



UNIVERSIDAD DE JAÉN
Nombre del Centro

Trabajo Fin de Grado

APLICACIÓN MÓVIL CON TECNOLOGÍA NFC PARA EL INTERCAMBIO DE INFORMACIÓN

Alumno: Antonio Segovia Pastor

Tutor: Dra. Da. Macarena Espinilla Estévez Dr.
D. Luis Martínez López

Dpto: Informática

Septiembre, 2018



Universidad de Jaén
Escuela Politécnica Superior de Jaén
Departamento de Informática

Dra. D^a. Macarena Espinilla Estévez y Dr. D. Luis Martínez López, tutores del Proyecto Fin de Carrera titulado: Aplicación Móvil con tecnología NFC para el intercambio de información, que presenta D. Antonio Segovia Pastor, autorizan su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, Septiembre de 2018

El alumno:

Antonio Segovia Pastor

Los tutores:

Macarena Espinilla Estévez

Luis Martínez López

Índice

1. Introducción	7
1.1. Motivación	7
1.2. Propósito del trabajo.....	8
1.3. Propuesta.....	8
1.4. Objetivos	9
1.5. Planificación	9
1.5.1. Planificación Temporal	11
2. Android	13
2.1. Introducción.....	13
2.2. Versiones	13
2.3. Android Studio.....	16
2.4. Arquitectura	18
2.5. Componentes	20
2.6. Activity.....	21
3. Tecnología NFC.....	25
3.1. Introducción.....	25
3.2. Tipologías de etiquetas NFC	26
3.3. Funcionamiento.....	27
3.3.1. Modo Activo/Pasivo	27
3.3.2. Modo Lectura/Escritura	28
3.3.3. Modo Peer to Peer	29
3.3.4. Modo Emulador	31
3.4. Formato NDEF	31
3.5. Aplicaciones de NFC	34
3.6. Comparativa con tecnologías similares.....	35
3.7. Ventajas y desventajas.....	38
3.8. Seguridad.....	39
4. Ingeniería del Software	41
4.1. Introducción.....	41
4.2. Esquema de Requerimientos.....	41
4.2.1. Funcionales.....	41
4.2.2. No Funcionales	42
4.3. Análisis del Sistema.....	42

4.3.1. Perfiles de usuarios	42
4.3.2. Casos de uso	43
4.3.3. Técnica Personas.....	48
4.3.4. Escenarios	49
4.4. Diseño	52
4.4.1. Diseño del sistema	52
4.4.1.1. Diagrama de clases	52
4.4.1.2. Diagrama de secuencia	56
4.4.1.3. Patrón de arquitectura software	60
4.4.2. Diseño de datos	61
4.4.3. Diseño de la interfaz.....	62
4.4.3.1. Metáforas	62
4.4.3.2. Estilo.....	66
4.4.3.3. Storyboard.....	67
4.5. Implementación y pruebas.....	70
4.5.1. Arquitectura	70
4.5.2. Lenguajes	71
4.5.3. Herramientas	72
4.5.4. Pruebas	73
4.5.5. Resultados.....	77
4.5.6. Encuesta.....	78
5. Conclusiones	83
Bibliografía	85
A. Anexos	89
a. Guía de instalación	89
b. Activación NFC	93
c. Manual de usuario.....	96
i. Escritura	97
ii. Uso cotidiano.....	102

Índice de ilustraciones y tablas:

Ilustración 1.1 Esquema de trabajo	10
Ilustración 1.2 Diagrama de Gantt	11
Tabla 2.1 Versiones de Android con sus principales modificaciones y cambios.	15
Ilustración 2.1 Cuota de mercado de las distintas versiones de Android a mayo de 2017. ...	16
Ilustración 2.2 Ventana de bienvenida de Android Studio.....	17
Ilustración 2.3 Área de trabajo en Android Studio.....	18
Ilustración 2.4 Capas de la arquitectura de Android.	19
Ilustración 2.5 Ciclo de vida de un Activity.....	23
Ilustración 3.1 Etiqueta NFC.....	25
Tabla 3.1 Resumen descriptivo de los tipos de Tags.....	26
Ilustración 3.2 Etiqueta pasiva NFC.....	28
Ilustración 3.3 Capas de la arquitectura del modo lectura/escritura.	28
Ilustración 3.4 Arquitectura Peer to Peer.	29
Ilustración 3.5 Capas de la arquitectura del modo Peer to Peer.	30
Ilustración 3.6 Arquitectura del modo emulador.....	31
Ilustración 3.7 Mensaje NDEF.....	32
Ilustración 3.8 Campos del registro NDEF.....	33
Ilustración 3.9 Etiqueta RFID.....	36
Ilustración 3.10 Ejemplo de código QR.....	36
Ilustración 3.11 Símbolo de la tecnología bluetooth.....	37
Ilustración 3.12 Símbolo de la tecnología Wifi.	38
Ilustración 4.1 Elementos de los casos de uso.	44
Ilustración 4.2 Diagrama frontera.	44
Ilustración 4.3 Diagrama de caso de uso: Realizar Llamada.	45
Ilustración 4.4 Diagrama de caso de uso: Enviar un mensaje.....	46
Ilustración 4.5 Diagrama de caso de uso: Escritura de un tag NFC.....	47
Ilustración 4.6 Elementos base de un diagrama de clases.	53

Ilustración 4.7 Diagrama de clases.....	55
Ilustración 4.8 Ejemplo de diagrama de secuencia.....	57
Ilustración 4.9: Diagrama de secuencia: Comunicación.....	58
Ilustración 4.10: Diagrama de secuencia: Escritura de contacto.....	59
Ilustración 4.11: Diagrama de secuencia: Escritura de acción.....	60
Ilustración 4.13: Dibujo del icono de la aplicación.....	63
Ilustración 4.14: Icono de la configuración.....	63
Ilustración 4.15: Dibujo de la configuración.	63
Ilustración 4.16: Icono de la información.	64
Ilustración 4.17: Icono de mensaje de texto.....	64
Ilustración 4.18: Icono para acceder a la agenda.	65
Ilustración 4.19: Icono de llamada.	65
Ilustración 4.20: Boceto original del fondo de pantalla.....	66
Ilustración 4.21: Primer storyboard de la aplicación.....	67
Ilustración 4.22: Storyboard de la aplicación en versión online.....	69
Ilustración 4.12: Separación Modelo-Vista-Controlador.....	70
Ilustración 4.23: Logo de java.....	71
Ilustración 4.24: Logo de XML.....	71
Ilustración 4.25: Icono de GitHub.	72
Tabla 4.1: Test 1. Cambio de vista.	73
Tabla 4.2: Test 2. Cambio de vista.	74
Tabla 4.3: Test 3. Acceso a la agenda.	74
Tabla 4.4: Test 4. Acceso a la agenda.	74
Tabla 4.5: Test 5. Escritura en una etiqueta.	75
Tabla 4.6: Test 6. Escritura en una etiqueta.	75
Tabla 4.7: Test 7. Lectura de una etiqueta.	75
Tabla 4.8: Test 8. Lectura de una etiqueta.	76
Tabla 4.9: Test 9. Realizar una llamada.	76
Tabla 4.10: Test 10. Realizar una llamada.	76

Tabla 4.11: Test 11. Enviar un sms.	77
Tabla 4.12: Test 12. Enviar un sms.	77
Tabla 4.13: Test 12. Resultados de los test.	78
Ilustración 4.26: Grafico pre-manual.	80
Ilustración 4.27: Grafico post-manual.	81
Ilustración 5.1: Mensaje de Instalación.	90
Ilustración 5.2: Permisos.	90
Ilustración 5.3: Advertencia sobre permisos.	91
Ilustración 5.4: Activación rápida de la tecnología NFC.	93
Ilustración 5.5: Acceso a las conexiones de nuestro terminal.	94
Ilustración 5.6: Acceso a las conexiones secundarias de nuestro terminal.	95
Ilustración 5.7: Vista principal de la aplicación.	96
Ilustración 5.8: Menú de configuración.	98
Ilustración 5.9: Menú de escritura de acción.	99
Ilustración 5.10: Mensaje informativo sobre la opción seleccionada.	100
Ilustración 5.11: Mensaje de confirmación de escritura.	100
Ilustración 5.12: Menú de escritura de contactos.	101
Ilustración 5.13: Ejemplo de comunicación NFC desde la aplicación.	103
Ilustración 5.14: Mensaje de error en la lectura.	103
Ilustración 5.15: Mensaje de confirmación en la lectura.	104

1. Introducción

En este apartado hablaremos sobre los motivos, los objetivos de realizar la aplicación y el planteamiento que se ha seguido.

1.1. Motivación

Envejecer es algo que nos ocurrirá a todos, por ello es conveniente que nos preparemos para ello además de hacerlo lo más cómoda posible, en este punto entra la tecnología, la cual puede hacer la vida mas fácil y entretenida [1].

Debido a la expansión de los dispositivos móviles, el asentamiento de la tecnología en la sociedad y el aumento de población envejecida y de la esperanza de vida, resulta muy importante emplear esta tecnología para mejorar la calidad de vida de dicha población.

Para ello se propone la creación de espacios accesibles a mayores, investigación en biomecánica, avances sociales y sanitarios, además de avances tecnológicos como biosensores o incluso casas inteligentes [2].

Según un informe realizado por Philips y el MIT señalan que las tecnologías conectadas son una buena forma de complementar el sistema sanitario, con aspectos como la prevención de problemas de salud o la atención domiciliaria, dando así mas posibilidad a la prevención de problemas de salud [3].

Según un estudio el 67% de los mayores usan el teléfono móvil con cierta asiduidad, aunque el 48% de los encuestados reciben ayuda de familiares o de personal cualificado. Esto nos indica que las personas mayores también son usuarios de la telefonía móvil, aunque pueden necesitar un poco de ayuda [4].

Por otro lado, las personas mayores no tienen como costumbre usar aplicaciones de fotografía ni redes sociales, hacen un uso escaso de los dispositivos táctiles, esto puede venir dado por la dificultad de uso, sobre todo al principio, y por la negativa al manejo de las nuevas tecnologías. Por ese motivo, es conveniente que las aplicaciones para este sector de la población sean muy intuitivas.

Para que la brecha tecnológica no sea tan grande, existen ciertas herramientas como launchers y aplicaciones que puede hacerles el uso del smartphone mucho mas simple [5].

Una de las utilidades más básicas y antiguas de la telefonía movil son las llamadas y los mensajes de texto, así que el proposito es facilitar su utilización a través de la tecnología NFC y facilitar la comunicación con sus seres queridos, para ello se usarán tarjetas o chips, los cuales los usaremos para mantener guardada la información necesaria.

1.2. Propósito del trabajo

El objetivo principal es simplificar la comunicación a través del dispositivo móvil entre las personas mayores o personas con problemas de movilidad o de visión no muy graves. Para ello, se hará uso de la tecnología NFC.

El prototipo de aplicación que vamos a desarrollar es una aplicación meramente de comunicación, a través de la cual podemos ponernos en contacto con una persona en concreto a través del uso de la tecnología NFC.

1.3. Propuesta

La propuesta de trabajo consiste en la realización de un prototipo de aplicación para los smartrphones, en concreto para android ya que es el más común, por el cual las personas mayores podrán realizar llamadas o mandar mensajes de texto a un contacto específico con solo acercar el teléfono a las tarjetas NFC correspondientes.

En la memoria se tratará de explicar el proceso de creación del software, para ello se ha dividido en cinco grandes apartados:

- Capítulo 1. Este capítulo se describirán la motivación de la aplicación propuesta en este trabajo y el planteamiento que se ha seguido.
- Capítulo 2. En este apartado se explicará los aspectos más importantes del sistema operativo en el cual operará nuestro prototipo de aplicación.

- Capítulo 3. En esta sección se revisarán los conceptos relativos a la tecnología NFC que será empleada en nuestro prototipo de aplicación.
- Capítulo 4. Este apartado se detallará el proceso de ingeniería del software que se ha seguido para el desarrollo del prototipo.
- Capítulo 5. Este capítulo se destinará a apuntar las conclusiones del trabajo fin de grado y a expresar las sensaciones derivadas tras la realización del mismo.

1.4. Objetivos

Los objetivos a seguir para completar el proyecto son los siguientes:

1. Planificación del desarrollo del prototipo de la aplicación.
2. Estudiar la tecnología NFC.
3. Desarrollar el prototipo de aplicación.
4. Implementar el prototipo de aplicación con la tecnología NFC y verificar el prototipo
5. Redactar la documentación del proyecto y realizar los manuales de usuario.

1.5. Planificación

Podemos ordenar y desgranar los objetivos en pasos a seguir para que sean más fácilmente comprensibles.

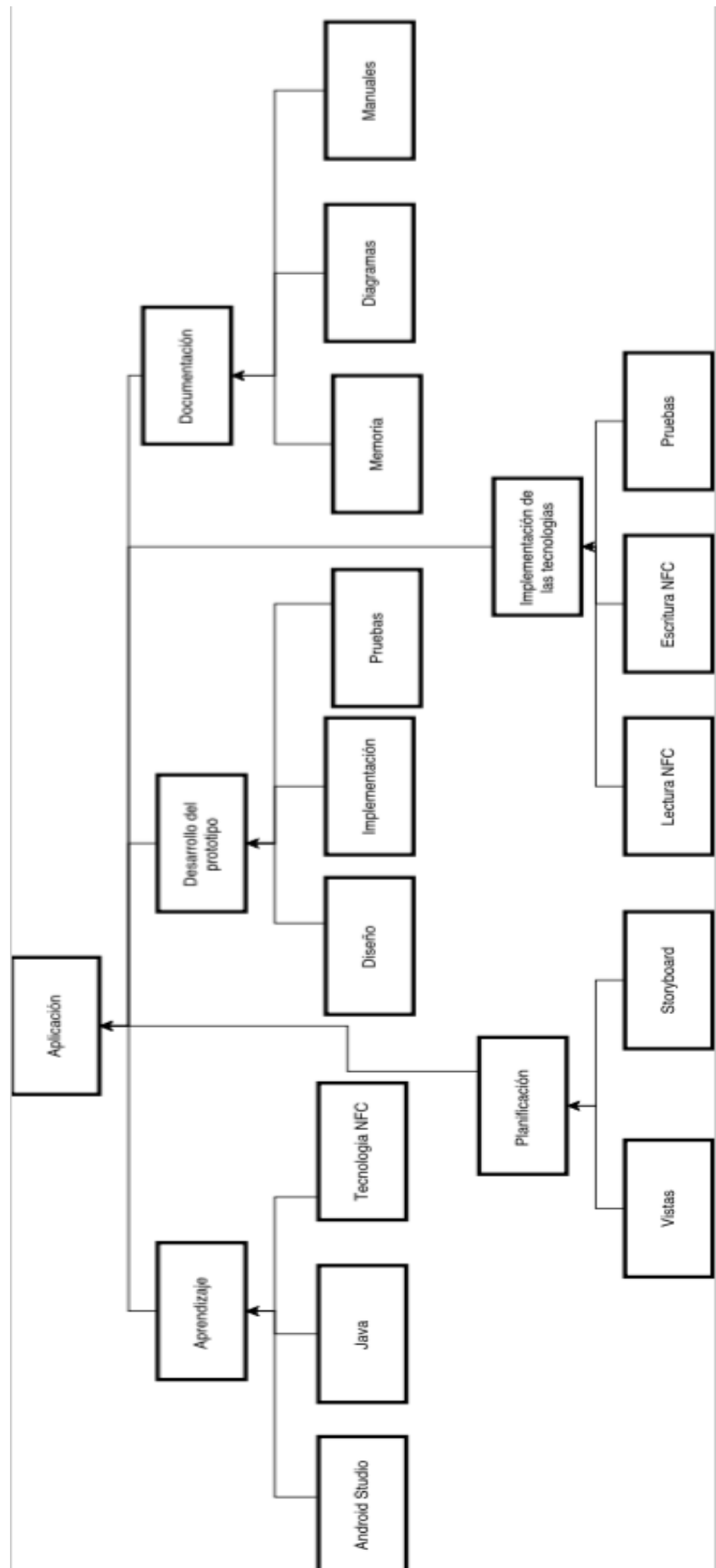


Ilustración 1.1 Esquema de trabajo

1.5.1. Planificación Temporal

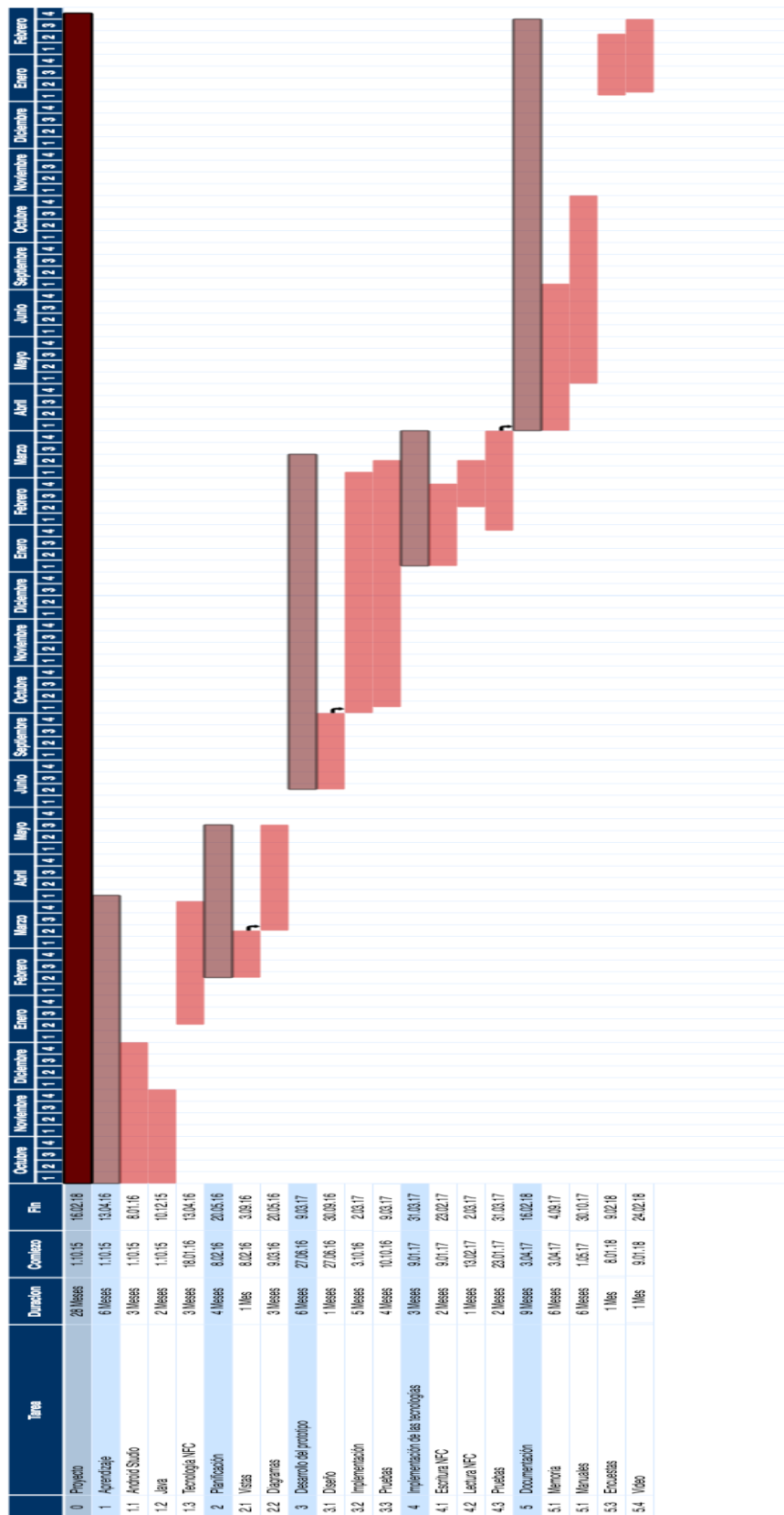


Ilustración 1.2 Diagrama de Gantt

En el diagrama de Gantt podemos observar algunas zonas anómalas como el estudio, tanto de java y android vienen de más atrás en el tiempo. En los meses de enero y junio suele ver saltos en el progreso de las actividades debido a factores académicos. Otro punto de solapamiento es durante el desarrollo de la escritura y lectura NFC, ya que una depende de la otra.

2. Android

En este capítulo se explicará los apartados mas importantes del sistema como las distintas versiones, su arquitectura y los componentes principales.

2.1. Introducción

Como ya comentamos anteriormente, el sistema operativo seleccionado para realizar el proyecto es Android ya que domina el nicho de mercado en los sistemas operativos móviles, este esta instalado entre el 84,7% de los dispositivos más vendidos de finales de 2015 [6] y aún estan presentes en el 81,7% de los dispositivos en 2017 [7].

2.2. Versiones

El sistema operativo de android ha pasado por diversas versiones desde su nacimiento, cada una de ellas suele corregir fallos o introducir nuevas funcionalidades al dispositivo.

Cada nueva versión tiene como nombre un postre, pero según el portavoz de google solo se trata de una broma entre empleados que terminó siendo un sello de identidad.

Nombre	Versión	Definición
Apple Pie	1.0	Primera versión comercial de Android en la que incluía Android Market, navegador web entre otros.
Banana Bread	1.1	Lanzada poco después que su predecesora para resolver fallos y alguna funcionalidad extra como las reseñas en los negocios.
Cupcake	1.5	Esta versión contenía mejoras como la inclusión de Widgets, transiciones animadas y la opción de auto-rotación pero lo más importante era el esqueleto de la API de android.

