



**Facultad
De
Ciencias**

**Seguridad utilizando dispositivos
NFC**
Security using NFC devices

Trabajo de Fin de Grado
para acceder al

GRADO EN INGENIERÍA INFORMÁTICA

Autor : Fidel Abascal López
Director : Domingo Gómez Pérez
Junio - 2016

Índice general

1. Introducción	9
2. APP: Autenticación en un sistema ficticio	11
2.1. Introducción	11
2.2. Materiales y tecnologías utilizadas	12
2.3. Metodología	16
2.4. Escenario	17
2.5. Requisitos y funcionalidades	19
2.6. Diseño y aspecto	27
2.7. Seguridad y criptología	29
2.8. Base de datos	29
2.9. Aplicación desarrollada	30
2.9.1. Inicio	31
2.9.2. Menú	31
2.9.3. Acerca de	31
2.9.4. Despliegue	31
Sistema	31
Base de datos	31
2.9.5. Nuevo usuario	31
2.9.6. Ver usuarios	31
2.9.7. Verificación	31
2.10. Pruebas	31

Índice de figuras

2.1.	Infraestructura del sistema ficticio.	12
2.2.	Smartphone One Plus One - Never Settle	14
2.3.	NFC Tag - NTAG213	15
2.4.	Diagrama Gantt - Subdivisión de elementos de la aplicación para su desarrollo	16
2.5.	Diagrama Gantt : Visualización Parte I	17
2.6.	Diagrama Gantt : Visualización Parte II	17
2.7.	Casos de uso	19
2.8.	Elementos de diseño: Muestra de la disposición espacial del menú y estruc- tura básica de la aplicación	28
2.9.	Representación de la base de datos implementada.	30
2.10.	Script para la creación de la base de datos.	30

Resumen

Hace décadas comunicarse mediante un dispositivo que estuviera conectado a una red cableada y tras mucha espera resultaba algo fantástico. Hoy en día tenemos la posibilidad de realizar un gesto en cualquier lugar y ponernos en contacto con alguien a cientos o miles de kilómetros. En este sentido, las comunicaciones han evolucionado de una forma increíble. La seguridad en estas comunicaciones basada en la criptografía es un elemento primordial para salvaguardar la privacidad de los usuarios y la del contenido.

La criptografía no se ha quedado atrás y durante el último siglo su devenir ha seguido el mismo camino. Desde los métodos más primitivos basados en cambiar una letra por la anexa; hasta complejos sistemas criptológicos (criptosistemas) que aprovechan ciertas propiedades matemáticas para preservar niveles de seguridad elevados con la necesidad de menos recursos (computacionales y de almacenamiento). Optimizar los recursos es esencial para ser competitivo y para ello la metodología de criptografía basada en curvas elípticas reduce exponencialmente la cantidad de almacenamiento necesaria respecto a otros algoritmos. De la mano va la tecnología NFC (*Near Field Communication*), la cual ha simplificado los dispositivos de comunicación a algo tan pequeño y barato que ha conquistado el planeta en forma de multitud de aplicaciones.

En este Trabajo Fin de Grado (TFG a partir de ahora), comprenderemos la posibilidad de elaborar sistemas seguros con dispositivos de comunicación de bajo coste y criptología avanzada. A su vez, se implementará una aplicación para *smartphones* que, gracias a algoritmos avanzados de criptografía, permitirán a un usuario crear y utilizar un *tag* NFC como dispositivo de autenticación de alta seguridad en un sistema ficticio.

Palabras clave: *Criptografía, Curvas elípticas, NFC, autenticación.*

Abstract

Hace décadas comunicarse mediante un dispositivo que estuviera conectado a una red cableada y tras mucha espera resultaba algo fantástico. Hoy en día tenemos la posibilidad de realizar un gesto en cualquier lugar y ponernos en contacto con alguien a cientos o miles de kilómetros. En este sentido, las comunicaciones han evolucionado de una forma increíble. La seguridad en estas comunicaciones basada en la criptografía es un elemento primordial para salvaguardar la privacidad de los usuarios y la del contenido.

La criptografía no se ha quedado atrás y durante el último siglo su devenir ha seguido el mismo camino. Desde los métodos más primitivos basados en cambiar una letra por la anexa; hasta complejos sistemas criptológicos (criptosistemas) que aprovechan ciertas propiedades matemáticas para preservar niveles de seguridad elevados con la necesidad de menos recursos (computacionales y de almacenamiento). Optimizar los recursos es esencial para ser competitivo y para ello la metodología de criptografía basada en curvas elípticas reduce exponencialmente la cantidad de almacenamiento necesaria respecto a otros algoritmos. De la mano va la tecnología NFC (*Near Field Communication*), la cual ha simplificado los dispositivos de comunicación a algo tan pequeño y barato que ha conquistado el planeta en forma de multitud de aplicaciones.

En este Trabajo Fin de Grado (TFG a partir de ahora), comprenderemos la posibilidad de elaborar sistemas seguros con dispositivos de comunicación de bajo coste y criptología avanzada. A su vez, se implementará una aplicación para *smartphones* que, gracias a algoritmos avanzados de criptografía, permitirán a un usuario crear y utilizar un *tag* NFC como dispositivo de autenticación de alta seguridad en un sistema ficticio.

Keywords: *Cryptography, elliptic curves, NFC, authentication.*

1

Introducción

Cuando una persona envía un mensaje a un destinatario con información que considera comprometida o personal, pretende realizarlo con el mínimo riesgo de que dicho mensaje llegue de forma alterada y que sea al destinatario indicado. Además, éste quiere que el canal sea seguro y que nadie más intercepte la información; y si ocurriese tal caso, que personas ajenas no sean capaces de interpretar el mensaje y usarlo en su contra de forma perjudicial -o simplemente no desea difundir la información a alguien que no sea el destinatario-. También es evidente la necesidad de sistemas que eviten de la mejor forma posible la suplantación de usuarios; por ello, la autenticación apoyada en la criptografía resulta indispensable.

Existe la consideración generalizada que aquellos elementos más seguros a la hora de identificar y autenticar a un usuario de un sistema son aquellos que implican el uso de parámetros biométricos. Por ejemplo, el algoritmo para el reconocimiento del iris patentado por el investigador de la universidad de Cambridge, John Daugman, se basa en el iris como elemento único e intransferible para cada persona [18]. De forma análoga se utilizan algoritmos basados en la estructura facial o huellas dactilares. Por otra parte, Manuel Lucena López, doctor en informática de la universidad de Jaén, asegura en su publicación [14] que esta clase de requerimientos biométricos se pueden reducir a problemas de autenticación basada en dispositivos. Es decir, una tarjeta puede actuar con el mismo compromiso de seguridad que dichos elementos biológicos.

