el chip se regrabe con otra información, incluyendo los sectores de bloqueo denominados sectores tráiler[30].

Memoria Disponible: Suele ser el elemento más importante para el comprador. Es la memoria en la que el usuario almacena datos en el chip[30].

Longitud URL: La longitud máxima del Localizador Uniforme de Recursos (URL) a almacenar, excluyendo la parte "http://"[30].

Longitud Texto: La longitud máxima del texto plano que se puede almacenar en el chip[30].

Compatibilidad Móvil: Si se utiliza una app Android para codificar las etiquetas NFC, algunas funcionalidades de bloqueo no funcionarán con el chip Standard 1K. Éste chip, al tener un formato propietario, es incompatible con dispositivos BlackBerry ya que éstos tan sólo pueden leer chips standard definidas por el NFC Fórum[30].

Tipo de Tag: Indica si el chip NFC es compatible con las especificaciones NFC Fórum, según la clasificación de estos en tipo1, tipo 2, tipo 3 y tipo 4.

Número de Serie: El chip contiene un número de serie único que permite identificarlo. Se requiere una aplicación específica para acceder a dicha información[30].

Criptografía: Una funcionalidad de seguridad en el chip que lo protege de ser clonado. Es una funcionalidad muy avanzada que requiere de conocimientos especializados y que no suele ser necesario en aplicaciones NFC standard[30][31][i].

Fortaleza de escaneo: Es una indicación de la distancia relativa de escaneo del chip, distancia máxima a la que el chip puede intercambiar información con un dispositivo habilitado con NFC. Es una medida aproxima ya que influyen factores como el material donde se coloca la etiqueta y la potencia que genera el dispositivo lector[30].

Acceso de Datos: es la indicación de que el Tag se puede leer y escribir, y reescribir. También se puede configurar para que sea de solo lectura.

Precio por unidad: Precio aproximado para las presentaciones básicas en forma de pegatina adhesiva. Se consideran de bajo costo aquellas cuyo precio es menor a 2 EUR, de 2 a 5 ERU en costo medio y mayor a 5 EUR se considera de alto costo.

3.3 INFORMACIÓN DETALLADA DE LAS ESPECIFICACIONES TÉCNICAS DE LOS TAGS NFC

3.3.1 STANDARD CARD IC MF1 IC S70

Características[32]**:**

Marca: MIFARE

- Interfaz de RF (ISO / IEC 14443 A).
- Transmisión sin contacto de datos y de suministro energía (no necesita batería).
- Distancia de funcionamiento: Hasta 100 mm (dependiendo en la geometría de la antena).
- Frecuencia de trabajo: 13,56 MHz
- Tasa de transferencia de datos: 106 kbit / s
- Alta integridad de los datos: 16 bits CRC, paridad, bits codificación, recuento de bits.
- Anticolisión
- Tiempo de transacciones de venta de entradas: <100 ms (incluyendo gestión de copia de seguridad)

EEPROM

- 4 Kbytes, organizado en 32 sectores con 4 bloques y 8 sectores con 16 bloques (un bloque consiste de 16 bytes)
- Las condiciones de acceso de usuario definibles para cada bloque de memoria.
- La retención de datos de 10 años.
- Rescritura 100.000 ciclos

Organización de la memoria

La memoria EEPROM 4 Kbytes está organizado en 32 sectores con 4 bloques y en 8 sectores con 16 bloques. Un bloque consta de 16 bytes. En el estado borrado las células EEPROM se leen como un "0" lógico, en el estado escrito como un "1" lógico.

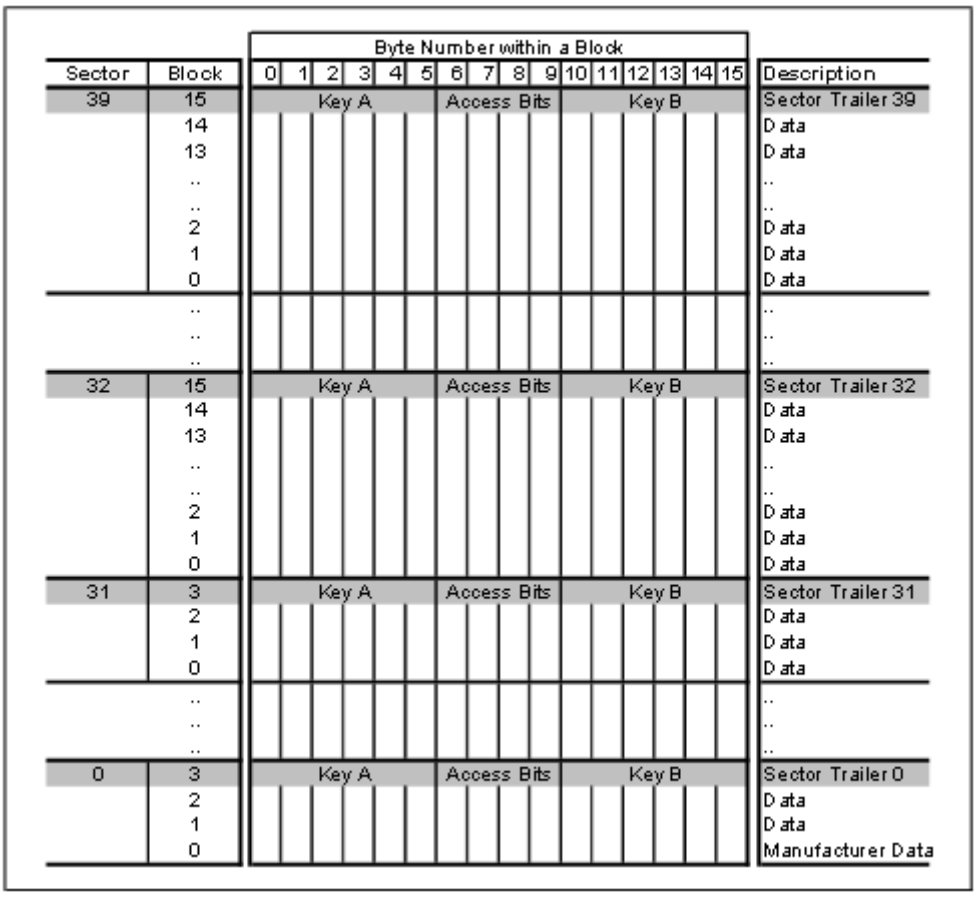


Figura 15. Organización de la memoria para el Tag IC MF1 IC S70[32].

3.3.2 MF3ICD21, ICD41, MF3ICD81[33]

MIFARE DESFire EV1 es ideal para los proveedores de servicios que deseen utilizar aplicaciones múltiples en tarjetas inteligentes, en sistemas de transporte, aplicaciones de administración electrónica o de identidad. Cumple totalmente con los requisitos para la transmisión rápida y de alta seguridad de datos, organización de la memoria flexible y la interoperabilidad con la infraestructura existente.

MIFARE DESFire EV1 se basa en estándares mundiales abiertos tanto para la interfaz de aire y métodos criptográficos. Es compatible con los 4 niveles de ISO / IEC 14443 y utiliza ISO / IEC 7816-4 y cuenta con algunos comandos opcionales.

Presenta un sistema de gestión de copia de seguridad en el chip y las tres fases de autenticación, una tarjeta MIFARE DESFire EV1 puede almacenar

hasta 28 aplicaciones diferentes y 32 archivos por aplicación. El tamaño de cada archivo se define en el momento de su creación, por lo que Mifare DESFire EV1 es un producto verdaderamente flexible y conveniente.

Cuenta con un mecanismo anti-desgarro automático, disponible para todos los tipos de archivos, que garantiza la integridad de los datos. Con MIFARE DESFire EV1, se pueden alcanzar velocidades de transferencia de datos de hasta 848 kbit, que permite la transmisión de datos rápida.

Las principales características de este dispositivo se designan por su nombre DESFire EV1: DES indica el alto nivel de seguridad mediante un motor criptográfico de hardware 3DES para cifrado de datos de transmisión y Fire indica su posición destacada como una forma rápida, innovadora, fiable y segura en el mercado de transacciones de proximidad sin contacto. Por lo tanto, MIFARE DESFire EV1 trae muchos beneficios a los usuarios finales. Los tarjetahabientes pueden experimentar sin inconveniente la venta de entradas al mismo tiempo tener la posibilidad de utilizar el mismo dispositivo para aplicaciones relacionadas, tales como pago en máquinas expendedoras, control de acceso y eventos de entradas. En otras palabras, la solución de silicio MIFARE DESFire EV1 ofrece un mayor diseño del sistema para el consumidor, en combinación con la seguridad y la fiabilidad. MIFARE DESFire EV1 ofrece el equilibrio perfecto entre velocidad, rendimiento, coste y eficiencia. Su concepto abierto permite futura integración de otros medios de venta de entradas tales como entradas inteligentes papel, llaveros, y venta de entradas para móviles basado en la Tecnología Comunicación (NFC). También es totalmente compatible con el MIFARE existentes plataforma de hardware lector. MIFARE DESFire EV1 es su boleto a los sistemas sin contacto en todo el mundo[33].

Organización de la memoria

La memoria se organiza mediante un sistema de archivos flexible. Este sistema de archivo permite un máximo de 28 aplicaciones diferentes en un MIFARE DESFire EV1. Cada aplicación proporciona hasta 32 archivos. Cada aplicación está representada por 3 bytes identificadores de la aplicación (AID).

Cada archivo se puede crear ya sea en la inicialización MIFARE DESFire EV1 (tarjeta producción / impresión de la tarjeta), a MIFARE DESFire EV1 personalización (máquinas expendedoras) o en el campo. Si un archivo o aplicación se vuelve obsoleto en funcionamiento, puede ser invalidado de forma permanente. Los comandos que tienen impacto en la estructura de

archivos en sí (por ejemplo, la creación o supresión de las aplicaciones, el cambio de claves) activan un mecanismo de reversión automática, que protege la estructura de archivos de ser dañada.

Si esta reversión es necesaria, se hace sin la interacción del usuario antes de llevar a cabo nuevos comandos. Para garantizar la integridad de los datos en el nivel de aplicación, una copia de seguridad orientada a transacciones se implementa para todos los tipos de archivos. Es posible mezclar tipos de archivos con y sin respaldo dentro de una aplicación. A medida que los comandos son los mismos para MF3ICD81, MF3ICD41 y MF3ICD21, los diferencia únicamente el tamaño de la memoria[33].

3.4 TAGS NFC DISPONIBLES EN EL GRUPO LÍDER DE LA UNIVERSIDAD DISTRITAL

3.4.1 NXP MIFARE NTAG 203



Figura 16. NXP NTAG 203[imagen propia]

Es un Tag tipo 2, fabricado por NXP semiconductor de acuerdo a las recomendaciones del NFC fórum y la norma ISO/IEC 14443.

La interacción entre el Tag y el dispositivo se lleva a cabo de acuerdo el esquemático(Fig. 17) y con una tasa de transmisión de datos de 106 Kb/s[34].

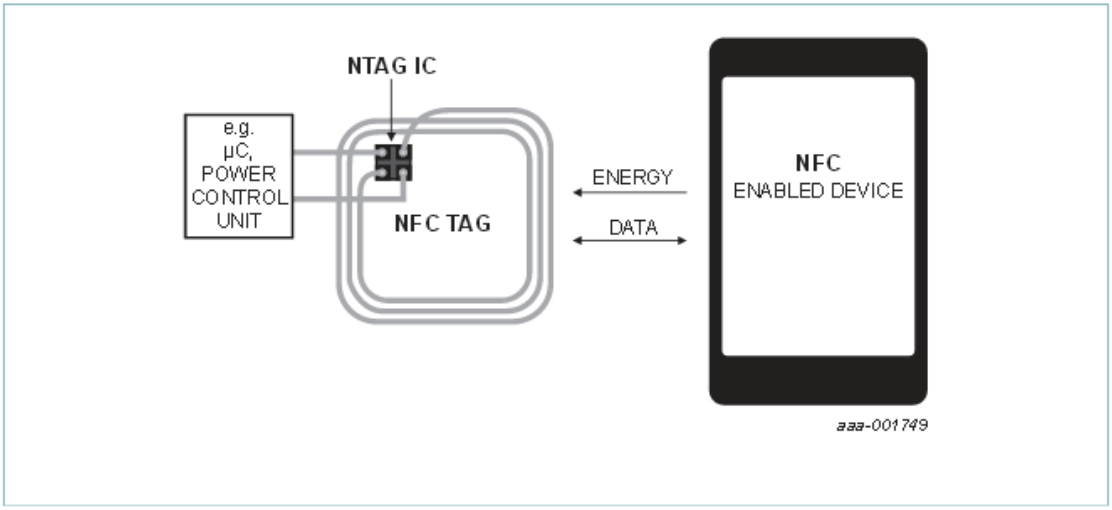


Figura 17. Interacción entre el Tag y un dispositivo NFC habilitado [34].

Características [34]

Interfaz de RF (ISO / IEC 14443A)

- Transmisión sin contacto de los datos y la energía de alimentación (no necesita batería).
- Distancia de funcionamiento: hasta 100 mm (en función de varios parámetros como por ejemplo campo la fuerza y la geometría de la antena).
- Frecuencia de trabajo: 13,56 MHz Rápida transferencia de datos: 106 kbit / s.
- Alta integridad de los datos: 16 bits CRC, paridad, bits de codificación, el conteo de bits.
- La función anticolisión
- Número de serie 7 byte (nivel 2 en cascada según la norma ISO / IEC 14443-3)

EEPROM

- 168 bytes de memoria total, divididas en 42 páginas (4 bytes cada uno)
- 144 bytes de área de r / w memoria de usuario, dividido en 36 páginas (4 bytes cada uno)
- El campo de sólo lectura programable de función de bloqueo por página 16 páginas (64 bytes) de la memoria
- El campo programable función por bloque de bloqueo de sólo lectura (2 páginas)
- área de 32 bits definida por el usuario de una sola vez programable (OTP)

- contador de 16 bits
- La retención de datos de 10 años
- 10.000 ciclos de escritura de resistencia.

3.4.2 NXP MIFARE CLASIC 1K

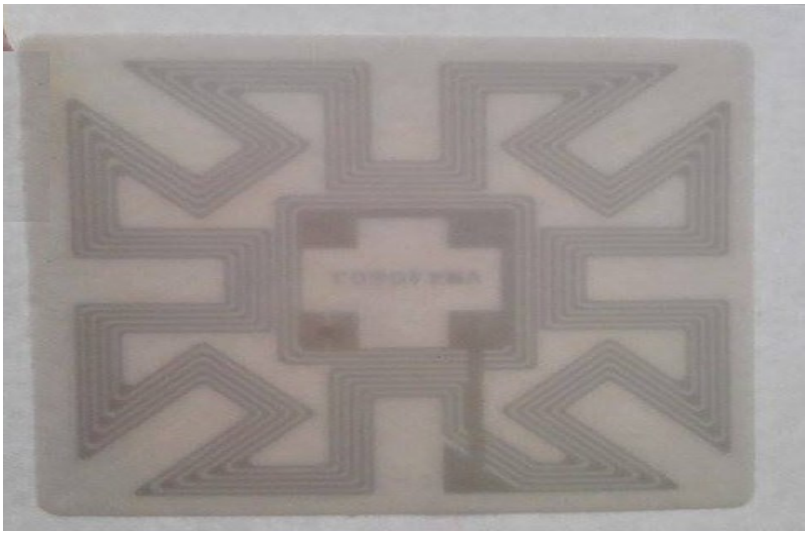


Figura 18. MIFARE CLASIC 1K [imagen propia].

Descripción General[35]
NXP Semiconductor ha desarrollado el NXP MIFARE CLASIC 1K para ser utilizado en una tarjeta inteligente sin contacto según la norma ISO / IEC 14443 Tipo A. Entre sus principales aplicaciones está el uso como billetes de transporte público, también se puede utilizar en otras aplicaciones.

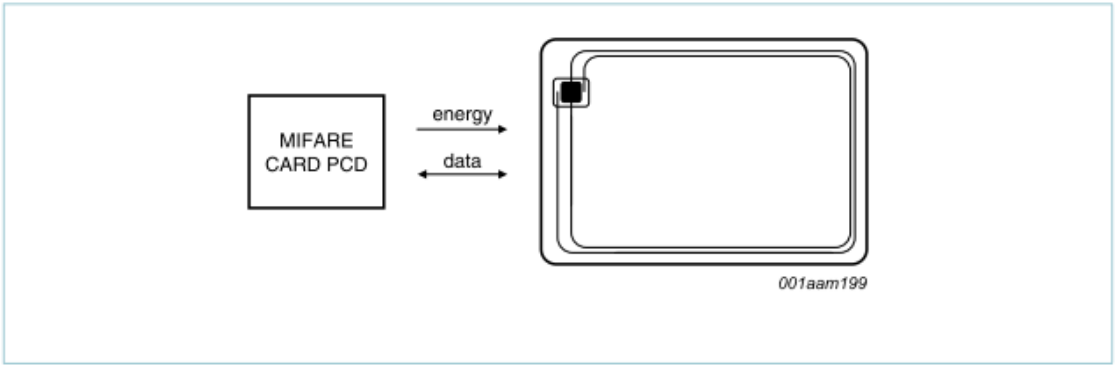


Figura 19. Interacción entre el Tag MIFARE CLASIC 1K y un dispositivo NFC habilitado [35].