Donut	1.6	Las mejoras más importantes de esta versión son las mejoras en la velocidad de búsqueda, la eficiencia de la cámara y la galería. Además de ampliar el framework de gestos.
Eclair	2.0-2.1	Las principales modificaciones de esta versión fueron corregir errores en la API, optimizaciones en la velocidad del hardware, bluetooth 2.1, además de mejoras en la cámara y en los teclados virtuales.
Froyo	2.2-2.2.3	Esta versión se caracteriza por tener varios parches de seguridad, optimizaciones de rendimiento, velocidad y seguridad.
Gingerbread	2.3-2.3.7	En este conjunto de actualizaciones se incluyeron mejoras como el soporte nativo para diversos sensores, mejor eficiencia de la batería arreglos en la API, un recolector de basura concurrente y un soporte para NFC permitiendo la lectura de etiquetas.
Honeycomb	3.0-3.2.6	Esta versión de android es exclusiva para TV y tablets.
Ice cream sandwich	4.0-4.0.4	Esta versión es "compatible" con los dispositivos 2.3 de Android, esta versión nos trae mejoras en la rotación de pantalla, una nueva API para los desarrolladores, optimización del uso de los datos y Android Beam que es una aplicación por la cual se explota la usabilidad de la tecnología NFC para poder compartir diversos datos entre dispositivos usando dicha tecnología [8].
Jelly Bean	4.1-4.3.1	En este caso, esta versión nos trae una mejora de la funcionalidad y rendimiento de la interfaz usando la anticipación táctil y mejorar la fluidez en la interfaz del usuario. Incluye soporte para bluetooth de baja energía e integra OpenGL.
Kit Kat	4.4-4.4.4	La principal característica de esta versión es que la mayoría de las modificaciones son soluciones de errores como fallos de seguridad en los SMS, sonidos que sonaban más altos unos que otros y un fallo de la barra en la pantalla de bloqueo, no hubo demasiados cambios para los usuarios pero se añadió la opción de acceder a tus fotos desde la cámara y la opción de ocultar el teclado al pulsar en un espacio vacío.
Lollipop	5.1-5.1.1	Mejora los elementos visuales, un cálculo del tiempo estimado de la carga de la batería, mejora en la recogida de basura, modificación en la pantalla de bloqueo, se añadieron 15 nuevos idiomas. Además de Tap and go que permite migrar la cuenta de google entre dispositivos usando NFC son algunas de las modificaciones de esta versión.
Marshmallow	6.0-6.0.1	Esta versión nos trae la posibilidad de administrar permisos por aplicación, lector de

		huellas dactilares, explorador de archivos de forma nativa, mejoras en la memoria RAM y en conectividad.
Nougat	7.0-7.1.2	Nougat nos ofrece la opción de multiventana, mejora las animaciones, el uso de la batería y mejoras en la interfaz.
Android Oreo	8.0	La última versión de Android nos traera límites para las aplicaciones en segundo plano con el objetivo de optimizar la batería, también nos permite diversificar las notificaciones de una aplicación y poder bloquear solo un tipo de estas, además de poder posponerlas, mejoras en las multiventanas, mayor compatibilidad de codecs, iconos adaptativos y mejoras en la API [9].

Tabla 2.1 Versiones de Android con sus principales modificaciones y cambios.

[10]

Pese a todas estas mejoras de funcionalidad, rendimiento o seguridad no todos los dispositivos están actualizados, ya sea por compatibilidad de los dispositivos o por que un usuario medio no suele hacerlo.

Como podemos observar en el siguiente gráfico, el uso de las distintas versiones está segmentada en varias versiones, acaparando sobre el 75% tres de estas (Marshmallow, Lollipop y Kit Kat), observamos que la nueva versión Nougat no está entre ellas y ni si quiera tiene una gran cuota de mercado [11].

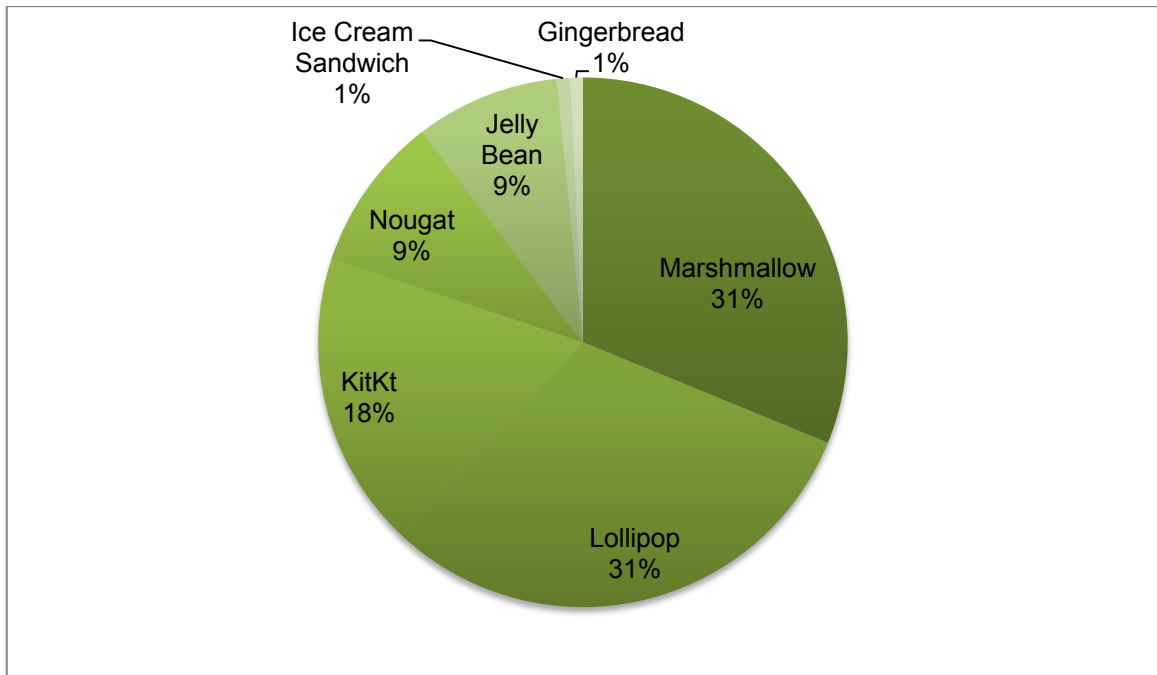


Ilustración 2.1 Cuota de mercado de las distintas versiones de Android a mayo de 2017.

Podemos deducir que los usuarios no siempre tienen la última versión, por lo tanto a la hora de desarrollar es mejor elegir la API de desarrollo adaptándose a los requisitos del software y no dejarse llevar por las últimas tecnologías.

2.3. Android Studio

Android Studio es el SDK oficial de Android para desarrollar aplicaciones pero no siempre fue así, antes del 2014 se usaba Eclipse como IDE oficial de desarrollo, el cambio supuso diversas mejoras como el compilador Graddle, mejoras del editor gráfico, creación de versiones, mejor distribución del código, etc. Sin embargo, también existen ciertas desventajas como menor número de plugins y una curva de aprendizaje más lenta, los cuales se pueden solventar con el paso del tiempo [12] [13] [14].

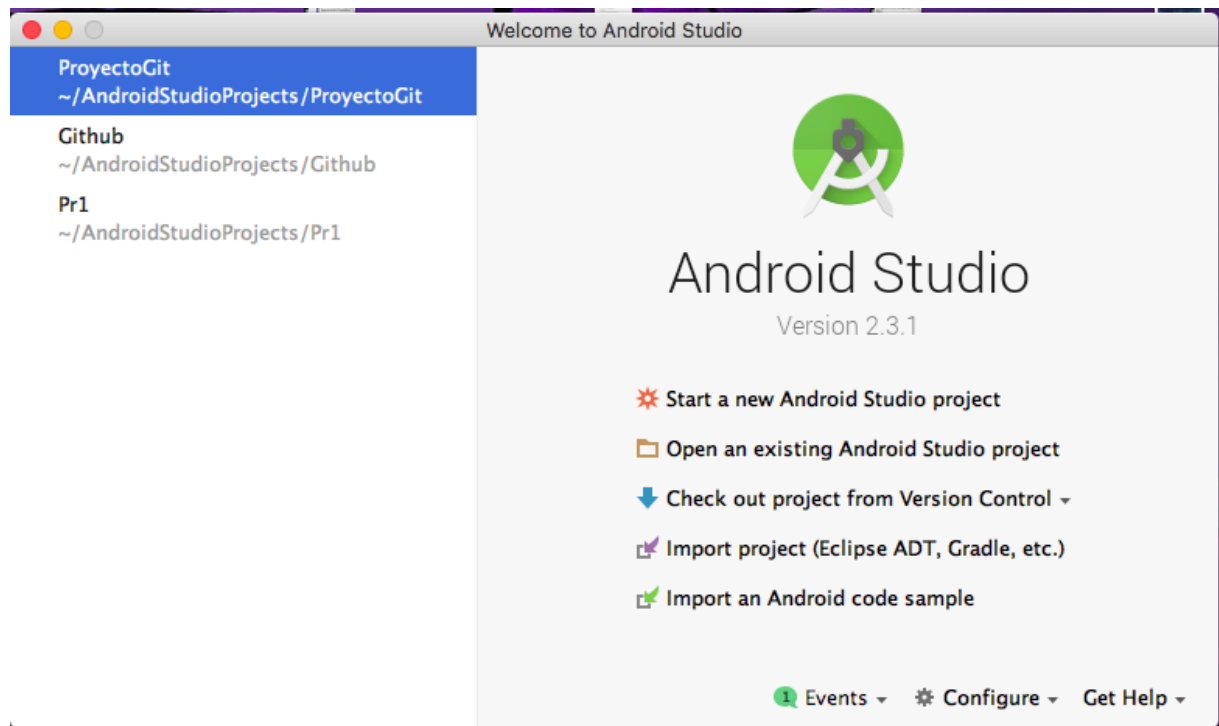


Ilustración 2.2 Ventana de bienvenida de Android Studio

Al iniciar Android Studio nos aparece una ventana inicial que nos permite abrir los proyectos recientes (izquierda) u otras acciones más comunes como empezar un nuevo proyecto, abrir otro existente o importarlo.

El área de trabajo de Android Studio tiende a ser similar a la de otros IDE de desarrollo para que la curva de aprendizaje no sea tan pronunciada, podemos observar las tres partes bien definidas:

La barra de herramientas, donde están las acciones más comunes como: guardar, copiar, pegar, compilar, ejecutar etc.

El área de desarrollo, es la parte central donde se introducen las líneas de código.

La barra lateral, la cual nos ayuda en la organización y gestión del proyecto.

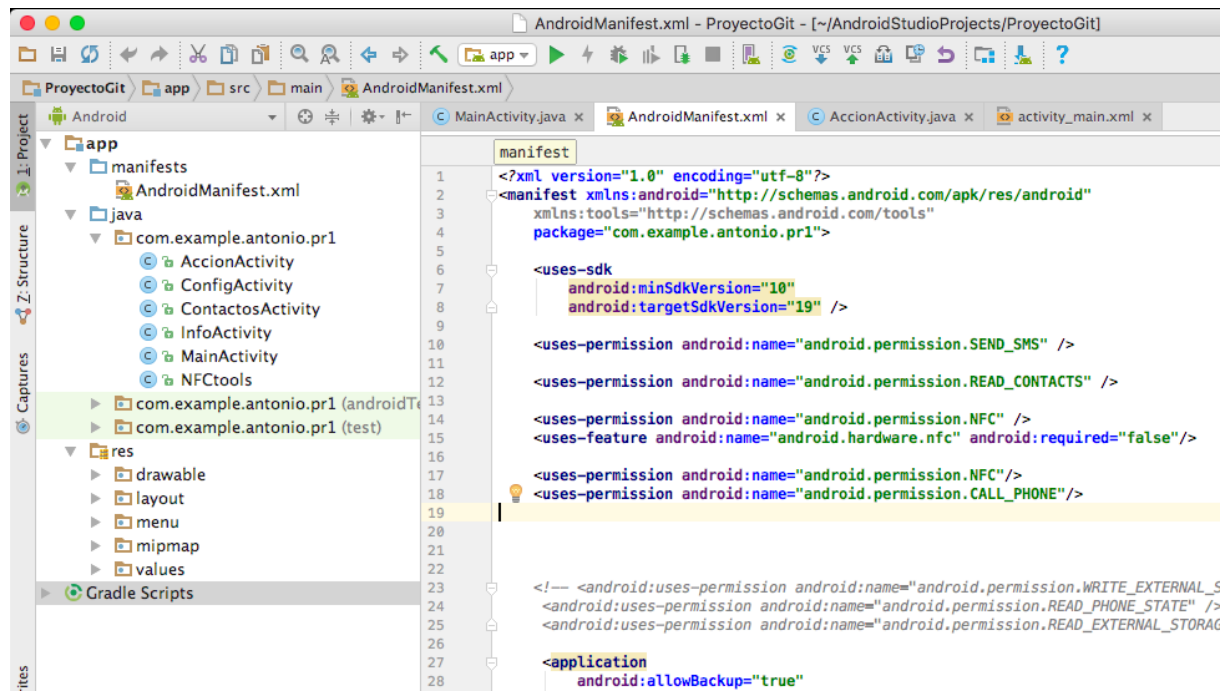


Ilustración 2.3 Área de trabajo en Android Studio

Como podemos ver en la ilustración anterior, hay tres carpetas principales en el proyecto:

- **Manifest:** Contiene el archivo `AndroidManifest` en el cual se definen los permisos de la aplicación, las actividades y diversas variables globales.
- **Java:** Es la carpeta donde se almacenan los documentos relacionados con el controlador, son los ficheros `.java`.
- **Res:** En esta carpeta encontramos los ficheros que son parte de las vistas del proyecto, como iconos, menús, imágenes o los ficheros `xml`.

2.4. Arquitectura

Para poder entender el funcionamiento de Android entraremos un poco más en profundidad en su arquitectura, esta se compone de cinco capas principales:

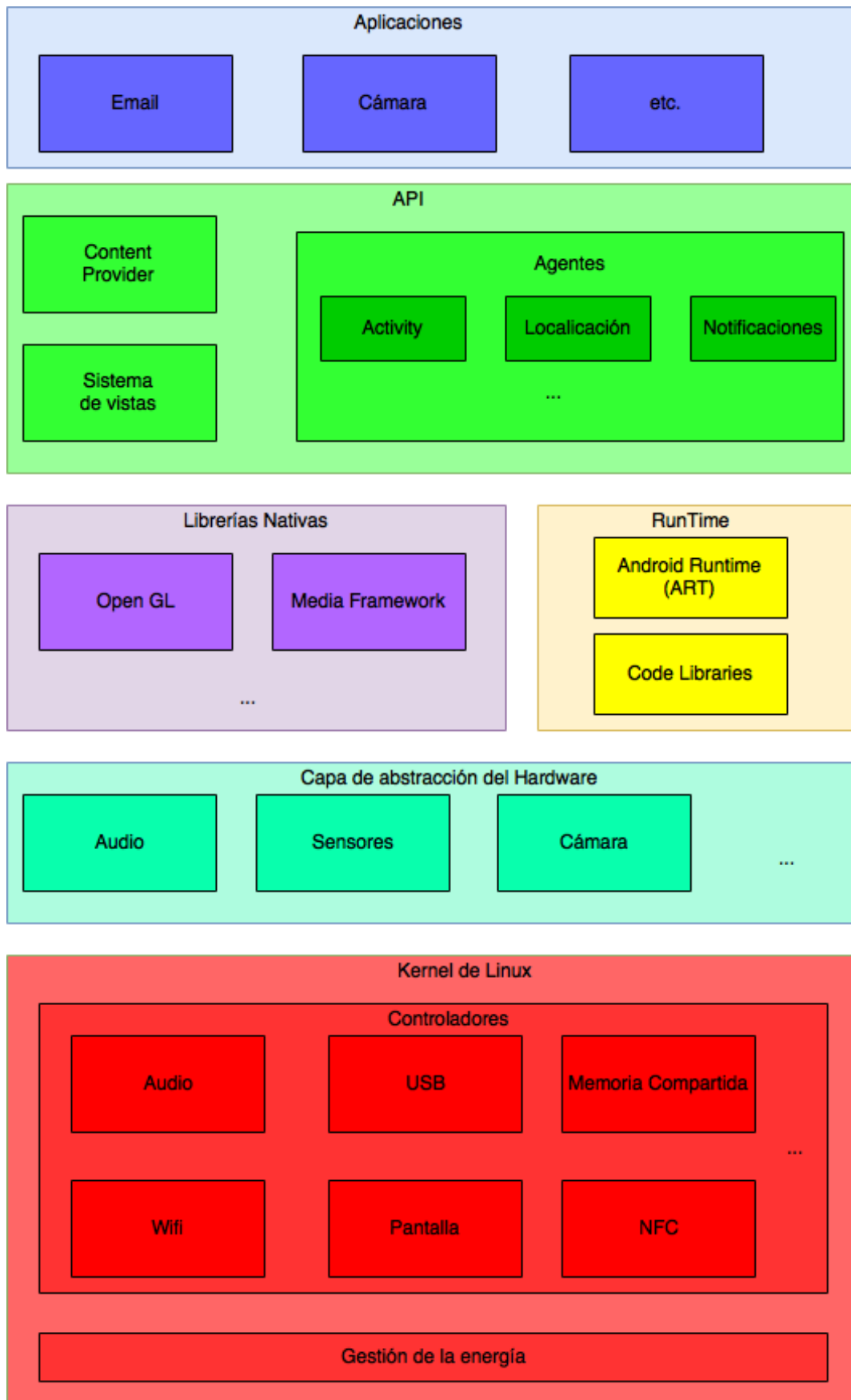


Ilustración 2.4 Capas de la arquitectura de Android.

- Comenzando desde la capa más baja encontramos el kernel de linux que actúa de intermediario entre el hardware y las capas superiores, ofreciendo soporte a las librerías más usadas.
- La capa justo por encima es una capa de abstracción del hardware que consiste en una serie de librerías de funciones específicas para cada componente hardware de manera individual.
- A la par podemos encontrar las capas de Android Runtime y las librerías nativas, el ART (android runtime) que está preparado para ejecutar máquinas virtuales optimizadas, muy útiles para la compilación anticipada, y la recolección de elementos no usados. Por otra parte están las librerías nativas escritas en c/c++, entre las más conocidas son las librerías de OpenGL, Media libraries y la librería SSL.
- La siguiente capa que encontramos es la capa de la API, esta nos proporciona la estructura base para la construcción de cualquier aplicación android, algunos de los ejemplos más comunes son: el sistema de vistas, administrador de actividades y la pila de navegación, content provider, ect.
- La última capa está formada por las distintas aplicaciones que nos ofrece el sistema operativo, además de las que se pueden descargar de terceros [15] [16] .

2.5. Componentes

Los componentes son los elementos básicos en la construcción de una aplicación móvil que viene proporcionada por la API, los principales elementos son:

- View o vistas: Son el elemento principal dentro de la interfaz gráfica con la cual interactuaremos con la aplicación, serán complementadas con diversos elementos como botones, cuadros de texto, imágenes, etc., todo ello fijado en un fichero XML.

- **Activity:** Representan las actividades que se van a realizar, representan a una vista y este determina la funcionalidad de la vista.
- **Service:** Los servicios son elementos lógicos sin representación gráfica, se ejecutan en segundo plano, los usos más comunes son las actualizaciones, notificaciones etc.
- **Intent:** Es el elemento que interconecta los distintos componentes de la aplicación entre sí, y así se facilita el intercambio de información entre estos.
- **Content provider:** Este componente es similar a un intent, ya que también se basa en el intercambio de información pero en este caso es entre distintas aplicaciones sin tener que acceder a la estructura o al almacenamiento interno de la otra aplicación.
- **Broadcast receiver:** Este elemento está diseñado para escuchar y actuar en consecuencia ante mensajes globales del sistema android del tipo “batería baja”, a diferencia de los anteriores este está enfocado a mensajes que no tienen un destinatario concreto, sino más bien son mensajes generales que pueden contener información interesante sobre el estado del dispositivo [17].

2.6. Activity

Los activity son los componentes principales, ya que son las encargadas de realizar cierto trabajo asociado a una vista, además son estas las que dictan el ciclo de vida de la aplicación.

El sistema controla una pila de actividades para que el usuario pueda pasar a la actividad anterior con el botón de “atrás” [18].

Una aplicación se ejecuta dentro de su propio proceso linux, aunque este puede ser requerido por el sistema, pudiendo así el sistema destruir el proceso si lo requiere.

Una actividad de Android puede estar en cuatro estados distintos y puede cambiar entre ellos gracias a siete métodos distintos:

- Activo: La actividad está visible y activa, es la actividad principal en pantalla y el usuario puede interactuar con ella.
- Visible o pausada: La actividad sigue estando visible, pero en este caso no podremos interactuar con ella ya que otra actividad tiene el foco a parte parcial y de manera momentánea.
- Parada o detenida: La actividad ya no es visible para el usuario pero sigue estando viva en el sistema, es conveniente guardar su estado ya que si el sistema lo requiere la puede eliminar.
- Destruída: La actividad es eliminada o por el sistema o por ella misma, ya no ocupa espacio en la pila de actividades ni tiene espacio en memoria.

Los metodos del activity tras los cuales se cambia de estado son:

- El metodo onCreate() es el método usado para la definición de variables y para cargar la vista asociada a la actividad.
- El metodo onStart() es el responsable de la inicialización tras el método onCreate() solo que este último no se ejecutara si la actividad vuelve del estado parado.
- El metodo onResume() es similar a los anteriores solo que este se ejecutará si la actividad viene del estado visible.
- El metodo onPause() es el método que se llama cuando la actividad que está activa va a pasar a segundo plano.
- El metodo onStop() es el responsable de pasar a parada la actividad y por lo tanto, no será visible para el usuario, el sistema puede llamar a esta funcion si lo necesitase.
- El método onRestart() se encarga de que la actividad pase de parada a activa.
- El metodo onDestroy() es el metodo que se ejecuta justo antes de que la aplicación pase a estado destruida, al igual que el metodo onStop(), puede ser llamada por el sistema si así lo precisa [19].

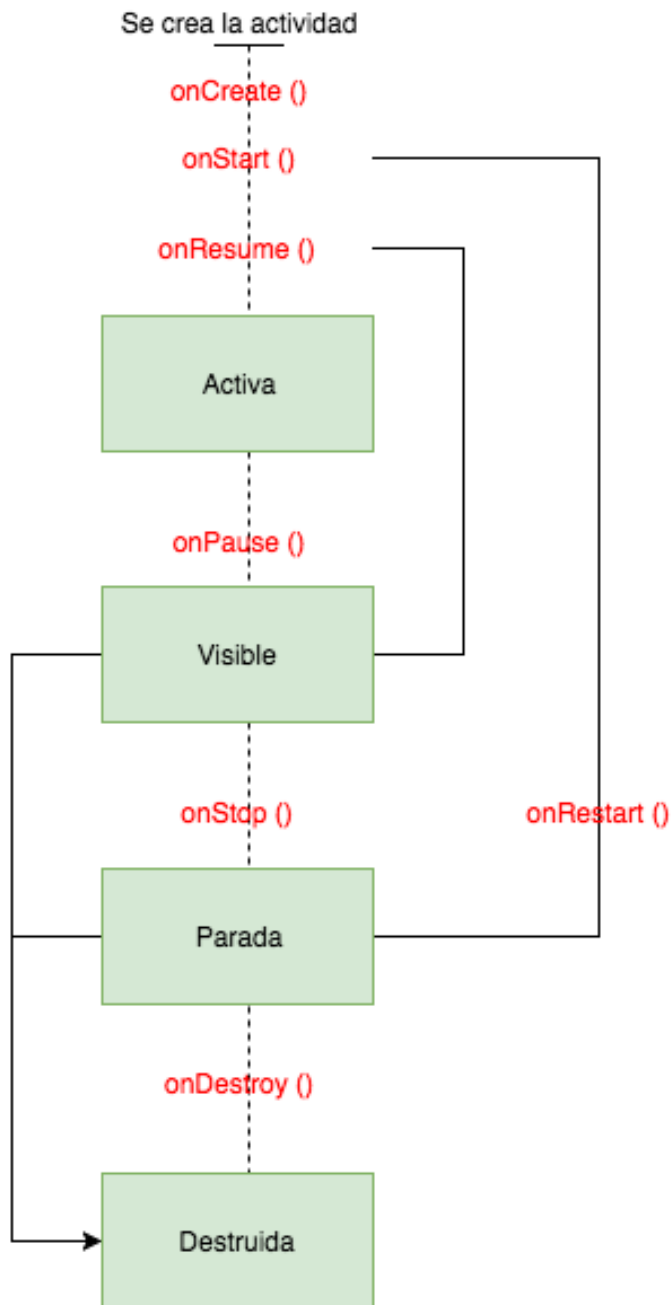


Ilustración 2.5 Ciclo de vida de un Activity.