Al igual, en los sistemas criptográficos (criptosistemas), la implementación más segura suele ser la menos eficiente. En la actualidad, la competencia hace de la optimización un objetivo en el que se invierten ingentes cantidades de recursos. En el campo de la automoción competitiva, un incremento de la velocidad punta de 3km/h puede implicar reducir el tiempo de un competidor en unas décimas vitales que podrían suponer la diferencia entre ser primero o ser segundo. La criptología y la seguridad de comunicaciones también están afectadas por este hecho y no están exentas de la voracidad por optimizar el trinomio de recursos, tiempo y resultados. Dentro de este contexto una tecnología tan

asequible como NFC (*Near Field Communication*), - dispositivos para la comunicación de información empleando radiofrecuencia de corto alcance- que ha conquistado numerosos ámbitos, será el soporte de estudio de este Trabajo de Fin de Grado (TFG a partir de ahora) como dispositivo de autenticación.

Como suele ser habitual, un dispositivo barato tiene ciertas desventajas. Respecto a la seguridad criptográfica con NFC el mayor inconveniente suele ser el tamaño de almacenamiento - inferior a 500 bits generalmente-. Este problema implica utilizar criptosistemas que puedan trabajar con tamaños reducidos de información sin comprometer la seguridad. La evolución en la criptografía es considerable desde que en la Segunda Guerra Mundial el proyecto ULTRA tratara de descifrar los mensajes del ejército alemán; quienes se encontraban a la vanguardia de la criptografía.

Con el paso del tiempo, se avanzó hacia criptosistemas más complejos. En 1985 Victor Miller y Neal Koblitz propusieron la utilización de curvas elípticas para criptología. Su estructura algebraica es notoriamente más compleja que la mayoría de los criptosistemas anteriores a ésta. Sin embargo su implementación la posiciona entre las más eficientes y capaces de conseguir claves más cortas con el mismo nivel de seguridad [14]. Por ello, se ha implementado esta tecnología en multitud de escenarios: desde ciertas tarjetas de crédito hasta el cifrado de *Whatsapp* [21] trabajan con las propiedades de la criptografía con curvas elípticas.

Finalmente, se verá la potencia de estos sistemas para que, utilizando chips NFC de capacidad reducida, se puede generar un sistema de autenticación seguro aplicable a un escenario real.

2

APP: Autenticación en un sistema ficticio

2.1. Introducción

Anteriormente se han explicado los conocimientos y términos generales relacionados con la seguridad en la autenticidad NFC basada en criptología con curvas elípticas. A continuación se muestra la aplicación de dichos conocimientos en un proyecto software experimental. El objetivo es demostrar la capacidad de los dispositivos NFC para, de la mano de la criptografía con curvas elípticas, llegar a desarrollar un sistema de autenticación óptimo y fiable.

Tanto los nombres, librerías, herramientas, así como el resto del material utilizado se describirán a continuación. A su vez, siguiendo un estándar de desarrollo software basado en metodologías ágiles, se mostrará la utilizada para éste proyecto. Para ello, el autor y director de este TFG (Fidel Abascal y Domingo Gómez) hemos actuado y ejercido tanto de cliente como de contratado para el desarrollo de la aplicación.

Dentro de un ámbito ficticio, se plantea una empresa llamada **Alpha - Consultora S.A.**^{*}, nueva potencia local dentro del campo de la seguridad bancaria, que ha cosechado unos excelentes resultados a lo largo de sus 2 años de existencia. Cuenta con más de 40 trabajadores y su crecimiento y expansión es notoria. Tanto es el éxito de esta compañía que, para dar cabida a su plantilla, ha decidido trasladarse a una nueva sede más moderna, amplia y mejor ubicada. La empresa, antes de instalarse en la nueva sede, decide contratar a unos expertos en seguridad para gestionar el control de accesos mediante un sistema de tarjetas y lectores en las entradas; teniendo en cuenta una inversión mínima pero garantizando un alto grado de seguridad.

Tras buscar incesantemente recurren a la empresa **F-NFC**. Una vez realizado el estudio por parte de F-NFC, se pone en consonancia un acuerdo para elaborar una aplicación que

^{*} Cualquier similitud con la realidad es mera coincidencia

genere información que autentifique a un usuario de la empresa *Alpha* y sea reconocido de forma unívoca para permitir su acceso a la sede. La infraestructura de la empresa propietaria de la nueva sede corresponde a la figura 2.1.

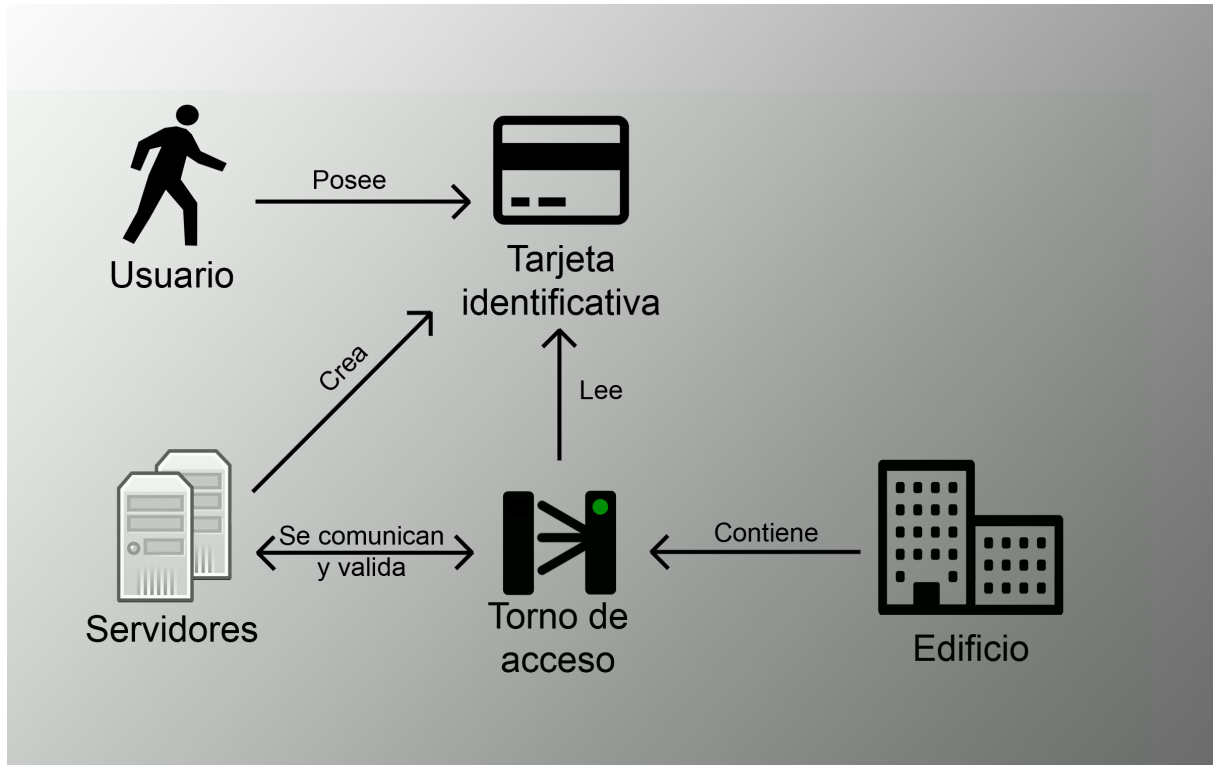


Figura 2.1: Infraestructura del sistema ficticio.

