ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

UNIVERSIDAD DE CANTABRIA

Trabajo Fin de Grado

Sistema de control de acceso mediante emulación de tarjeta con teléfono inteligente

(Access control system based on Smart card emulation on smartphone)

Para acceder al Titulo de

Graduado en
Ingeniería de Tecnologías de Telecomunicación

Autor: Daniel Muñoz Guerra

Enero- 2021

E.T.S. DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACION

GRADUADO EN INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN

CALIFICACIÓN DEL TRABAJO FIN DE GRADO

Realizado por: Daniel Muñoz Guerra
Director del TFG: Jorge Lanza Calderón
Título: "Sistema de control de acceso mediante emulación de tarjeta con teléfono inteligente"
Title: "Access control system based on Smart card emulation on smartphone"

Presentado a examen el día: 12 de Enero del 2021

para acceder al Título de

GRADUADO EN INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN

Composición del Tribunal:
Presidente (Apellidos, Nombre): Jesús Ibáñez Díaz
Secretario (Apellidos, Nombre): Jorge Lanza Calderón
Vocal (Apellidos, Nombre): Alberto Eloy García Gutiérrez

Este Tribunal ha resuelto otorgar la calificación de: .....

Fdo.: El Presidente

Fdo.: El Secretario

Fdo.: El Vocal

Fdo.: El Director del TFG
(sólo si es distinto del Secretario)

Vº Bº del Subdirector

Trabajo Fin de Grado Nº
(a asignar por Secretaría)

Índice:

Índice de figuras

Índice de tablas

Lista de acrónimos:

AID .- Application Identifier

APDU .- Application Protocol Data Unit

ATR .- Answer To Reset

BLE .- Bluetooth Low Energy

CPU .- Central Processing Unit

EEPROM .- Electrically Erasable Programmable Read Only Memory

GND.- Ground

GSM .- Global System for Mobile communications

HCE .- Host Card Emulation

I2C .- Inter Integrated Circuit

ICSP .- In Chip Serial Programmer

ISO .- International Organization for Standardization

MISO .- Master Input Slave Output

MOSI .- Master Output Slave Input

NFC .- Near Field Communication

NFCIP-1 .- Near Field Communication Interface and Protocol 1

OSI .- Open System Interconnection

P2P .- Peer to Peer

PC .- Personal Computer

PIN .- Personal Identification Number

PWM .- Pulse Width Modulation

QR .- Quick Response

RAE .- Real Academia Española

RAM .- Random Access Memory

RFID .- Radio Frequency Identification

ROM .- Read Only Memory

SCK .- Serial Clock

SCL .- Slave Chip Select

SE .- Secure Element

SIM .- Subscriber Identity Module

SPI .- Serial Peripheral Interface

SSH .- Secure Shell

SVN.- Salto Virtual Network

TPDU .- Transmission Protocol Data Unit

TUI.- Tarjeta Universitaria Inteligente

UART.- Universal Asynchronous Receiver Transmitter

UID.- User Identification

URL.- Uniform Resource Locator

USB.- Universal Serial Bus

Resumen

Hoy en día, hay recintos, como pueden ser parkings privados, laboratorios de investigación, etc., en los cuales se necesita algún método de control de acceso para evitar accesos no autorizados a dichos lugares.

Normalmente para poder acceder a estos tipos de recintos se suele emplear una llave, una tarjeta, o incluso se puede llegar a utilizar datos biométricos.

Basándose en esto y en la evolución de la tecnología móvil que permite emular tarjetas inteligentes, se pretende desarrollar una aplicación móvil que actúe como una tarjeta inteligente para controlar el acceso, mejorando la usabilidad, ya que el usuario solo tendrá que emplear un smartphone como elemento de autenticación, dispositivo que actualmente para la mayoría de la población es su mayor prioridad. De esta forma se sigue la tendencia iniciada por los sistemas de pagos bancarios.

La solución a desarrollar buscará operar en la medida de lo posible de forma sin conexión, siendo el propio terminal del usuario el habilitador de conectividad en caso de ser requerida.