3. Tecnología NFC

En esta sección detallaremos como funciona la tecnología que usaremos en la aplicación, exponiendo los distintos tipos de etiquetas, definiendo tecnologías similares, comparandolas y viendo las ventajas y desventajas de esta tecnología.

3.1. Introducción

La tecnología NFC (Near Field Communication) pasó a formar parte del estándar ISO/IEC en 2003.

Dicha tecnología se basa en la comunicación inalámbrica de corto alcance (inferior a 10 cm) alta velocidad (hasta 848 Kbits/s) y que trabaja entre las frecuencias 13-56 MHz sin necesidad de licencia [20] [21] [22].

Esta tecnología está basada en el funcionamiento del sistema RFID (Radio Frequency Identification) pero el objetivo de esta tecnología es transmitir el identificador del objeto mediante radiofrecuencia [23].

De forma análoga a la tecnología NFC, estas también usan unos pequeños dispositivos que contienen una antena que permite la comunicación por radiofrecuencia que no necesitan alimentación eléctrica (etiquetas pasivas) por otro lado, los dispositivos encargados de la comunicación si que necesitan alimentación eléctrica (etiquetas activas).



Ilustración 3.1 Etiqueta NFC.

3.2. Tipologías de etiquetas NFC

Las etiquetas NFC o Tags son microchips con una pequeña antena acoplada sin alimentación que se usa para almacenar una cantidad no muy grande de datos para su posterior lectura, hay dos grandes tipos que son los NTAG y los Mifare, ambos se subdividen en varios tipos:

- Serie NTAG: Su característica principal es que son compatibles con todos los dispositivos móviles, existen varios tipos como el NTAG203 o el NTAG213 que se describirán más adelante.
- Mifare Classic: Son etiquetas con una memoria con más capacidad pero requiere usar el protocolo mifare para su uso.
- Mifare Ultralight: Estas etiquetas son más compatibles que las anteriores pero tienen una capacidad y un costo menor [24].

Para poder observar mejor las diferencias entre los distintos tipos y poder ver sus características, se han resumido distintos tipos de etiquetas NFC en la siguiente tabla:

Características	NTAG203	NTAG213	NTAG216	Mifare 1K	Mifare Ultralight
Memoria Disponible	137 bytes	180 bytes	888 bytes	716 bytes	888 bytes
Número de serie	Sí	Sí	Sí	Sí	Sí
Cifrado	No	No	No	Sí	No
Usos frecuentes	Cualquier uso, recomendado para publicidad.	Chip muy versátil, listo para reemplazar al anterior.	Chip más avanzado de dicha serie, perfecto para V-card	Para usos con mucha memoria, necesita cifrado	Cualquier uso, recomendado para publicidad.
Compatibilidad tipo 2	Sí	Sí	Sí	No	Sí
Campos V-card	4-5	4-5	10+	9-10	1-2

Tabla 3.1 Resumen descriptivo de los tipos de Tags

La principal diferencia entre los tipos son la capacidad de almacenaje, la compatibilidad y el cifrado [25].

3.3. Funcionamiento

A continuación pasaremos a explicar los distintos modos de funcionamiento del protocolo NFC, teniendo en consideración que no son excluyentes entre sí.

3.3.1. Modo Activo/Pasivo

Este modo de funcionamiento no es un protocolo como tal, más bien son las dos maneras que tienen de funcionar los distintos dispositivos NFC:

Modo Activo: Es el modo con el cual funcionan las terminales, ya sean de pago o los propios teléfonos móviles, la cualidad principal de este modo es que el dispositivo genera su propio campo electromagnético necesario para transmitir datos, es necesario que al menos uno de los dispositivos use este modo.

Modo pasivo: En este modo el dispositivo no genera campo electromagnético para la comunicación, sino que usa la carga del otro dispositivo para la comunicación, este tipo de modo suelen usarlo las etiquetas sin baterías que suelen estar en tarjetas, pegatinas, llaveros etc [26].



Ilustración 3.2 Etiqueta pasiva NFC.

3.3.2. Modo Lectura/Escritura

Un modo de funcionamiento del protocolo NFC es el modo lectura/escritura en el cual la comunicación con el tag será tanto de lectura como de escritura de datos, teniendo una arquitectura basada en capas.

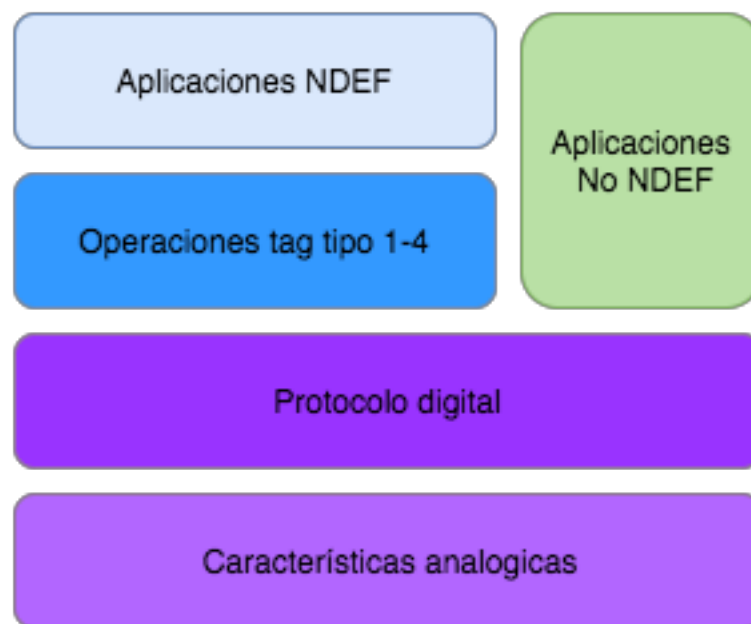


Ilustración 3.3 Capas de la arquitectura del modo lectura/escritura.

- Capa NDEF: Aplicaciones que usan el formato NDEF como vCard, SmartPoster o URL para guardar los datos.
- Capa de operaciones tipo1-4: Como la anterior esta capa se encarga de regular la interoperabilidad de la funcionalidad de los distintos tipos de tags.
- Capa no NDEF: Aplicaciones que no usan el formato NDEF para guardar sus datos.
- Capa digital: Esta capa se ocupa de coordinar la comunicación entre distintos tipos de dispositivos NFC.
- Capa analógica: La capa de más bajo nivel se encarga de controlar la comunicación a nivel de radiofrecuencia.

3.3.3. Modo Peer to Peer

El modo peer to peer o punto a punto es un protocolo de comunicación también usado en otro tipo de comunicaciones, se basa en la comunicación bidireccional entre dos dispositivos activos, normalmente usado para transmitir ficheros.

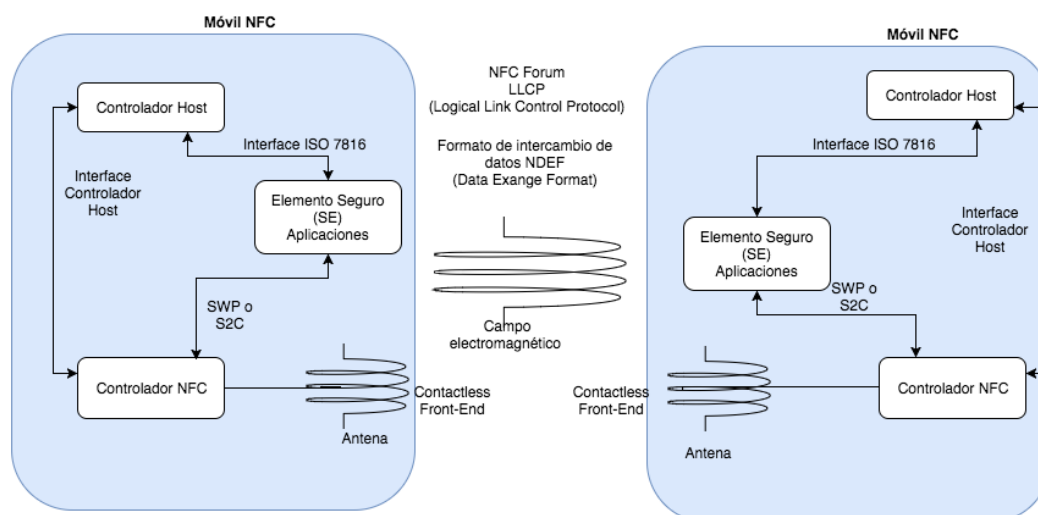


Ilustración 3.4 Arquitectura Peer to Peer.

Como podemos observar en el esquema de la figura 3.4, la aplicación se comunica usando el controlador NFC, empleando a posteriori el formato NDEF y en este caso en concreto una capa LLCP (Logical Link Control Procol).

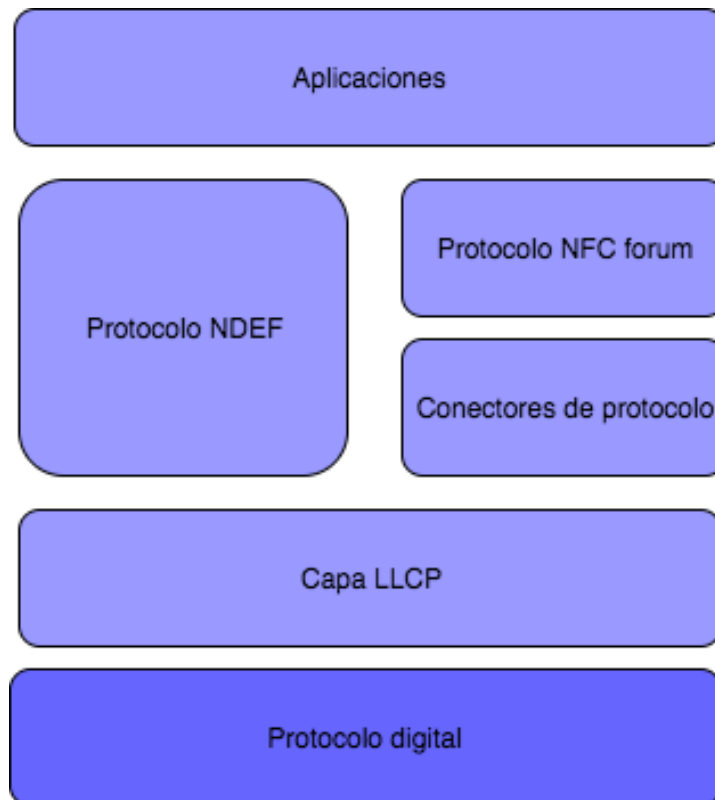


Ilustración 3.5 Capas de la arquitectura del modo Peer to Peer.

- En la capa de aplicaciones podemos encontrar el software que usan dicho modo para la comunicación NFC.
- La capa del protocolo NDEF encapsula los datos en mensajes para que pueda ser identificado más fácilmente.
- La capa del protocolo NFC forum es similar a la capa de operaciones tipo 1-4, por lo que su función es regular la funcionalidad de los distintos tipos de tags.
- La capa de conectores de protocolo es una capa cuya función es la de permitir y preparar la comunicación entre las capas adyacentes.
- La capa LLCP administra las conexiones bidireccionales, dando soporte así a aplicaciones de intercambio de datos siendo similar a ciertos protocolos de red.
- La capa digital tiene una funcionalidad similar en todos los modos de funcionamiento.

3.3.4. Modo Emulador

Este modo de funcionamiento se basa en usar el propio smartphone como tarjeta o tag, el principal uso de este modo es el pago.

El modo emulador comparte la capa física de los protocolos de comunicación con las tarjetas inteligentes, siendo la emulación más parecida a una de estas tarjetas inteligentes que a una etiqueta NFC.

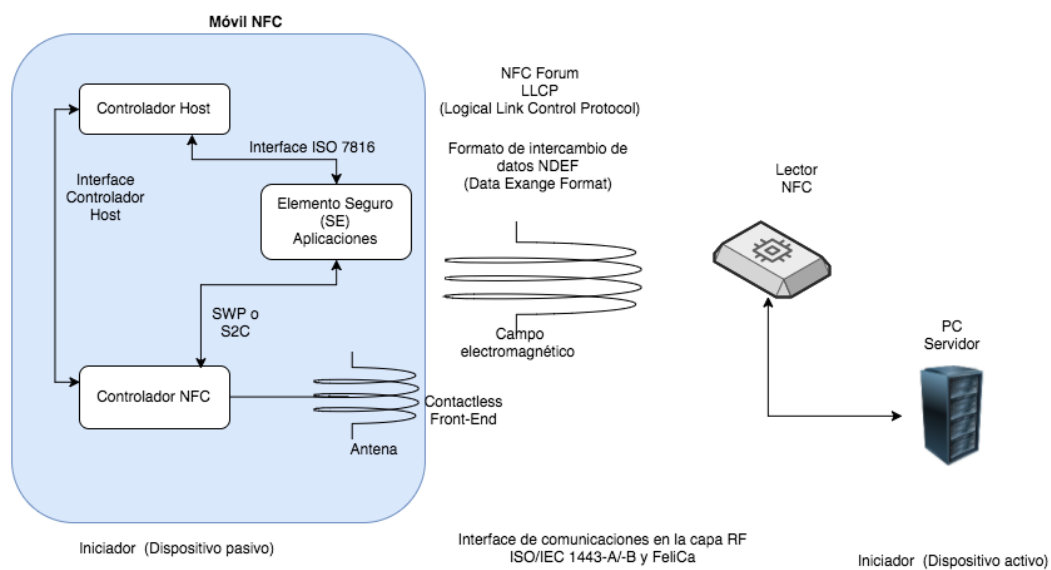


Ilustración 3.6 Arquitectura del modo emulador.

La comunicación se lleva a cabo por un hardware específico (SE Elemento Seguro) diseñado para compatibilizar con la tecnología NFC, este elemento está protegido o por el sistema operativo o por el fabricante del dispositivo, son estos quien a última instancia deciden quien puede tener acceso y quien no [27].

3.4. Formato NDEF

El formato NDEF (NFC Data Exchange Format) es el formato creado para el intercambio de datos en la comunicación NFC de tipo binario creado para empaquetar uno o varios payloads adaptables.

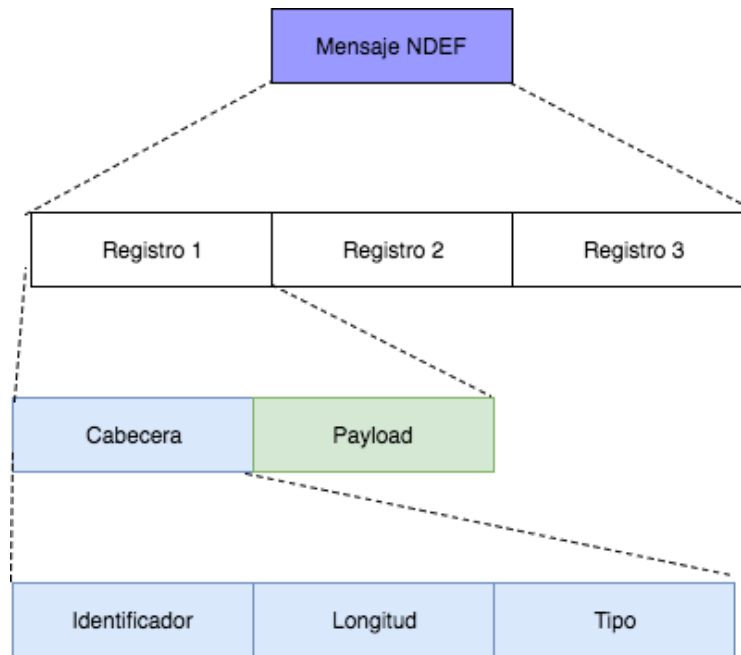


Ilustración 3.7 Mensaje NDEF.

Para conseguir esa versatilidad se han de incluir campos de cabecera para poder leer exactamente los datos como por ejemplo el tipo de payload y el tamaño, además de otros campos.

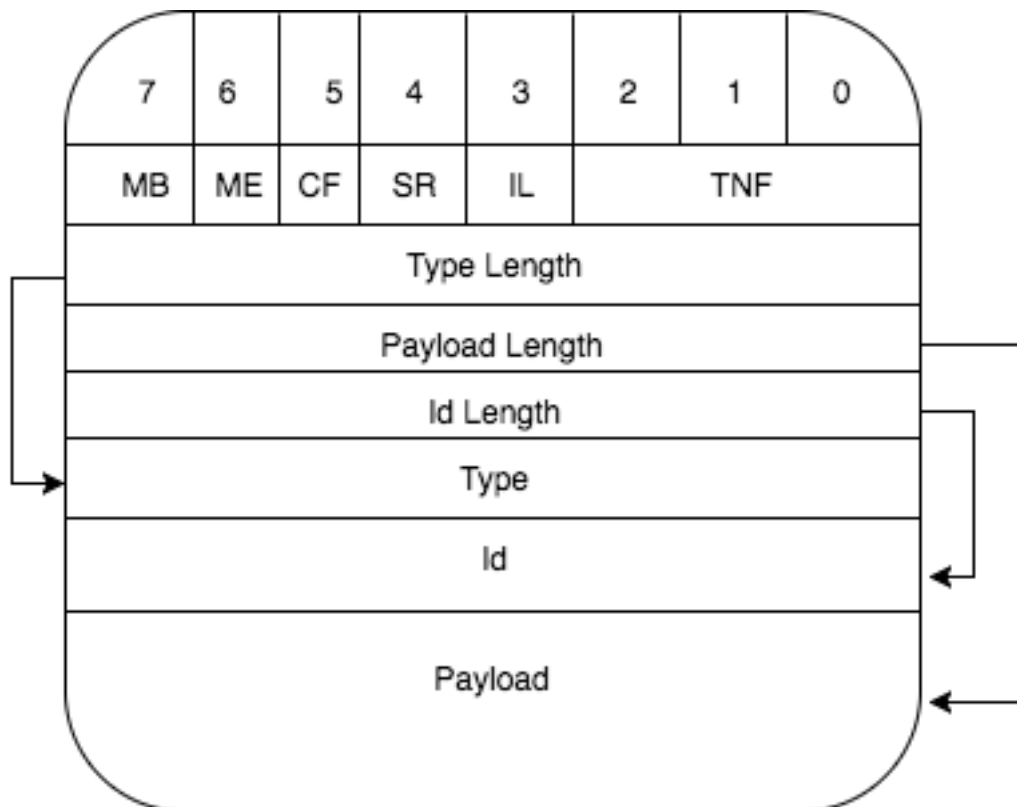


Ilustración 3.8 Campos del registro NDEF.

El cometido de los campos de la cabecera son los siguientes:

- MB: (Message Begin): Campo de 1 bit que indica el inicio de un mensaje NDEF
- ME: (Message End): Campo de 1 bit similar al anterior pero que indica el final de un mensaje.
- CF: (Chunk Flag): Campo de 1 bit que nos indica si es el primer paquete o es uno de los paquetes segmentados.
- SR: (Short Record): Campo de 1 bit que nos muestra que el campo Payload Length ocupa un solo octeto o si por el contrario ocupa cuatro.
- IL: (Id_Length is present) Es un campo de 1 bit que nos informa de la presencia del campo id_length.
- TNF: (Type Name Format) Campo de 3 bits que indica que estructura tendrá el campo type:
 - 0x00 significa que no hay datos asociados al registro
 - 0x01 tipo NFC Forum
 - 0x02 tipo de medios
 - 0x03 tipo URI

- 0x04 tipo NFC Forum externo
- 0x05 tipo desconocido
- 0x06 tipo sin cambios
- 0x07 Reservado
- Type Length: Es un campo de 8 bits que determina el tamaño del campo type.
- ID_LENGTH: Este campo de 8 bits determina el tamaño del campo ID cuando el campo IL esté activo.
- PAYLOAD_LENGTH: Este campo, como los anteriores, es de 8 bits y nos indica el tamaño del campo payload.
- Type: Este campo de tamaño variable contiene el tipo de dato que estará almacenado en el campo payload.
- ID: En este campo variable que se usa como identificador del mensaje en forma de URI.
- Payload: Este campo es el campo principal de la información a transmitir [28].

3.5. Aplicaciones de NFC

Pese a que la tecnología NFC lleva en activo desde el 2003, no tiene un uso muy extendido pero poco a poco vamos viendo como se usa cada vez un poco más.

- El principal uso de esta tecnología es la del pago a través del teléfono o mediante las tarjetas inteligentes o smart tags, ya que los verdaderos propulsores de dicha tecnología fueron las instituciones bancarias para poder sustituir las tarjetas convencionales, pese a que google wallet no está disponible en España podemos usar las aplicaciones propias de cada banco para realizar pagos a través de su cuenta bancaria.
- Esta tecnología puede ser muy útil para automatizar tareas complejas en un solo contacto con uno de los tags como generar varios perfiles que se pueden cambiar con un solo contacto, ya sea para el coche, activar el GPS y conectarlo a través del bluetooth sería muy útil o ponerlo en modo noche para que se active la alarma y se desactive los sonidos de llamadas o notificaciones además de la asistencia a personas mayores.

- Otro de los usos más empleados es la identificación, utilizando el teléfono como tarjeta inteligente podemos identificarnos en el check-in en las compañías aéreas e incluso se puede usar para fichar en una empresa usando el teléfono o una simple tarjeta.
- Además también puede ser muy útil en el mundo de la publicidad, podemos acceder a ciertos datos a través de un tag NFC que está en el anuncio para poder acceder a información ampliada, aunque no solo puede usarse con fines comerciales, también podemos acceder a los horarios de los autobuses o de actividades en algún evento.
- Otros de los usos es el de la conectividad, esta tecnología al ser muy rápida se puede usar para conectarse a otras tecnologías que requieren autenticación, como puede ser una red wifi o incluso conectar cierto dispositivo bluetooth [29].