Los detalles de implementación de la aplicación seguirán un objetivo didáctico y experimental que cumplirán los requerimientos de la situación propuesta; adecuándose a la carencia real de la infraestructura anteriormente mencionada. Se explicará en las secciones siguientes en detalle todos los puntos implicados en la consecución de este objetivo. Más concretamente en el apartado de escenario de la aplicación 2.4.

2.2. Materiales y tecnologías utilizadas

Para el desarrollo del proyecto se ha utilizado, ante todo, software de carácter libre junto a imágenes, textos, estructuras y contenido sin restricciones de uso. Ya sea debido al tipo de licencia de cada elemento o por ser de autoría propia.

El desarrollo se ha realizado en el lenguaje de programación *Java* con el kit de desarrollo *Java* (JDK) 1.8.65 (*Java 8 update 65*) de *Oracle Corporation* [2].

El entorno de desarrollo integrado o *IDE* de la aplicación es el oficial de Google para el desarrollo para dispositivos *Android*: *Android Studio*, versión 1.5.1 [9]. Este programa provee las herramientas básicas de desarrollo; entre ellas, un administrador de los kit de desarrollo o *SDKs* para las diferentes versiones y varios elementos descritos a continuación:

- Plataformas SDK
 - API 22: Para la versión Android 5.1.1.
- Herramientas SDK
 - Android Build Tools : Para la construcción de la aplicación.
 - Android SDK Platform Tools v23.1: Soporte para el desarrollo.
 - Repositorio de ayuda Android, rev 30.
 - Librería de ayuda Android, rev 23.2.1 : Ayudas en la retrocompatibilidad de elementos de interfaz de usuario.
 - Google USB Driver, rev 11 : Conexión entre el servicio de ejecución de la aplicación y los dispositivos USB.
 - Intel x86 HAXM, rev 6.0.1 : Aceleración *hardware* para la emulación.

Estos componentes de *Android Studio* hacen posible el desarrollo de la aplicación objetivo. Este programa, con carencias notorias en ciertos apartados, ha hecho complicado, de cierta forma, la selección de componentes a instalar, versiones , etcétera debido a diversos motivos de compatibilidades de librerías y actualizaciones desfasadas entre los componentes y el propio IDE.

La esencia de la aplicación reside en la criptografía mediante curvas elípticas. Para la implementación de ello ha sido necesario utilizar una librería externa llamada *Bouncy Castle* [16] versión 1.54. Dicha librería suple las deficiencias de la implementación base de la propia API (*Application Programming Interface*) de *Java* en su paquete de *java.security* [1]. La cual tiene limitaciones claras a la hora de la generar curvas elípticas. Gracias a ésta librería se ha podido realizar la parte crítica de la aplicación con una mejora de rendimiento notoria si se tuviera de haber utilizado la API de *Java*.

Respecto al almacenaje de datos se ha optado por utilizar una pequeña base de datos en *SQLite* [19] descrita más adelante. Para el almacenamiento de unos pequeños registros que utiliza la aplicación. El uso de una base de datos de mayor capacidad y funcionalidades no se ha contemplado factible.

En cuanto a los elementos gráficos de la aplicación, un alto porcentaje son de elaboración propia mediante programas de edición de imágenes. El resto son de libre uso comercial. La iconografía de la aplicación es autoría de *Google Inc.* [11]. Dichos iconos han de ser vectoriales debido a la optimización del tratamiento de imágenes y su renderizado, por lo que en este aspecto, *Google* provee estos iconos en formato *SVG* (*Scalable Vector Graphics*) y *PNG*; también *Android Studio* contiene iconografía de forma nativa pero no actualizada. A la hora de incluir estos elementos externos se ha transformado las descripciones vectoriales *SVG* en formato *XML* (*eXtensible Markup Language*) interpretable de forma sencilla por *Android Studio* y fácilmente modificables en los casos que se ha requerido. En el apartado 2.6 de diseño y aspecto de la aplicación se comenta en detalle el resto de la disposición, motivación y elaboración gráfica de la aplicación.

La API de objetivo del proyecto *Android* ha sido la número 22. Desarrollada para la versión 5.1. (*LOLLIPOP MR1*). Se ha decidido utilizar esta API debido a las mejoras sustanciales en cuanto al trabajo del adaptador NFC implementadas desde la versión 5.0 [7] y mejoradas en ésta [8].

Para el testado de la aplicación *Android Studio* se dispone de la tecnología *AVD* para la virtualización de dispositivos *Android*. Sin embargo, el rendimiento es pobre en comparación con el testeo y *debug* en un dispositivo físico. Por lo que se ha utilizado un dispositivo *Android smartphone One Plus One* de la compañía americana *Never Settle*, el cuál dispone de la versión *Android* 5.1.1 y *Cyanogen OS* 12.1.1 en el momento de la elaboración de la aplicación.



Figura 2.2: Smartphone One Plus One - Never Settle

Por último, el proyecto ha necesitado de dos elementos físicos principales: el dispositivo *Android* mencionado anteriormente y de tarjetas NFC. Debido a la carencia de un presupuesto, se ha optado por utilizar etiquetas adhesibles (desde ahora NFC-T) de bajo coste. Se trata del chip *NTAG213* que siguen el estándar ISO 14443-3 [12]. Las características de estas etiquetas son las siguientes:

- Tipo de etiqueta : ISO 14443-3A.
- Descripción : NXP MIFARE Ultralight (Ultralight C) - NTAG213.
- Tecnología : NfcA, Ndef, MifareUltralight.
- Formato de datos: NFC Forum Type 2.
- Diámetro: 25 mm.
- Identificadores del chip.
 - Valor ATQA: 0x0044.

- Valor SAK: 0x00.
- Firma: NXP Public Key.
- Tamaños y capacidad.
 - Memoria: 45 páginas de 4bytes por página (180 bytes).
 - Tamaño: 137 bytes.
 - *UID* (Identificador del contenido): 7 bytes [4].
 - *Byte 7*: Valor *UTF-8*, codificación.
 - *Byte 6*: Valor 0, reservado para uso futuro.
 - *Byte 5-0*: Tamaño del código del lenguaje *IANA*.
 - Tamaño utilizable: 130 bytes (Tamaño menos *UID*).

Estas etiquetas se pueden adherir en una superficie que le haga de soporte. Por ejemplo, se podría plastificar junto a dos tapas que cubran el chip y tener la apariencia de una tarjeta común. En la figura 2.3 se muestra una de estas etiquetas.