Abstract

Nowadays, there are some places, such as private parking, investigation laboratories, etc., in which some access control is required to prevent unauthorized access to these places.

Usually, to be able to access to those enclosures is needed to use a key, a card or even some biometrical data.

Based on this and the development of the mobile technology able to emulate smart cards, it is pretend to develop a mobile application which act like a smart card for the access control, increasing usability, because the user only must use a smartphone like an authentication element, device that it is the highest priority for grand part of the population nowadays. In this way, it is followed the trend started by the bank payment systems.

The solution to be developed will seek to operate without connection, been the user terminal that allow the connectivity if it were necessary.

1. Introducción

La tendencia hacia la seguridad ha llevado a un desarrollo de métodos para controlar y garantizar dicha seguridad en los diferentes aspectos en los que se puede requerir.

La seguridad se puede establecer mediante un control de acceso que consiste en la verificación de si un usuario, vehículo, etc. se le permite el acceso a un recinto, sala, recurso, etc. denegándosele en caso de no disponer de acceso, otorgando así la seguridad de que entidades no deseadas no tengan a disposición el recurso.

Estos sistemas de control de acceso se pueden dar en muchos casos del día a día, y de diferentes formas. Se puede encontrar un control de acceso desde el portero automático de un edificio, hasta el inicio de sesión en un entorno web, siendo cada particular diferente, dependiendo de la seguridad que se le quiera dar al recurso a acceder y de la tecnología utilizada para tal. Sea como fuera el control, este ha de garantizar que en el proceso haya autenticación y autorización de la entidad que solicita el acceso. Así, la autenticación garantiza la identidad mediante un proceso basado en emplear algo que se sabe, que se tiene o que se es. El proceso mediante algo que se sabe consiste en valores, claves, contraseñas, etc. que el usuario tiene en conocimiento. Un ejemplo de este tipo de proceso es el control de acceso mediante un PIN, donde el usuario conoce un código PIN para acceder a un lugar y ha de introducir en el terminal de acceso. En cambio, el proceso mediante algo que tengo se trata de aquello que el usuario puede presentar para garantizar que es el y que dispone de dicho acceso. Un control mediante algo que se tiene es el control mediante una tarjeta, ya sea magnética o inteligente, basándose en que el usuario dispone de una tarjeta la cual es la que le permite el acceso. Por último, el proceso mediante algo que se es se basa en usar datos del usuario, como pueden ser la huella dactilar, iris, etc., siendo estos únicos e intransferibles, haciendo este tipo de autenticación de los más seguros. Un ejemplo de este tipo de control es el control de acceso mediante huella dactilar, que se basa, como su nombre indica, en la autenticación mediante la huella del usuario para poder acceder a un recinto.

Por su parte, la autorización define los permisos asignados a entidad autenticada para acceder a un determinado recurso.

Con la idea de la seguridad, y el desarrollo de las tecnologías actualmente muchas entidades bancarias han decidido emular en sus aplicaciones móviles tarjetas inteligentes vinculadas a sus cuentas bancarias, permitiendo a los usuarios pagar con sus teléfonos. Esto tiene como objetivo conseguir una mayor seguridad ante robo o la perdida de la tarjeta, siendo necesario simplemente llevar consigo el smartphone, tener la aplicación descargada y activar la emulación de la tarjeta.

Además, actualmente las personas son dependientes del smartphone, convirtiéndose éste en un elemento esencial en sus vidas, usándolo para cualquier cosa de la vida cotidiana, como puede ser hacer la compra, consultar cuentas bancarias, conocer gente, sacar fotografías, ver series y películas, leer, chatear, jugar, o incluso ligar. Y es por eso,

que se busca desarrollar esta tecnología para hacer al usuario su día a día más fácil y cómodo.

Siguiendo las premisas anteriores, el escenario que se plantea en este proyecto es desarrollar un sistema de control de acceso, en el cual el usuario se pueda autentificar mediante una tarjeta inteligente emulada a través de un terminal móvil. El proyecto planteado sigue la idea de la seguridad mediante el control de acceso que emplea la autenticación basada en dos factores, ya que dispondría de la tarjeta (algo que se tiene) y de una contraseña para desbloquear la misma (algo que se sabe). Con este trabajo también se sigue la línea de hacer el día a día más fácil al usuario siguiendo la tendencia de tener todo en el teléfono móvil, creando un control de acceso más simple y rápido.