3.6. Comparativa con tecnologías similares

En este apartado compararemos la tecnología NFC con otras tecnologías similares ya extendidas para evaluar su potencial.

Es imposible no comparar la tecnología NFC con la tecnología RFID, ya que están basada en la misma idea, pero tienen diferencias como el rango de actuación: (NFC tiene un rango de 10 cm mientras RFID tiene un rango de 3 metros) y sobre todo la usabilidad: (NFC tiene un uso más humano como puede ser la conectividad o el pago a través del teléfono, mientras que RFID es usado para obtener información de los objetos en los cuales está adherido) [30] [31].



Ilustración 3.9 Etiqueta RFID.

Otra tecnología que se asemeja por la sencillez son los códigos QR, esta se basa en un código impreso en dos dimensiones y con la aplicación necesaria se puede acceder a la información.



Ilustración 3.10 Ejemplo de código QR

Este método de almacenamiento es más económico que los tags NFC pero para su lectura se necesita una aplicación especial y enfocar al código, al contrario en la tecnología NFC solo es preciso acercar la etiqueta al lector.

Otra de las grandes diferencias son los usos, si es verdad que los códigos QR se usan mucho para marketing, publicidad e identificación en eventos la tecnología NFC tiene más variedad de usos, más seguridad y por lo tanto más potencial [32] [33] .

En este apartado veremos la comparación entre las tecnologías NFC y bluetooth. La tecnología NFC tiene un menor rango de actuación pero posee un tiempo de establecimiento de conexión muy bajo, muy útil para transferencia de una pequeña cantidad de datos de manera muy rápida.

En cambio la tecnología bluetooth cuenta con un alcance mayor, llegando hasta los 30 metros incluso con una velocidad mayor pero con un tiempo de establecimiento más alto, esta tecnología es muy usada para el intercambio de información con otros dispositivos como un ordenador o algún periférico inalámbrico, en ocasiones la tecnología NFC es usada para la identificación entre dispositivos bluetooth [34].



Ilustración 3.11 Símbolo de la tecnología Bluetooth.

La tecnología NFC se puede comparar con la tecnología beacon, esta se basa en la tecnología bluetooth low energy o de baja energía, su funcionamiento se basa en emitir un mensaje hasta unos 50 metros de distancia, por otra parte NFC solo transmite a unos cuantos centímetros de distancia.

En el caso de la tecnología beacon los dispositivos pueden recibir la señal fácilmente mientras que en el caso de NFC tienes que interactuar con la etiqueta, eso

hace que dicha tecnología sea mas invasiva, para conseguir este enlace mas sencillo la tecnología bluetooth de baja energia posee una pila de protocolo nueva que hace que la conexión sea mas rapida [35].

Por otra parte los dispositivos beacon necesitan de alimentación externa aunque son dispositivos eficientes ya que la batería suele durar unos dos años, en el ámbito de la seguridad beacon tiene algunos puntos de ataques, como la posibilidad de descifrar los códigos de identificación aunque es complejo ya que usa el protocolo HTTS que hace muy complicado el robo de información [36].

En el caso de la tecnología wifi es similar al caso de la tecnología bluetooth, la tecnología wifi tiene mayor alcance, mayor velocidad y posibilidad de crear infraestructuras de dispositivos, también tiene un tiempo de establecimiento mas alto, igualmente se pueden convinar ambas tecnologías para que la autenticación sea mas comoda, pero en este caso es menos usado [37].



Ilustración 3.12 Símbolo de la tecnología Wifi.

3.7. Ventajas y desventajas

La tecnología NFC tiene varias ventajas y desventajas que hacen que sea una tecnología para unos usos específicos.

- Principales ventajas:
 - La conexión entre los dispositivos es practicamente automática.

- La versatilidad de la tecnología hace que pueda tener múltiples usos.
- La sencillez de su uso, ya que con solo aproximar el tag al lector se produce el intercambio de información.
- Las etiquetas no requieren alimentación eléctrica.
- Principales desventajas:
 - La conexión es siempre entre dos dispositivos y no se permiten crear redes de comunicación.
 - Esta tecnología no se encuentra en todos los dispositivos del mercado aunque cada vez está disponible en un mayor número de estos.
 - La distancia necesaria para la comunicación es a la vez una ventaja y una desventaja, ya que nos obliga a que la comunicación sea de una distancia corta pero a la vez eso dificulta que alguien intercepte esa comunicación [38] [39].

3.8. Seguridad

La seguridad es siempre uno de los aspectos importantes en una tecnología de comunicación y más aun cuando con ella podemos acceder a la banca online.

Dada la naturaleza de la comunicación y al necesitar que la comunicación sea muy cercana, además de que la comunicación es punto a punto, resulta casi imposible que un tercero pueda suplantar la identidad de alguno de los comunicantes.

Desde el punto de vista interno, el chip NFC no tiene acceso a los datos del teléfono, en el caso contrario, sería necesario que el chip pidiera los datos necesarios para realizar una transferencia, por lo tanto un troyano en el dispositivo no saldría provechoso [40].

4. Ingeniería del Software

Este apartado se detallará el proceso de ingeniería del software, el cual lo podemos definir como el planteamiento metódico del desarrollo de un software, el cual nos ofrece metodos y técnicas para que el el software sea de calidad y para que su mantenimiento sea facil [41].

4.1. Introducción

La ingeniera del software nos ayuda a que nuestro software sea más eficiente, más seguro, y más facil de mantener. Además que nos proporciona una base para poder empezar el desarrollo de forma más cómoda.

Los procesos más importantes de la ingenieria del software son:

- Requerimientos.
- Sistema
- Diseño
- Implementación

4.2. Esquema de Requerimientos

Los requerimientos de un sistema describe lo que dicho sistema puede realizar y lo que no puede realizar.

4.2.1. Funcionales

Los requerimientos funcionales nos explica la esencia del sistema y su funcionalidad.

- Lectura de etiquetas NFC: El sistema es capaz de leer el contenido de los tags NFC y poder realizar la acción escrita al contacto escrito en el otro tag, lo nombraremos RF1.
- Realizar la acción de comunicación: El sistema reconoce el código de acción y decide en base a este si realiza una llamada o manda un sms al contacto suministrado, lo nombraremos RF2.

- Escritura de etiquetas NFC: El sistema tiene la capacidad de escribir en los tags el número de contacto con el cual nos queremos identificar o la capacidad de escribir el código de comunicación y el mensaje es caso de ser necesario, lo nombraremos RF3.

4.2.2. No Funcionales

Los requerimientos no funcionales muestran como debe ser el sistema.

Requerimientos hardware:

- El sistema se podrá ejecutar en un smartphone.
- Deberá ejecutarse en un sistema android 2.3.3 o superior.
- El dispositivo ha de ser compatible con la tecnología NFC.

Requerimientos software:

- La interfaz del sistema se ha diseñado para ser lo más intuitiva posible, ya que el usuario medio no será experto.
- El sistema se comunicará con el usuario para que este sepa si a ocurrido algún fallo o por el contrario todo se ha realizado con éxito.

4.3. Análisis del Sistema

El análisis del sistema es un proceso en el cual se definen los perfiles de los usuarios, el uso que se le dará al sistema y los escenarios donde este se ejecutará.

4.3.1. Perfiles de usuarios

Estudiar el perfil de los usuarios es un paso importante ya que nos ayuda a decidir cómo se ha de realizar el diseño.

- Usuarios Inexpertos: Este tipo de usuario será el que tendremos que tener más en consideración, ya que posiblemente sea los que usen más la aplicación.

- Los conocimientos del uso de un smartphone son muy limitados o nulos.
- En ciertas ocasiones pueden tener limitaciones visuales.
- Usuarios Intermedios: Este tipo de usuario tendrá un papel secundario, ya que serán los posibles encargados de realizar la escritura de los datos en los tags para su posterior uso.
 - Los conocimientos de uso de un smartphone son medios.
 - Los conocimientos del uso de la tecnología NFC pueden ser escasos debido a su poco uso.

4.3.2. Casos de uso

Los casos de uso son una descripción de como un usuario se supone que interactuará con el sistema, siendo los usos principales similares a los requerimientos funcionales.

Para la definición de los casos de uso usaremos a los actores principales de interacción, las condiciones de entrada y condiciones de salida cuando sea preciso, los eventos que ocurren y las excepciones que puedan ocurrir.

Para la representación de los diagramas emplearemos diversas formas para representar a los diversos elementos:

- Los usuarios vendrán representados por un sencillo dibujo de humano.
- Los casos de uso estarán representados a través de un óvalo con el nombre de este en el interior.
- El sistema o escenario será representado por un gran rectángulo que englobará a todos los casos de usos que tengan relación con el.
- Las relaciones entre casos de uso están representadas por flechas que pueden ser de varios tipos:
 - Flecha punteada: pueden representar tanto dependencias como instancias que son relaciones entre clases.
 - Flechas huecas: representan generalizaciones que se usan cuando un caso de uso hereda de otro caso de uso.
 - Flechas finas: son usadas para representar la asociación entre el actor y los casos de uso con los que va a interactuar.

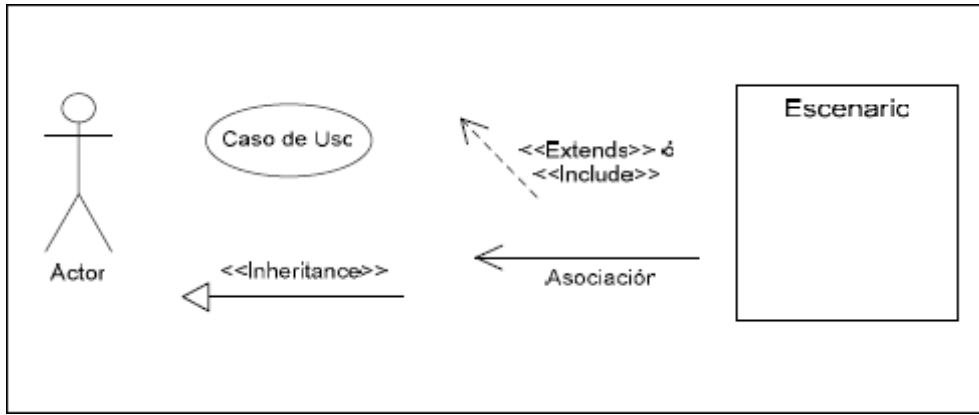


Ilustración 4.1 Elementos de los casos de uso.

Para comenzar tendremos el diagrama frontera que nos muestra un resumen de como el usuario interactua con el sistema.

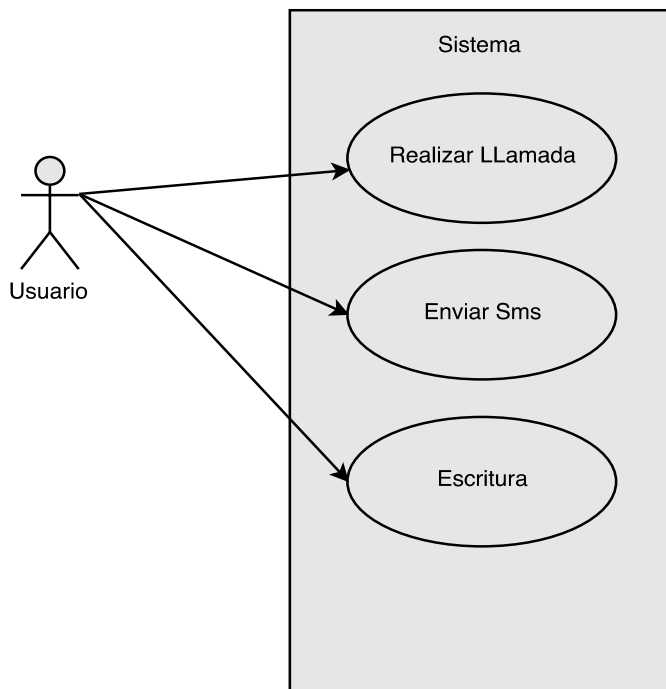


Ilustración 4.2 Diagrama frontera.

A continuación, nos disponemos a desarrollar los distintos casos de uso uno por uno.

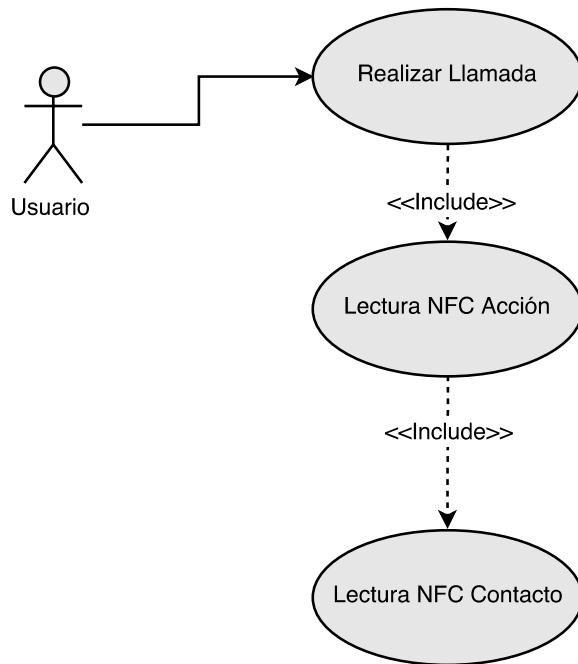


Ilustración 4.3 Diagrama de caso de uso: Realizar Llamada.

Caso de uso: Realizar Llamada.

- Actuador: Usuario
- Condiciones de entrada: Los tags deberán estar configurados correctamente y el smartphone deberá soportar la tecnología NFC.
- Eventos:
 - El usuario ha de comunicarse con el tag de las acciones en este caso estará configurado para que sea una llamada.
 - El usuario deberá comunicarse con el tag del contacto.
 - La aplicación realizará una llamada al número de teléfono dado por el tag NFC.
- Excepciones: El sistema puede fallar al leer los tag NFC, por lo tanto le faltarán datos para realizar la operación. En este caso el sistema mostrará un mensaje de error y el usuario deberá intentar la comunicación de nuevo.
- Condiciones de salida: El sistema emitirá una llamada al contacto seleccionado.

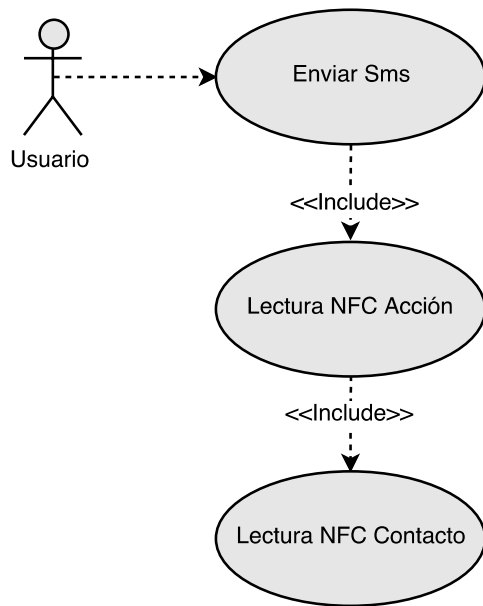


Ilustración 4.4 Diagrama de caso de uso: Enviar un mensaje.

Caso de uso: Enviar un mensaje.

- Actuador: Usuario
- Condiciones de entrada: Los tags deberán estar configurados correctamente y el smartphone deberá soportar la tecnología NFC.
- Eventos:
 - El usuario tendrá que establecer comunicación con el tag de las acciones, en este caso estará configurado para que se envíe un mensaje de texto.
 - El usuario deberá comunicarse con el tag del contacto.
 - La aplicación enviará un mensaje de texto al número de teléfono dado por el tag NFC.
- Excepciones: El sistema puede fallar al leer los tag NFC, por lo tanto le faltarán datos para realizar la operación, en este caso el sistema mostrará un mensaje de error y el usuario deberá intentar la comunicación de nuevo.
- Condiciones de salida: El sistema enviará un mensaje de texto al contacto seleccionado.

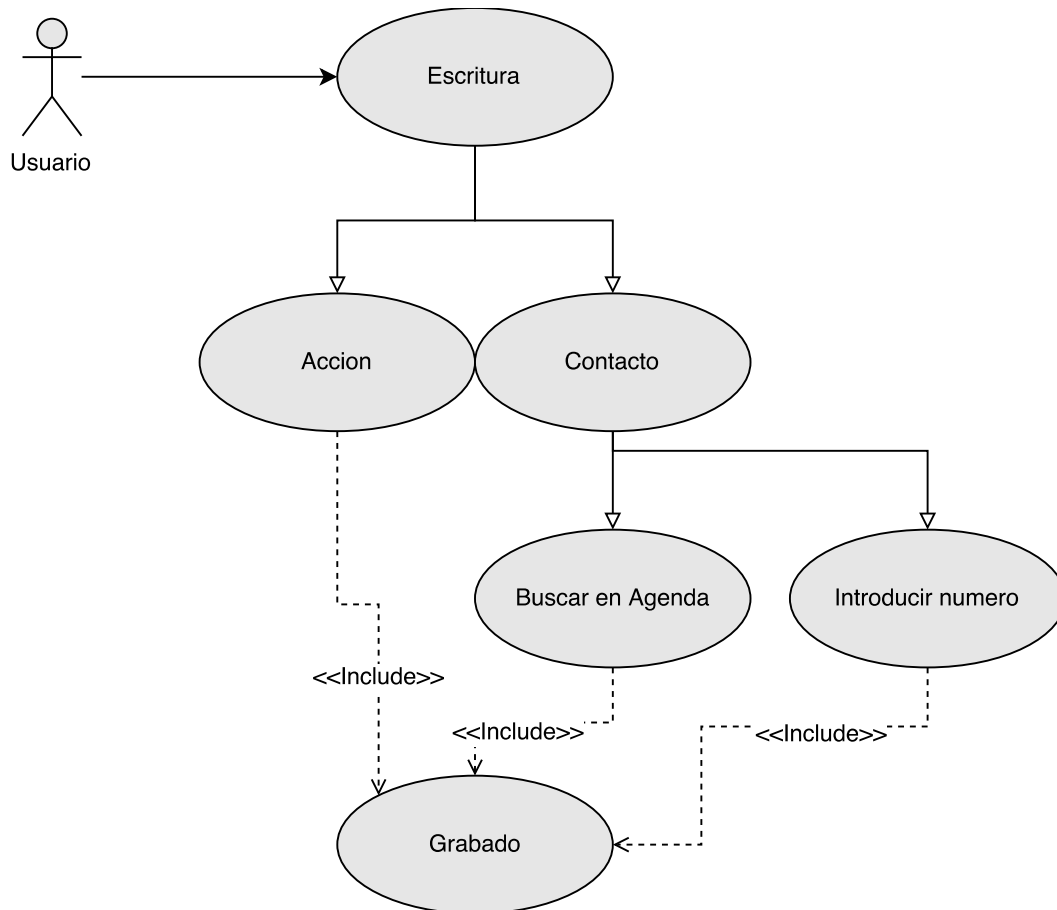


Ilustración 4.5 Diagrama de caso de uso: Escritura de un tag NFC.

Caso de uso: Escritura de un tag NFC.

- Actuador: Usuario
- Condiciones de entrada: el smartphone deberá soportar la tecnología NFC y los tags no deberán estar protegidos contra escritura.
- Eventos:
 - El usuario elige entre escribir en el tag una acción o un contacto.
 - En el caso de escribir la acción, el usuario elegirá cual es la que más le conviene y solo en el caso que sea un mensaje este lo introducirá en el cuadro de texto.
 - En el caso de escribir un contacto, el usuario podrá buscarlo desde la agenda o introducir el número de teléfono manualmente.
 - El usuario tendrá que establecer comunicación con el tag para que el sistema pueda guardar los datos en el tag para su posterior uso.

- Excepciones: El sistema puede fallar al escribir los tag NFC, en este caso el sistema mostrará un mensaje de error y el usuario deberá rehacer la operación.
- Condiciones de salida: El sistema mostrará un aviso sobre el éxito o el fracaso de la operación.

4.3.3. Técnica Personas

La Técnica personas es una técnica muy útil para conocer a los usuarios potenciales y sus necesidades.

Dicha técnica se basa en crear un grupo de usuarios ficticios que representen a la población que usará nuestro software con el objetivo tener más conocimientos sobre el usuario y poder simular la reacción en determinadas circunstancias de los usuarios.

- Nombre: Juan Ortiz Perez.
- Profesion: Jubilado (antiguo barrendero).
- Edad: 76
- Habilidades: Usuario medio de teléfonos convencionales, los smartphones le vienen grande.
- Herramientas: BQ Aquaris U
- Trasfondo: Juan es un hombre que vive solo porque está viudo y sus hijos están independizados, tiene problemas de cadera pero es autosuficiente, posee cierta limitación visual así que le es costoso el manejo del smartphone, le gusta pasear y visitar a sus familiares y amigos cercanos.

- Nombre: Maria Azuaga Martin.
- Profesion: Jubilada (antigua dependienta).
- Edad: 80
- Habilidades: Usaria principiante en el manejo tanto de teléfonos tradicionales como de smartphones.
- Herramientas: Blackview BV6000s
- Tranfondo: Maria es una mujer que vive con su marido, sus hijos viven en el extranjero, por lo que la comunicación con ellos es por teléfono. Maria tiene problemas de rodilla, poseé cierta limitación visual y dificultad de usar las manos para tareas de mucha precisión, Maria suele andar hasta la tienda más cercana.

4.3.4. Escenarios

La técnica de escenarios consiste en crear una historia con la inclusión de actores, acontecimientos, elementos y entornos. Es una herramienta que nos ofrece reflejar varios puntos de vista sobre el uso de nuestro software.

Escenario 1:

- Actores: Juan
- Elementos: Smartphone
- Acontecimientos:
 - Juan al levantarse del sofa sufre un dolor en la espalda que le impide moverse con normalidad.
 - Juan sufre dolores en la espalda y decide que debe ponerse en manos de un profesional.
 - Juan al no poder moverse con normalidad por el pequeño accidente intenta usar su teléfono para llamar a su hijo pero con el dolor la tarea es más tediosa.
- Entorno: El suceso ocurre en la casa de Juan.

Escenario 2:

- Actores: Juan
- Elementos: Smartphone con nuestra aplicación.
- Acontecimientos:
 - Juan al levantarse del sofa sufre un pinchazo en la espalda que le impide moverse con normalidad.
 - Juan sufre dolores en la espalda y decide que debe ponerse en manos de un profesional.
 - Juan solo tiene que colocar brevemente su smartphone sobre las pegatinas que tiene encima de la mesa y podrá llamar a su hijo comodamente.
- Entorno: El suceso ocurre en la casa de Juan.