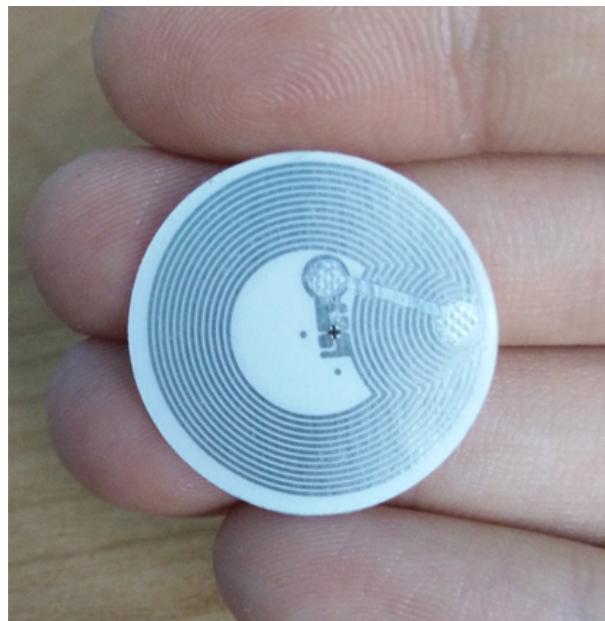


Figura 2.3: NFC Tag - NTAG213

Hubiera sido preferible utilizar chips *MIFARE* [15] de alta capacidad y mayores funcionalidades como los *MIFARE classic* 1K y 4K del productor *NXP*. Estos chips están implementados en una enorme cantidad de aplicaciones en todo el mundo, un ejemplo claro de su uso son en las tarjetas de transporte o hasta en las tarjetas de crédito y la mayoría de chips NFC actuales que impliquen la necesidad de encriptación (RSA comúnmente) y seguridad. En estas tarjetas es realmente complicado acceder a su contenido ya que se precisan ciertas claves para la lectura y decodificación. Sin embargo, en las utilizadas en este proyecto se escribe en texto plano (siguiendo el objetivo didáctico); en el apartado de seguridad y criptología 2.7 se explica cómo afectaría esta carencia al sistema final.

Todos los elementos descritos en este apartado conforman lo necesario para simular la estructura descrita en la figura 2.1 y la implementada definida en el apartado 2.4.

2.3. Metodología

Se ha decidido implementar una metodología de desarrollo software basada en iteraciones de funcionalidades de forma incremental. Para una planificación estructurada se han programado los hitos y la consecución de tareas a completar en cada hito de validación.

Gracias a la herramienta Gantt [20] para la elaboración de diagramas de planificación de proyectos, se han propuesto gráficamente la duración de cada tarea valorando su dificultad y las posibles modificaciones que hubieran podido surgir de las mismas en cada hito de validación. La duración de cada tarea implica los días necesarios para su finalización; sin tener relación con las horas laborales ya que, aunque no venga reflejado, se ha realizado trabajo en días no laborables. En las figuras 2.4, 2.5 y 2.6 se observa la organización inicial planteada. Cada apartado principal de la aplicación precedida de un hito se considera una iteración incremental del proyecto para adecuarse a la metodología planteada.



Nombre	Fecha de inicio	Fecha de fin
Base	23/02/16	29/02/16
Entorno	23/02/16	26/02/16
Estructura	29/02/16	29/02/16
Componentes iniciales	29/02/16	29/02/16
Diseño	1/03/16	2/03/16
Logo	1/03/16	1/03/16
Paleta de colores	2/03/16	2/03/16
Estilo	2/03/16	2/03/16
Validación	3/03/16	3/03/16
BBDD	3/03/16	4/03/16
Definición	3/03/16	3/03/16
Valores predeterminados	3/03/16	4/03/16
Validación	7/03/16	7/03/16
Portada	7/03/16	7/03/16
Menú lateral	8/03/16	9/03/16
Acerca de	10/03/16	10/03/16
Validación	11/03/16	11/03/16
Administración	11/03/16	7/04/16
Base de Datos	11/03/16	14/03/16
Sistema	15/03/16	28/03/16
Validación	29/03/16	29/03/16
Nuevo Usuario	29/03/16	4/04/16
Listado de usuarios	5/04/16	7/04/16
Validación	8/04/16	8/04/16
Verificación NFC	8/04/16	21/04/16
Validación	22/04/16	22/04/16
Pruebas	22/04/16	27/04/16
Despliegue	28/04/16	29/04/16

Figura 2.4: Diagrama Gantt - Subdivisión de elementos de la aplicación para su desarrollo