Características y beneficios [35].

- Transmisión sin contacto de datos y el suministro de energía

- Distancia de funcionamiento de hasta 100 mm, dependiendo de la geometría de la antena y configuración del lector
- Frecuencia de trabajo de 13,56 MHz
- La transferencia de datos de 106 kbit/s
- Integridad de datos de 16 bits CRC, paridad, bits de codificación, el conteo de bits anticolisión
- Típico tiempo de la transacción de venta de entradas de menos de 100 ms (incluyendo la gestión de copia de seguridad).

Organización de la memoria

- La memoria EEPROM está organizado en 16 sectores de 4 bloques. Un bloque contiene 16 bytes.

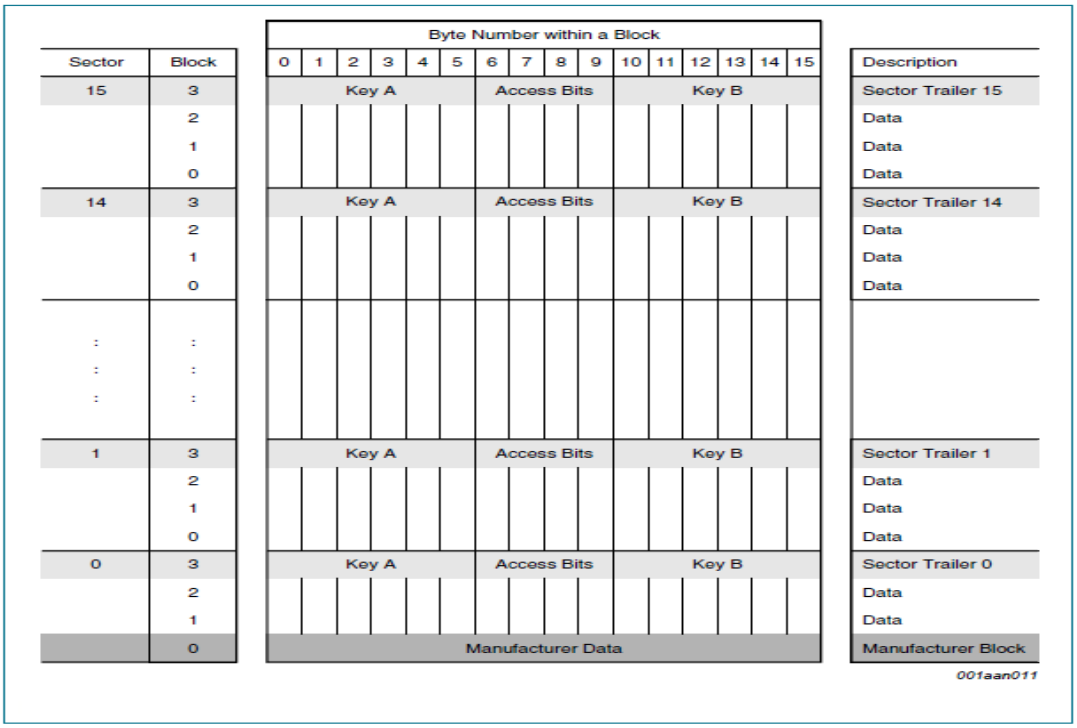


Figura 20. Organización de la memoria para el Tag MIFARE CLASIC 1K [35].

EEPROM

- 1 kB, organizados en 16 sectores de 4 bloques (un bloque consta de 16 bytes)
- Condiciones de acceso definibles por usuario para cada bloque de memoria
- Tiempo de retención de datos de más de 10 años
- Resistencia de escritura 100.000 ciclos

BLOQUES DE DATOS

Todos los sectores contienen 3 bloques de 16 bytes para almacenar datos (sector 0 contiene sólo dos bloques de datos y el bloque de fabricante de sólo lectura).

Los bloques de datos se pueden configurar por los bits de acceso como:

- Bloques de lectura y escritura
- Bloques de valores

Los bloques de valores pueden utilizarse para aplicaciones de monedero electrónico, donde se proporcionan los comandos adicionales como incremento y decremento para el control directo del valor almacenado.

3.4.3 NXP MIFARE CLASIC 4K

Descripción general[36]

NXP Semiconductor ha desarrollado el MIFARE CLASIC 4K para ser utilizado en una tarjeta inteligente sin contacto según la norma ISO / IEC 14443 Tipo A. El MIFARE Classic 4K se utiliza como billetes de transporte público y en otras aplicaciones.

MIFARE CLASIC 4K tiene incorporada una función anticolisión inteligente que permite operar más de una tarjeta en el campo simultáneamente. El algoritmo anticolisión selecciona cada tarjeta individual y garantiza que la ejecución de una transacción con una tarjeta seleccionada se lleva a cabo correctamente sin la interferencia de otra tarjeta en el campo.

Fácil integración y la comodidad del usuario.

Dentro de los tacs MIFARE CLASIC 4K está la serie MF1S70yyX está diseñado para una integración sencilla y la comodidad del usuario que permite a las transacciones de venta de entradas completas para manejarse en menos de 100 ms.

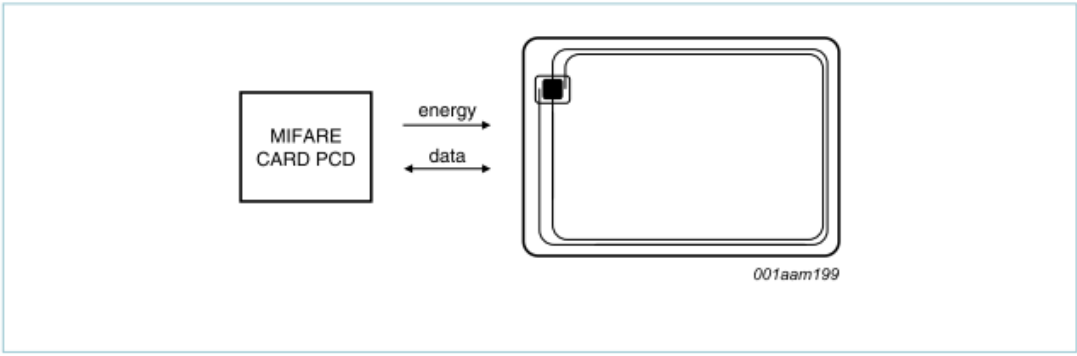


Figura 21. Interacción entre el Tag MIFARE CLASIC 4K y un dispositivo NFC habilitado [36].

Características y beneficios[36]

- Transmisión sin contacto de datos y suministro de energía
- Distancia de funcionamiento de hasta 100 mm, dependiendo de la geometría de la antena y configuración del lector
- Frecuencia de trabajo de 13,56 MHz La transferencia de datos de 106 kbit / s Integridad de datos de 16 bits CRC, paridad, bits de codificación, el conteo de bits
- Función anticolisión
- Típico tiempo de la transacción menor a 100 ms (incluyendo la gestión de copia de seguridad)

Organización de la memoria

- La memoria EEPROM 4096 × 8 bits está organizado en 32 sectores de 4 bloques y 8 sectores de 16 bloques. Un bloque contiene 16 bytes.

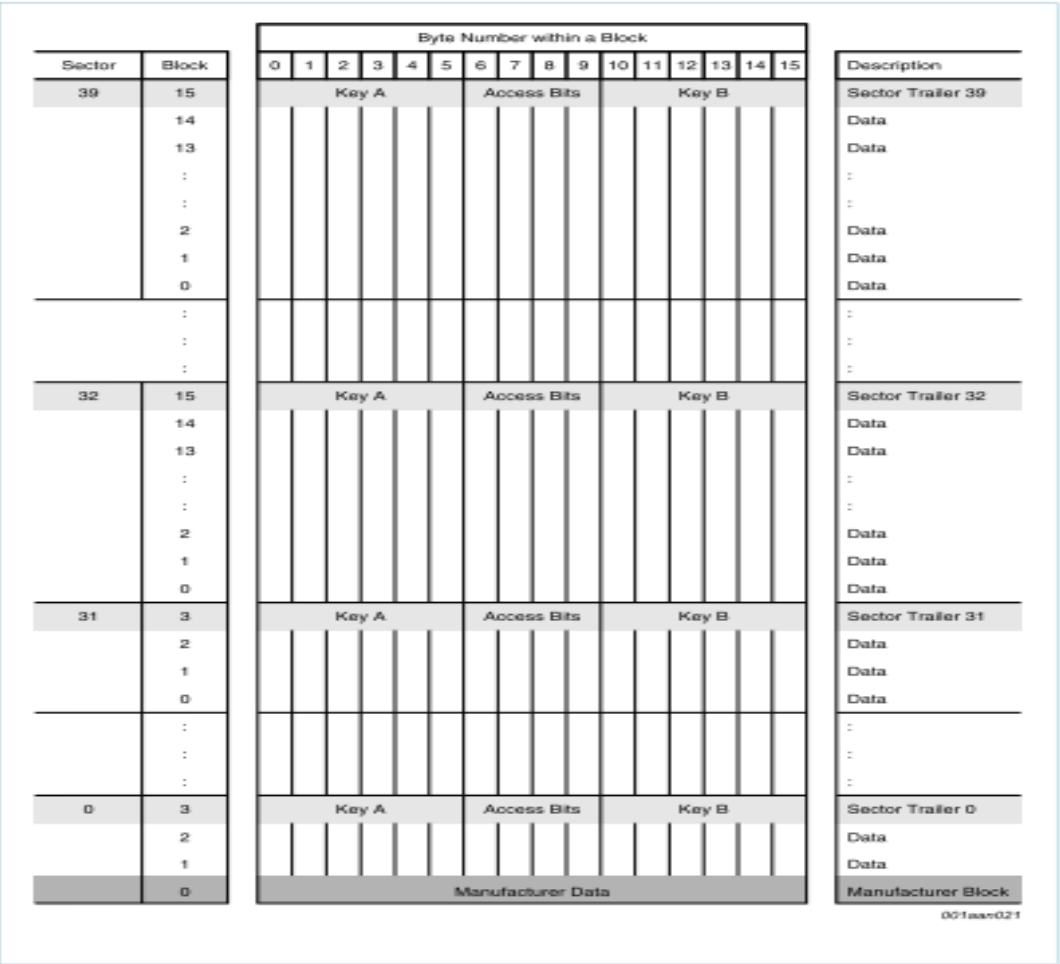


Figura 22. Organización de la memoria para el Tag MIFARE CLASIC 4K [36].

EEPROM[36]

- Contiene 4 kB, organizados en 32 sectores de 4 bloques y 8 sectores de 16 bloques (un bloque consta de 16 bytes)
- Condiciones de acceso definibles por el usuario para cada bloque de memoria
- Tiempo de retención de datos de 10 años
- Resistencia a la escritura 100000 ciclos

BLOQUES DE DATOS

Un bloque consta de 16 bytes. Los primeros 32 sectores contiene 3 bloques y los últimos 8 sectores contienen 15 bloques para almacenar datos (sector 0 contiene sólo dos bloques de datos y el bloque de fabricante de sólo lectura).

Los bloques de datos se pueden configurar por los bits de acceso como

- Bloques de lectura/escritura
- Bloques de valores.

Bloques de valores pueden utilizarse en aplicaciones tipo monedero electrónico, donde se proporcionan los comandos adicionales para incremento/decremento y control directo del valor almacenado.

3.5 TAG SELECCIONADO PARA LA APLICACIÓN

Teniendo en cuenta las especificaciones técnicas de los tags mencionados anteriormente, se toma la decisión de trabajar con el tag NXP MIFARE ULTRALIGHT NTAG 203 el cual presenta las siguientes características:

Tamaño: 25mm*35mm
Tipo de chip: Ntag203
Memoria: 144 bytes
Frecuencia de operación: 13.56MHz
Reescritura: 100,000+ ciclos
Retención de datos: Más de 10 años.
Rango de operación: 0 ~ 10cm (dependiendo del dispositivo NFC)
Compatibilidad: Compatible con todos los dispositivos NFC
Costo: Bajo.

CAPITULO 4

4 DISEÑO E IMPLEMENTACIÓN DE LA APLICACIÓN

4.1 ANALISIS DE REQUERIMIENTOS

ID	DESCRIPCIÓN	PRIORIDAD
R1	Leer TAG	**ALTA**
R2	Escribir TAG	**ALTA**
R3	Realizar Inventario	**ALTA**
R4	Crear Base de Datos	**ALTA**
R5	Crear Sistema de almacenamiento	**ALTA**
R6	Autenticación por contraseña	**ALTA**
R7	Ingresar datos del inventario de libros (idnfc, titulo, autores, isbn, edición, editorial, fecha, país, páginas, idioma).	ALTA
R8	Ingresar datos del usuario (nombre, identificación, institución, estado, estatus).	ALTA
R9	Modificar datos del libro.	ALTA
R10	Eliminar datos del libro.	ALTA
R11	Modificar datos del usuario.	ALTA
R12	Eliminar datos del usuario.	ALTA
R13	Consultar libro por idnfc.	ALTA
R14	Consultar usuario por identificación	ALTA
R15	Color de interfaz gráfica.	BAJA
R16	Color de la letra.	BAJA

Tabla 6. Análisis de requerimientos.