1.1. Objetivos

El objetivo principal de este trabajo es conseguir que el usuario propietario de una tarjeta de acceso a un recinto pueda acceder a este sin necesidad de llevar consigo la tarjeta físicamente, sino que puede emplear su teléfono móvil como dispositivo que, a través de una aplicación previamente instalada, emule dicha tarjeta garantizando el acceso a la instalación.

Dentro de este trabajo se pueden diferenciar tres objetivos secundarios:

- El diseño e implementación de un dispositivo físico que instalado en un recinto detecte la tarjeta, se comunique con ella, y sea capaz de determinar si el usuario tiene o no acceso.
- La creación de un sistema de gestión de información en el que estén registrados los usuarios de la aplicación, y en la que un administrador sea el encargado de señalar cuál de ellos tiene o no acceso al recinto.
- La creación de una aplicación móvil que permita el registro y consulta de la información de un usuario específico, y que emule el comportamiento de la tarjeta como acreditación de acceso.

Para realizar estos objetivos y poder lograrlos correctamente, se deben afrontar otros más teóricos como, por ejemplo:

- Comprensión del funcionamiento de las tarjetas inteligentes y la comunicación de estas con los terminales.
- Programación de aplicaciones Android, así como las herramientas específicas para el objetivo principal de esta.
- Desarrollo de soluciones sobre equipos embebidos.

1.2. Estructura del Trabajo

Este trabajo consta de siete capítulos, tal y como se enumera a continuación:

- Capítulo 1, Introducción. Se trata del capítulo actual en el cual se describe el enfoque del trabajo, así como la idea del trabajo y los objetivos del mismo.

- Capítulo 2, La Tarjeta inteligente. Se definen las bases teóricas en las que se apoya el proyecto, se profundiza en los conocimientos necesarios para llevar a cabo el trabajo y las tecnologías actuales que han ayudado a implementar cada desarrollo.

- Capítulo 3, Sistemas de control de acceso. Se detallan los diferentes tipos de controles de acceso que se pueden encontrar hoy día, además, se nombran algunas de las empresas del sector que se pueden encontrar en el mercado.

- Capítulo 4, Equipo embebido de control de acceso. En este capítulo se detallan las características del sistema embebido que habilita el control de acceso, tanto desde el punto de vista de programación como de conexiones con hardware adicional.

- Capítulo 5, Aplicación móvil. En el quinto capítulo se describe el desarrollo de la aplicación móvil que permite emular el comportamiento de una tarjeta inteligente.

- Capítulo 6, Integración. En este capítulo se presentan los desarrollos para llevar a cabo la integración de los desarrollos realizados en los capítulos anteriores, desde la configuración de la comunicación entre ellos, hasta la validación del correcto funcionamiento global.

- Capítulo 7, Conclusión y líneas futuras. En este último capítulo se analizan los objetivos iniciales del trabajo y se exponen los resultados obtenidos una vez implementado todo el proyecto, así como posibles líneas futuras de trabajo.

2. La tarjeta inteligente

En este capítulo se describen las bases en las que se apoya el proyecto. Se explica qué son las tarjetas inteligentes, y sus tipos, profundizando en su comportamiento, etc. También se hablará de las metodologías de emulación de tarjetas inteligentes empleando la tecnología actual para entrar en contexto a la hora de programar la aplicación.

Las tarjetas inteligentes ofrecen funciones de almacenaje y procesamiento de datos de manera segura verificando la identidad y los derechos de acceso.

Se componen de un chip con microprocesador capaz de procesar la información. El microprocesador proporciona la inteligencia, que las diferencia de las tarjetas de banda magnética. Además, incorporan sistemas lógicos de alta seguridad, que impiden la manipulación no autorizada [1].