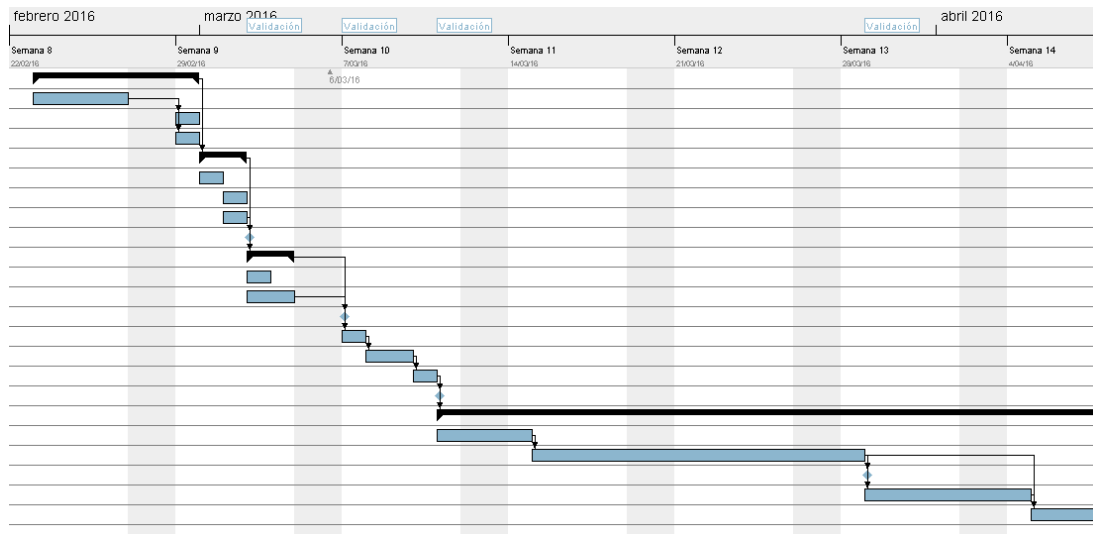


Figura 2.5: Diagrama Gantt : Visualización Parte I

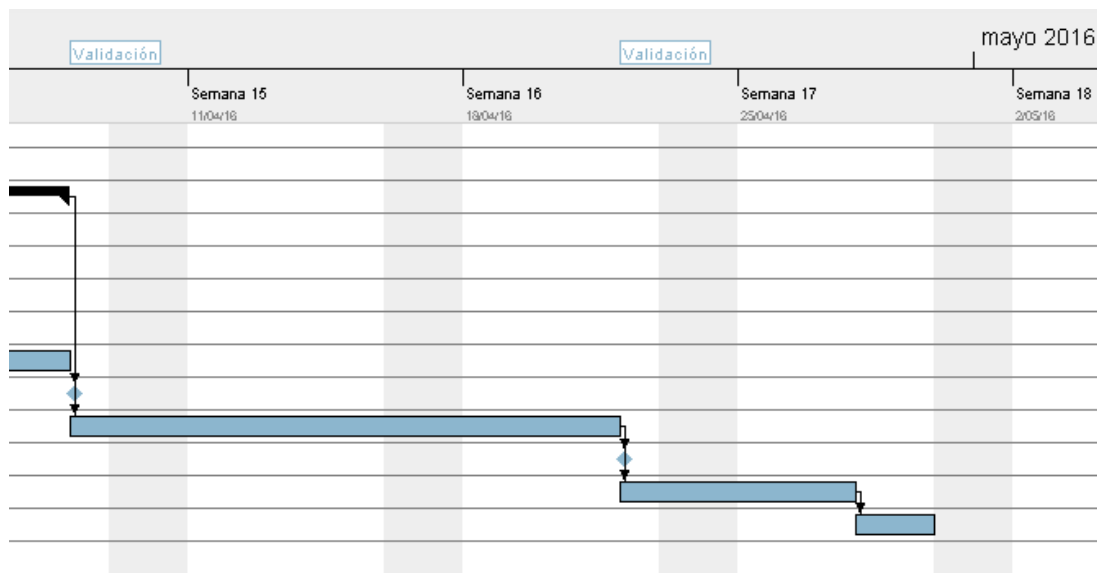


Figura 2.6: Diagrama Gantt : Visualización Parte II

Como se puede observar, la duración final del proyecto ha sido de 17 semanas aproximadamente. Con pequeñas variaciones, los hitos se han ido finalizando satisfactoriamente hasta comprobar que todas las tareas y partes del proyecto se han completado correctamente. A esta planificación habría que añadir el tiempo previo necesario para el estudio y preparación para el uso de las herramientas y la tecnología utilizada.

2.4. Escenario

La idea principal del escenario de la aplicación se ha comentado en el apartado 2.1, en donde se comenta que el objetivo de la aplicación es meramente didáctico y experimental. Alejándose de los modelos pragmáticos del desarrollo profesional.

La complejidad añadida de implementar un sistema cercano a la realidad aplicaría un coste elevado de recursos sin tener relevancia notoria en la finalidad comentada. Elementos tales como: servidores, *WebServices*, *frameworks* de desarrollo, certificación, optimización, etcétera. Dado que la esencia de este TFG se centra en el elemento teórico y didáctico de la criptología en curvas elípticas, la creación de una aplicación de carácter altamente profesional le añadiría beneficios nimios, los cuales no compensan la complejidad agregada. Por tanto, se han realizado labores de simulación y descarte de las partes que se han considerado prescindibles.

La arquitectura de la aplicación representada en la figura 2.1 muestra un escenario típico que da pie a la implementación de este tipo de seguridad. Dentro de este proyecto virtual se dispone de un edificio el cuál posee tornos o puertas de acceso. Estas puertas son capaces de leer el contenido de las tarjetas NFC de cada usuario del sistema. El contenido de las tarjetas se transmite a los servidores de validación de la empresa (sin la necesidad de que se encuentren físicamente dentro del mismo edificio). Los servidores validarán la información recogida en el torno de acceso y validará o no el contenido. Si resulta validado el torno recogerá de los servidores la nueva información a escribir en la tarjeta para hacer válida el paso la próxima vez. Finalmente, el usuario por medio de ésta tarjeta unipersonal podrá acceder al edificio autenticándose en la entrada.

Descrito el escenario virtual, es necesario comentar que éste proyecto se ha realizado. Contando con los elementos del apartado 2.2 se agrupan las funcionalidades del escenario virtual en el dispositivo *Android*. Por lo tanto, no hay conexiones a elementos externos y toda la estructura se compone únicamente del dispositivo *Android* y las tarjetas NFC. A su vez, cuenta con las siguientes funciones que simulan las labores del escenario virtual:

- Información sobre los usuarios del sistema: En lugar de contar con un acceso a una base de datos que recoja la información, se utiliza una pequeña base de datos dentro del dispositivo que contiene la información básica de los usuarios (identificador y nombre del usuario).
- Información del sistema de encriptación: El dispositivo también contiene la información básica del criptosistema implementado (labor de información de servidores).
- Validación: Tras leer el contenido de la tarjeta NFC, denegará o validará la información obtenida (labor de validación y acceso).
- Funciones de administración del sistema: El dispositivo es capaz de restaurar los valores por defecto del sistema, junto a la información inicial. También permite crear nuevas definiciones del sistema de seguridad. Por último, también asignará usuarios al sistema de seguridad que no se encuentren dentro (labor de creación de tarjetas).

Todos los elementos de información se describen en el apartado 2.8 sobre la base de datos de la aplicación; y las funcionalidades del sistema dentro del apartado 2.5 de requisitos y funcionalidades.

2.5. Requisitos y funcionalidades

La toma de requisitos, especificaciones y funcionalidades de la aplicación real, que cumple con los simulados en el contexto virtual descrito en la introducción 2.1, se ha llevado a cabo mediante constantes reuniones entre las partes implicadas (dentro del escenario real, véase el apartado 2.4).

Siguiendo la metodología ágil basada en iteraciones comentada en el apartado 2.3, desde el primer momento se ha ido mostrando el avance del proyecto al supuesto cliente. Éste ha aceptado según sus requerimientos iniciales o ha concretado cambios que se adecuen en el marco de las funcionalidades principales. Iteración a iteración se han ido realizando los pasos siguiendo hitos a validar. El diagrama *Gantt* mostrado en la figura 2.4 muestra en detalle la división de las tareas y apartados del proyecto realizado en este TFG.

Cada una de las tareas a realizar han seguido el objetivo de los casos de uso referidos a las funcionalidades de la aplicación. Los casos de uso de la aplicación se muestran en la figura 2.7, realizada mediante la herramienta *online* gratuita para la creación de diagramas UML de este tipo: yUML [17].

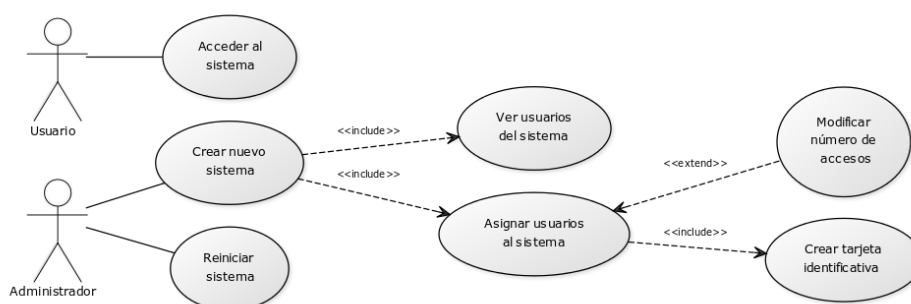


Figura 2.7: Casos de uso

Los casos de uso, cumpliendo las funcionalidades acordadas con el cliente, se corresponden con los requisitos funcionales de la aplicación y la infraestructura de la aplicación referida en la figura 2.1:

- Usuario

- RF01 - Un usuario, supuesto trabajador de la empresa, poseedor de una tarjeta de identificación proveída por la administración del sistema, es capaz de autenticarse.

- Administrador

- RF02 - La administración del sistema define las características del criptosistema que utiliza la aplicación.
 - RF03 - Puede restaurar la información a los valores por defecto.
 - RF04 - Puede ver los usuarios pertenecientes al sistema.

- RF05 - Es capaz de asignar usuarios al sistema creando las tarjetas identificadores.
- RF06 - Valida la información recogida de cada tarjeta y sobrescribe con la nueva información acorde al criptosistema empleado.