4.2 IDENTIFICACION DE ACTORES.

- ADMINISTRADOR.
- USUARIO1
- USUARIO2

4.3 DIGRAMA DE CASOS DE USO

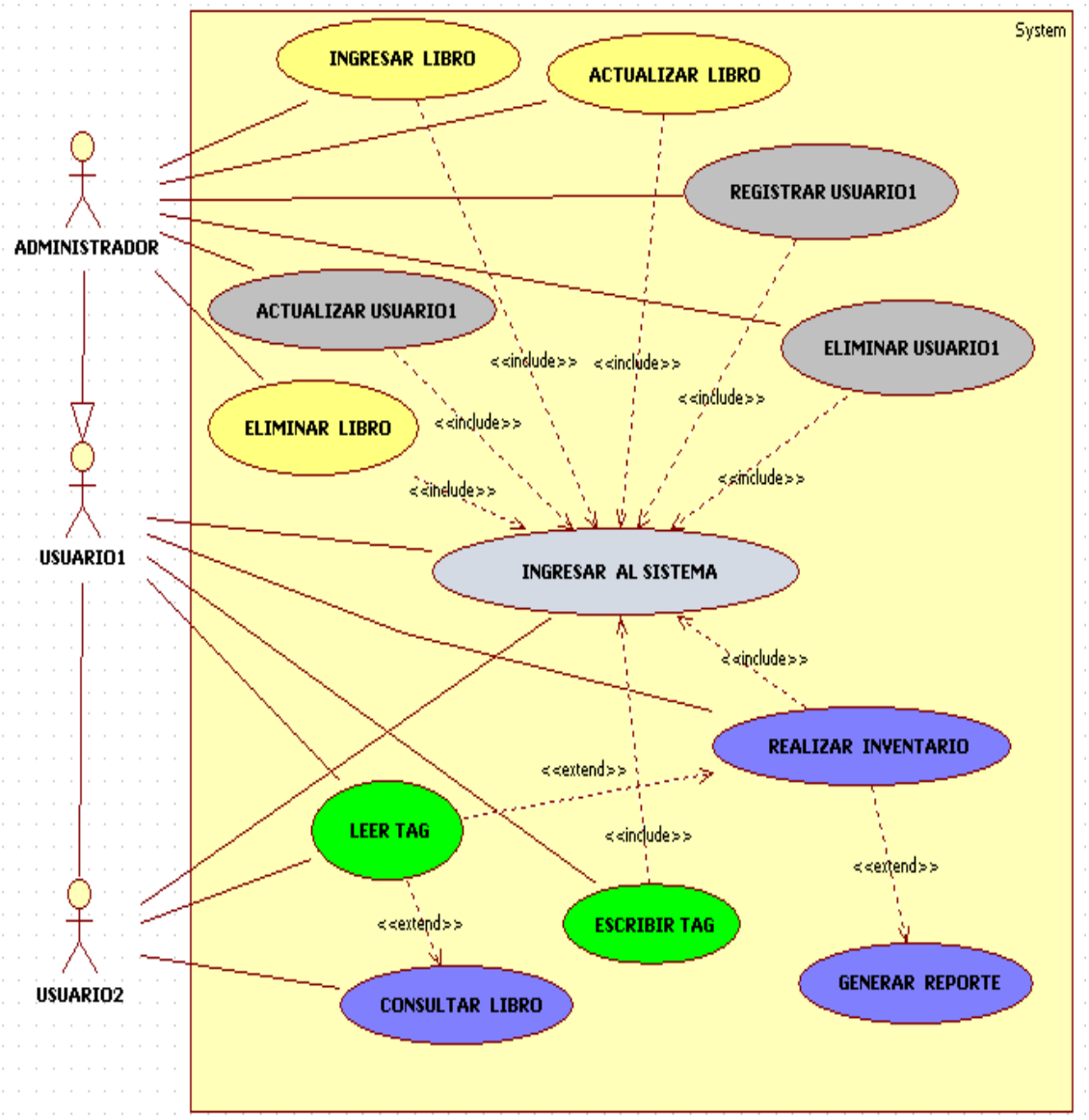


Figura 23. Diagrama de casos de uso

4.4 DIAGRAMA DE FLUJO

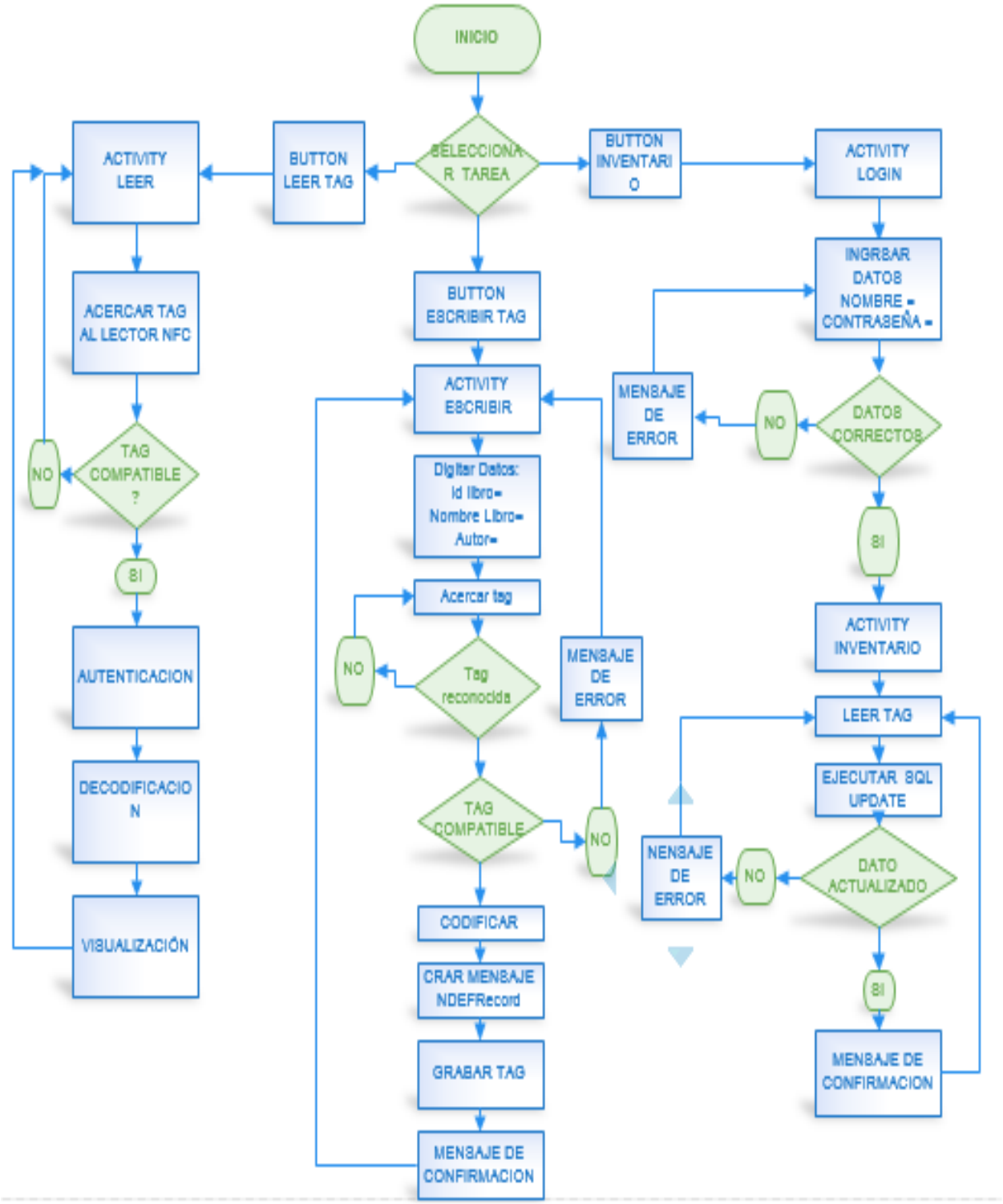


Figura 24. Diagrama de Flujo

4.5 DIGRAMA DE CLASES

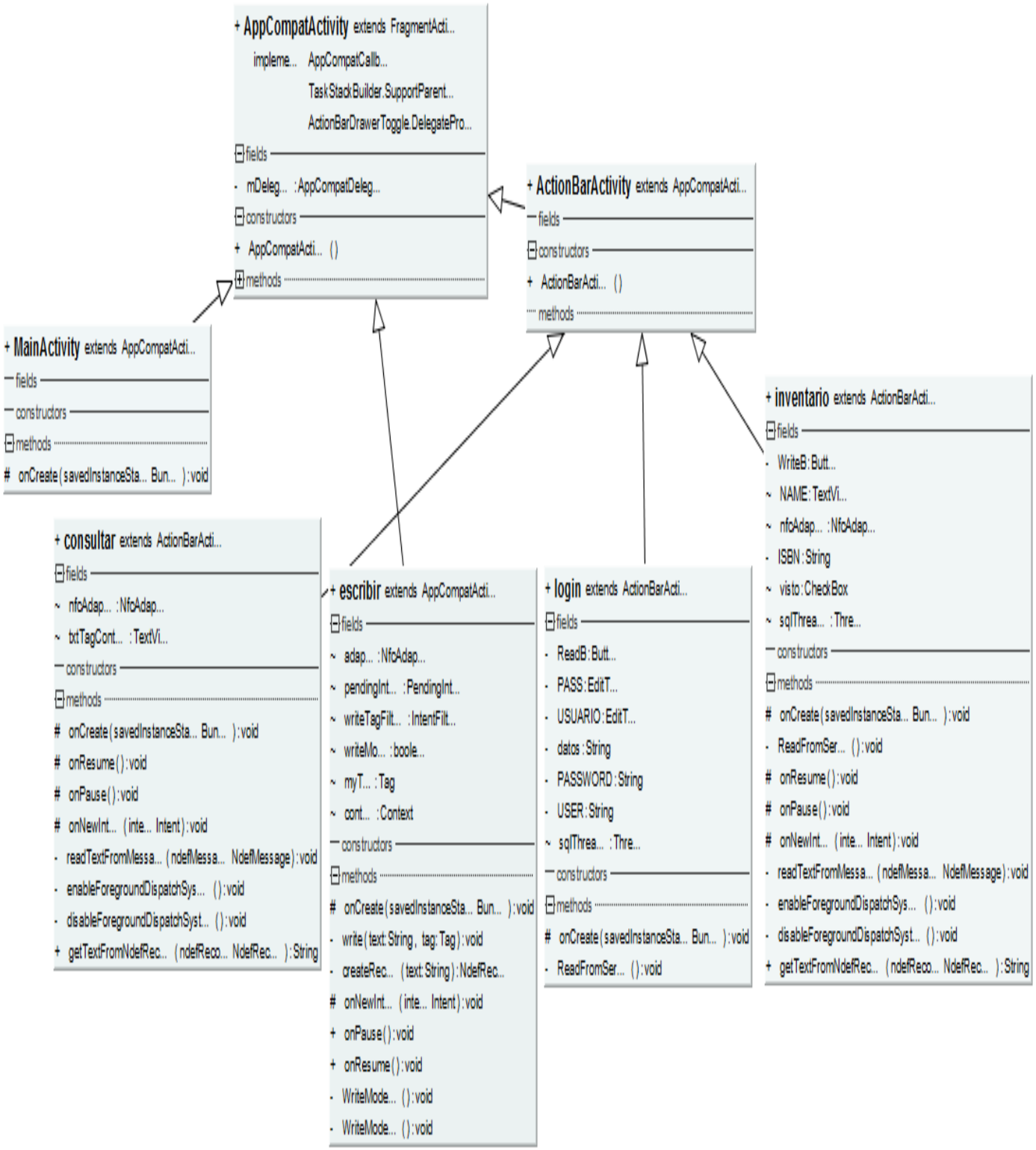


Figura 25. Diagrama de clases

4.6 DISEÑO D E INTERFAZ DE LA APLICACIÓN

La interfaz de usuario es el medio con que el usuario puede comunicarse con una máquina, equipo, o dispositivo, y comprende todos los puntos de contacto entre el usuario y el equipo. Normalmente suelen ser fáciles de entender y fáciles de accionar, amigables e intuitivas.

4.6.1 Vista principal

En la ventana principal la aplicación mostrara en la parte superior un mensaje de bienvenida, en la parte central se visualizara el escudo de la universidad distrital Francisco José de Caldas y en la parte inferior estarán ubicados tres botones que permitirán al usuario escoger entre leer tag, escribir tag y realizar inventario.

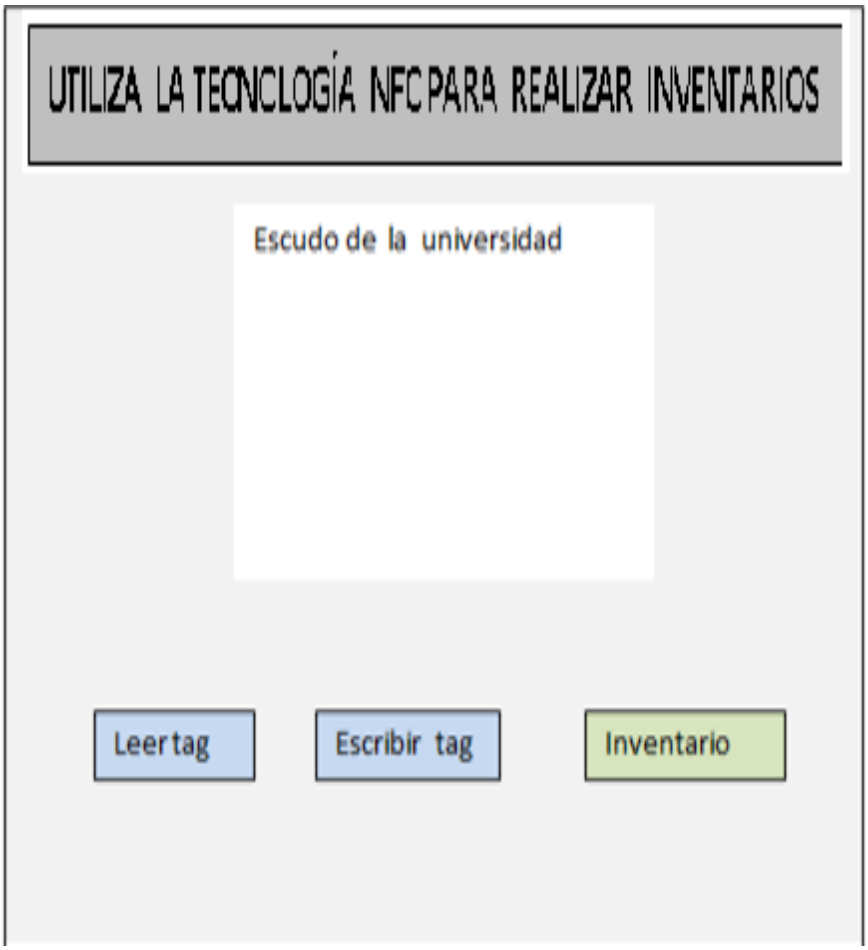


Figura 26. Vista principal

4.6.2 Leer tag

En esta pantalla el usuario podrá visualizar la información contenida en el tag NFC.

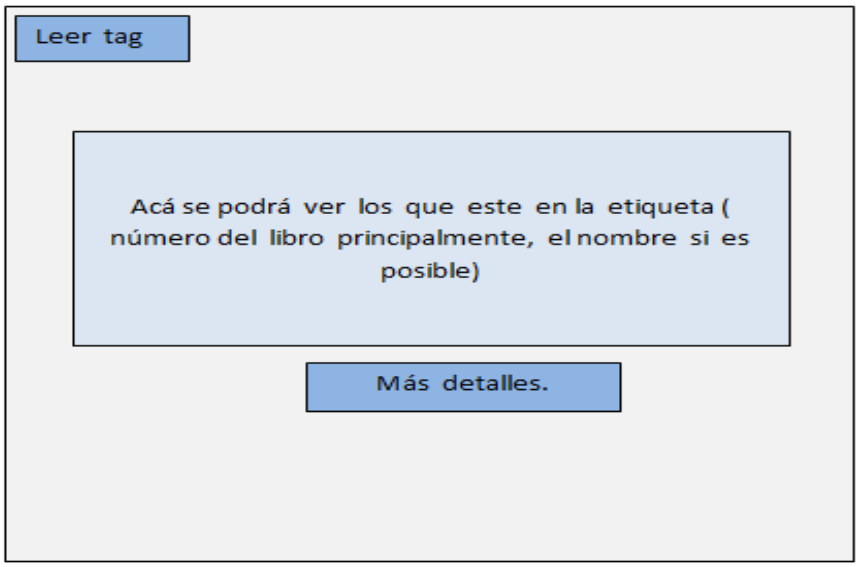


Figura 27. Vista de leer Tag

4.6.3 Escribir tag

En esta pantalla el usuario podrá digitar la información que desee escribir en el tag.

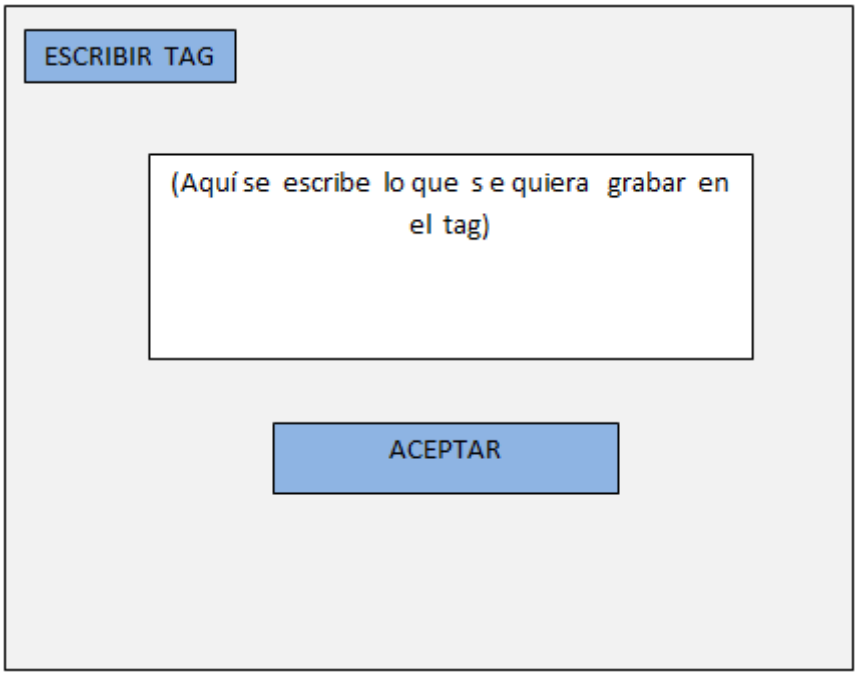


Figura 28. Vista de escribir Tag

4.6.4 Opción inventario

Esta opción estará compuesta de dos pantallas, en la primera el usuario realizara un login con el fin de tener acceso a los recursos de la pantalla inventario.

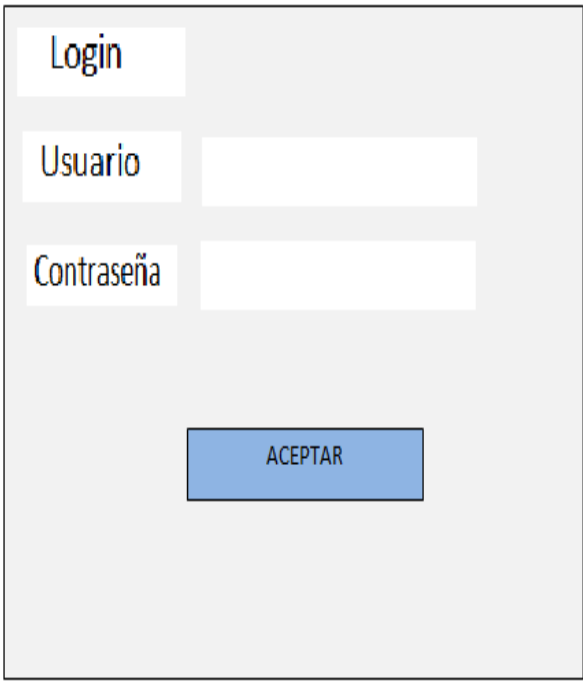


Figura 29. Pantalla de login

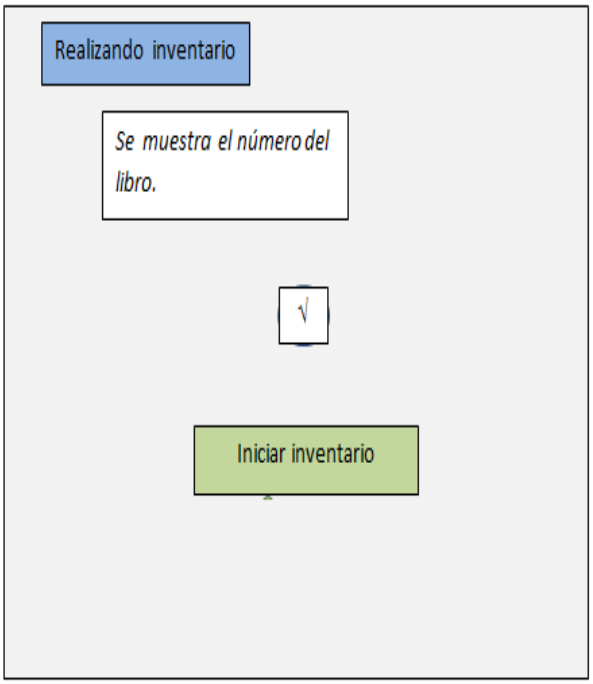


Figura 30. Pantalla de inventario

4.7 IMPLEMENTACIÓN DE LA APLICACIÓN

4.7.1 REQUERIMIENTOS DE HARDWARE SOFTWARE

4.7.1.1 ANDROID STUDIO

Android Studio es la IDE oficial para el desarrollo de aplicaciones para Android, basado en IntelliJ IDEA. La primera versión de beta fue lanzada en noviembre de 2007 con el nombre de build 0.1, a partir de entonces las cosas han avanzado rápidamente y han salido nuevas versiones[37], [38] .