Escenario 3:

- Actores: Juan
- Elementos: Smartphone
- Acontecimientos:
 - Juan llega a casa después de dar un agradable paseo.
 - Juan decide intentar mandar un mensaje de texto a su hijo diciéndole que ha llegado a casa de su paseo, comunicándole que se encuentra bien.
 - Juan se hace un lío y envía un mensaje ilegible a su hijo.
 - El hijo de Juan recibe un mensaje raro y se preocupa.
 - El hijo de Juan va a ver a su padre preocupado pero se da cuenta de que está perfectamente.
- Entorno: El suceso ocurre en la casa de Juan.

Escenario 4:

- Actores: Juan
- Elementos: Smartphone con nuestra aplicación.
- Acontecimientos:
 - Juan llega a casa después de dar un agradable paseo.
 - Juan decide mandar un mensaje de texto a su hijo diciéndole que ha llegado a casa de su paseo, expresando que se encuentra perfectamente.
 - Juan usa la aplicación que su hijo le configuró para que la comunicación le resultara más sencilla.
 - Juan envía un mensaje predeterminado a su hijo.
 - El hijo de Juan recibe el mensaje y sabe con certeza que su padre se encuentra bien.
- Entorno: El suceso ocurre en la casa de Juan.

Escenario 5:

- Actores: Maria
- Elementos: Smartphone.
- Acontecimientos:
 - El marido de Maria sale a hacer la compra.
 - Maria se da cuenta que no les queda leche en la despensa.
 - Maria intenta llamar a su marido pero le resulta complicado.
 - El marido de Maria llega de hacer la compra sin leche.
- Entorno: El suceso ocurre en la casa de Maria.

Escenario 6:

- Actores: Maria

- Elementos: Smartphone con nuestra aplicación.
- Acontecimientos:
 - El marido de Maria sale a hacer la compra.
 - Maria se da cuenta que no les queda leche en la despensa.
 - Maria usa la aplicación NFC para llamar a su marido y avisarle sobre el producto que debe comprar.
 - El marido de Maria llega de hacer la compra con leche.
- Entorno: El suceso ocurre en la casa de Maria.

4.4. Diseño

El diseño del software es de vital importancia, ya que proporciona una ruta a seguir en el transcurso del desarrollo del software. Además se pueden detectar fallos o puntos a mejorar antes de que sea más costosa su reparación.

El diseño se ha dividido en tres apartados, diseño del sistema, diseño de datos y diseño de la interfaz.

4.4.1. Diseño del sistema

En este apartado se detalla el diseño del sistema a través de un diagrama de clases y los diagramas de secuencias.

4.4.1.1. Diagrama de clases

Los diagramas de clases son esquemas que nos muestran como están estructuradas y la forma en la que están relacionadas las clases en un sistema basado en el paradigma orientado a objetos.

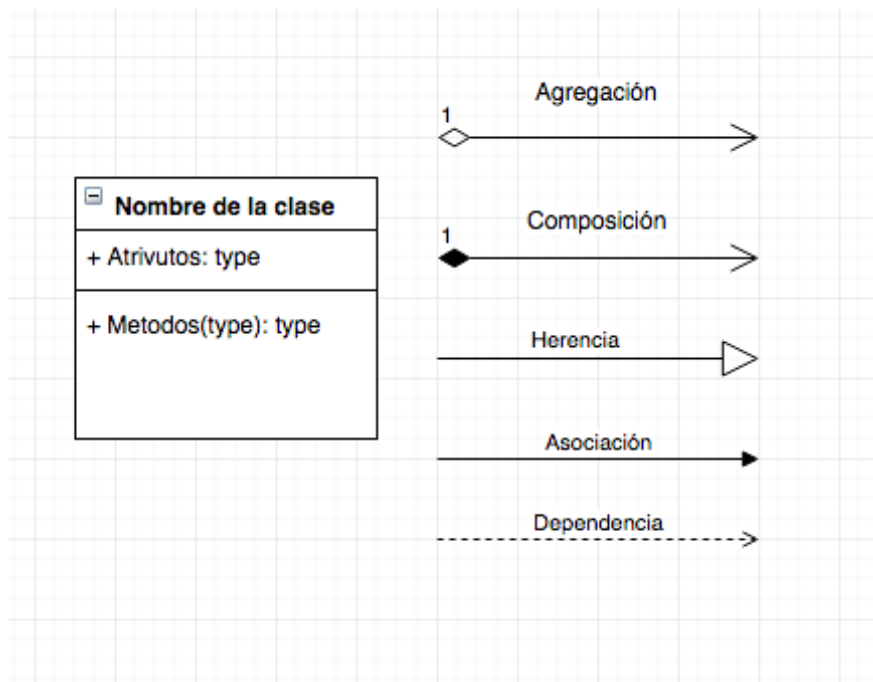


Ilustración 4.6 Elementos base de un diagrama de clases.

- Las clases vienen representadas por un rectángulo dividido en tres partes:
 - En la parte superior podemos observar el nombre de la clase, el cual debe tener un nombre representativo.
 - Justo debajo tendremos una lista de los atributos de la clase, en caso de que sean muchos estarán los más representativos.
 - En la parte inferior tendremos los métodos más importantes de dicha clase.
- Las relaciones están representadas por flechas que conectan dos clases, dependiendo de que tipo de flecha tendrán una relación u otra:
 - Asociación: Es una relación entre dos clases que puede ser bidireccional (en este caso no hay flecha)
 - Agregación: La clase que está junto al rombo blanco posee objetos de la clase agregada, pero con cierta libertad de existencia.
 - Composición: Es similar a la agregación solo que la existencia de la clase con menos peso no tiene sentido sin la clase global.

- Dependencia: Es un tipo de relación débil, la relación suele basarse en el uso de ciertos métodos de otra clase para evitar duplicarlos.
- Herencia: En esta relación la subclase posee los atributos y métodos de la superclase que se pueden identificar porque es a la que señala la flecha.

El diagrama de clases usado en nuestra aplicación en el siguiente:

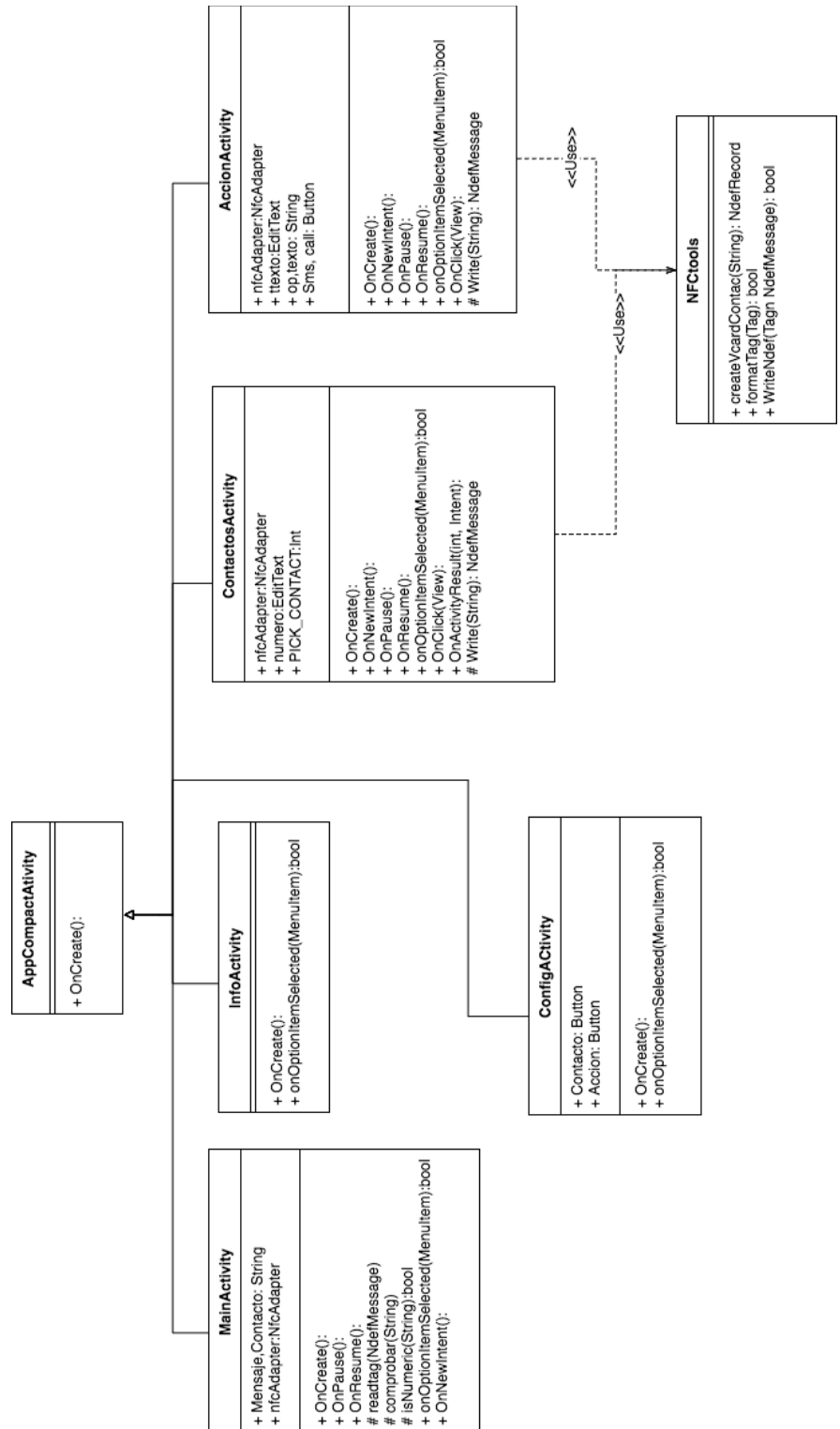


Ilustración 4.7 Diagrama de clases.

Como podemos observar en nuestro diagrama de clases, disponemos de cinco clases que heredan de “AppCompatActivity” y otra clase extra que es usada por dos de las otras clases:

- **MainActivity:** Es la clase principal y por tanto es la encargada del tránsito hacia otras vistas como de la lectura de los tags NFC y de las comprobaciones pertinentes para evitar errores como por ejemplo: comprobar que el contacto suministrado sea de tipo numérico o que la lectura de la acción tenga el formato usado en la aplicación.
- **InfoActivity:** Esta clase se encarga de controlar la vista que nos proporciona información sobre la aplicación.
- **ConfigActivity:** Es la encargada de regir la vista en la cual está la configuración, desde esta podemos optar por escribir una acción nueva o un contacto nuevo en un tag libre.
- **ContactosActivity:** Esta clase es la responsable de su vista asociada, además de llamar al content provider de la agenda para poder acceder a un contacto determinado y posteriormente grabarlo en una etiqueta.
- **AccionActivity:** Dicha clase es similar a la anterior, salvo que en este caso los datos a grabar en las etiquetas son la acción que se quiere efectuar, además de la inclusión de un texto en el caso que la acción fuese enviar un mensaje de texto. Ambas acciones son parte de la configuración pero no tienen relación con la clase ConfigActivity, aunque se necesite acceder a esta antes.
- **NFCtools:** Esta clase es una clase auxiliar usada por las dos clases anteriores, ya que muchas de las funciones que usan para la escritura en las etiquetas NFC son las mismas se han decantado por encapsularlas en una clase distinta.

4.4.1.2. Diagrama de secuencia

Una buena forma de poder entender cómo interactúan los objetos entre sí es realizando diagramas de secuencia, ya que proporcionan una visión sobre el tiempo del uso y la vida de las distintas clases que intervienen en cada actividad.

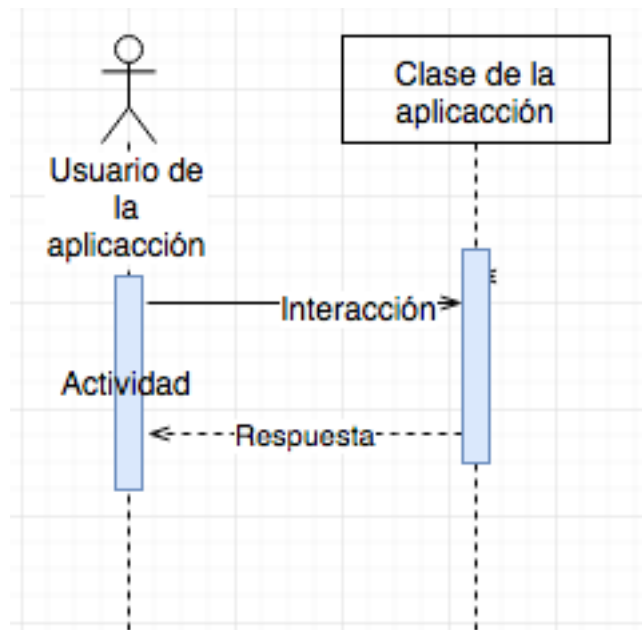


Ilustración 4.8 Ejemplo de diagrama de secuencia.

Para comprender mejor los diagramas de secuencia es importante conocer los elementos básicos de este:

- El actor: Se simboliza con un dibujo de un humano y representa al usuario de la aplicación.
- Clases de la aplicación: Son las distintas clases que entran en juego en la actividad representada por el diagrama.
- Tiempo de vida: Viene representado por la línea de rayas que baja desde el actor o de las clases, define la duración de la vida de la clase o de la actividad del usuario.
- Actividad: Representado por rectángulos sobre la línea del tiempo de vida de una clase o un actor simbolizan los tramos de tiempo en los cuales hay una actividad relevante.
- Mensajes: Están simbolizados con una flecha que sale desde una actividad y llega a otra actividad. Los mensajes principales se representan mediante flechas lisas, suelen ser llamadas o peticiones a otras clases, las respuestas se simbolizan por flechas intermitentes, suelen ser una devolución de datos o una confirmación.

A continuación están representados los diagramas de secuencia de las principales acciones de nuestra aplicación:

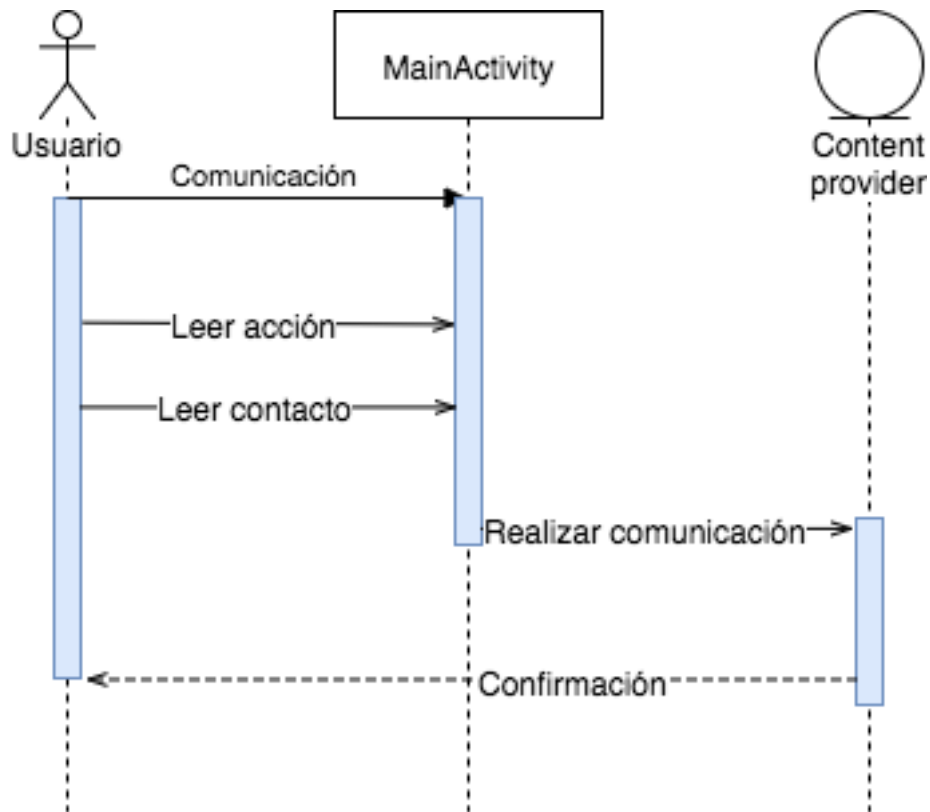


Ilustración 4.9: Diagrama de secuencia: Comunicación.

El usuario a través de las etiquetas NFC le proporciona la acción y el contacto a la clase MainActivity y esta se encarga de llamar al content provider de la función de llamadas o de la función de mensajería para que realice la comunicación. El usuario recibirá una confirmación si se ha realizado o no.

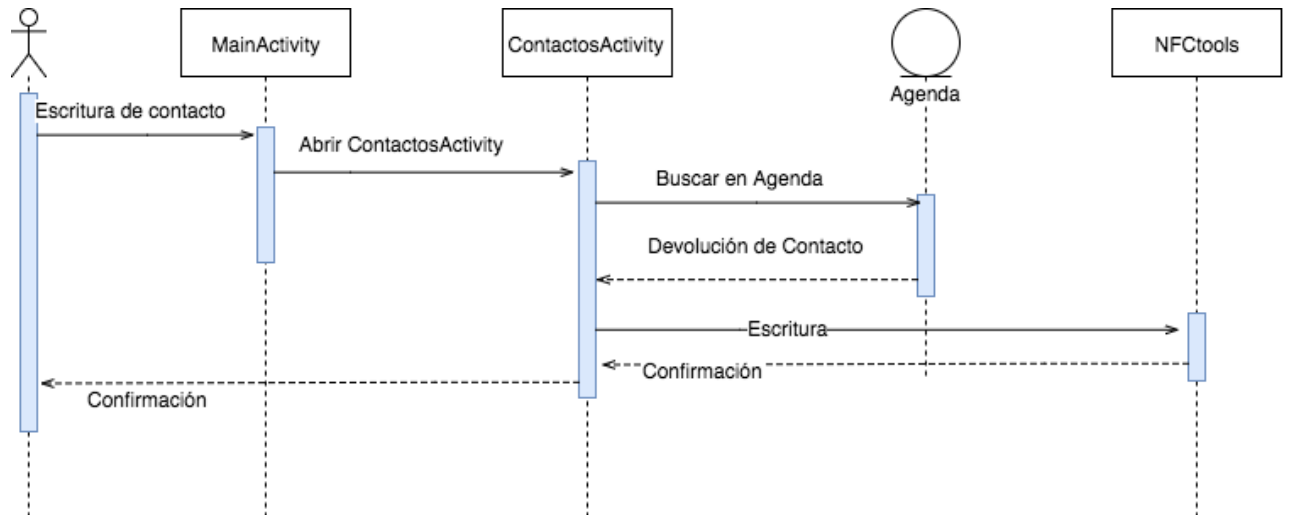


Ilustración 4.10: Diagrama de secuencia: Escritura de contacto.

El usuario entra en la aplicación y se comunicará con la clase MainActivity, para poder acceder a la configuración de los contactos es necesario la clase ContactosActivity. Esta, a su vez, podrá acceder a la agenda para que el usuario busque un contacto específico o puede escribirlo directamente.

Para terminar de efectuar la escritura se requiere de la clase NFCtools para que la realice la escritura. El usuario irá recibiendo confirmaciones o mensajes de error en el proceso de escritura.

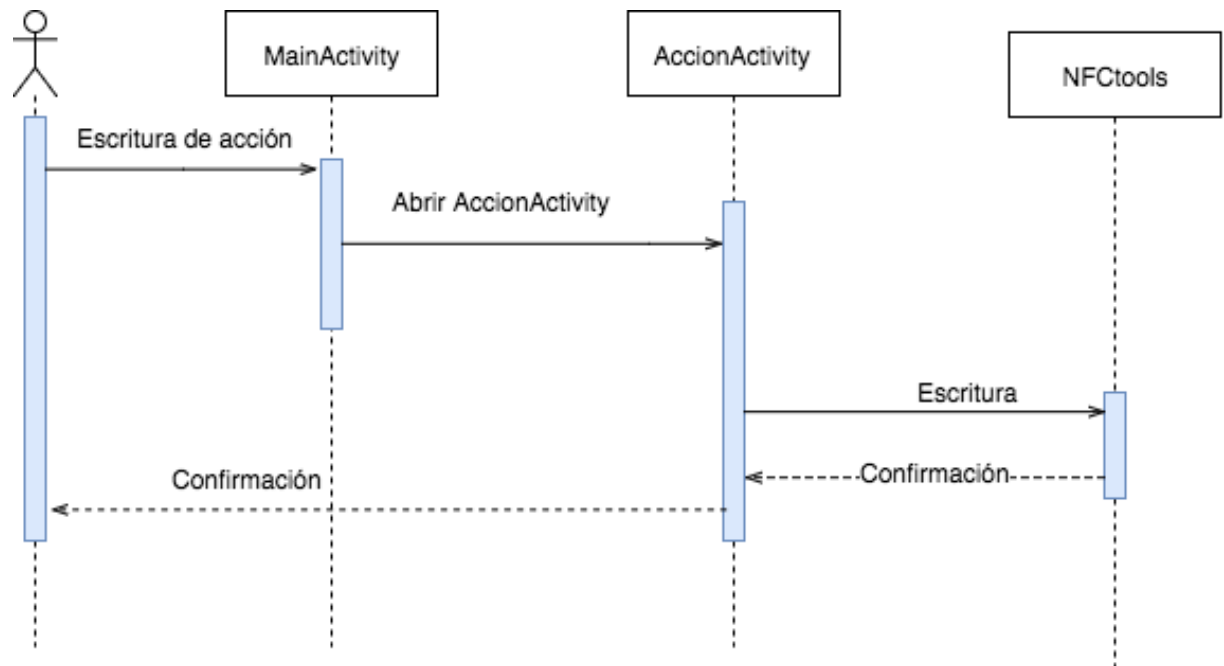


Ilustración 4.11: Diagrama de secuencia: Escritura de acción.

Esta situación es muy similar a la anterior, ya que en lugar de escribir un contacto se escribe la acción a realizar.

El usuario a través del MainActivity accederá al AccionActivity, en el cual configurará de manera sencilla que acción se asociará a la etiqueta NFC. De manera similar a la anterior se usará la clase NFCtools para que realice la escritura de los datos en la etiqueta.

Entre las clase MainActivity y las clases ContactoActivity y AccionActivity se llama a la clase ConfigActivity que se encarga de gestionar la vista que nos permite bifurcarnos para escoger entre la configuración de contactos o de acciones. La clase ConfigActivity no se incluye en ningun diagrama de secuencia porque su aportación es mínima.