Como requisitos no funcionales se definen principalmente los siguientes:

- Plataforma: Se utiliza un dispositivo *Android* que cumpla las funciones de administración y validación.
- Criptosistema: Compuesto por criptografía con curvas elípticas de alta seguridad y el sistema *SKEY* para validaciones de un solo uso (*véase apartado sobre 2.7 la seguridad de la aplicación*). La plataforma utilizada puede modificar la curva elíptica implementada por el criptosistema.
- Elementos identificativos: Las tarjetas NFC contendrán la información asignada por el criptosistema. La información se escribirá a la hora e la asignación del usuario al sistema o en el momento de una validación correcta.

Las definiciones en profundidad de estos casos de uso se muestran a continuación.

Caso de uso	Acceder al sistema
<i>Actor principal:</i>	Usuario/Empleado
<i>Stakeholders e interesados:</i>	<ul style="list-style-type: none"> ■ Usuario/Empleado: Validar su identidad para acceder a las instalaciones de la empresa. ■ Empresa: Autenticar inequívocamente a quienes acceden a emplazamiento de la empresa.
<i>Precondiciones:</i>	<ul style="list-style-type: none"> ■ El usuario dispone de una tarjeta NFC creada por el sistema. ■ La aplicación se encuentra instalada en el dispositivo <i>Android</i>. ■ La aplicación posee un sistema de seguridad desplegado. ■ El dispositivo cuenta con la tecnología NFC y en estado activo.

Postcondiciones:

- La aplicación valida el contenido de la tarjeta NFC e informa al usuario del resultado.
- La aplicación reescribe la tarjeta NFC preparándola para su siguiente uso.

Escenario principal de éxito:

- 1 Se inicia el dispositivo.
- 2 Se aproxima la tarjeta NFC al terminal para la lectura del contenido.
- 3 La aplicación muestra un mensaje de validación confirmada y escribe el contenido oportuno para la próxima ocasión.

Extensiones:

- 1 Se inicia la aplicación:
 1. Se despliega el menú
 2. Se selecciona la opción Verificar NFC
- 2 Se aproxima una tarjeta NFC que no es del tipo correcto o el contenido es vacío:
 1. La aplicación muestra un mensaje de aviso
 2. Vuelta al paso 1 del escenario principal.
- 3 Datos de la tarjeta NFC incorrectos:
 1. El sistema muestra el de validación incorrecta.
 2. Vuelta al paso 1 del escenario principal.

Frecuencia de ocurrencia: Cada vez que un usuario del sistema desee autenticarse mediante su tarjeta NFC.

Caso de uso	Crear nuevo sistema
<i>Actor principal:</i>	Administrador
<i>Stakeholders e interesados:</i>	<ul style="list-style-type: none"> ■ Administrador: Definir un nuevo sistema de seguridad suplantando al actual.

Precondiciones:

- La aplicación se encuentra instalada en el dispositivo *Android*.
- La aplicación posee un sistema de seguridad desplegado.

Postcondiciones:

- La aplicación implementa una nueva definición del sistema utilizado. Cambiando las características de la curva elíptica a emplear en el futuro.
- No hay nuevos usuarios asignados en el nuevo sistema.

Escenario principal de éxito:

- 1 Se inicia el dispositivo.
- 2 Se inicia la aplicación.
- 3 Se abre el menú lateral.
- 4 Se selecciona la opción Despliegue.
- 5 Se selecciona la definición de una curva elíptica utilizando el desplegable.
- 6 Se pulsa en crear.
- 7 Se muestra un cuadro de confirmación.
- 8 Tras aceptar, se prepara el sistema para asignar nuevos usuarios con la nueva definición.

Extensiones:

4 Se pasa directamente al punto 6 del escenario principal:

1. La definición por elegida es la listada por defecto.

7 Pulsar en Cancelar:

1. Se cierra el cuadro de confirmación.
2. Se vuelve al punto 4 del escenario principal.

8 Error en la creación del sistema:

1. Se muestra un mensaje de error
2. Vuelta al paso 4 del escenario principal.

Frecuencia de ocurrencia: Cada vez que se desee reiniciar los datos de la aplicación con una nueva definición de seguridad.

Caso de uso	Reiniciar sistema
<i>Actor principal:</i>	Administrador
<i>Stakeholders e interesados:</i>	<ul style="list-style-type: none"> ■ Desarrollador: Revertir todos los cambios respecto a la definición de ejemplo.
<i>Precondiciones:</i>	<ul style="list-style-type: none"> ■ La aplicación se encuentra instalada en el dispositivo <i>Android</i>. ■ La aplicación posee un sistema desplegado.
<i>Postcondiciones:</i>	<ul style="list-style-type: none"> ■ La aplicación recupera la implementación del sistema de ejemplo.
<i>Escenario principal de éxito:</i>	

- 1 Se inicia el dispositivo.
- 2 Se inicia la aplicación.
- 3 Se abre el menú lateral.
- 4 Se selecciona la opción Despliegue.
- 5 Se selecciona la pestaña de Base de datos.
- 6 Se pulsa en el botón Restaurar Base de Datos.
- 7 Se muestra un cuadro de confirmación.
- 8 Tras aceptar, se recupera el sistema de ejemplo inicial.

Extensiones:

- 7 Pulsar en Cancelar:
 1. Se cierra el cuadro de confirmación.
 2. Se vuelve al punto 5 del escenario principal.
- 8 Error en la restauración del sistema:
 1. Se muestra un mensaje de error
 2. Vuelta al paso 5 del escenario principal.

Frecuencia de ocurrencia: En cualquier momento que se desee.

Caso de uso	Ver usuarios del sistema
<i>Actor principal:</i>	Administrador
<i>Stakeholders e interesados:</i>	<ul style="list-style-type: none"> ■ Administrador: Listar la información de los usuarios del sistema.
<i>Precondiciones:</i>	<ul style="list-style-type: none"> ■ La aplicación se encuentra instalada en el dispositivo <i>Android</i>. ■ La aplicación posee un sistema desplegado.
<i>Postcondiciones:</i>	<ul style="list-style-type: none"> ■ La aplicación muestra los usuarios activos del sistema .

Escenario principal de éxito:

- 1 Se inicia el dispositivo.
 - 2 Se inicia la aplicación.
 - 3 Abrir el menú lateral.
 - 4 Seleccionar la opción Ver Usuarios.
 - 5 Se lista la tabla con la información de seguridad de los usuarios activos en el sistema.
-

Extensiones:

- 4 No hay usuarios activos en el sistema:
 1. Se muestra un mensaje indicándolo
-

Frecuencia de ocurrencia: En cualquier momento que se desee.

Caso de uso	Asignar usuarios al sistema
<i>Actor principal:</i>	Administrador
<i>Stakeholders e interesados:</i>	<ul style="list-style-type: none"> ■ Administrador: Crear (escribir) tarjetas NFC para autenticar dentro del sistema generado a un empleado de la empresa. ■ Usuario/Empleado: Obtener una tarjeta NFC con el contenido adecuado.