Versión 1.0 Apple Pie - Salió en septiembre del 2008.
Versión 1.1 Banana Bread - Salió en febrero 2009.
Versión 1.5 Cup Cake - Salió en abril 2009

Versión 1.6 Donut - Salió en septiembre 2009
Versión 2.0 Eclair - Salió en octubre 2009
Versión 2.2 Froyo - Salió en mayo 2010
Versión 2.3 Gingerbread - Salió en diciembre 2010
Versión 3.0 Honeycomb - Salió en febrero 2011
Versión 4 .0 Ice Cream Sandwich - Salió en octubre 2011
Versión 4.1 Jelly Bean - Salió en julio 2012
Versión 4.2 Jelly Bean (Gummy Bear) salió - octubre de 2012
Version 4.3 Jelly Bean (Michel) – Salió julio de 2013
Versión 4.4 KitKat - Salió en octubre 2013
Versión Android 5.0 Lollipop – Salió noviembre de 2014
Versión Android 6.0 Marshmallow - Salió 5 de octubre de 2015

El nuevo entorno de desarrollo Android Studio cuenta con un sencillo asistente de configuración, permite importar ejemplos y plantillas, permite ver, editar y pre visualizar los diseños de Android a través de múltiples tamaños de pantalla, idiomas e incluso versiones de API, analizar el rendimiento de las aplicaciones y mucho más herramientas que ayudarán a los desarrolladores a crear con mayor facilidad mejores aplicaciones para Android.

Características principales de Android Studio

Editor de código inteligente

En el núcleo de Android Studio es un editor de código inteligente capaz de finalización avanzada, refactorización, y análisis de código. Esta característica ayuda en al ahorro de tiempo a la hora de escribir código.

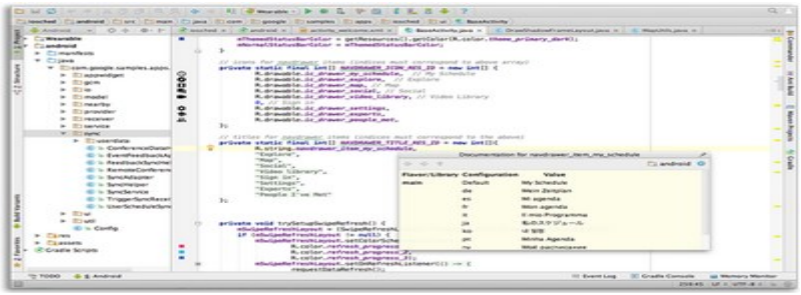


Figura 31. Editor de código inteligente[39]

Plantillas de código y la integración GitHub

Los nuevos asistentes de proyectos hacen que sea más fácil que nunca para iniciar un nuevo proyecto.

Iniciar proyectos utilizando el código de plantilla para los patrones como los cajones de navegación y ver su buscapersonas, e incluso importar ejemplos de código de Google desde GitHub.

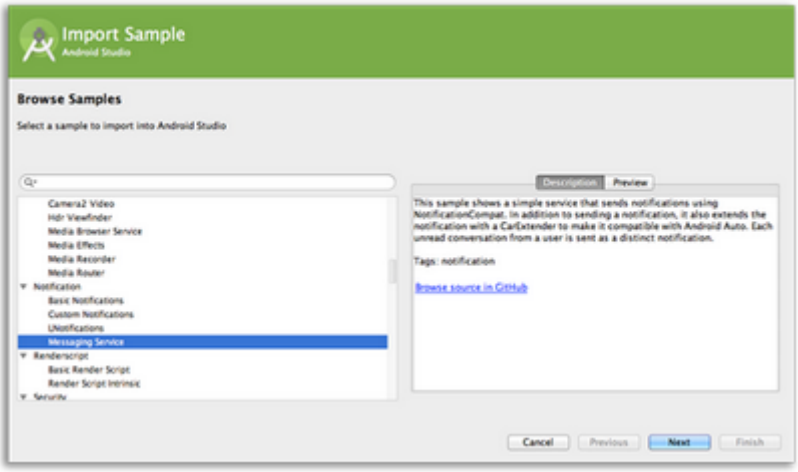


Figura 32. Plantillas de código[39]

Desarrollo de aplicaciones para múltiples pantallas

Construir aplicaciones para teléfonos Android, tabletas, Android Wear, TV Android, Android auto y Google Glass. Con el nuevo proyecto Android Ver y módulo de soporte en Android de estudio, es más fácil gestionar proyectos de aplicaciones y recursos.

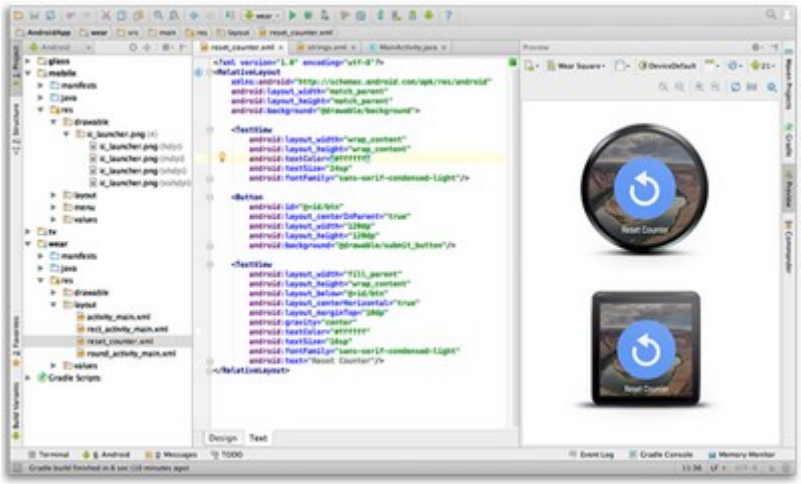


Figura 33. Soporte para múltiples pantallas[39].

Dispositivos virtuales para todas las formas y tamaños

Android Studio Viene pre-configurado Una imagen optimizada con emulador. El Administrador de Dispositivos virtuales Actualizada y simplificada

proporciona Perfiles de Dispositivo predefinidos párrafo Dispositivos Comunes Android

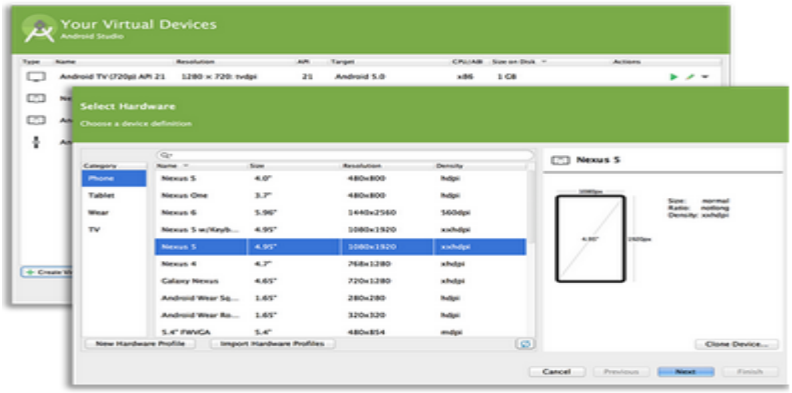


Figura 34. Diferentes dispositivo virtuales[39]

Otras características de Android Studio

- Construido sobre IntelliJ IDEA Community Edition, el IDE Java populares por JetBrains.
- Sistema de construcción a base de Gradle flexible.
- Construir variantes y generación APK múltiple. Apoyo plantilla ampliado para los servicios de Google y varios tipos de dispositivos.
- Editor de diseño Rich con soporte para la edición de tema.
- Herramientas para facilidad de uso, compatibilidad de versiones, y otros problemas.
- Incorporado soporte para Google Cloud Platform, lo que facilita la integración de Google Cloud Mensajería y App Engine.

Windows

Microsoft® Windows® 7.8 / Vista (32 o 64 bits)
2 GB de RAM mínimo, 4 GB de RAM recomendado
400 MB de espacio en disco duro
Al menos 1 GB para Android SDK, imágenes del sistema emulador, y cachés
1280 x 800 resolución de pantalla mínima
Java Development Kit (JDK) 7
Opcional para emulador acelerada: procesador Intel® con soporte para Intel® VT-x, Intel® EM64T (Intel® 64) y Execute Disable (XD) la funcionalidad de bit

Mac OS X

Mac® OS X® 10.8.5 o superior, hasta 10,9 (Mavericks)
2 GB de RAM mínimo, 4 GB de RAM recomendado
400 MB de espacio en disco duro
Al menos 1 GB para Android SDK, imágenes del sistema emulador, y cachés
1280 x 800 resolución de pantalla mínima
Java Runtime Environment (JRE) 6
Java Development Kit (JDK) 7

Opcional para emulador acelerada: procesador Intel® con soporte para Intel® VT-x, Intel® EM64T (Intel® 64) y Execute Disable (XD) la funcionalidad de bit

En **Mac O**S, ejecute Estudio Android con Java Runtime Environment (JRE) 6 para la renderización de fuentes optimizado. A continuación, puede configurar el proyecto para utilizar Java Development Kit (JDK) 6 o JDK 7.

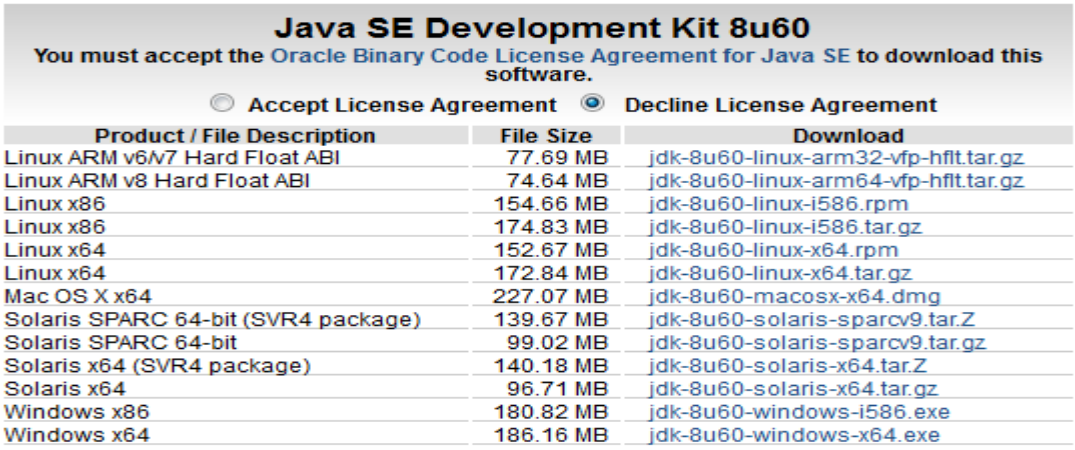
Linux

Escritorio GNOME o KDE
GNU C Library (glibc) 2.15 o posterior
2 GB de RAM mínimo, 4 GB de RAM recomendado
400 MB de espacio en disco duro
Al menos 1 GB para Android SDK, imágenes del sistema emulador, y cachés
1280 x 800 resolución de pantalla mínima
Oracle Java Development Kit (JDK) 7

Probado en Ubuntu® 14.04, de confianza Thar (distribución de 64 bits capaz de ejecutar aplicaciones de 32 bits).

4.7.1.2 JAVA JDK

Las librerías de Java JDK (Java Development Kit) son indispensables para ejecutar el emulador de Android y algunas herramientas de depuración. Hay que tener en cuenta que no es suficiente con instalar el Java JRE (Java Runtime Edition) que se usa comúnmente para elementos de internet, es necesario instalar el JDK 5.0 o superior completo para tener acceso a todas las herramientas de desarrollo.
El instalador del JDK se puede encontrar en el sitio de descargas de Oracle, donde se debe aceptar el acuerdo de licencia para escoger la versión correspondiente a al sistema y realizar la descarga[40].



Java SE Development Kit 8u60

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

○ Accept License Agreement ◉ Decline License Agreement

Product / File Description	File Size	Download
Linux ARM v6/v7 Hard Float ABI	77.69 MB	jdk-8u60-linux-arm32-vfp-hflt.tar.gz
Linux ARM v8 Hard Float ABI	74.64 MB	jdk-8u60-linux-arm64-vfp-hflt.tar.gz
Linux x86	154.66 MB	jdk-8u60-linux-i586.rpm
Linux x86	174.83 MB	jdk-8u60-linux-i586.tar.gz
Linux x64	152.67 MB	jdk-8u60-linux-x64.rpm
Linux x64	172.84 MB	jdk-8u60-linux-x64.tar.gz
Mac OS X x64	227.07 MB	jdk-8u60-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	139.67 MB	jdk-8u60-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	99.02 MB	jdk-8u60-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	140.18 MB	jdk-8u60-solaris-x64.tar.Z
Solaris x64	96.71 MB	jdk-8u60-solaris-x64.tar.gz
Windows x86	180.82 MB	jdk-8u60-windows-i586.exe
Windows x64	186.16 MB	jdk-8u60-windows-x64.exe

Figura 35. Versiones de JDK disponibles para descargar en el sitio oficial de Oracle [40].

Configuración de java

Una vez terminado de instalar el JDK, lo debemos configurar para lo cual se realiza lo siguiente:

- Damos **Clic Derecho a Mi PC y nos vamos a Propiedades.** Y nos aparecerá una ventana donde aparece la información de nuestro sistema, **Damos clic en Configuración Avanzada del Sistema**.

- Nos vamos a la pestaña de **Opciones Avanzadas** y damos clic en **Variables de Entorno.**

- Buscamos en las **Variables del sistema**, la variable llamada **Path** y se le da clic en **Editar.**

- Se abrirá una nueva ventana y se le agrega esto: **;C:\Program Files\VERSION DE TU JAVA\bin** si es que se deja la ruta por defecto del instalador, si se instala en otra carpeta, se debera poner la dirección donde se encuentra instalado el JDK. Y se da Clic en **Aceptar**.

- Después ahí mismo en Variables del Sistema, se creara una nueva variable llamada Classpath y tendrá la dirección de nuestro src.zip en este caso sería esta: **;C:\Program Files\Java\VERSION DE TU JAVA\src.zip** y se le da clic en **Aceptar**.

Luego se da Clic en Aceptar en Variables de Entorno y en la Ventana de Opciones Avanzadas.

Ahora en la **Consola de Comandos de Windows 7**, o al famoso CMD, una vez abierto escribimos javac y deberá mostrar este contenido.

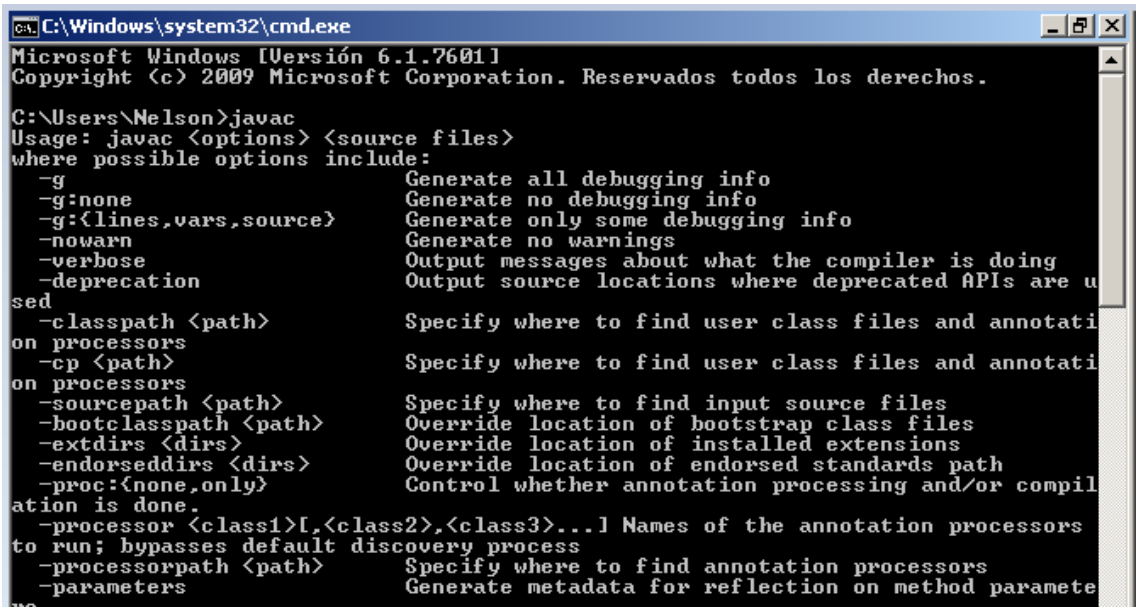


Figura 36. Vista de la consola de Windows si el java está configurado

Y si se muestra esto, quedó instalado el Java JDK en nuestro sistema de Windows 7, y podemos crear los programas y compilarlos utilizando un IDE como Android Studio, Netbeans o JCreator.