4.4.1.3. Patrón de arquitectura software

El patrón de arquitectura usado para el desarrollo de nuestro prototipo está basada en el modelo-vista-controlador, ya que las propias herramientas usadas para el desarrollo del software te incentivan a separar la lógica de la interfaz.

Este patrón de arquitectura se basa en la separación de la lógica, los datos y la interfaz de usuario, con el objetivo de separar conceptos y facilitar su comprensión.

4.4.2. Diseño de datos

En este apartado se describirá el diseño de los datos de la aplicación, aunque en nuestro caso no tratamos con una gran cantidad de datos, analizaremos como han sido diseñados para el uso de la aplicación.

- Datos guardados en la etiqueta de acción: Se ha creado una correspondencia entre acción a ejecutar en un futuro y un sencillo código:
 - El número uno corresponde a la acción de llamada telefónica.
 - El número dos corresponde a la acción de mensajería básica de un teléfono que son los sms.
 - El número tres corresponde a la acción de mensajería por whatsApp, pero se encuentra desactivado.

En el caso de ser un mensaje es necesario guardarlo en la etiqueta, por lo tanto se ha decidido transformar el código en cadenas de texto, añadirles una marca para que se controle la separación entre el código de acción y el mensaje en caso de ser necesario.

Ejemplo: "1#" "2#he llegado a casa"

- Datos guardados en la etiqueta de contactos: En el caso de los contactos podemos acceder a más información si accedemos a través de nuestra agenda del teléfono, pero para nuestra aplicación solo necesitaremos el número de teléfono para efectuar futuras comunicaciones.

La forma efectuada se basa en una consulta a la agenda telefónica del usuario, del contacto seleccionado se extrae solo el número de teléfono, a posteriori se mostrará en el cuadro de texto. Al usuario se le da la

opción de introducir el número del contacto manualmente en el cuadro de texto, desde los datos introducidos en el cuadro de texto se pasarán a la etiqueta NFC correspondiente para su posterior uso.

4.4.3. Diseño de la interfaz

La interfaz es una parte muy importante de un proyecto software, ya que es el método de comunicación del software con el usuario y viceversa, por tanto es importante realizar un diseño de la interfaz intuitivo y fácil de usar para que la aplicación no caiga en desuso.

Se han separado los elementos de la interfaz de usuarios en tres apartados: metáforas, estilo y storyboard.

4.4.3.1. Metáforas

Las metáforas consisten en aplicar palabras a conceptos que realmente no tienen ese nombre, pero con ello se puede facilitar la comprensión. En el ámbito de la informática se suele usar para facilitar la relación entre una acción con un diseño.

Unas metáforas usadas en la informática son las tijeras que representan la función de cortar o a pesar de estar obsoleta la metáfora entre el disco floppy y la función de guardar los cambios realizados está muy presente. Además de las metáforas nuevas en los smartphones como los deslizamientos y el zoom.

A continuación se van a mostrar las metáforas usadas en la aplicación y el motivo de su elección.



Ilustración 4.13: Dibujo del icono de la aplicación.

Este símbolo se ha usado en el icono principal de la aplicación, ya que el símbolo del teléfono expresa que se trata de una aplicación sobre comunicación pero que no es la aplicación convencional.



Ilustración 4.14: Icono de la configuración.

Para la sección de configuración se ha optado por la rueda dentada que simboliza las entrañas de una gran maquinaria, pero que en nuestro caso no es así. Otra opción válida hubiera sido utilizar como símbolo un martillo y una llave inglesa pero me he decantado por este por su simplicidad.



Ilustración 4.15: Dibujo de la configuración.

Se le ha querido dar utilidad a un botón que suele aparecer en la parte inferior derecha para poder acceder a la configuración por otro camino, pero en este caso como el botón ya tenía un contorno diferenciable se ha optado por poner un símbolo similar al anterior solo que más sencillo, para que en caso de no estar claro, la otra opción pueda resultar más clara.



Ilustración 4.16: Icono de la información.

Para acceder a la sección de información de nuestra aplicación se ha optado por usar un icono similar al botón principal de configuración, ya que los dos estarán localizados en la barra superior de nuestra aplicación, como dibujo para simbolizar dicha sección se ha escogido la letra "I" ya que es muy común que sea ese símbolo elegido para dicha sección.



Ilustración 4.17: Icono de mensaje de texto.

A la hora de seleccionar el sistema de mensajería sms como opción buscaremos un icono que represente al servicio de mensajería, en este caso me he decantado por un par de diálogos de texto dentro de un círculo azul teniendo uno de

ellos tres puntos, el cual se asemeja mucho al icono que se suele usar para este tipo de servicio.



Ilustración 4.18: Icono para acceder a la agenda.

En el caso de la agenda de contactos me he decantado por usar como icono la típica libreta antigua donde se escribían los contactos, para representar la funcionalidad de entrar en tu agenda personal para seleccionar el contacto deseado. En mi opinión, creo que este logo es más representativo que muchos otros que se usan para la misma función en otros smartphones.



Ilustración 4.19: Icono de llamada.

A la hora de seleccionar el icono para la opción de llamada he sido simple y práctico, he buscado un icono similar a los iconos de llamadas que ya existen usualmente en los smartphones, debido a que representan muy bien la utilidad del botón con el símbolo del teléfono antiguo.

Para seleccionar los símbolos me he basado en mi propia experiencia, al observar diversos sitios web y aplicaciones que usan unas metáforas similares, ya que si el usuario está acostumbrado a un tipo de metáforas es mejor rehusarlas a tener que volver a entrenar al usuario, de esta manera el aprendizaje es mucho más rápido.

4.4.3.2. Estilo

Para el estilo general de la aplicación móvil me he decantado por usar como color primario el azul y como fondo a pesar de que se ha usado un fondo de pantalla el color predominante es el blanco.

Para mejorar el uso de la aplicación se ha añadido un dibujo como fondo de pantalla, explicando de manera sencilla la forma de uso, en los casos que se pueden hacer uso de la tecnología NFC, aunque dicho fondo se ha suavizado para que no resalte demasiado.

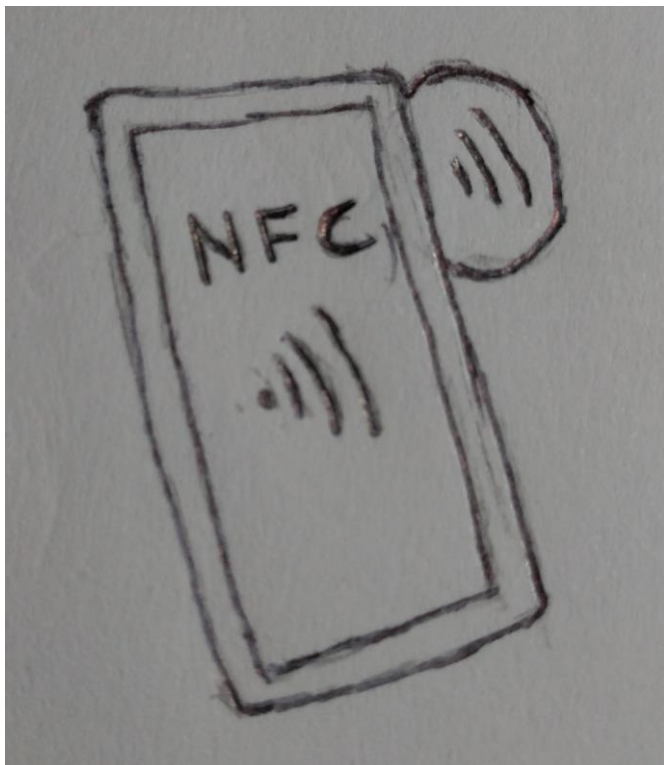


Ilustración 4.20: Boceto original del fondo de pantalla.

A la hora de las fuentes de texto la estrategia a seguir ha sido muy simple, mantener la fuente y tamaño por defecto ya que no se usa demasiado, se ha buscado que la aplicación sea sencilla, por tanto el objetivo ha sido usar iconos siempre cuando fuera posible.

4.4.3.3. Storyboard

El storyboard es un conjunto de ilustraciones usadas para mostrar una historia o un deslencance, es usado en muchos campos y también empleado en el campo de la informática para poder crear un prototipo de manera muy fácil y económica de como será la aplicación.

Crear un storyboard en la fase de diseño nos permite descubrir posibles errores de diseño de nuestro sistema y poder repararlos fácilmente, además de poder tener un entregable para saber si el cliente está satisfecho o no con la idea de producto.

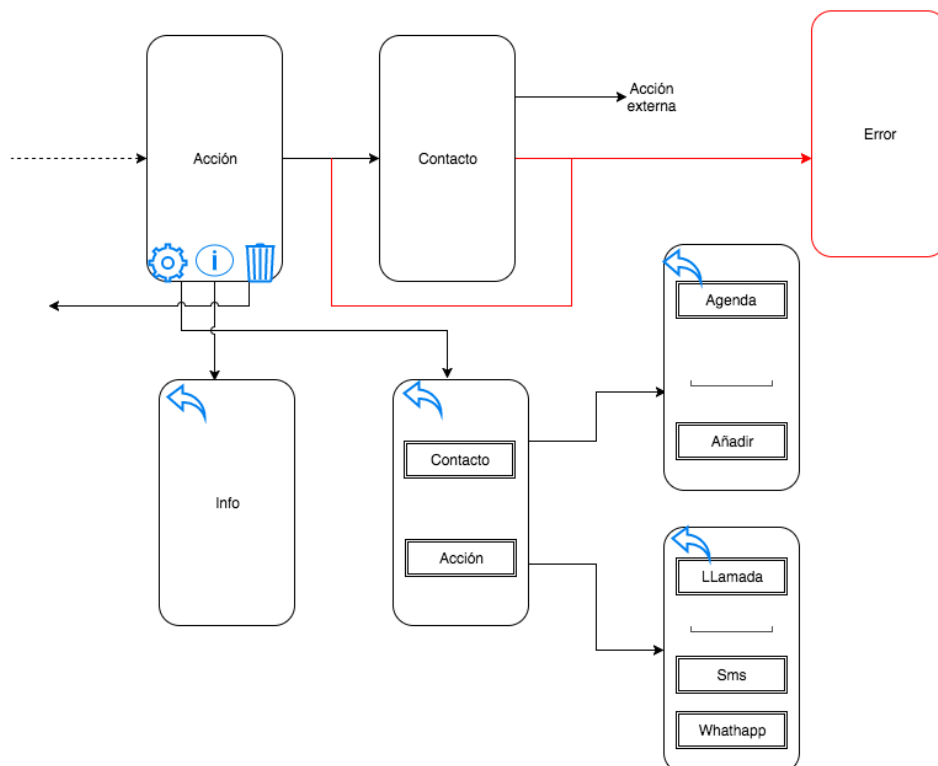


Ilustración 4.21: Primer storyboard de la aplicación.

Desde un principio ha estado más o menos claro el diseño de la aplicación, pero se ha tenido que adaptar dada la existencia de otras herramientas como por ejemplo: mostrar los mensajes de error en un cuadro de diálogo es más cómodo que tener una vista por cada error que pueda ocurrir, o colocar los botones de configuración en la parte superior y no en la inferior.

A posteriori se ha decidido tener un prototipo un poco más profesional por lo que me decante por usar un servicio online para realizar prototipos, el resultado es similar al anterior solo que dispone de la posibilidad de interactuar con el prototipo.

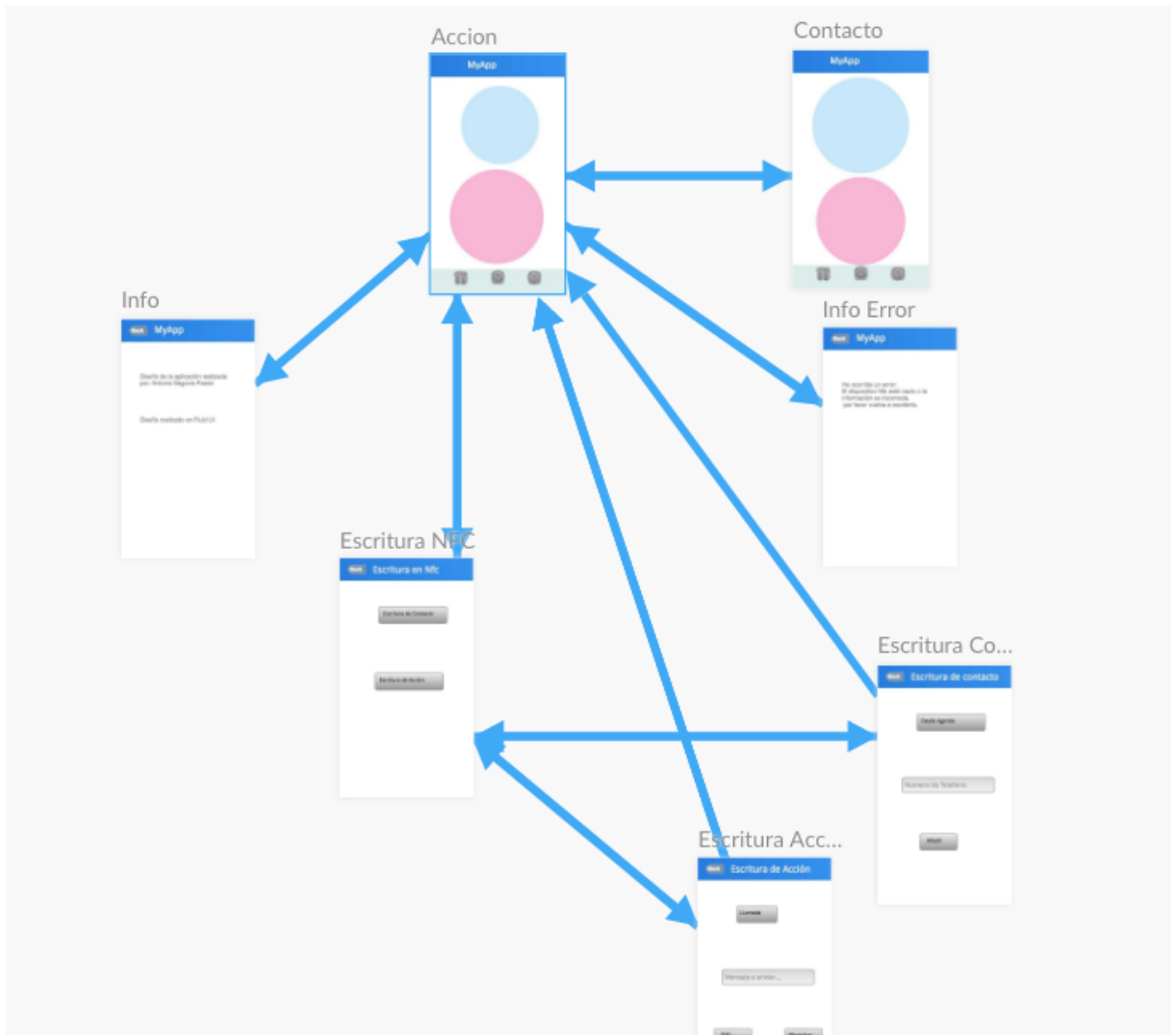


Ilustración 4.22: Storyboard de la aplicación en versión online.

El storyboard realizado en la aplicación es similar al realizado a mano solo que tiene la ventaja de ser fácilmente testeado, ya que tiene una versión interactiva que se muestra a continuación:

https://www.fluidui.com/editor/live/preview/p_NBttKaTXiEnCghDaU6Bfvnhe2uW4aTJF.14987703e35748

4.5. Implementación y pruebas

En la fase de implementación se usan los requisitos y el diseño para desarrollar el software usando unas determinadas herramientas un determinado lenguaje y si es preciso una determinada arquitectura.

4.5.1. Arquitectura

Para la arquitectura del prototipo de software se ha usado el patron de arquitectura modelo-vista-controlador aunque se ha adaptado a las necesidades del prototipo.

Los datos al estar almacenados en los tags tienen una separación natural, en este caso se ha desacoplado ciertas funciones de escritura NFC, ya que eran usadas desde varias clases distintas.

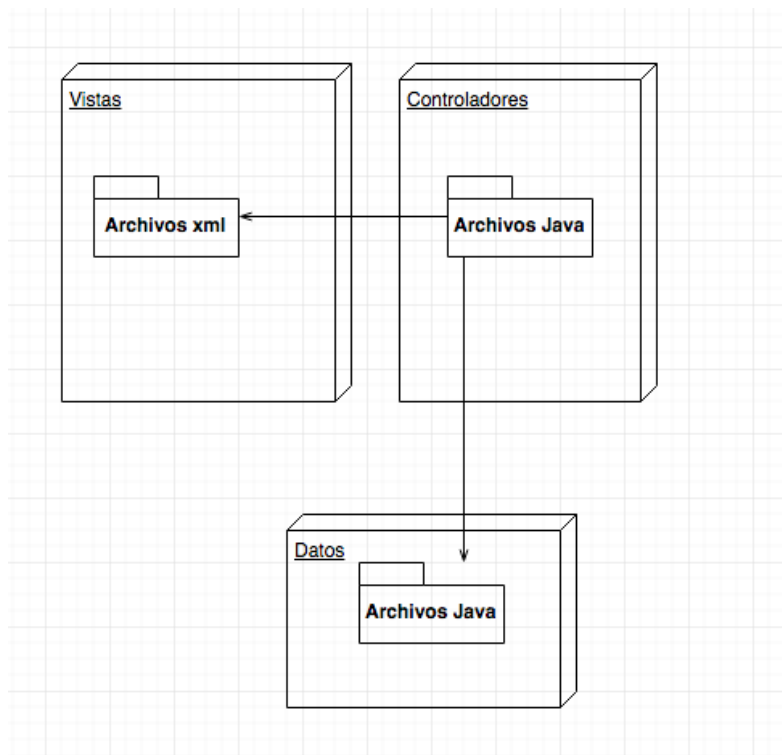


Ilustración 4.12: Separación Modelo-Vista-Controlador.

Ya desde la fase de diseño la separación entre las distintas entidades era clara y se ha diseñado la lógica por separado de la interfaz o los datos.

4.5.2. Lenguajes

Los lenguajes de programación usados para el desarrollo de la aplicación han sido los lenguajes específicos para este tipo de software, que son tanto java como xml.



Ilustración 4.23: Logo de java.

- Java es un lenguaje de programación orientado a objetos y su principal ventaja es que se puede ejecutar en cualquier dispositivo, su sintaxis es parecida a la de c++ pero se omiten las funciones a bajo nivel, es utilizado para la lógica del software [42].



Ilustración 4.24: Logo de XML.

- XML es un lenguaje de marcado desarrollado por el W3C usado para guardar datos de forma comprensible, pudiendo ser usado para casi

cualquier objetivo, en nuestro caso se usa para la maquetación de las vistas y para la definición de parametros [43].

4.5.3. Herramientas

Las herramientas empleadas para el desarrollo de la aplicación han sido principalmente Android Studio para el desarrollo, Github como repositorio, draw.io para la creación de los diagramas y fluidui para crear el storyboard

- Android Studio es un entorno de desarrollo usado para el desarrollo de aplicaciones android de forma oficial desde el 2014, ha sido difundido de forma gratuita.

Android Stuidio está disponible para los principales sistemas operativos y sus requisitos son de 3 GB de RAM como mínimo 8 GB de RAM recomendado, 500 MB de espacio en disco duro y 1.5 GB para el SDK.

- GitHub es un repositorio usado para alojar y compartir código o proyectos completos, usa el sistema de control de versiones de git para facilitar el desarrollo a los programadores.



Ilustración 4.25: Icono de GitHub.

En mi caso el objetivo de usar esta herramienta ha sido el de tener una copia de seguridad en la nube, ya que se ha desarrollado el prototipo en solitario y por lo tanto no hay problemas en compartir código.

- Draw.io es una herramienta web que nos permite crear una gran variedad de diagramas, resulta muy útil ya que estos diagramas se pueden guardar de forma local o se pueden exportar a otros servicios en la nube como GitHub o dropbox.

Esta herramienta me ha sido de mucha utilidad ya que he realizado todos los diagramas del proyecto con esta herramienta y si necesidad de tener que instalar software adicional en mi unidad de trabajo.

- Fluid ui es una herramienta que permite crear prototipos de diversas plataformas como IOS, Android, aplicaciones de escritorio etc.

Esta herramienta la he usado para crear un prototipo de mi aplicación de manera online para poder tener un diseño con el cual se pueda interactuar.

4.5.4. Pruebas

En este apartado se documentaran las pruebas realizadas durante la fase de implementación, las pruebas a realizar se han elegido por tener mayor relevancia en los objetivos de la aplicación.

- Test 1: Cambio de vista.

Este test se efectúa incorrectamente tratando de cambiar a otra vista mediante un botón.

Condiciones	El prototipo debe estar operativo.
Acción	El usuario pretende cambiar a otra vista.
Respuesta	El sistema no responde cuando el usuario presiona un botón

Tabla 4.1: Test 1. Cambio de vista.

- Test 2: Cambio de vista.

Este test se realiza correctamente tratando de cambiar a otra vista mediante un botón.

Condiciones	El prototipo debe estar operativo.
Acción	El usuario pretende cambiar a otra vista.
Respuesta	El sistema cambia de vista cuando el usuario presiona el botón correspondiente.

Tabla 4.2: Test 2. Cambio de vista.

- Test 3: Acceso a la agenda.

Este test se lleva a cabo incorrectamente tratando de acceder a la agenda personal del usuario.

Condiciones	El prototipo debe estar operativo, y aceptar los permisos necesarios para el acceso a la agenda.
Acción	El usuario pretende abrir la agenda para extraer un contacto.
Respuesta	La agenda no devuelve información alguna al seleccionar un contacto.

Tabla 4.3: Test 3. Acceso a la agenda.

- Test 4: Acceso a la agenda.

Este test se efectúa correctamente tratando de acceder a la agenda personal del usuario.

Condiciones	El prototipo debe estar operativo, y aceptar los permisos necesarios para el acceso a la agenda.
Acción	El usuario pretende abrir la agenda para extraer un contacto.
Respuesta	La agenda devuelve información alguna al seleccionar un contacto.

Tabla 4.4: Test 4. Acceso a la agenda.

- Test 5: Escritura en una etiqueta.

Este test se realiza incorrectamente tratando de escribir tanto una acción como un contacto en una etiqueta, este test esta relacionado con el requerimiento funcional RF3.

Condiciones	La tecnología NFC deberá estar operativa, tener una etiqueta NFC y el prototipo debe estar operativo.
Acción	El usuario desea guardar datos en una etiqueta NFC.
Respuesta	El prototipo tiene un fallo y se cierra.