Precondiciones:

- La aplicación se encuentra instalada en el dispositivo *Android*.
- La aplicación posee un sistema desplegado.
- El dispositivo cuenta con la tecnología NFC y en estado activo.
- La tarjeta NFC objetivo contiene información por defecto (no está vacía) para evitar fallos de escritura.
- La aplicación dispone de un listado de usuarios generales de la empresa.

Postcondiciones:

- Se vuelca el contenido de autenticación de un usuario recién añadido al sistema a una tarjeta NFC.

Escenario principal de éxito:

- 1 Se inicia el dispositivo.
- 2 Se inicia la aplicación.
- 3 Pulsar en abrir el menú lateral.
- 4 Seleccionar la opción Nuevo Usuario.
- 5 Seleccionar un usuario que no esté asignado al sistema actual.
- 6 Pulsar el botón Continuar.
- 7 Se muestra un mensaje de confirmación.
- 8 Un mensaje de de información indica que la aplicación está preparada para escribir en la tarjeta. Se acerca la tarjeta para escribir.
- 9 La aplicación ha escrito correctamente la información en la tarjeta NFC.

Extensiones:

7 Pulsar en Cancelar:

1. Se cierra el cuadro de confirmación.
2. Se vuelve al punto 5 del escenario principal.

8 Fallo al escribir:

1. Se muestra un mensaje indicando el fallo de escritura.
2. No se añade el usuario al nuevo sistema.
3. Se prepara la escritura nuevamente, paso al punto 8 del escenario principal.

Frecuencia de ocurrencia: Cuando se requiera asignar un usuario al sistema.

2.6. Diseño y aspecto

La aplicación cuenta con un contenedor principal que ocupa la visión principal del dispositivo. En la parte superior dispone de barra principal de la aplicación que contiene el título de cada apartado y un acceso al menú. El menú se muestra desde la parte izquierda ocupando gran parte de la pantalla. Muestra una cabecera de menú y el listado de apartados de la aplicación.

Principalmente estos elementos son los que definen el diseño principal de la aplicación. La disposición sigue las normas básicas para el diseño de una aplicación *Android* utilizando *material design* de Google [10]. Tanto la sombra de la barra principal, la sombra de fondo al mostrar el menú, el menú que se superpone a la barra principal y por debajo de los indicadores del dispositivo, la organización de los elementos del menú, el espaciado entre los elementos, la iconografía y demás elementos siguen en su mayoría las indicaciones de *material design*. En la figura 2.8 se puede apreciar gran parte de estos elementos de diseño.

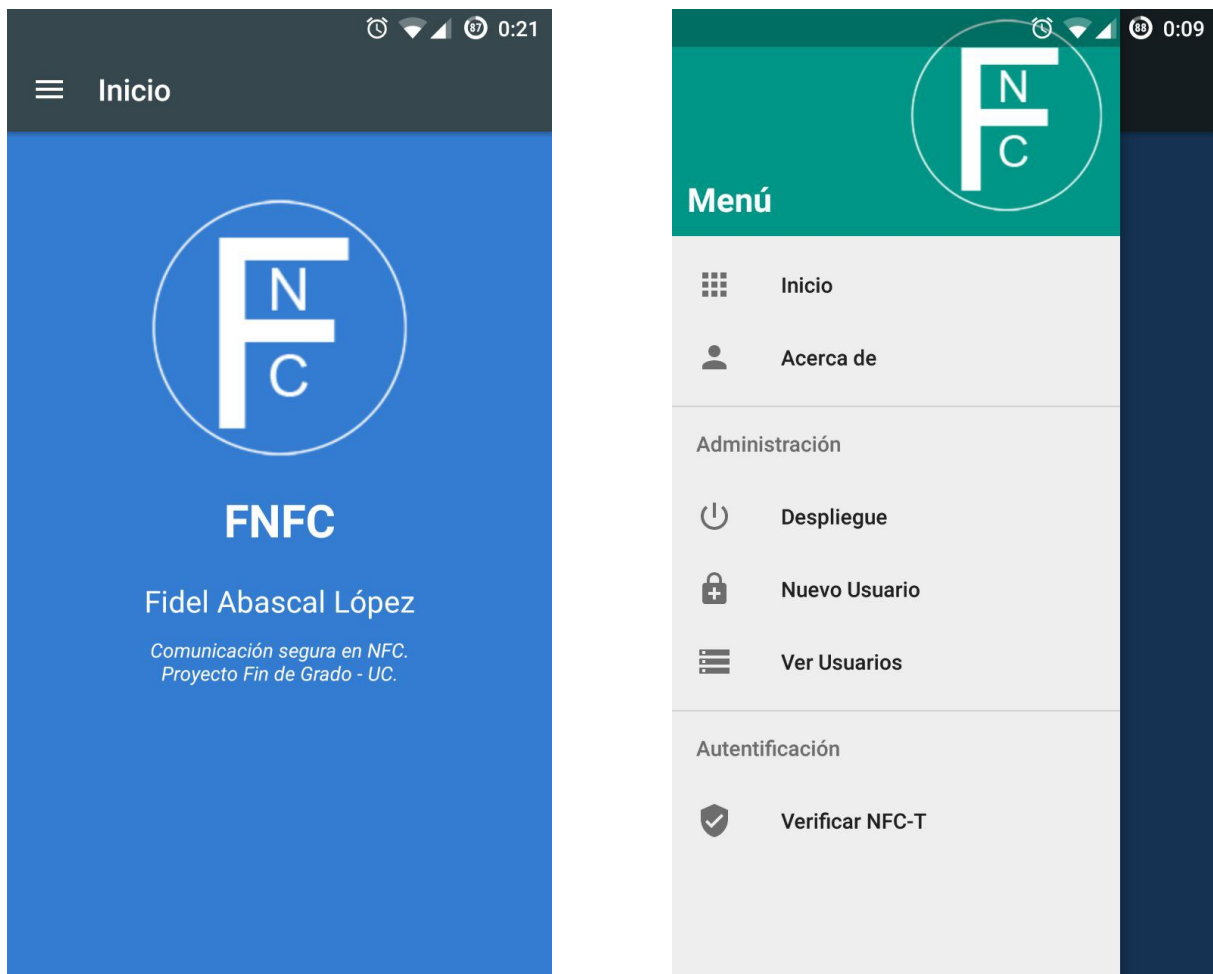


Figura 2.8: Elementos de diseño: Muestra de la disposición espacial del menú y estructura básica de la aplicación

Es oportuno mencionar la paleta de colores utilizada, tanto para el fondo como para las acentuaciones o el color primario y su homónimo oscuro :

- Color primario: #009688 .
- Color primario oscuro: #37474f .
- Color acentuado: #ff4081 .
- Color base principal: #367cd3 .

Se han elegido manteniendo la relación de colores de *material desing* y siguiendo normas conocidas de *marketing* y presentación de elementos visuales a consumidores. En estas normas y patrones es común observar que, por ejemplo, el color rojo implica pasión, efervescencia, impulso; ideal para representar ciertos elementos y estimular sensaciones acordes. Nótese que en la mayoría de marcas de comida, el color rojo predomina, ya que lo hace más apetecible al subconsciente humano y predispone al potencial consumidor a querer hacerse con ello.