4.7.1.3 MySQL WorkBench 6.2 CE

Los inicios de MySQL se remontan a la necesidad de conectar el gestor [mSQL] a las propias tablas de [MySQL AB], para lo que se comenzó con unas rutinas a bajo nivel. Después de diversas pruebas se hizo más evidente que mSQL no brindaba la suficiente flexibilidad para estos requerimientos, motivo por el cual se comenzaron a desarrollar nuevas funciones, que dieron lugar a una interfaz SQL totalmente compatible con MySQL. Desde sus comienzos se han ido desarrollando multitud de versiones, siendo actualmente la versión estable 5.6.27 . El origen del nombre “MySQL” no se conoce con certeza, parece ser que durante los últimos años sus librerías han llevado el prefijo “my”, lo que justamente coincide con el nombre de la hija de un miembro del equipo de desarrollo[41].

Requerimientos de Hardware:

- CPU: Intel Core o Xeon 3GHz (o Dual Core 2GHz) o su equivalente CPU AMD.

- Núcleos: Simple (se recomiendan núcleos Dual/Quad Core).
- RAM: 4 GB (se recomienda 6 GB).
- Aceleradores Gráficos: nVidia o ATI con soporte de OpenGL 1.5 o mayor.
- Resolución de la pantalla: se recomienda 1920×1200, 1280×1024 como mínimo.

Sistemas Operativos

- Linux: x86_64. Los usuarios deben descargar y compilar los binarios de 32 bits de las fuentes dadas.
- Mac OS: x86 (solamente 32 bits)
- Windows: x86 y x64_64

Para usuarios Windows:
Para Windows XP SP3 y Windows Vista instalar el .Net 4.0 framework (Solamente perfil full o incluso el perfil de cliente).
Para usuarios de sistemas Windows superiores instalar el .Net 4.0 framework con el perfil de cliente.
Para la tarjeta gráfica se necesitan los últimos controladores de nVidia o ATI.
Instalar el Microsoft Visual C++ 2010 Redistributable Package (x86). [Frequent Asked Questions, “System Requirements”[41].

4.7.2 INTERFAZ DE USUARIO

La interfaz de usuario es el medio con que el usuario puede comunicarse con una máquina, equipo, o dispositivo, y comprende todos los puntos de contacto entre el usuario y el equipo.
Normalmente suelen ser fáciles de entender y fáciles de accionar, amigables e intuitivos. A continuación se dará a conocer algunos de los principales elementos que componen una aplicación en Android Studio:

AndroidManifest.xml

Este fichero se crea automáticamente a la hora de crear un nuevo proyecto en Android Studio para desarrollar una aplicación Android. Éste archivo, se encuentra en la raíz de los directorios de nuestro proyecto y es uno de los

archivos más importantes de nuestra aplicación ya que en él se declaran todos los componentes que utilicemos en nuestra aplicación Android.

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="stocktaking"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".MainActivity"
            android:label="stocktaking" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name=".consultar"
            android:label="consultar" >
        </activity>
        <activity
            android:name=".escribir"
            android:label="escribir" >
        </activity>
        <activity
            android:name=".login"
            android:label="@string/title_activity_login" >
        </activity>
        <activity
            android:name=".inventario"
```

Figura 37. Archivo manifest.xml

Todas las aplicaciones deben contener este archivo por convención. El nombre debe permanecer intacto, ya que se usa como referencia para el **parsing** de nuestra aplicación. El **nodo raíz** de este documento se representa con la etiqueta <manifest> y por obligación debe contener un hijo de tipo <application>.

Elementos básicos del AndroidManifest.xml[42]**.**

Para empezar el *AndroidManifest.xml* empieza con una definición del XML en el que se codifica y va seguido de la etiqueta. Dentro de ésta etiqueta podemos encontrar los siguientes atributos:

- **Xmlsn android**. No se debe cambiar nunca, ya que es el **namespace** del archivo.
- **package:** es el nombre del paquete JAVA de nuestra aplicación. Con éste, se referencia a nuestra aplicación en el *Google Play Store* y en nuestro dispositivo Android.
- **Android versionCode:** hace referencia al número de versión de desarrollo nuestra aplicación, cada versión final que deseemos publicar tiene que tener un número de versión distinto.
- **Android versionName:** es el número de versión de nuestra aplicación.

La siguiente etiqueta que nos encontramos en un fichero AndroidManifest.xml es <application></application> y dentro de esta etiqueta se crean por defecto los siguientes atributos:

- **Android allowBackup:** permite o no según su valor (true o false) hacer una copia de seguridad y restaurar la infraestructura.
- **Android icon:** hace referencia al icono de nuestra aplicación a través de un recurso *Drawable*.
- **Android label:** hace referencia al nombre de nuestra aplicación. Es simplemente una etiqueta. Para cambiar el nombre de nuestra aplicación debemos hacerlo desde el fichero *strings.xml.* Está referenciado por un recurso de tipo *String.*
- **Android theme:** es el tema (apariencia) que se le aplica a nuestra aplicación Android.

Dentro de la etiqueta se define la actividad principal de nuestra aplicación Android. Tenemos que añadir cada Actividad que tengamos en nuestra aplicación y podemos asignarle o no atributos. Dentro de la etiqueta podemos tener los siguientes atributos:

- **android:name:** es el nombre de la clase JAVA de nuestra actividad.
- **android:label:** es el texto que aparecerá en la barra de acción (Action Bar) de nuestra aplicación. Está referenciado por un recurso de tipo *String*.
- **android:theme:** es el tema (apariencia) que se le aplica a nuestra aplicación Android.
- **android:configChanges:** con éste atributo la aplicación no detectará los cambios de orientación y por ocultación del teclado físico.
- **android: screenOrientation:** *c*on éste atributo podemos hacer que nuestro d*ispositivo solo tenga disposición en modo Port*rait o en modo *Landscape.*

Otras etiquetas que son frecuentemente utilizadas en el desarrollo de una aplicación son:

- Cuando nuestra aplicación necesita usar una característica especial de nuestro dispositivo añadimos un permiso con la etiqueta <uses-permission>.
- El usuario puede crear sus propios permisos e incluye las características de éste en la etiqueta <permission>.
- Cuando queremos declarar una característica de hardware o software que requiere ser utilizada por la aplicación usamos la etiqueta.

Activity

En Android una actividad es un conjunto de acciones (tocar la pantalla para apretar un botón, para escribir con el teclado, etc) que son una iteración directa con el usuario y que afectan a una parte de la aplicación. Una actividad se caracteriza por tener un **ciclo de vida**

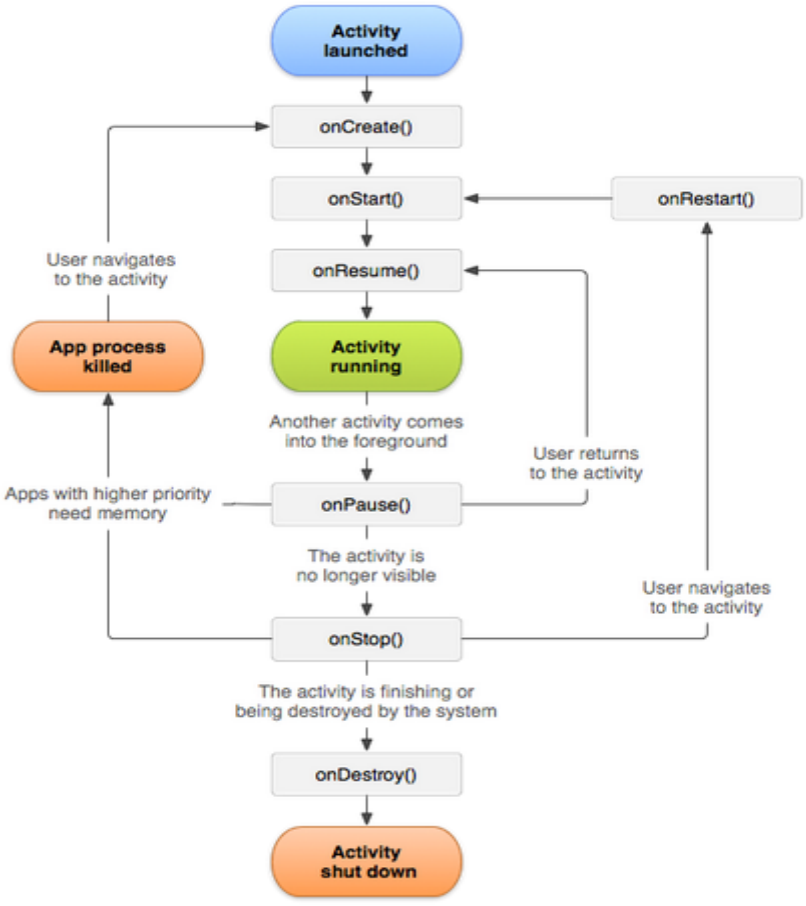


Figura 38. Ciclo de vida de una de una activity [30]

Archivo strings.xml

Dentro de la carpeta “res” encontraremos todos aquellos recursos tercerizados para nuestra aplicación. Esta práctica de excluir los atributos de la aplicación a

través de archivos externos, permite reducir la complejidad de diseño en las interfaces.

Uno de los recursos más relevantes es el archivo strings.xml que se encuentra dentro de la subcarpeta valúes. Este fichero almacena todas las cadenas que se muestran en los **widgets**(controles, formas, botones, vistas y demás) de nuestras actividades[43].

Por ejemplo, si tuvieses un botón cuyo título es *"Presiona aquí"*, es recomendable incluir dicha cadena en tu archivo strings.xm

Al abrir el archivo, verás que se encuentran tres nodos del tipo <string>: "app_name", "action_settings" y "hello_world".

```
<resources>
    <string name="app_name">stocktaking</string>

    <string name="hello_world">Hello world!</string>
    <string name="action_settings">Settings</string>
    <string name="textView">BIENVENIDO A LA  BIBLIOTECA DE LA  UDFJDC</string>
    <string name="imageView"> </string>
    <string name="imageButton"> </string>
    <string name="imageButton2"> </string>
    <string name="imageButton3"> </string>

    <string name="title_activity_consultar">consultar</string>
    <string name="textView2">Informaciòn Bàsica</string>
    <string name="textView3"></string>
    <string name="imageButton4"> </string>


    <string name="title_activity_escribir">escribir</string>
    <string name="textView4">Datos de identificaciòn del libro(IdNFC)</string>
    <string name="editText">  </string>
    <string name="imageButton5"> </string>

    <string name="title_activity_login">login</string>
    <string name="textView5">Usuario</string>
    <string name="textView6">Contraseña</string>
    <string name="editText2">  </string>
    <string name="editText3">  </string>
    <string name="button">Aceptar</string>