Tabla 4.5: Test 5. Escritura en una etiqueta.

- Test 6: Escritura en una etiqueta.

Este test se efectúa correctamente tratando de escribir tanto una acción como un contacto en una etiqueta, este test esta relacionado con el requerimiento funcional RF3.

Condiciones	La tecnología NFC deberá estar operativa, tener una etiqueta NFC y el prototipo debe estar operativo.
Acción	El usuario desea guardar datos en una etiqueta NFC.
Respuesta	El prototipo escribe los datos en la etiqueta y nos confirma la escritura.

Tabla 4.6: Test 6. Escritura en una etiqueta.

- Test 7: Lectura de una etiqueta.

Este test se produce incorrectamente tratando de obtener información de una etiqueta NFC, este test esta relacionado con el requerimiento funcional RF1.

Condiciones	La tecnología NFC deberá estar operativa, etiqueta NFC con datos de nuestro prototipo.
Acción	El usuario pretende leer los datos de una etiqueta NFC
Respuesta	El prototipo no es capaz de extraer los datos.

Tabla 4.7: Test 7. Lectura de una etiqueta.

- Test 8: Lectura de una etiqueta.

Este test se lleva a cabo correctamente tratando de obtener información de una etiqueta NFC, este test esta relacionado con el requerimiento funcional RF1.

Condiciones	La tecnología NFC deberá estar operativa, etiqueta NFC con datos de nuestro prototipo.
Acción	El usuario pretende leer los datos de una etiqueta NFC.
Respuesta	El prototipo extrae los datos y los muestra solo en versiones de prueba.

Tabla 4.8: Test 8. Lectura de una etiqueta.

- Test 9: Realizar una llamada.

Este test se efectúa incorrectamente tratando realizar una llamada al contacto leído, este test esta relacionado con el requerimiento funcional RF2.

Condiciones	La tecnología NFC deberá estar operativa, etiqueta NFC con datos de nuestro prototipo, obtener los permisos necesarios.
Acción	El usuario pretende realizar una llamada usando la aplicación.
Respuesta	El prototipo no tiene la capacidad de llevar a cabo la operación.

Tabla 4.9: Test 9. Realizar una llamada.

- Test 10: Realizar una llamada.

Este test se lleva a cabo correctamente tratando realizar una llamada al contacto leído, este test esta relacionado con el requerimiento funcional RF2.

Condiciones	La tecnología NFC deberá estar operativa, etiqueta NFC con datos de nuestro prototipo y obtener los permisos necesarios.
Acción	El usuario pretende realizar una llamada usando la aplicación.
Respuesta	El prototipo tiene la capacidad de realizar una llamada al contacto dado.

Tabla 4.10: Test 10. Realizar una llamada.

- Test 11: Enviar un sms.

Este test se realiza incorrectamente tratando enviar un mensaje de texto al contacto leído, este test esta relacionado con el requerimiento funcional RF2.

Condiciones	La tecnología NFC deberá estar operativa, etiqueta NFC con datos de nuestro prototipo y obtener los permisos necesarios.
Acción	El usuario pretende enviar un mensaje de texto usando los datos de las etiquetas NFC.
Respuesta	El prototipo no tiene la capacidad de llevar a cabo la operación.

Tabla 4.11: Test 11. Enviar un sms.

- Test 12: Enviar un sms.

Este test se efectúa correctamente tratando enviar un mensaje de texto al contacto leído, este test esta relacionado con el requerimiento funcional RF2.

Condiciones	La tecnología NFC deberá estar operativa, etiqueta NFC con datos de nuestro prototipo y obtener los permisos necesarios.
Acción	El usuario pretende enviar un mensaje de texto usando los datos de las etiquetas NFC.
Respuesta	El prototipo realiza la operación con el texto y al contacto suministrado por las etiquetas NFC.

Tabla 4.12: Test 12. Enviar un sms.

4.5.5. Resultados

Tras realizar todos los test sobre el funcionamiento de nuestro prototipo, durante su implementación, pudimos obtener algunos errores que fueron solventados. A continuación se detallan cómo se solventaron los errores que aparecieron en algunos test:

Test	Problema	Solución	Resultado
Test 1	Había un error al llamar a la clase correspondiente en la función de cambiar de vista.	Escribir correctamente el nombre de dicha clase	Ok
Test 2			Ok
Test 3	Había errores al intentar extraer toda la información del contacto	Extraer solo y exclusivamente el número de contacto, que es lo que realmente usaremos	Ok
Test 4			Ok
Test 5	Creación incorrecta del mensaje usando "VcardContact"	Se usó un "VcardText" para crear el mensaje	Ok
Test 6			Ok
Test 7	Se usaba una función errónea para leer los datos.	Usar otra función detallando la codificación que se usa Utf-8 y Utf-16	Ok
Test 8			Ok
Test 9	Se realizaba una llamada a un contacto sin número.	Se controló que se realizara la acción después de obtener todos los datos necesarios.	Ok
Test 10			Ok
Test 11	Se realizaba a través de un "intent"	Se opto por usar "SmsManager"	Ok
Test 12			Ok

Tabla 4.13: Test 12. Resultados de los test.

4.5.6. Encuesta

Para conocer el impacto de la aplicación se ha decidido realizar una encuesta sobre el uso del prototipo de aplicación con personas mayores de mi entorno, el sistema utilizado para la evaluación es el "The System Usability Scale (SUS)".

Se ha elegido este método porque es un sistema de evaluación de software que funciona muy bien con muestras pequeñas, y podemos diferenciar un sistema usable de uno no usable, lo cual es perfecto debido al limitado acceso a personas para su testeo.

Este método consiste en la evaluación del uso de un software mediante diez preguntas, las cuales se responden mediante una escala Likert que tienen un rango entre uno y cinco, siendo este entre “Muy de acuerdo” y “Muy en desacuerdo” [44] [45].

- Las preguntas que se van a usar para la evaluación son las siguientes:
- Me gustaría usar este sistema.
- El sistema resultó ser complejo.
- El sistema fue fácil de usar.
- Creo que necesito ayuda técnica para el manejo del sistema.
- Creo que la funcionalidad estaba bien integrada.
- Creo que el sistema es inconsistente.
- Creo que el sistema es fácil de usar por la mayoría de personas.
- Creo que el sistema es incómodo de usar.
- Me sentí seguro usando el sistema.
- Tuve que aprender muchas cosas antes de poder usar el sistema.

Para poder observar la efectividad del manual de usuario, los encuestados han realizado el test antes (pretest) y después (postest) de tener acceso al manual de usuario para poder comparar los resultados.

Para el cálculo de los resultados no se han necesitado complejas fórmulas matemáticas, por cada pregunta impar debemos restarle 1 a la puntuación, y por cada pregunta par restaremos el resultado de la pregunta al valor 5, a continuación se sumaran todos los resultados y lo multiplicaremos por 2,5 para obtener un resultado sobre cien.

A la hora de conocer cómo de usable es el sistema tendremos en cuenta que los valores obtenidos de las encuestas están comprendidos entre los siguientes valores:

- Si el resultado es superior a 80 el sistema es muy cómodo para los usuarios.

- Si el resultado está por debajo de 68 significa que el sistema está bien, pero hay ciertos aspectos que podrían mejorar.
- Sin embargo, si el resultado es inferior a 51 significa que el sistema no es usable y es preciso mejorar ese punto [46].

El resultado de las encuestas realizadas se representa mediante un gráfico de barras 100% para que se observe la proporción de respuestas dadas según el grado de coincidencia con la pregunta.

La encuesta realizada antes de la lectura del manual de usuario tuvo una media de 63,3 puntos, lo cual significa que el sistema está bien pero se podría mejorar.

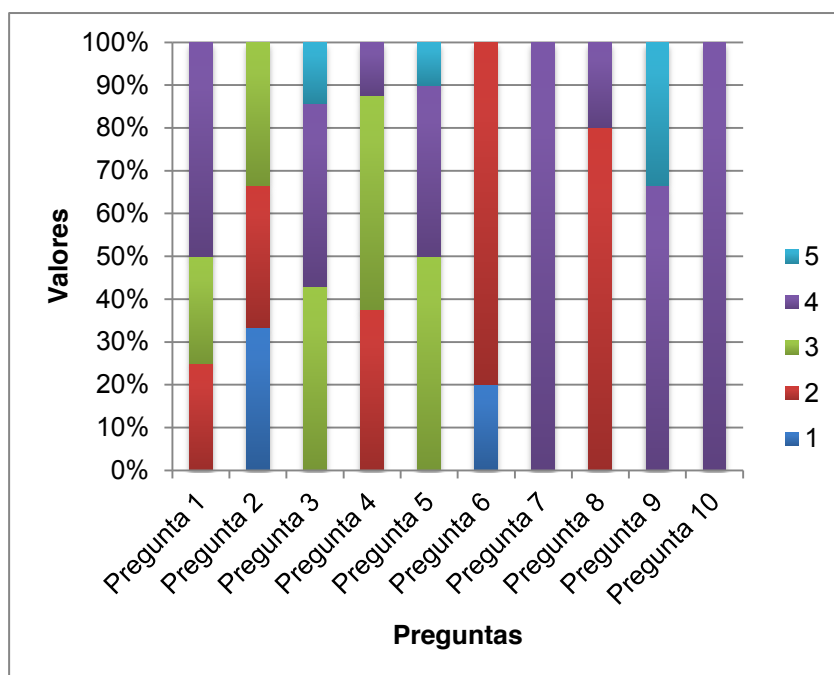


Ilustración 4.26: Gráfico pre-manual.

Para mejorar la experiencia de los usuarios se les pasó el manual de usuarios para que pudieran entender mejor el funcionamiento del sistema, para así poder hacer que su experiencia fuese más satisfactoria.

En este caso el resultado de la segunda tanda de encuestas subió la media a 78,1 por lo que podemos deducir que tras leer el manual de usuario la satisfacción de los usuarios al usar el sistema es mayor.

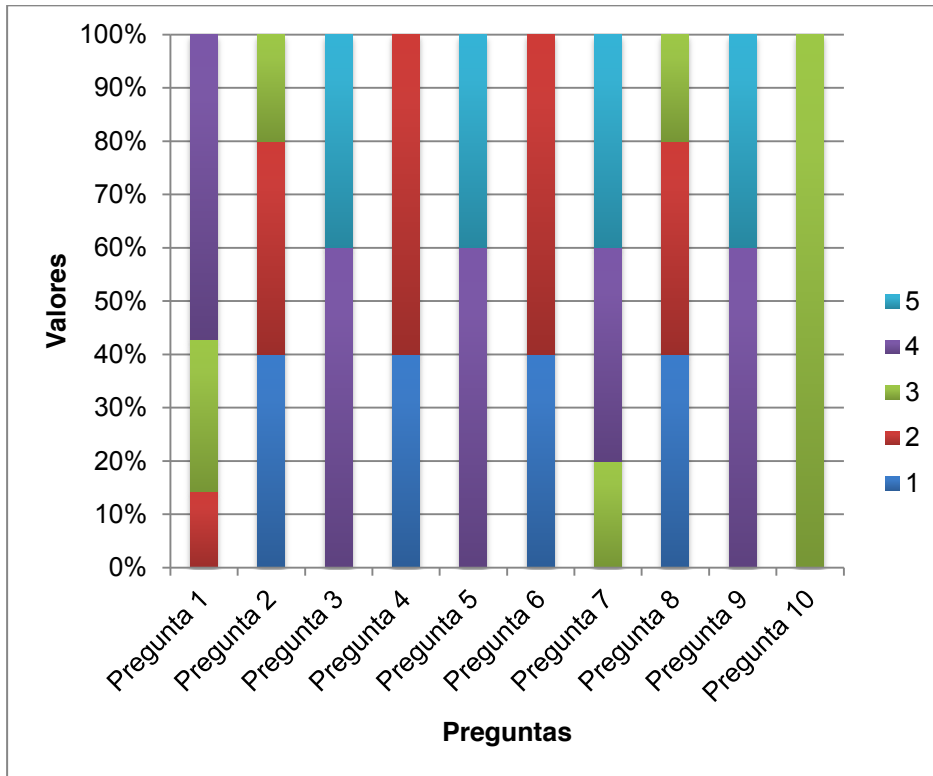


Ilustración 4.27: Grafico post-manual.

5. Conclusiones

Esta sección esta destinada a resumir el proceso de desarrollo del prototipo de la aplicación móvil y a expresar las sensaciones derivadas tras realizar tanto el software con el proceso de ingeniería del software como la documentación de dicho proyecto.

Dada la creciente población envejecida y la dificultad de proporcionarles la suficiente atención en muchos casos es necesario usar la tecnología para facilitar y mejorar la calidad de vida de dicha población y mejorar la comunicación con sus seres queridos, este es el principal motivo por el cual este proyecto me resultó interesante.

La razón por la cual realizo esta aplicación es para asistencia a personas mayores, la tecnología NFC puede serles muy útil ya que se pueden automatizar tareas que para ellos puede resultar complejas debido a limitaciones visuales que pueden tener. El proyecto es un prototipo de aplicación móvil que permite al usuario comunicarse con sus seres allegados de una forma más fácil y sencilla. El prototipo de aplicación móvil ha sido realizada para los sistemas android ya que son los más extendidos, y usando la tecnología NFC que es la que permite darle cierta facilidad de uso a nuestra aplicación.

El uso es sencillo, solo se ha de colocar el dispositivo sobre la etiqueta que representa la acción que se quiere llevar a cabo, a continuación se hará lo mismo pero con la etiqueta del contacto con el cual queramos comunicarnos.

En el desarrollo de este proyecto se han empleado los conocimientos de muchas de las asignaturas de la carrera, por lo que he visto aplicado mis conocimientos adquiridos durante estos últimos años. El proyecto me ha resultado muy atractivo dado que no es solo un software, es un prototipo de aplicación que puede ayudar a la gente en su día a día y eso es algo muy satisfactorio.

Pese a lo gratificante del desarrollo, durante la realización del proyecto han surgido algunos problemas como el enfrentarse a una tecnología totalmente nueva pero que a su vez he podido aprender sobre dicha tecnología. También comentar

otros problemas que tuvieron que quedar sin solución como es el de poder activar la tecnología NFC desde la aplicación cuando esta se encuentra desactivada pero no era posible por motivos de seguridad.

Uno de los aspectos a mejorar ha sido el poder añadir funcionalidad extra como poder elegir whatsapp u otro servicio de mensajería como otra opción, otra funcionalidad que podría estar en una segunda versión es la de poder lanzar el prototipo con tan solo hacer contacto con una de las tarjetas NFC que usa el prototipo de aplicación.

El prototipo de la aplicación ha sido evaluado por personas mayores de mi entorno con resultados favorables, las personas que realizaron la prueba realizaron un manejo de la aplicación acertado, además quedaron satisfechos por la simplicidad de la aplicación.

A carácter personal, estoy satisfecho con el trabajo realizado debido a que he podido poner en práctica muchas de las habilidades y conocimientos adquiridos durante la carrera, además de poder ampliar mis conocimientos sobre el desarrollo de software.

Bibliografía

- [1] Artículo sobre envejecimiento y tecnología. [Online]. <http://www.ageinglab.com/index.php/noticias/54-el-arte-de-envejecer-ayudado-por-los-ambientes-inteligentes>
- [2] Artículo sobre el envejecimiento. [Online]. [http://www.ageinglab.com/index.php?option=com_content&view=article&id=45&Itemid=203\(=es](http://www.ageinglab.com/index.php?option=com_content&view=article&id=45&Itemid=203(=es)
- [3] Artículo sobre sanidad y tecnología. [Online]. <http://www.larazon.es/atusalud/salud/tecnologias-conectadas-e-integradas-para-el-futuro-HA15517770>
- [4] Artículo sobre el uso de la tecnología por parte de los mayores. [Online]. <http://www.lavanguardia.com/internet/20110125/54106504302/el-67-de-los-mayores-usa-el-movil-y-el-14-navega-por-internet.html>
- [5] www.nobbot.com/. [Online]. <http://www.nobbot.com/tecnologia/aplicaciones-moviles-tecnologia/smartphone-y-personas-mayores-existen-maneras-de-facilitar-su-uso/>
- [6] Estudio sobre el mercado móvil. [Online]. <https://www.xataka.com/moviles/gartner-el-98-del-mercado-movil-es-de-ios-y-android-crecio-un-15-gracias-a-los-mercados-emergentes>
- [7] Estudio sobre el mercado móvil 2017. [Online]. <https://andro4all.com/2017/02/ventas-2016-android-vs-iphone>
- [8] Artículo sobre Android Beam. [Online]. <https://elandroidelibre.lespanol.com/2014/07/que-es-y-para-que-sirve-android-beam.html>
- [9] Artículo sobre novedades en Android Oreo. [Online]. <https://www.cnet.com/es/analisis/google-android-o-novedades/>
- [10] Wiki sobre versiones de Android. [Online]. [es.wikipedia.org/wiki/Anexo:Historial de versiones de Android](es.wikipedia.org/wiki/Anexo:Historial_de_versiones_de_Android)
- [11] Estudio de mercado sobre versiones android. [Online]. <https://www.xatakandroid.com/mercado/android-6-0-marshmallow-es-ahora-la-version-mas-usada>
- [12] Wiki sobre Android Studio. [Online]. [https://es.wikipedia.org/wiki/Android Studio](https://es.wikipedia.org/wiki/Android_Studio)
- [13] Blog sobre programación. [Online]. <https://academiaandroid.com/android-studio-v1-caracteristicas-comparativa-eclipse/>
- [14] Comparativa entre Android Studio y eclipse. [Online]. <https://didesweb.com/android/android-studio-vs-eclipse/>

- [15] Artículo sobre la arquitectura de Android. [Online].
<https://sites.google.com/site/swcuc3m/home/android/generalidades/2-2-arquitectura-de-android>
- [16] Guía sobre la arquitectura de Android. [Online].
<https://developer.android.com/guide/platform/index.html>
- [17] Blog de componentes android. [Online]. <http://www.sqliver.net/blog/componentes-de-una-aplicacion-android/>
- [18] Guía sobre las actividades en android. [Online].
<https://developer.android.com/guide/components/activities.html?hl=es-419>
- [19] Tutorial sobre los ciclos de vida de una actividad. [Online].
<http://www.androidcurso.com/index.php/tutoriales-android/37-unidad-6-multimedia-y-ciclo-de-vida/158-ciclo-de-vida-de-una-actividad>
- [20] Wiki sobre la tecnología NFC. [Online]. https://es.wikipedia.org/wiki/Near_field_communication
- [21] Artículo sobre tecnología NFC. [Online]. <https://www.xataka.com/moviles/nfc-que-es-y-para-que-sirve>
- [22] Trabajo sobre tecnología NFC. [Online]. <http://www.monografias.com/trabajos101/tecnologia-nfc/tecnologia-nfc.shtml>
- [23] Blog sobre RFID. [Online]. <http://blog.enzocard.eu/2012/09/27/rfid/>
- [24] Comparativa de etiquetas. [Online]. <https://www.shopnfc.it/es/content/11-como-elegir-la-etiqueta-nfc>
- [25] Comparativa de etiquetas. [Online]. <https://www.shopnfc.it/es/content/6-caracteristicas-tecnicas-de-etiquetas-nfc>
- [26] Artículo sobre NFC. [Online]. http://www.mibgyyo.com/articulos/2015/07/23/nfc-funcionamiento-utilidades/#/vanilla/discussion/embed/?vanilla_discussion_id=0
- [27] Artículo sobre funcionamiento de NFC. [Online].
<https://www.fqingenieria.com/es/conocimiento/tecnologia-nfc-modalidades-operativas-y-aspectos-tecnicos-47>
- [28] Artículo sobre el formato NDEF. [Online]. <http://www.dummies.com/consumer-electronics/nfc-data-exchange-format-ndef/>
- [29] Listado de usos con el NFC. [Online]. <https://elandroidelibre.lespanol.com/2015/01/todo-lo-que-puedes-hacer-con-el-nfc-de-tu-smartphone.html>
- [30] Wiki sobre RFID. [Online]. <https://es.wikipedia.org/wiki/RFID>
- [31] Sitio web sobre NFC. [Online]. <https://mundonfc.wordpress.com/2012/02/08/diferencia-entre->

[nfc-y-rfid/](#)

- [32] Blog sobre tecnologías. [Online]. <http://www.trustpointinnovation.com/blog/2015/03/26/qr-codes-vs-nfc-tags/>
- [33] Artículo de comparación entre Qr y NFC. [Online]. <http://nearfieldcommunication.org/qr-codes.html>
- [34] Comparativa entre NFC y bluetooth. [Online]. <http://tecnologia21.com/50804/nfc-vs-bluetooth>
- [35] Breve explicación sobre bluetooth de baja energia. [Online]. <https://www.easycontext.com/que-es/tecnologia-bluetooth-baja-energia-ble/>
- [36] Artículo de comparación entre beacon y nfc. [Online]. http://www.setfile.net/noticias/ble_nfc.php
- [37] Artículo sobre tecnologías similares. [Online]. <http://www.marketing-movil-sms.com/marketing-codigos-qr/nfc-qr-bluetooth-y-wifi-%C2%BFtecnologias-complementarias-o-sustitutivas/>
- [38] Entada de un blog sobre ventajas y desventajas de NFC. [Online]. <http://gigatecno.blogspot.com.es/2014/09/ventajas-y-desventajas-de-nfc.html>
- [39] Revista sobre las ventajas de la tecnología NFC. [Online]. <https://slidedoc.es/ventajas-de-la-technolog%C3%ADa-near-field-communication-nfc-como-sistema-de-pago-electronico-pdf>
- [40] Artículo sobre seguridad NFC. [Online]. <http://www.androidpit.es/NFC-es-seguro-Asi-protege-nuestros-datos-esta-nueva-tecnologia>
- [41] Guia sobre la Ingeniería de software. [Online]. https://www.ecured.cu/Ingenier%C3%ADa_de_software
- [42] Wiki sobre java. [Online]. [https://es.wikipedia.org/wiki/Java_\(lenguaje_de_programaci%C3%B3n\)](https://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n))
- [43] Wiki sobre xml. [Online]. https://es.wikipedia.org/wiki/Extensible_Markup_Language
- [44] Broke J, "Sus-a quick and dirty usability scale.," *Usability Evaluation in Industry*, vol. 189, no. 194, pp. 4-7, 1996.
- [45] Sauro J Lewis JR, "The factor structure of the system usability scale," *Human Centered Design*, pp. 95-103, 2009.
- [46] Sitio web sobre utilidades tecnológicas. [Online]. <https://usabilitygeek.com/how-to-use-the-system-usability-scale-sus-to-evaluate-the-usability-of-your-website/>

A. Anexos

a. Guía de instalación

Antes de explicar como es el proceso de instalación, detallaremos los ficheros que nos encontraremos:

- Videos: Podemos encontrar varios videos que nos muestra como se usa el prototipo de aplicación.
- NFCComunicator.apk: Con el cual podremos iniciar la instalación del prototipo en nuestro terminal.
- Código fuente: Son los ficheros que contienen las lineas de código necesarias para la ejecución del prototipo de aplicación.
- La memoria: Es el documento en el cual se explica como se ha desarrollado el prototipo de aplicación.