Siguiendo el mismo ejemplo, el verde en marcas de alimentos se utiliza para representar lo natural, la limpieza y frescura; muy empleado en marcas ecológicas. El color base de la aplicación es el azul (hex:#367cd3); representa la serenidad, responsabilidad, sinceridad y verdad; apto para ser acogedor y mostrar fiabilidad. Es deliberado el uso de este color al igual que también lo es en la mayoría de las redes sociales. Buscan ganarse la confianza del usuario y dar un ambiente en el que se tenga una experiencia relajante y comfortable. Por todo esto, el azul acerca la seguridad del criptosistema utilizado.

2.7. Seguridad y criptología

hablar de qué ocurre al tener texto plano en las nfc-t en vez de las mifare classic
Contenido de las nfc de ejemplo, funcionamiento del SKEY y las curvas elípticas

2.8. Base de datos

La base de datos, mencionada en el apartado 2.2, es una pequeña *SQLite*. Se utiliza este tipo de base de datos debido a la comodidad y simplicidad con la que trabaja en un proyecto *Android*.

En el escenario virtual de la aplicación, en vez de esta base de datos se utilizaría una centralizada en los servidores de la empresa. Se realizarían conexiones seguras para no comprometer la información a enviar. Sin embargo, ésto no es necesario dentro del ámbito real ya que toda la información que se precisa de los servidores ficticios de la empresa se encuentran dentro de esta pequeña base de datos.

Dentro de esta base de datos se encuentra la información de la curva elíptica implementada y la clave privada asociada (*véase el apartado sobre seguridad y criptología 2.7*). También está la información de parte de los usuarios ficticios de la empresa. Por último, se encuentra una tabla referida a la información de cada usuario que compone los datos para poder validar el contenido de cada tarjeta NFC de cada usuario. En la figura 2.9 se observa la distribución de tablas y la relación entre ellas (creado mediante la herramienta online *GenMyModel* [6]). Posteriormente en la figura 2.10 se observa el *script* de creación en lenguaje *SQL*.

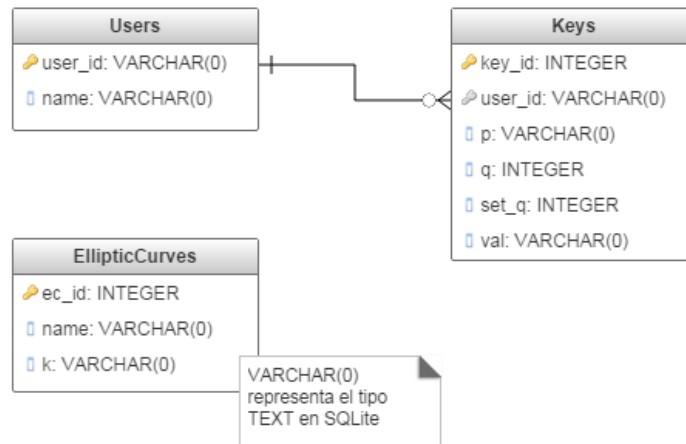


Figura 2.9: Representación de la base de datos implementada.

```

CREATE TABLE Users(
    user_id TEXT PRIMARY KEY,
    name TEXT
);
CREATE TABLE Keys(
    key_id INTEGER PRIMARY KEY AUTOINCREMENT,
    user_id TEXT,
    p TEXT,
    q INTEGER,
    set_q INTEGER,
    val TEXT,
    FOREIGN KEY(user_id) REFERENCES Users(user_id) ON DELETE CASCADE
);
CREATE TABLE EllipticCurves(
    ec_id INTEGER PRIMARY KEY AUTOINCREMENT,
    name TEXT,
    k TEXT
);

PRAGMA foreign_keys=ON;
CREATE INDEX user_id_index ON Keys(user_id);
  
```

Figura 2.10: Script para la creación de la base de datos.

El contenido por defecto del sistema se adecua a la implementación de la curva elíptica *c2pnb163v1* (cuyos detalles se pueden encontrar en [3]). La clave por defecto es un número entero de 160 bits que es el siguiente: 838828326113658401440043399564525405856963575389. Los distintos valores de los campos P , kP , Q y $Set\ Q$, QkP corresponden con datos válidos del sistema utilizado y de puntos aleatorios de la curva generados bajo la clave mencionada.

2.9. Aplicación desarrollada

La aplicación final se compone principalmente de una actividad principal que dispone del menú lateral, la barra superior de acciones y un contenedor que va modificándose en función de la situación. En las siguientes secciones se muestra el contenido, utilización,

significado y ejemplos de cada uno de estos apartados que compondrán los datos que se muestran en el contenedor.

2.9.1. Inicio

Pagina XX

2.9.2. Menú

Pagina XX

2.9.3. Acerca de

Pagina XX

2.9.4. Despliegue

Pagina XX

Sistema

Pagina XX

Base de datos

Pagina XX

2.9.5. Nuevo usuario

Pagina XX

2.9.6. Ver usuarios

Pagina XX

2.9.7. Verificación

Pagina XX

2.10. Pruebas

Para la comprobación y consecución de pruebas unitarias se ha elaborado un proyecto independiente para el testeo de la parte crítica de la aplicación. Ésta parte es la referida a la implementación de la seguridad y criptografía (*véase apartado 2.7*).

En cada uno de los métodos de la clase principal se ha testado que su ejecución en

distintos ámbitos y condiciones devuelve los resultados esperados cubriendo un alto porcentaje de código. Las pruebas se han centrado en el *framework* de pruebas unitarias *JUnit* [13] en su versión 4 elaborada para *Eclipse* [5].

La descripción básica de la clase y su documentación es la recogida por la interfaz que implementa...

<Crear interfaz y mostrar pruebas junit de esa clase>

Bibliografía

- [1] Oracle Corporation. Java security package.
- [2] Oracle Corporation. Software java.
- [3] Technische Universität Darmstadt. Curves over $gf(2^m)$ defined by ansi x9.62.
- [4] NFC Forum. Nfc forum specifications.
- [5] Eclipse Foundation. Eclipse.
- [6] GenMyModel. Genmymodel modeling platform.
- [7] Google Inc. Android apis - about version 5.0.
- [8] Google Inc. Android apis - about version 5.1.
- [9] Google Inc. Android studio - the official ide for android.
- [10] Google Inc. Material design guidelines.
- [11] Google Inc. Material design icons.
- [12] ISO/IEC. Iso/iec standard 14443-3.
- [13] JUnit. Junit.
- [14] Manuel José Lucena. Criptografía y seguridad en computadores. *versión 0.7*, 5, 1999.
- [15] MIFARE. Mifare ic - contactless smart cards.
- [16] Legion of the Bouncy Castle Inc. Bouncy castle - api criptográfica.
- [17] Pocketworks. yuml diagrams.
- [18] Mohamad Ramli, Nurul Akmar, Muhammad Saufi Kamarudin, and Ariffuddin Joret. Iris recognition for personal identification. 2008.
- [19] SQLite. Sqlite - public domain embedded sql database engine.
- [20] GanttProject Team. Ganttproject.
- [21] Whatsapp. Whatsapp encryption overview.