    <string name="title_activity_inventario">inventario</string>
    <string name="textView7">IdNFC</string>
    <string name="textView8"> </string>
    <string name="checkBox"> </string>
    <string name="button2">REPORTE</string>
```

Figura 39. Vista del archivo string.xml de la aplicación.

El archivo strings.xml es muy útil para los desarrolladores. Una de sus grandes utilidades es facilitar el uso de **múltiples idiomas** en tu aplicación. Esto se debe a que puedes externalizar las cadenas del código java y seleccionar la **versión** del archivo strings.xml con el lenguaje necesitado.

Layout[44]

En la carpeta layout encontrarás los archivos de diseño de todas tus actividades. Este archivo representa el diseño de la interfaz dela aplicación. En él se establecerán todos los widgets que van a utilizar.

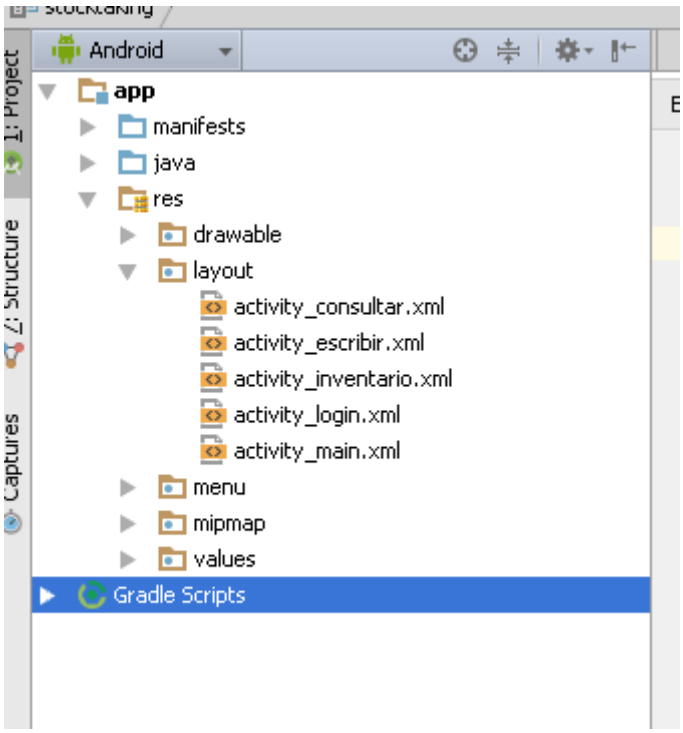


Figura 40. Archivo layout de la aplicación.

.

Este archivo de diseño comienza con un nodo raíz llamado <RelativeLayout>. Un Layout es el contenedor principal que define el orden y secuencia en que se organizarán los widgets en nuestra actividad. Existen varios tipos de Layouts, como por ejemplo el LinearLayout, GridLayout, FrameLayout.

Android Studio crea por defecto un RelativeLayout que permite crear un grupo de componentes con ubicaciones relativas. Quiere decir que se ubicaran por referencias y no por valores absolutos. Esto permite ajustar nuestras aplicaciones a cualquier tipo de pantalla para dispositivos móviles.

Utilidad de los atributos para <RelativeLayout>[44]**:**

- layout_width: Es el ancho que tendrá el layout dentro de la actividad. Aunque se puede especificar con unidades personalizadas, es recomendable usar match_parent para ajustarlo al ancho del dispositivo.
- layout_height: Representa la dimensión vertical del layout. Usa match_parent para ajustarla al dispositivo.
- paddingLeft, paddingRight: Es el espacio lateral existente entre el contorno del Layout y los widgets. Su valor apunta al archivo de recursos dimens.xml ubicado en la carpeta “values”. Este archivo contiene nodos de tipo <dimen> con valores density-indepent píxeles (dp). Para estos paddings se usa el nodo “activity-horizontal-margin”, cuyo valor estándar son 16dp.
- paddingTop, paddingBottom: Es el espacio vertical existente entre el contorno del Layout y los widgets. Su valor es igual a el nodo “activity-vertical-margin” de dimens.xml.
- context: Define el nombre del archivo Java que contiene la actividad donde el Layout será acogido.

/app/build.gradle [44]

Contiene información necesaria para la compilación del proyecto, por ejemplo la versión del SDK de Android utilizada para compilar, la mínima versión de Android que soportará la aplicación, referencias a las librerías externas utilizadas.

4.8 IMPLEMENTACIÓN DE LA INTERFAZ GRÁFICA

Esta se realiza en Android Studio Versión 1.3.2 y utilizando el jdk1.7.0_79. El soporte mínimo es para la API 16.



Figura 41. Interfaz principal

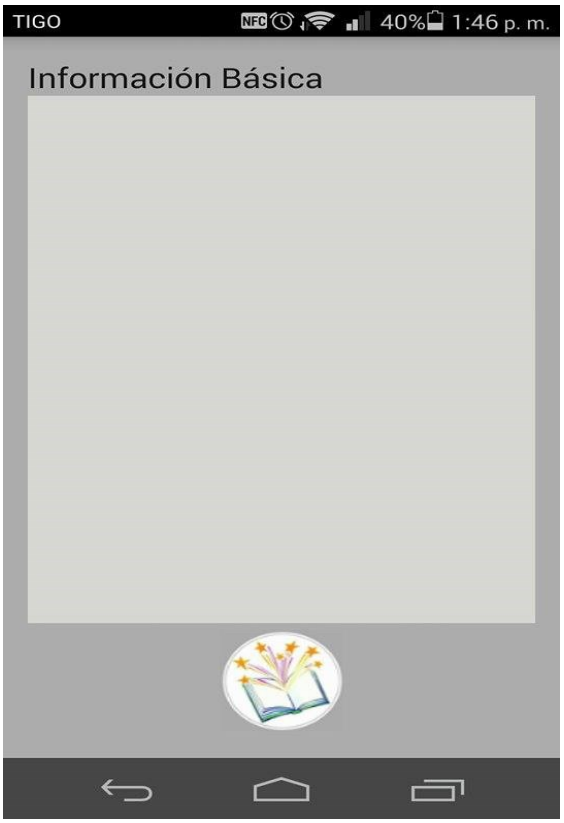


Figura 42. Interfaz de lectura

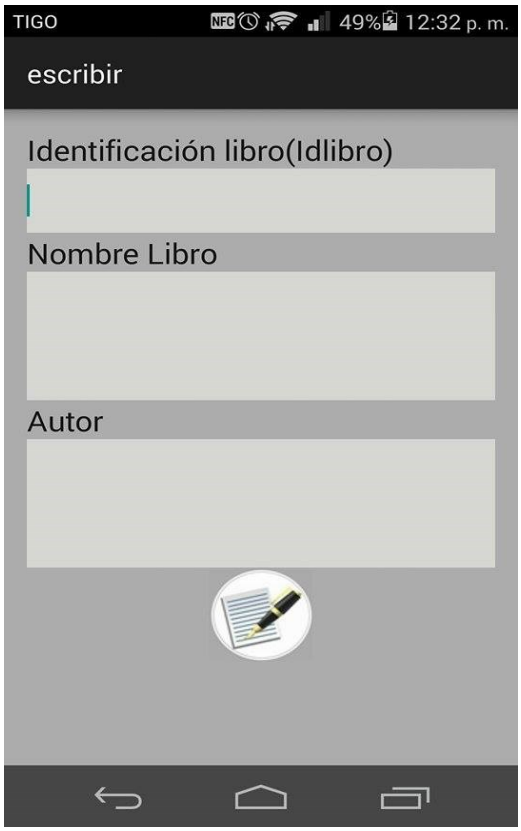


Figura 43. Interfaz de escritura

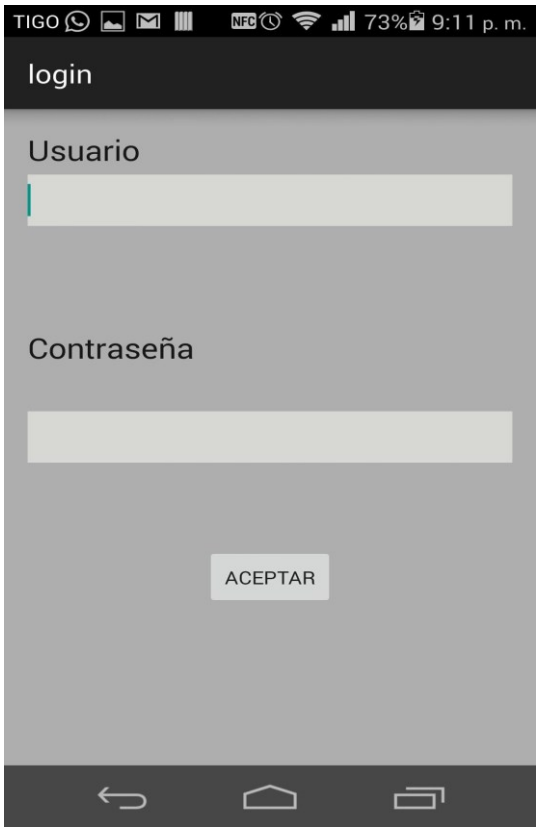


Figura 44. Interfaz de login

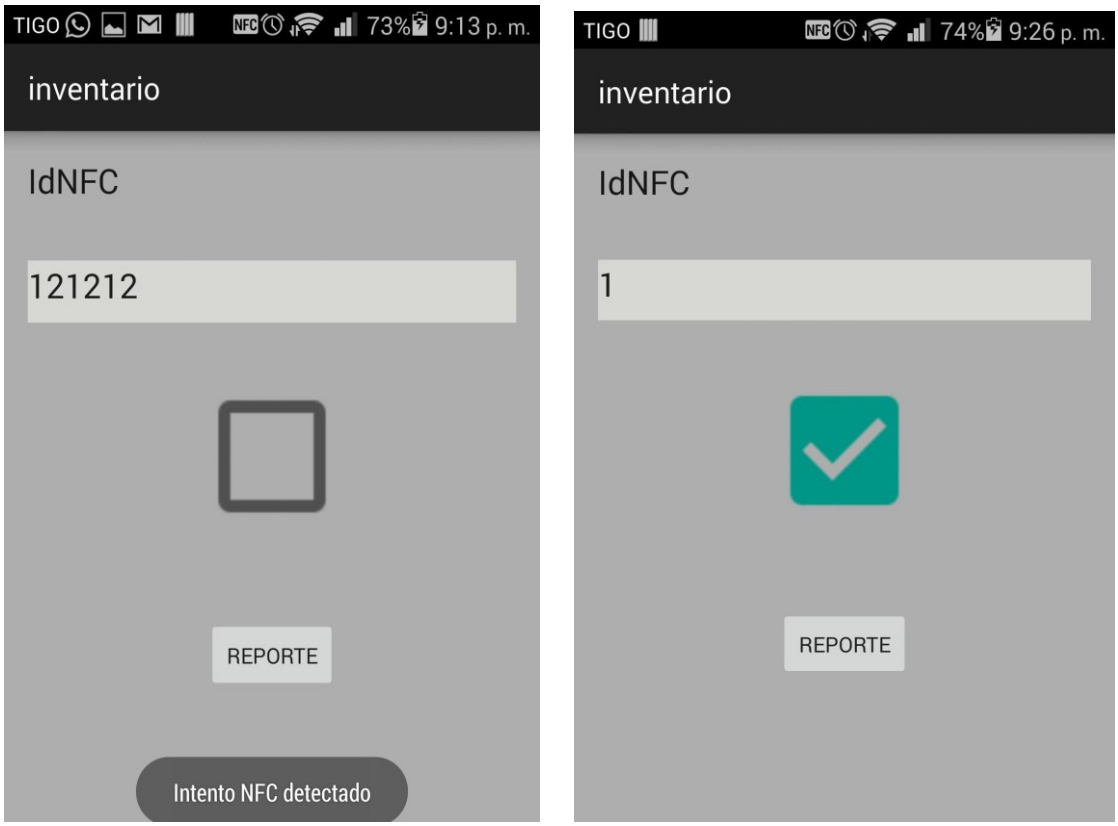


Figura 42. Interfaz para realizar inventario

4.9 PROGRAMACION DE LA APLICACIÓN

4.9.1 Archivo Manifest.xml

```
</application>
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.NFC"/>
```

Resumen: En el archivo manifest.xml se agregan los permisos que necesite la aplicación, en este caso los correspondientes al internet conexión y el uso del terminal NFC.

4.9.2 Pantalla principal

```
ImageButton consulta = (ImageButton) findViewById(R.id.imageButton);
ImageButton escribe = (ImageButton) findViewById(R.id.imageButton2);
ImageButton invetaria = (ImageButton) findViewById(R.id.imageButton3);


consulta.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent abrir = new Intent(getApplicationContext(),consultar.class);
        startActivity(abrir);
    }
});
escribe.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent abrir1 = new Intent(getApplicationContext(),escribir.class);
        startActivity(abrir1);
    }
});
invetaria.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent abrir2 = new Intent(getApplicationContext(),login.class);
        startActivity(abrir2);
    }
});
```

Resumen: Lo primero que se hace es relacionar los elementos creados en el XML con la clase correspondiente(Main Activity), y para lo cual se usa el método **findViewById()**.

Luego se crea un objeto anónimo para recoger los eventos de la clase View.OnclickListener(). En particular se crean e instancian los objetos(consulta, escribe e inventaría) encargados de llamar las activitys a través del método onClick(), el cual realiza las tareas correspondientes.

4.9.3 Lectura de tags NFC

```
public class consultar extends Activity {
    NfcAdapter nfcAdapter;
    TextView txtTagContent;

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

        nfcAdapter = NfcAdapter.getDefaultAdapter(this);

        txtTagContent = (TextView) findViewById(R.id.textView3);

    }
```

Resumen: Se crean la variable *txtTagContent* y se relaciona con su texView correspondiente, la variable *nfcAdapter* será la encargada de conectar con el lector NFC del dispositivo.

```
@Override
protected void onNewIntent(Intent intent) {
    super.onNewIntent(intent);

    if (intent.hasExtra(NfcAdapter.EXTRA_TAG)){
        Toast.makeText(this, "Intento NFC detectado", Toast.LENGTH_SHORT).show();

        {
            Parcelable[] parcelables = intent.getParcelableArrayExtra(NfcAdapter.EXTRA_NDEF_MESSAGES);
            if (parcelables !=null && parcelables.length>0){
                readTextFromMessage((NdefMessage)parcelables[0]);

            }else{
                Toast.makeText(this, "No se encontró mensaje NDEF en la Tag", Toast.LENGTH_SHORT).show();
            }

        }


    }
}
```

Resumen: Este método se encarga de detectar el tag, seguidamente se crea una variable tipo arreglo para recuperar la información, la información recuperada se coloca en formato NDF.

```
public String getTextFromNdefRecord(NdefRecord ndefRecord)
{
    String tagContent = null;
    try {
        byte[] payload = ndefRecord.getPayload();
        String textEncoding = ((payload[0] & 128) == 0) ? "UTF-8" : "UTF-16";
        int languageSize = payload[0] & 0063;
        tagContent = new String(payload, languageSize + 1,
                payload.length - languageSize - 1, textEncoding);
    } catch (UnsupportedEncodingException e) {
        Log.e("getTextFromNdefRecord", e.getMessage(), e);
    }
    return tagContent;
}
```

Resumen: Cada NdefMessage que se extrae contiene uno o varios NdefRecord, que son las entidades que contienen los mensajes de interés.

En cada objeto NdefRecord, el mensaje se almacena como una tabla de bytes, llamada Payload, a la que se accede mediante el método getPayload() del objeto NdefRecord. Es preciso convertir estos datos en función del formato del registro NdefRecord. Para ello, el objeto NdefRecord expone los métodos: getTnf(), que devuelve el formato del NdefRecord, y getType(), que devuelve el subtipo en el caso de que el tipo de NdefRecord seaTNF_WELL_KNOWN.

En el caso de un NdefRecord cuyo subtipo sea RTD_TEXT, los primeros bytes del payload contienen información acerca de la codificación del contenido.
Un primer elemento define la codificación, UTF-8 o UTF-16, así como la longitud del código que le sigue.

El siguiente elemento define el código del lenguaje utilizado. La extracción del texto del registro consiste en determinar la codificación de la cadena de caracteres útiles de payload y, a continuación, transcribir la tabla de bytes como una cadena de caracteres en función de su codificación.

```
private void readTextFromMessage(NdefMessage ndefMessage) {
    NdefRecord[] ndefRecords = ndefMessage.getRecords();

    if (ndefRecords !=null && ndefRecords.length>0){
        NdefRecord ndefRecord = ndefRecords[0];


        String tagContent = getTextFromNdefRecord(ndefRecord);


        txtTagContent.setText(tagContent);

    }else{
        Toast.makeText(this, "No se encontró ningun registro en la Tag", Toast.LENGTH_SHORT).show();
    }
}
```

Resumen: el método *readTextFromMessage* determina si había mensajes en el tag de ser verdadero procede a mostrarlos en el *Texview*.

4.9.4 Escritura de tags NFC

```
btnWrite.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        try{

            if(myTag == null){

                Toast.makeText(getBaseContext(),"tag no encontrado",Toast.LENGTH_LONG).show();
            }else{

                write(message.getText().toString(), myTag);

                Toast.makeText(getBaseContext(),"tag escrito correctamente",Toast.LENGTH_LONG).show();
            }
        }catch(IOException e){

            Toast.makeText(getBaseContext(),"error de escritura",Toast.LENGTH_LONG).show();
            e.printStackTrace();
        }catch(FormatException e){

            Toast.makeText(getBaseContext(),"error de escritura",Toast.LENGTH_LONG).show();
            e.printStackTrace();
        }
    }
});
```

Resumen: Como en la lectura, la escritura de un tag se realiza, habitualmente, tras comprobar la compatibilidad del tag detectado por la intención.

```
private void write(String text, Tag tag) throws IOException, FormatException{

    NdefRecord[] records = {createRecord(text)};

    NdefMessage message = new NdefMessage(records);

    Ndef ndef = Ndef.get(tag);
    ndef.connect();
    ndef.writeNdefMessage(message);
    ndef.close();
}
```

Resumen: Se crea un mensaje NFC con formato NdefMessage, que es, independiente de la tecnología de tags subyacente. La escritura de un tag sigue la estructura de un objeto de tipo NdefMessage: el mensaje incluye uno o varios registros (objetos de tipo NdefRecord), que pueden ser de distintos tipos.

El método get(tag) puede devolver null, en caso de que el tag no esté soportado por la tecnología NDEF.

Para realizar la escritura de un tag se inicializa la operación de escritura mediante el Método connect(), escribir el tag con el método writeNdefMessage (NdefMessage) y cerrar la conexión invocando al método close().

```
@SuppressLint("NewApi") private NdefRecord createRecord(String text) throws UnsupportedEncodingException{
    String lang = "us";
    byte[] textBytes = text.getBytes();
    byte[] langBytes = lang.getBytes("US-ASCII");
    int langLength = langBytes.length;
    int textLength = textBytes.length;
    byte[] payLoad = new byte[1 + langLength + textLength];

    payLoad[0] = (byte) langLength;

    System.arraycopy(langBytes, 0, payLoad, 1, langLength);
    System.arraycopy(textBytes, 0, payLoad, 1+langLength, textLength);

    NdefRecord recordNFC = new NdefRecord(NdefRecord.TNF_WELL_KNOWN, NdefRecord.RTD_TEXT, new byte[0], payLoad);

    return recordNFC;

}
```

Resumen: La construcción de un mensaje NdefRecord utiliza el siguiente formato:

```
NdefRecord record = new NdefRecord (short tnf, byte[] type, byte[] id, byte[] payload);
```

Dónde:

- **tnf** representa el formato del registro creado.
- **type** indica el tipo de registro.
- **id** permite asignar un identificador al registro (resulta interesante si el mensaje incluye varios registros).
- **payload** es una tabla de bytes que contiene el mensaje propiamente dicho.

El payload de un registro de tipo texto debe comenzar con la información relativa a la codificación del texto y, a continuación, el propio texto codificado.

- El primer bit define el formato del texto:UTF-8 (bit a 0) o UTF-16 (bit a 1).
- El segundo bit vale 0 (inutilizado en la norma actual).
- Los 6 siguientes bits indican la longitud de la codificación del idioma.
- A continuación, se incluye el valor de la codificación del idioma (codificado en "US ASCII", por ejemplo), y a continuación el propio texto.

- La longitud total de la tabla de bytes que constituye el payload será, por tanto, la suma del byte de información acerca de la codificación, el tamaño del código de idioma y el tamaño del mensaje.

4.9.5 Pantalla de login

```
    private void ReadFromServer(){
        SERVER= URL.getText().toString();
        DATABASE=DB.getText().toString();
        UID=USER.getText().toString();
        PASSWORD=PASS.getText().toString();
        ExecutorService ejecutorSub = Executors.newCachedThreadPool();
        ejecutorSub.execute(sqlThreadR);
        ejecutorSub.shutdown();
    }
Thread sqlThreadR = new Thread() {
    public void run() {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            Connection conn = DriverManager.getConnection(
                    SERVER + DATABASE, UID, PASSWORD);
            Statement st = conn.createStatement();
            ResultSet rs = st.executeQuery("select * from `Name`;");
            int c = 1;
            while (rs.next()) {
                Name=rs.getString("name");
                System.out.println("Nombre leido: "+ Name);
            }
            conn.close();
            runOnUiThread(() -> { NAMER.setText(Name); });
        } catch (SQLException se) {
            runOnUiThread(() -> {
                    Toast.makeText(getApplicationContext(), "Imposible conectar con el servidor: "+ SERVER,
                            Toast.LENGTH_LONG).show();
            });
        } catch (ClassNotFoundException e) {
            System.out.println("oops! No se encuentra la clase. Error: "
                    + e.getMessage());
        }
    }
};
```

Resumen: Para realizar la conexión con la base de datos creada en Mysql lo primero que se debe agregar es el conector de mysql jdbc.jar en la carpeta “libs”. Se crea el método ReadFromServer() que se encargara de la tarea de la operación completa de conexión, dentro de este método se ejecuta una tarea secundaria(sqlTreadR) el cual recibirá la cadena de conexión*(server + database + user + password)*. Lo que sigue es realizar una consulta en formato sql, se captura el resultado y se cierra la conexión.

4.9.6 Realización del inventario

```
private void readTextFromMessage(NdefMessage ndefMessage) {
    NdefRecord[] ndefRecords = ndefMessage.getRecords();

    if (ndefRecords !=null && ndefRecords.length>0){
        NdefRecord ndefRecord = ndefRecords[0];


        String tagContent = getTextFromNdefRecord(ndefRecord);


        NAME.setText(tagContent);

        ReadFromServer();


    }else{
        Toast.makeText(this, "No se encontró ningun registro en la Tag", Toast.LENGTH_SHORT).show();
    }

}
```

Resumen: En la realización de inventario se combinan los métodos creados para leer los tag, realizar la conexión a la base de datos y las consulta sql.