Para poder instalar el prototipo de la aplicación en un dispositivo se realiza a través de un fichero con extensión apk ya que aún no está de manera oficial en el market oficial de Android, debido a eso debemos activar la posibilidad de su instalación.

Con el objetivo de permitir la instalación del prototipo en formato apk deberemos abrir la opción de ajustes, posteriormente dirigimos a la sección de seguridad y desde ahí activaremos la opción orígenes desconocidos para que nos permita instalar el prototipo.

Solo con descargar el archivo apk debería empezar la instalación del prototipo, si no es así deberemos buscar el archivo mediante el administrador de archivos, una vez empiece la instalación aparecerá un mensaje emergente para realizar dicha instalación.

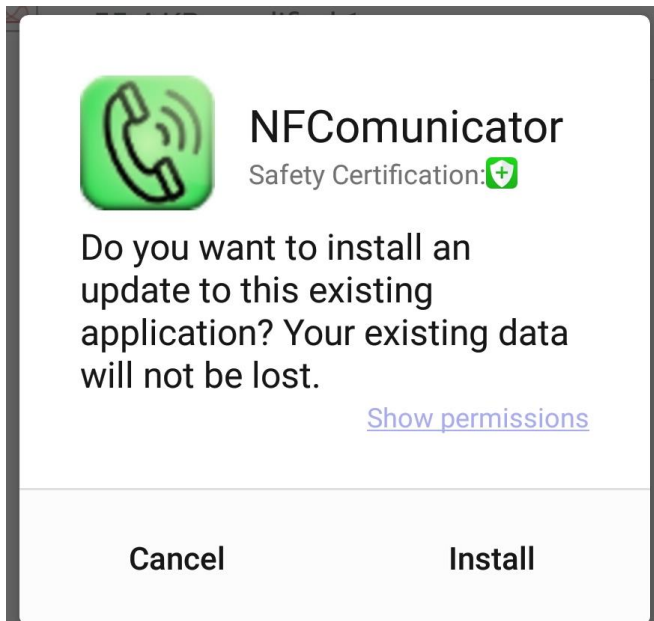


Ilustración 5.1: Mensaje de Instalación.

En el caso de estar interesados en los permisos que necesita la aplicación para funcionar podremos verlos a través de la opción mostrar permisos, aquí se nos mostrará una lista con los permisos que son: acceder a tus contactos, realizar llamadas y enviar mensajes de texto, algo que es normal teniendo en cuenta el funcionamiento del prototipo de la aplicación.

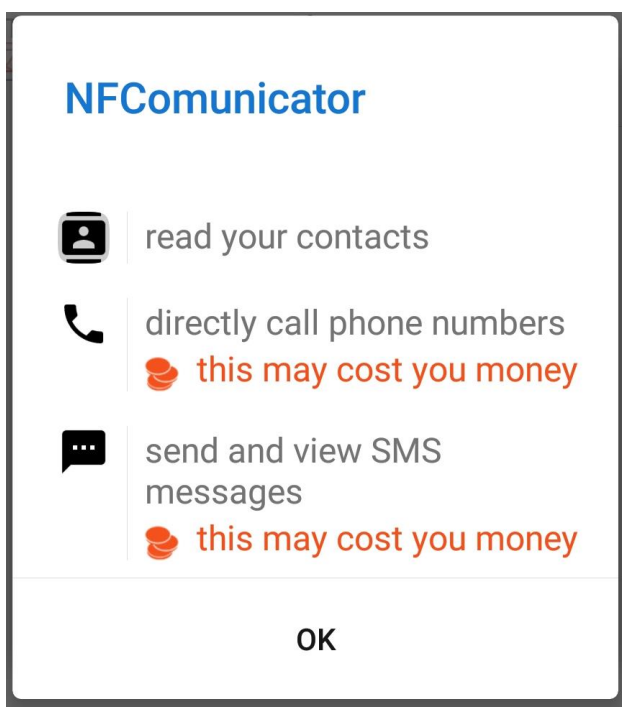


Ilustración 5.2: Permisos.

Una vez instalado el prototipo, nos aparecía otro cuadro de texto que nos permite abrir la aplicación o cambiar los permisos, en ese caso la aplicación no funcionará correctamente.

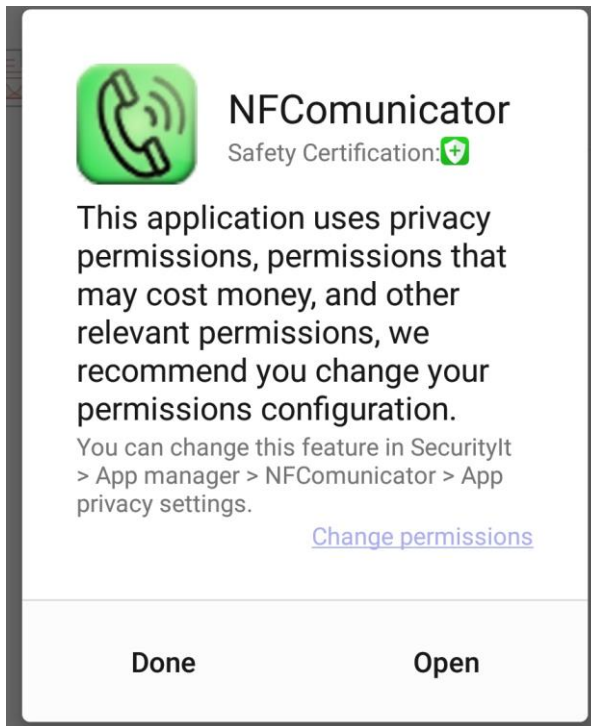


Ilustración 5.3: Advertencia sobre permisos.

Una vez terminado el proceso ya tendremos nuestro prototipo de aplicación listo para funcionar en nuestro smartphone.

b. Activación NFC

Para el funcionamiento de la aplicación necesitaremos activar la tecnología NFC, podremos hacerlo desde el menú de notificaciones como se muestra en la siguiente imagen

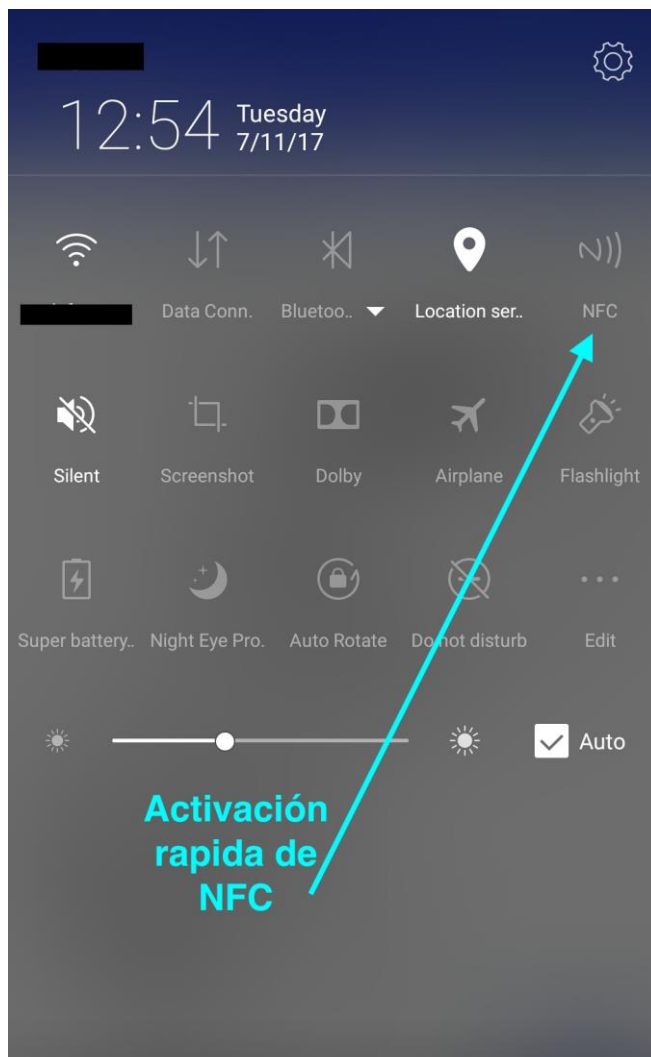


Ilustración 5.4: Activación rápida de la tecnología NFC.

Pero en el caso de no estar disponible en dicho apartado, podremos buscarlo en los ajustes generales de nuestro smartphone.

Para ello abriremos los ajustes y nos dirigiremos a la primera sección que es donde suelen estar las tecnologías de comunicación, desde hay podremos activarlo, en caso de no ser visible, deberemos acceder al submenú "Más" para acceder al resto de tecnologías de comunicación.

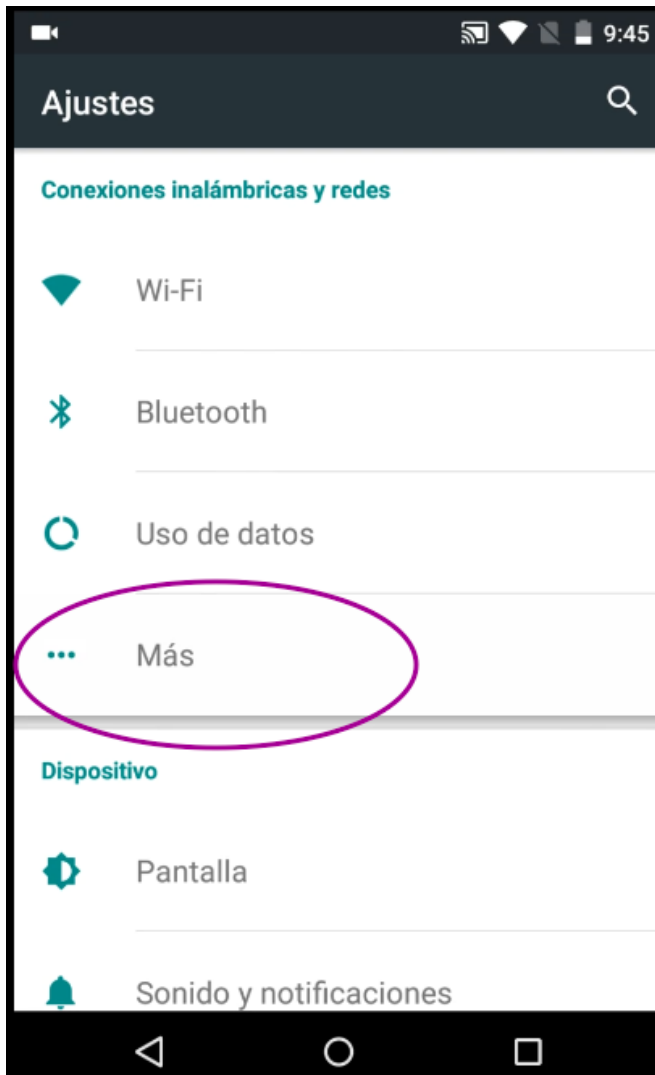


Ilustración 5.5: Acceso a las conexiones de nuestro terminal.

Una vez dentro de dicho submenú deberíamos ver la opción de activar y desactivar la tecnología NFC, en el caso de no encontrarla significará que nuestro dispositivo no tiene dicha tecnología y por lo tanto la aplicación no podrá funcionar.

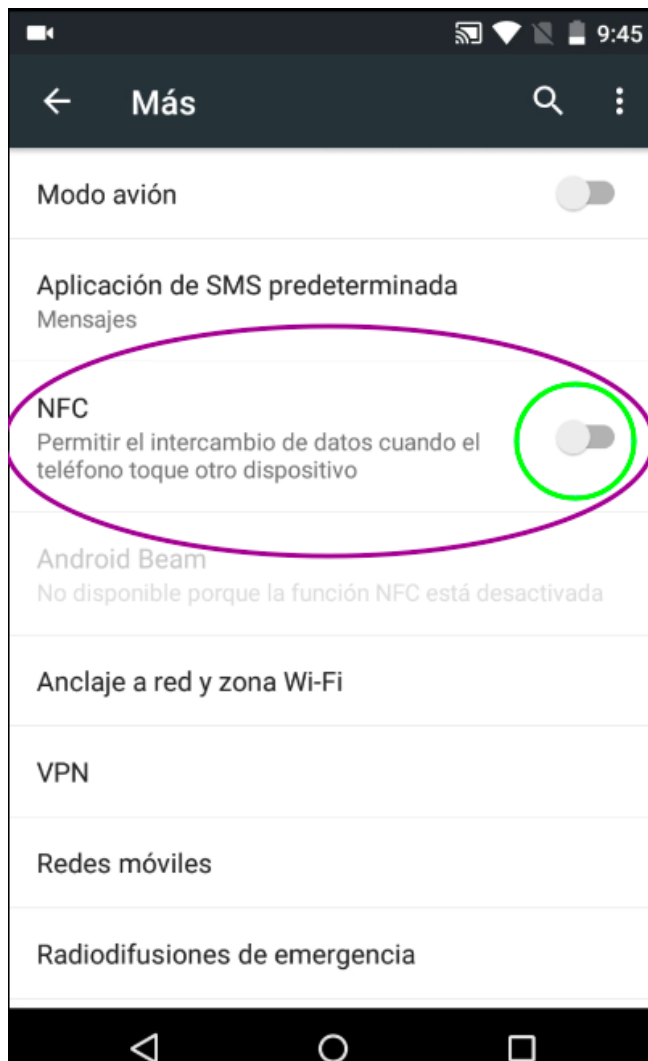


Ilustración 5.6: Acceso a las conexiones secundarias de nuestro terminal.

c. Manual de usuario

Para usar el prototipo deberemos familiarizarnos con su forma de trabajo, la cual no es complicada, esta se basa en acercar o tocar una etiqueta NFC con el dispositivo para poder llamar a una persona en concreto, o para enviar un mensaje.

En la parte superior de la vista principal, tenemos la barra de acciones en la que tenemos acceso al apartado de información y al menú de configuración, a este último también podremos acceder desde un botón extra en la parte inferior derecha.



Ilustración 5.7: Vista principal de la aplicación.

El apartado de información nos proporciona datos básica sobre la aplicación, la cual no es relevante. El menú de configuración se usará para la escritura de las etiquetas que veremos a continuación.

Como podemos observar nos aparece un error de NFC, esto aparece porque es necesario activar dicha tecnología, dicho proceso se ha detallado en el apartado anterior.

i. Escritura

En esta sección se explicará cómo se realiza la escritura en las etiquetas NFC a través de nuestro prototipo de aplicación, para ello tendremos que acceder al menú configuración.

Podemos acceder a dicha sección desde los botones con forma de rueda dentada situado en la barra superior o mediante un botón en la parte inferior derecha, una vez allí podemos elegir entre añadir una acción o añadir un contacto pulsando sobre el botón correspondiente, como se muestra en la imagen siguiente.



Ilustración 5.8: Menú de configuración.

A partir de aquí debemos escoger que etiqueta escribiremos primero, en este caso nos decantaremos por la escritura de una acción, que será si deseamos llamar a alguien o enviar un mensaje de texto, más adelante podremos ver como añadir contactos a los cuales podremos llamarles o enviarles un mensaje.

Una vez dentro de dicho submenú encontraremos dos grandes botones, un cuadro de texto y una lista desplegable en el centro.

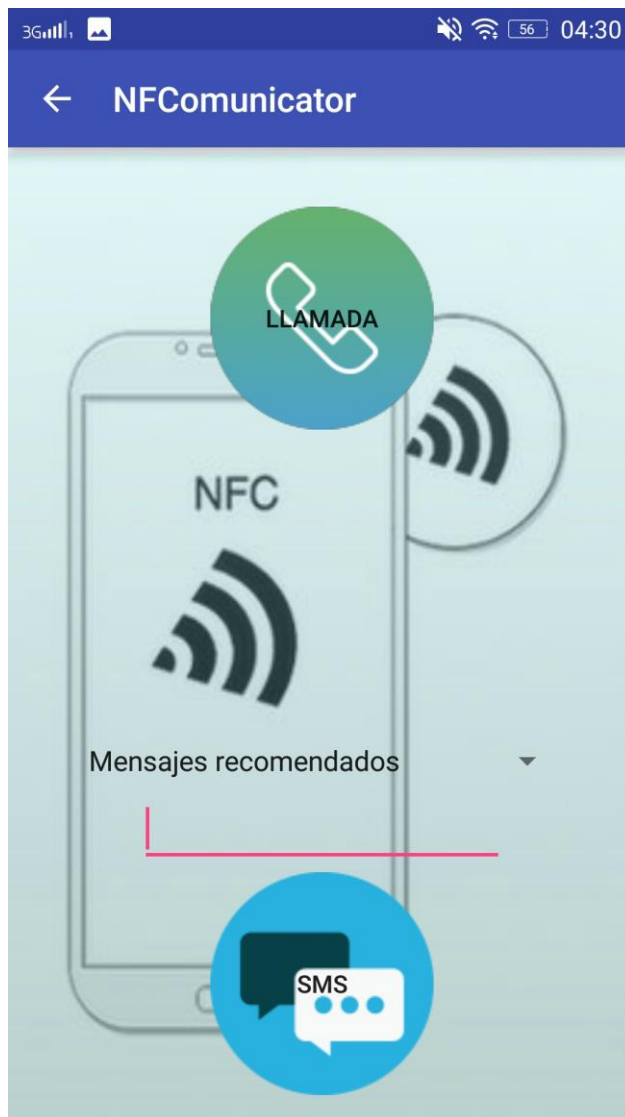


Ilustración 5.9: Menú de escritura de acción.

El botón superior nos permite preparar la escritura de una etiqueta con la opción de llamadas, en este caso solo será necesario dicha opción para poder escribir en la etiqueta.

El botón situado en la parte inferior prepara la escritura para la opción de mensaje, pero en este caso será necesario escribirlo en el cuadro de texto correspondiente o seleccionar uno de los mensajes predefinidos por el sistema.



Ilustración 5.10: Mensaje informativo sobre la opción seleccionada.

Una vez elegida una de las opciones el prototipo informará al usuario sobre qué opción se ha escogido, entonces es cuando se podrá escribir en la etiqueta de la misma manera que se lee, acercando el teléfono a la etiqueta, el prototipo de aplicación informará si ha ocurrido algún error o no.

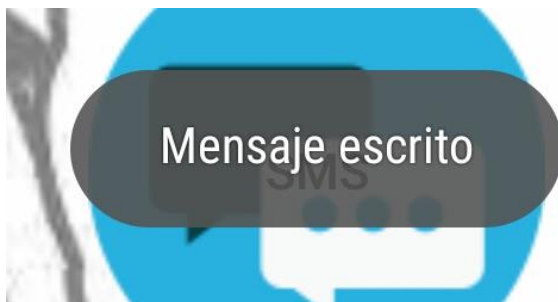


Ilustración 5.11: Mensaje de confirmación de escritura.

Como se ha expresado anteriormente, es el turno de la escritura de los contactos, aunque es similar a la escritura de acciones.

Podemos observar en este submenú tenemos un solo botón y un cuadro de texto que solo permite números, ya que será usado para los números de teléfono.



Ilustración 5.12: Menú de escritura de contactos.

El botón agenda nos permitirá acceder a nuestra agenda personal desde la cual podremos seleccionar el contacto que desemos guardar en la etiqueta para comunicarnos con el posteriormente, una vez seleccionado aparecerá su número de teléfono en el cuadro de texto.

También podremos introducirlo manualmente, pero es necesario marcar los espacios en el formato XXX XX XX XX.

Para escribir el contacto en la etiqueta NFC se realizará de la misma manera que se ha realizado para la escritura de la acción, una vez que nos muestre

información sobre el contacto elegido en el cuadro de texto, deberemos acercar el teléfono a la etiqueta para que se realice el grabado.

ii. Uso cotidiano

En este apartado se explicará el funcionamiento principal de la aplicación o cómo funciona la interacción entre el dispositivo y las etiquetas NFC en el uso cotidiano, en resumen el uso básico del prototipo.

El chip NFC suele estar colocado en la parte superior del teléfono, pero no es necesario que la comunicación se efectúe en el mismo punto.

Nuestra intención es comunicarnos con una persona en concreto, ya sea un familiar o un amigo, para ello usaremos el prototipo de aplicación teniendo preparadas las etiquetas correspondientes.

El primer paso es escoger la acción a realizar mediante la etiqueta NFC, podemos elegir entre una llamada de teléfono o enviarle un mensaje de texto concreto.

Solo con acercar el teléfono a la etiqueta debería establecerse la comunicación, pero suele ser cómodo colocar el smartphone sobre la etiqueta tal y como se muestra en la siguiente imagen.



Ilustración 5.13: Ejemplo de comunicación NFC desde la aplicación.

En el caso de encontrar un mensaje que sea erróneo o que no sea el correspondiente, el prototipo mostrará un mensaje diciendo que no se ha podido reconocer el mensaje, en ese caso debemos usar otra etiqueta o escribirla de nuevo.

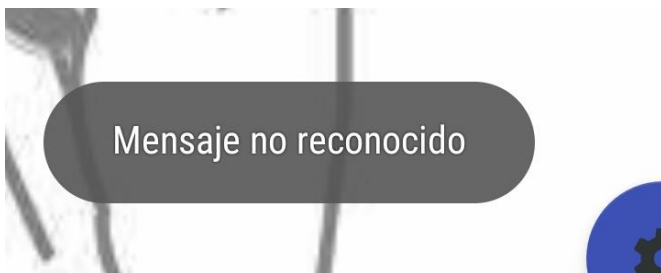


Ilustración 5.14: Mensaje de error en la lectura.

En caso contrario mostrará un mensaje sobre qué código se ha guardado, por lo tanto podremos pasar al siguiente paso, que será elegir el destinatario de la llamada o del mensaje de texto.



Ilustración 5.15: Mensaje de confirmación en la lectura.

Para elegir el contacto con el cual queremos comunicarnos debemos realizar una acción similar a la anteriormente realizada, debemos acercar el teléfono a la etiqueta NFC donde tenemos el contacto guardado.

Una vez reconocida el método con el que nos queremos comunicar y el contacto de la persona con la cual nos queremos comunicar, el prototipo de aplicación ejecutará de manera automática la comunicación, ya sea enviando el mensaje de texto o llamando a la persona deseada.