```
Statement st = conn.createStatement();

 rs = st.executeUpdate("UPDATE libros SET idNFC = '1' WHERE idlibros="+ISBN);

conn.close();
```

Resumen: Una vez leído el tag y realizada la conexión a la base de datos se envia la sentencia SQL update con el fin de Insertar el idNFC correspondiente al libro que está siendo inventariado.

4.10 Diseño de la Base de datos.

Después de realizar un estudio sobre los principales requerimientos del sistema de almacenamiento, se determinaron los campos que debe llevar la base de datos.

Nombre de la tabla	**Libros**
Campo 1	Idlibros (int) PK
Campo 2	idNFC (TEXT)
Campo 3	Titulo (TEXT)
Campo 4	ISBN (TEXT)
Campo 5	Edicion (TEXT)
Campo 6	Edditorial (TEXT)
Campo 7	Fecha (DATE)
Campo 8	Pais (TEXT)
Campo 9	Paginas(INT)
Campo 10	Idioma (TEXT)
Campo 11	Autor (FK)

Tabla 7. Campos de la tabla libros

Nombre de la tabla	**Usuarios**
Campo 1	Idusuarios(int) PK
Campo 2	nombre(TEXT)
Campo 3	identificacion (INT)
Campo 4	Institución (TEXT)
Campo 5	Status (BINARY)

Tabla 8. Campos de la tabla usuarios

Nombre de la tabla	**autores**
Campo 1	Idautor(int) PK
Campo 2	nombre(VARCHAR)
Campo 3	Nacionalidad (VARCHAR)

Tabla 9. Campos de la tabla autor

Nombre de la tabla	**prestamos**
Campo 1	Fecha_prestamo (DATETIME)

Campo 2	Fecha_ devolucion(DATETIME)
Campo 3	Devuelto (CHAR)
Campo 4	Idlibro (FK)
Campo 5	idestudiante (FK)
Campo 6	idusuario (FK)

Tabla 10 Campos de la tabla préstamos

Nombre de la tabla	**Estudiantes**
Campo 1	Idestudiante (INT)
Campo 2	Nombre (VARCHAR)
Campo 3	Código (INT)
Campo 4	Proyecto curricular (VARCHAR)
Campo 5	Dirección (VARCHAR)
Campo 6	Teléfono(INT)

Tabla 11. Campos de la tabla estudiantes.

4.11 IMPLEMENTACION DE LA BASE DE DATOS

La implementación de la base de datos se realizó en el software MysqlWorbech 6.3, y se obtuvo el siguiente modelo relacional Figura 45.

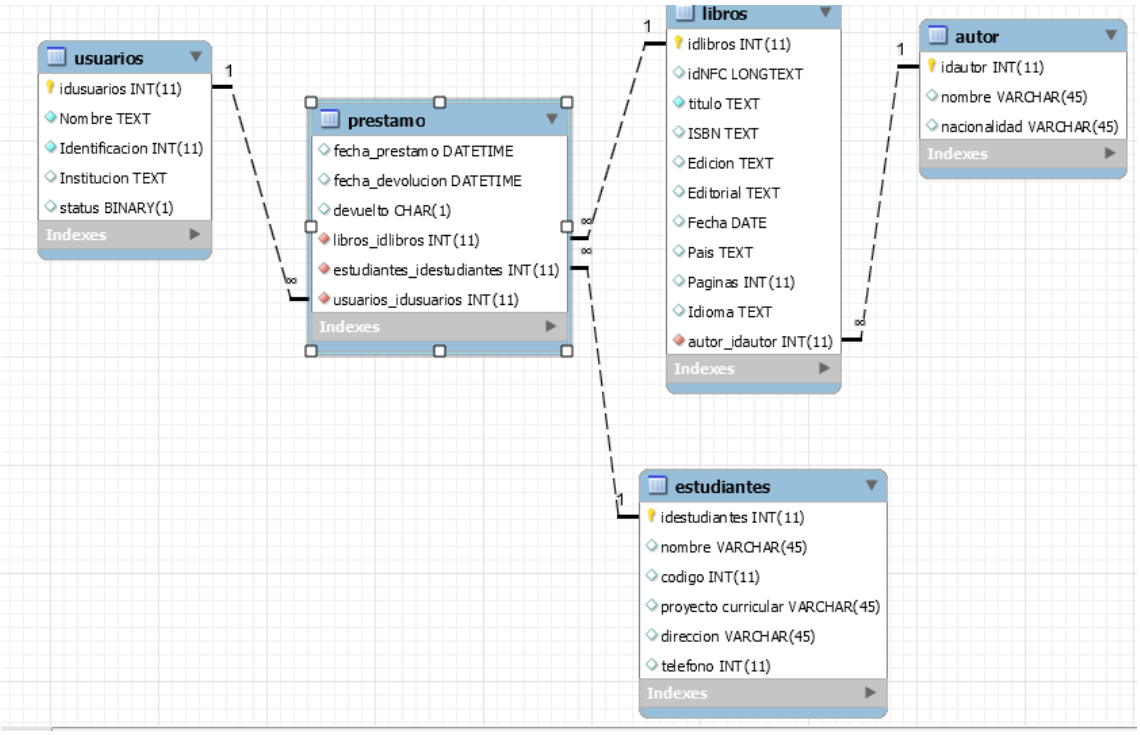


Figura 45. Modelo relacional de la base de datos

Capítulo 5

5 Pruebas y análisis de resultados

5.1 Pruebas

5.1.1 Prueba de escritura

La siguiente prueba se realiza con los tag NXP MIFARE Ultralight (Ultralight C) – NTAG203 y NXP mifare classic 1k con el dispositivo móvil Huawai P7.

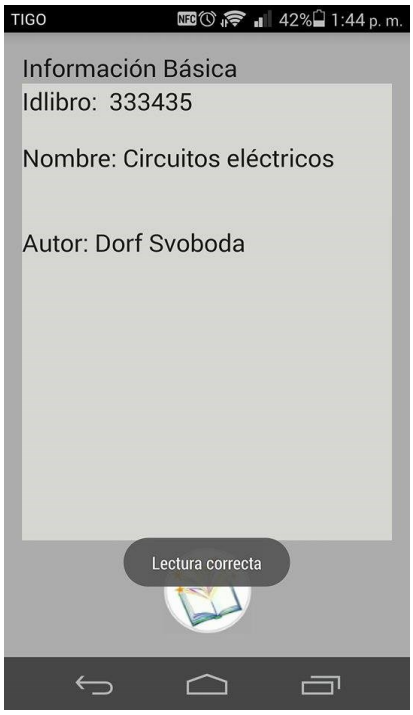




Referencia de Tag	Resultado
Ntag 203	Tag escrita!
Ntag 203	Tag escrita!
Ntag 203	Tag escrita!
Ntag 203	Tag escrita!
Ntag 203	Tag escrita!
Classic 1k	Tag escrita!
Classic 1k	Tag escrita!
Classic 1k	El tag no puede ser Null!
Classic 1k	Tag escrita!
Classic 1k	Tag escrita!
Ntag 203	Tag escrita!
Ntag 203	Tag escrita!
Ntag 203	Tag escrita!
Ntag 203	Tag escrita!
Ntag 203	Tag escrita!
Classic 1k	Tag escrita!
Ntag 203	Tag escrita!
Ntag 203	Tag escrita!
Ntag 203	Tag escrita!
Ntag 203	Tag escrita!
Ntag 203	Tag escrita!

Classic 1k	Tag escrita!

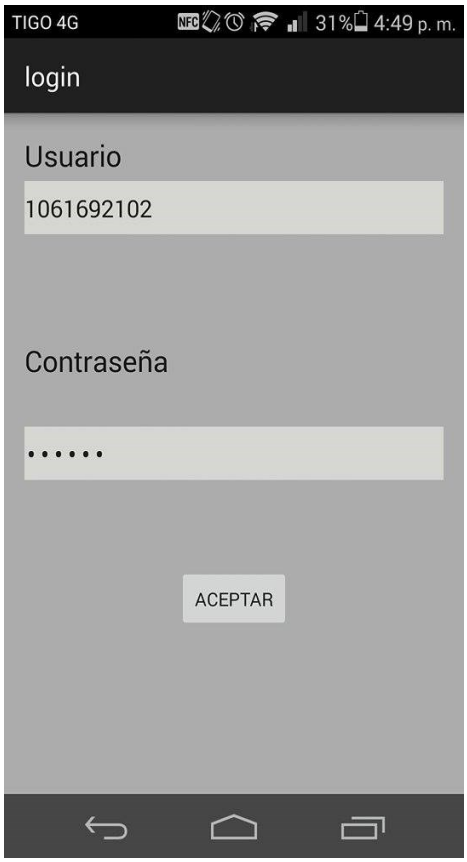
5.1.2 Prueba de lectura

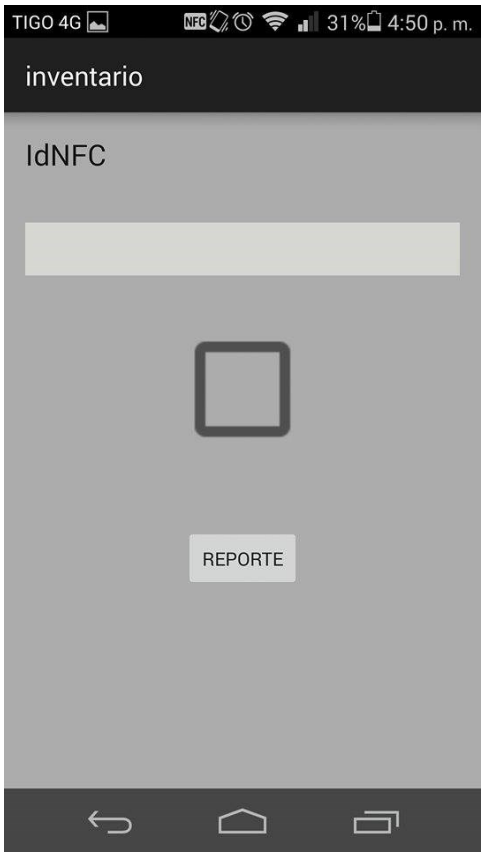


Referencia de Tag	Resultado
Ntag 203	Lectura correcta
Ntag 203	Lectura correcta
Ntag 203	Lectura correcta
Ntag 203	Lectura correcta
Ntag 203	Lectura correcta
Ntag 203	Lectura correcta
Ntag 203	Lectura correcta
Classic 1k	Lectura correcta
Classic 1k	Lectura correcta
Classic 1k	Lectura correcta
Classic 1k	Lectura correcta
Classic 1k	Lectura correcta
Classic 1k	Lectura correcta
Classic 1k	Lectura correcta
Classic 1k	Lectura correcta

5.1.3 Prueba de conexión

La prueba se realizó, alojando la base de datos en un servidor externo (sql3.freesqldatabase.com).





Nombre usuario	Resultado
Usario1	Conexión exitosa
Usario1	Conexión exitosa
Usario1	Conexión exitosa
Usario1	Conexión exitosa
Usario1	Conexión exitosa
Usario1	Error de conexión
Usario1	Conexión exitosa
Usario1	Conexión exitosa
Usario1	Conexión exitosa
Usario1	Conexión exitosa
Usario1	Conexión exitosa
Usario1	Conexión exitosa
Usario1	Conexión exitosa
Usario1	Conexión exitosa

Usario1	Conexión exitosa
Usario1	Error de conexión
Usario1	Conexión exitosa
Usario1	Conexión exitosa
Usario1	Servidor no encontrado
Usario1	Conexión exitosa





Referencia de Tag	Resultado
Ntag 203	Registro actualizado
Ntag 203	Registro actualizado
Ntag 203	Registro actualizado
Ntag 203	Registro actualizado
Ntag 203	Registro actualizado
Ntag 203	Registro actualizado
Ntag 203	Registro actualizado
Ntag 203	Registro actualizado
Ntag 203	Registro actualizado
Ntag 203	Registro actualizado

5.2 Análisis de resultados

- Se realizo la prueba de lectura y escritura con diferentes marcas de tags y se evidenció un correcto funcionamiento de la aplicación. Los errores que se presentaron fueron debido a que el tag no quedaba dentro del campo de operación del dispositivo.

- Teniendo en cuenta que la base de datos se aloja en un servidor externo es necesario una buena conexión a internet para obtener buenos resultados en la prueba de inventario.

- En las pruebas realizadas anteriormente se puede observar que la aplicación presenta un 95% de efectividad, lo que demuestra que la aplicación cumple con los requerimientos específicos.

Capítulo 6

6 CONCLUSIONES Y TRABAJOS FUTUROS

Después de desarrollar el proyecto se llegó a las siguientes conclusiones y recomendaciones para posibles trabajos futuros.

6.1 CONCLUSIONES

- El desarrollo del proyecto permitió introducir un concepto básico de la tecnología NFC y exponer sus principales características, ventajas y desventajas, para que el usuario pueda escoger libremente la que mejor se acople a sus necesidades teniendo en cuenta que la tecnología NFC es compatible con otras tecnologías principalmente con Bluetooth, Wifi, RFID dando lugar a que se pueda utilizar la infraestructura existente.

- La tecnología NFC comenzó a desarrollarse en el año 2002 pero es hasta este momento que ha empezado a expandirse de forma acelerada, esto gracias a que los fabricantes de teléfonos móviles la están incluyendo en los dispositivos.
- La tecnología NFC presenta un importante ventaja frente las tecnologías tradicionales como los códigos de barras y QR, y es que permite obtener información útil de forma off-line, debido a su capacidad de almacenamiento.

- La rapidez con que se establece la comunicación NFC está ligada a la aceptación implícita cuando dos dispositivos entran en el campo de operación, superando así el cuello de botella de otras tecnologías donde es necesario la aceptación explicita de la comunicación y que en algunos casos requiere de la autenticación con contraseña.

- Gran parte de la seguridad de la tecnología NFC radica en su corto alcance, quien trate de hacer una intercepción deberá esta ubicado a menos de 10 cm. Además de esto, los tag NFC disponen de campos de seguridad configurables para seleccionar quien puede leer, escribir o modificar los datos existentes.

- Al desarrollar un proyecto con tags NFC se debe tener en cuenta la cantidad de información que se desea almacenar en estos, ya que no sobrepasan unos cuantos Kbytes, lo que dificulta su uso donde se requiera gran capacidad de almacenamiento.

- La aplicación se desarrolló utilizando la plataforma Android Studio, el cual facilita el soporte para diferentes tamaños de dispositivos móviles, además, utiliza el compilador JAVA (JDK), lo que hace posible que la aplicación sea compatible con diferentes sistemas operativos.
- Una parte importante del desarrollo fue el manejo correcto de la conexión con la base de datos así como las consultas y modificaciones llevadas a cabo en ella.

- El desarrollo del proyecto permite demostrar que un avance tecnológico puede ser utilizado para dar solución a diferentes problemas de la vida diaria, sin tener en cuenta el fin específico para la cual fueron creados y es ahí donde nosotros como ingenieros estamos llamados a explotarlas.

6.1 TRABAJOS FUTUROS

Este proyecto se desarrolló con el apoyo del Grupo de Investigación LIDER de la Universidad Distrital y a continuación se expone un conjunto de propuestas para ser desarrolladas en un futuro:

- Utilizar la aplicación para Implementar un sistema de control y acceso de los estudiantes de la Universidad Distrital aprovechando que los carnets que la universidad da a sus miembros traen incorporada la tecnología NFC

- La aplicación permite la lectura y escritura de solo texto por lo tanto se propone la modificación de la misma para grabar imágenes, archivos multimedia y direcciones web.

- Utilizar la aplicación para desarrollar proyectos en sensorica.

- Modificar la aplicación para que se pueda utilizar para realizar inventarios de otros artículos como son los artículos de laboratorios.

ANEXO

Estudio para el uso de la tecnología NFC en la biblioteca de la universidad distrital.

El sistema de Información que utiliza la biblioteca de la universidad distrital se llama **ALEPH 500**, actualmente versión **20.** A través de este sistema se gestionan tanto los procesos administrativos (procesamiento técnico o descripción bibliográfica y los inventarios, registros de usuarios) así como los de servicio (préstamos, sanciones, convenios). El sistema puede ser consultado por protocolo *http, z3950, consultas sql, Web Services.*

Se basa en una arquitectura cliente-servidor y utiliza la base de datos de Oracle.

Permite la personalización y adición de módulos con el fin de cumplir con el mayor número de requerimientos, permite la conectividad con otros sistemas y bases de datos, maneja interfaces graficas intuitivas y soporta diferentes idiomas. Las constantes actualizaciones permiten una depuración del sistema y un mejoramiento en la prestación de servicios. A continuación se muestra la interfaz gráfica para acceder a los servicios de la biblioteca.

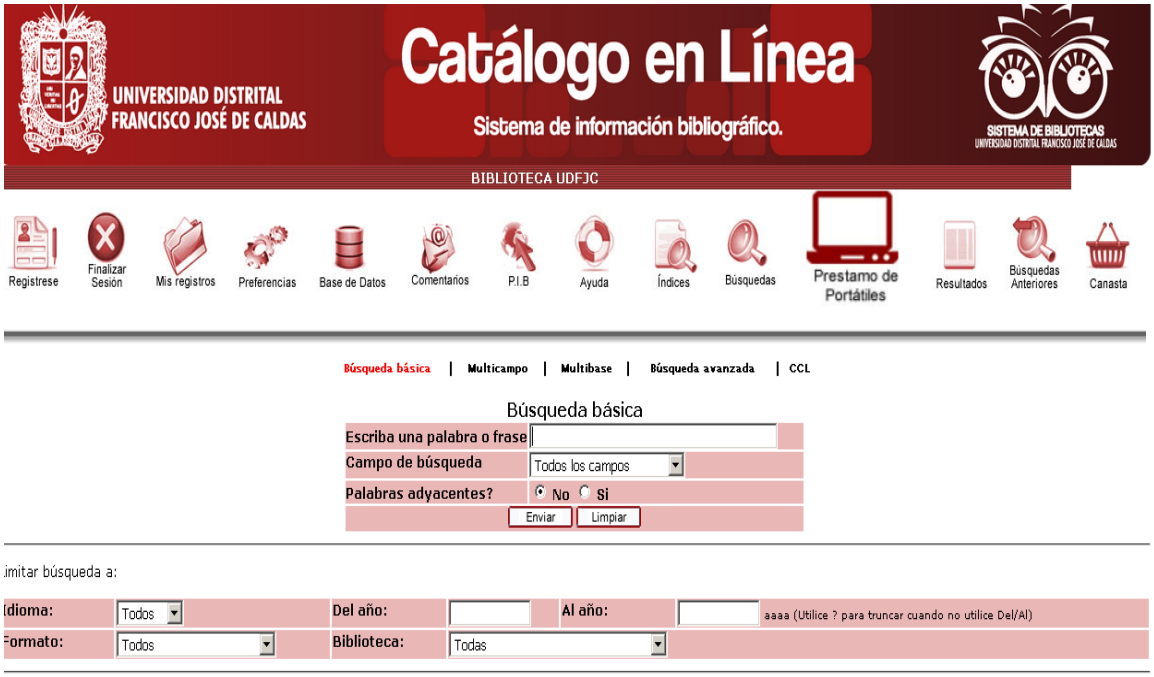


Figura 46. Sistema de información bibliográfico de la UDFJC.

Teniendo en cuenta los servicios que la biblioteca tiene implementados y la manera como se realizan dichos procesos es posible utilizar la tecnología para realizar las siguientes tareas:

Control de inventarios

El control de inventarios que se realiza en la biblioteca se basa en la lectura del código de barras que contiene un numero identificador de cada libro. Como se mencionó en el proyecto, esta etiqueta se coloca en la parte interna del libro y para su lectura es necesario abrir el libro. En cuanto a esto se puede utilizar los tags NFC donde se guardara el número de identificación del libro y aprovechando que esta es una tecnología de campo podrá ser leída sin necesidad de abrir el libro. Además tiene un tiempo de respuesta menor comparado con el lector óptico de código de barras.

No. sis.	000029171
Formato	BK
ISBN	9701054725
Clas. Dewey	621.3815/S33c
Autor principal	Sedra, Adel S..
Título	Circuitos microelectrónicos /
Edición	5a ed.
Ciudad	México :
Editorial	: McGraw-Hill,
Fecha	2006.
Descripción	xxix, 1283 [100] p. : il. ; 28 cm. + 1 CD-ROM
Idioma	spa eng
Nota	Incluye apéndices
Bibliografía	Incluye bibliografía e índice
Notas	Microelectronic circuits
Tema - Otros	Circuitos eléctricos
	Microelectrónica
	Circuitos integrados
	Transistores
Acc. secundario	Smith, Kenneth C..
	Pineda Rojas, Eloy, tr..
	Navarro Salas, Rodolfo, tr..
Ejemplares	TECNO MULTI 621.3815 S33c TECNOLOGICA Coleccion de audioviduales, CD, Diapositivas, videos, etc
	TECNO GENER 621.3815 S33c TECNOLOGICA Coleccion general
	INGEN MULTI 621.3815 S33c INGENIERIA Coleccion de audioviduales, CD, Diapositivas, videos, etc
	INGEN GENER 621.3815 S33c INGENIERIA Coleccion general
Existencias	Todos los ejemplares
Biblioteca	INGENIERIA
Biblioteca	TECNOLOGICA

Figura 47. Vista Completa de un registro utilizando catalogación estándar.

Como se aprecia en la figura 47, cada ejemplar tiene un número de identificación en el sistema, que permite diferenciarlo de los demás, y poder llevar el control del inventario.

Utilizando la capacidad de almacenamiento que tiene las etiquetas NFC es posible almacenar la vista completa de catalogación de un ejemplar y no solo en número de identificación como ocurre actualmente con los códigos de barras.

Manual de usuario de equipos

Utilizando los tags NFC con mayor prestación de memoria sería de gran interés poder grabar el manual del equipo en un tag y pegarlo al equipo de tal manera que el usuario pueda acceder a esta información con solo pasar el dispositivo lector sobre la etiqueta, de esta manera podrá adquirir la información necesaria para poder hacer un uso correcto del equipo. La implementación de esto sería de gran ayuda ya que en la actualidad para solicitar esta información es necesario acudir a las personas encargadas, en ocasiones se requiere de un permiso por escrito, y en otras ocasiones el manual no se encuentra disponible.

Acceso a los equipos (usuario y contraseña)

Los equipos de la Universidad Distrital están configurados con usuario y contraseña, cuando un estudiante va a acceder a ellos requiere de dicha información, cuando se hacen actualizaciones de software o se cambia esta información como parte de la implementación de la seguridad, el usuario no podrá acceder al equipo, debido a esto el personal encargado del mantenimiento de los computadores opta por dejar el mismo usuario y contraseña de los equipos, exponiendo el sistema a ataques de seguridad. Aprovechando que los tags NFC se pueden re-escribir, se podría adherir un tag a cada equipo donde se guardara la información de acceso y el usuario podrá acceder a esta información acercando su dispositivo lector, sin importar cada cuando se actualice dicha información.

Control de los equipos para que no sean sacados de la institución

Se podrá incluir un tag NFC dentro de los equipos y con la información que indiquen que el equipo pertenece a la institución, de tal manera que el personal encargado de la seguridad pueda verificar la pertenencia de forma rápida y sin temor a equivocaciones.

Solicitar préstamo de libros

Esto va ligado a la tecnología NFC que utilizan los carnets actuales, de tal manera que el usuario pueda escoger el libro de su preferencia y a través de su dispositivo lector pueda validar la información y ser cargado a su cuenta.

Consulta de libros

El sistema de información de la biblioteca permite realizar diferentes tipos de búsqueda. Una búsqueda básica, una búsqueda multicampos, una búsqueda multibase, una búsqueda avanzada y una búsqueda por comandos.

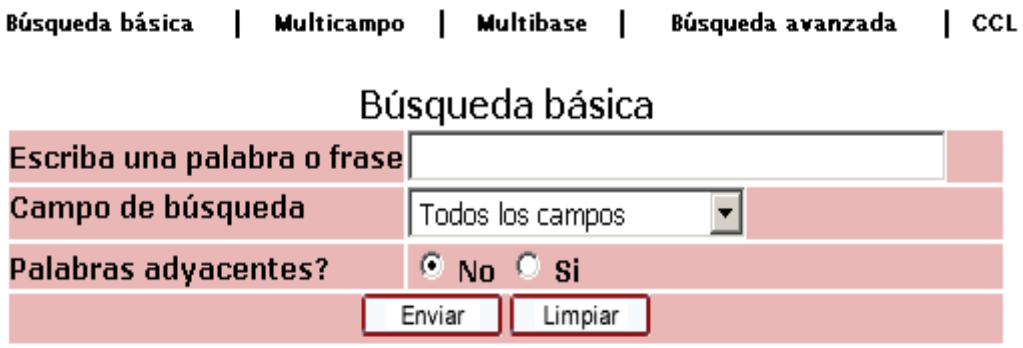


Figura 48.Busquedas implementadas en el sistema de información bibliográfico de la UDFJC.

Un complemento a este servicio, consiste en grabar la tabla de contenido en una etiqueta NFC, con el fin de que el usuario escriba una palabra clave del tema de su preferencia, acerque su dispositivo lector al libro y obtenga información acerca de si ese libro contiene temas relacionados con su búsqueda.

Proceso de implementación

La implementación de la tecnología NFC en la biblioteca requiere de una inversión elevada debido al gran número de ejemplarles que se maneja, de tal manera que un cambio total y de manera rápida no sería viable. Se recomienda ir implementando la tecnología NFC de manera paulatina, comenzando con los nuevos ejemplares aprovechando que esta tecnología es

compatible con las tecnologías existentes, lo que significa que se puede utilizar la infraestructura existente.

Tecnología	Costo/unidad	Número de ejemplares Susceptibles a NFC	Tiempo de vida útil	Costo total
Código de barras	$ 100	121 000	Mayor a 10 años	$12 100 000
NFC	$ 1 500	121 000	Mayor a 10 años	$181 500 000

Tabla 12. Comparación de costos entre la tecnología de código de barras y la tecnología NFC.

En cuanto al hardware y software necesario para la implementación de la tecnología NFC se advierte que el grupo LIDER cuenta con estos recursos, y que se podría llegar a un eventual acuerdo para ser utilizados.

BIBLIOGRAFÍA

[1] M. Reveilhac and M. Pasquet, “Promising secure element alternatives for NFC technology,” *Proc. - 2009 1st Int. Work. Near F. Commun. NFC 2009*, pp. 75–80, 2009.

[2] R. Ramanathan and J. Imtiaz, “NFC in industrial applications for monitoring plant information,” *2013 4th Int. Conf. Comput. Commun. Netw. Technol. ICCCNT 2013*, pp. 4–7, 2013.

[3] D. E. U. N. a Web and Q. U. E. No, “RUIDERAe: Revista de Unidades de Información. Número 5 (1er semestre 2014) ISSN 2254-7177,” vol. 5, pp. 1–18, 2014.

[4] S. Paper and A. C. Emulation, “NFC - Vulnerabilities and Defense,” no. 1, pp. 35–38, 2014.

[5] “A new simple Wi-Fi direct connection method using NFC on remote control and DTV.” .

[6] P. Teh, P. K. Ahmed, S. Cheong, A. H. S. Chan, and W. Yap, “Knowing What a User Likes : Mobiquitous Home with NFC Smartphone,” pp. 121–125, 2013.

[7] A. Albi and D. F. Piles, “Antonio Albi ñ ana Martinez Emilio Cardona Caballero David Ferrer Piles.”

[8] S. H. Wu and C. Yang, “Promoting collaborative mobile payment by using NFC-micro SD technology,” *Proceedings - IEEE 10th International Conference on Services Computing, SCC 2013*. pp. 454–461, 2013.

[9] I. Near and F. Comunication, “Informe Near Field Comunication (NFC),” 2012.

[10] C. a. Opperman and G. P. Hancke, “A generic NFC-enabled measurement system for remote monitoring and control of client-side equipment,” *Proc. - 3rd Int. Work. Near F. Commun. NFC 2011*, pp. 44–49, 2011.

[11] V. Patil and N. Varma, “NFC Based Health Monitoring And Controlling System,” pp. 133–137, 2014.

[12] L. Name, F. Name, O. Training, P. Training, C. Darin, R. O. Training, M. Kimberly, G. Deepa, E. Board, E. Principal, I. Primary, F. Systems, E. B. Study, and N. Co-investigator, “Modulacion ASK,” *Igarss 2014*, 2014. [Online]. Available: http://www.udb.edu.sv/udb/archivo/guia/electronica-ingenieria/sistemas-de-comunicacion-ii/2013/ii/guia-5.pdf.

[13] L. Name, F. Name, O. Training, P. Training, C. Darin, R. O. Training, M. Kimberly, G. Deepa, E. Board, E. Principal, I. Primary, F. Systems, E. B. Study, and N. Co-investigator, “Codigo Manchester,” *Igarss 2014*, 2014. [Online]. Available: http://ayudaelectronica.com/codigo-manchester/.

[14] C. Robledo Sosa, *Redes de Computadoras*. 2002.

[15] L. Steller, “Codificaci ó n Manchester y Manchester Diferencial ¿ Qu é es la codificaci ó n ?”

[16] K. Xiao and L. Luo, “A Novel Mobile Device NFC Stack Architecture,” *2013 IEEE 11th Int. Conf. Dependable, Auton. Secur. Comput.*, pp. 169–173, 2013.

[17] J. Shen and X. C. Jiang, “A proposed architecture for building NFC tag services,” *Proc. - 6th Int. Symp. Comput. Intell. Des. Isc. 2013*, vol. 2, pp. 48–52, 2013.

[18] M. Roland and J. Langer, “Digital signature records for the NFC data exchange format,” *Proceedings - 2nd International Workshop on Near Field Communication, NFC 2010*. pp. 71–76, 2010.

[19] M. Saeed and C. Walter, “Off-line NFC Tag Authentication,” *Internet Technol. Secur. …*, pp. 730–735, 2012.

[20] “NFC Fórum.”Disponible en linea: http://nfc-forum.org/.

[21] “Tecnología NFC Near field communication.” .

[22] C. T-, “Capa de enlace Capa de Enlace : Objetivo,” 2012.

[23] “NFC Forum Technical Specifications.” Disponible en linea:

http://members.nfc-forum.org/specs/spec_list/#rtds.

[24] D. A. Chavarría Chavarría, “Tecnología de comunicaciones de campo cercano (NFC) y sus aplicaciones,” 2011.

[25] “Todo lo que necesitas saber sobre NFC.” .

[26] “Escoge tu TAG NFC.” Disponible en linea: http://www.nfc-mifare.com/tag.html.

[27] “NFC-Mifare.” Disponible en linea: http://nfc-mifare.tumblr.com/page/2.

[28] “NFC-Mifare — Cómo seleccionar la etiqueta NFC correcta.” Disponible en linea: http://nfc-mifare.tumblr.com/post/65462671205/c%C3%B3mo-seleccionar-la-etiqueta-nfc-correcta-para-tu.

[29] “NFC First Steps - Home.” Disponible en linea: https://nfcfirststeps.codeplex.com/wikipage?title=%C2%BFQu%C3%A9 es un TAG NFC?
Disponible en linea: http://www.etiquetas-nfc.es/tipos-de-chip-nfc.

[31] H. Dunant, *Libro Blanco sobre la aplicación de la Tecnología NFC en el Transporte Público*. 2013.

[32] Nxp, “MF1S703x MIFARE Classic 4K - Mainstream contactless smart card IC for fast and easy solution development,” no. December, pp. 1–35, 2010.

[33] M. Desfire, “MF3ICD21, MF3ICD41, MF3ICD81 MIFARE DESFire EV1 contactless multi-application IC,” no. March, pp. 1–15, 2009.

[34] N. X. P. Semiconductors, “NTAG203 NFC Forum Type 2 Tag compliant IC with 144 bytes user memory - NTAG203_SDS.pdf,” no. October, pp. 1–30, 2011.

[35] M. Classic, “MF1S503x MIFARE Classic 1K - Mainstream contactless smart card IC for fast and easy solution development,” no. February, pp. 1–37, 2011.

[36] NXP Semiconductors, “MF1S50yyX/V1 MIFARE Classic EV1 1K - Mainstream contactless smart card IC for fast and easy solution

development," no. December, p. 40, 2014.

[37] "Historial de versiones de Android." Disponible en linea: https://es.wikipedia.org/wiki/Anexo:Historial_de_versiones_de_Android#V.C3.A9ase_tambi.C3.A9n.

[38] "Introducción a Android." Disponible en linea: http://www.desarrolloweb.com/articulos/introduccion-android.html.

[39] "Android Developers." Disponible en linea: http://developer.android.com/index.html.

[40] "Oracle." Disponible en linea: http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html.

[41] "MYSQL." Disponible en linea: https://www.mysql.com/.

[42] "Tutoriales programación." Disponible en linea: http://programandolo.blogspot.com.co/2013/09/el-archivo-androidmanifestxml.html.

[43] "Programacion en android." Disponible en linea: http://www.hermosaprogramacion.com/2014/08/android-studio-proyecto-en/.

[44] "Configurar Layouts y Views En Android Studio." Disponible en linea: http://www.hermosaprogramacion.com/2014/09/android-layouts-views/.

[45] "imagenes codigo Manchester." Disponible en linea: http://docente.ucol.mx/al000408/public_html/MANCHE.gif.