En los años 80 se explotaron al máximo las tarjetas inteligentes, también conocidas por su denominación anglosajona smart cards, siguiendo el modelo de Ronald Moreno con la identificación personal (PIN). Su uso hoy día se extiende desde la idea inicial de dispositivo de almacenamiento en memoria a, tarjetas bancarias, prepago de telefonía, etc. Los avances tecnológicos en los circuitos integrados y la criptografía han hecho posible tarjetas inteligentes más potentes, seguras, y baratas, convirtiéndose en el elemento idóneo para almacenamiento y procesado de información confidencial. Esto habilita poder usar las smart cards como llave de acceso, identificador en telefonía móvil, o para otros múltiples servicios [3].

2.1. Tipos de tarjetas con chips

Las tarjetas inteligentes pueden dividirse en dos grupos, como se ve en la Figura 1[2], según su tipo de chip, y según el método de transmisión. Dentro del primer grupo podemos encontrar otros dos grupos, los cuales se diferencian tanto en funcionalidad como en precio:

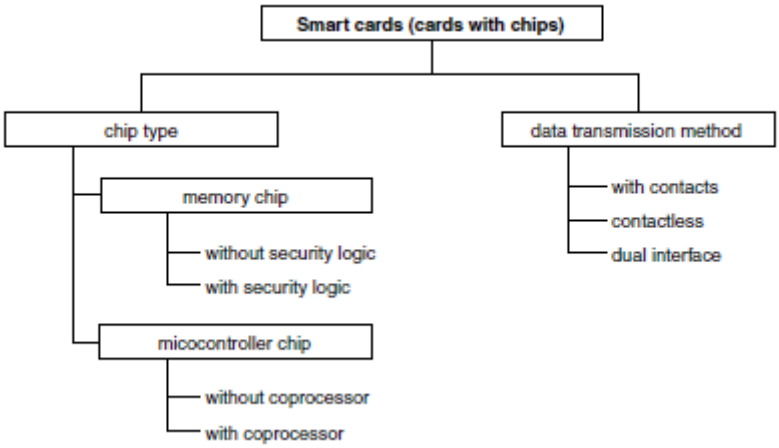

Figura 1: Clasificación para tarjetas con chips en función del tipo de chip y el método de trasmisión de datos [2].

1. Tarjetas de memoria (Memory Cards): tarjetas en las que el chip únicamente almacena datos con un acceso a memoria controlado por una lógica de seguridad, consisten en protección de escritura y de borrado, aunque también existen chips de memoria con una lógica de seguridad más compleja que pueden realizar tareas simples.
   La funcionalidad de las tarjetas de memoria está, generalmente, optimizada por una aplicación en particular. A pesar de que eso restrinja la flexibilidad, las hace que sean muy económicas.
   Un ejemplo de este tipo de tarjetas son las tarjetas telefónicas que almacenan el saldo restante, mientras que otra aplicación exterior es la encargada del control de los datos, o las tarjetas de seguro médico [2].
2. Tarjetas con microprocesador (Microprocessor Cards): El chip de estas tarjetas es un procesador rodeado de cuatro bloques funcionales: La ROM que contiene el sistema operativo del chip, la EEPROM que es la memoria no volátil desde la que se pueden escribir y leer datos bajo el control del sistema operativo, la RAM que es la memoria volátil de trabajo del procesador y la interfaz de E/S por la cual se transfieren los datos.
   Las tarjetas con microprocesador tienen un uso muy flexible. Los sistemas operativos modernos permiten utilizar varias aplicaciones diferentes integrados en una sola tarjeta.
   Se utilizan medidas de hardware y software especiales para prevenir que las condiciones de seguridad de las aplicaciones individuales sean violadas [2].

Por otro lado, también cabe hablar de las tarjetas inteligentes en función de las características del interfaz de entrada y salida, es decir con y sin contacto.

Frente a la alta tasa de fallos de las tarjetas con contacto surgen las tarjetas inteligentes sin contacto o contactless. Disponen de un circuito integrado el cual las permite ejecutar comandos y operar con los datos que posee o con los que pueda suministrar algún dispositivo externo. Estas tarjetas pueden transferir datos mediante el contacto de su superficie o a través de campos magnéticos.

Además de las ventajas técnicas de las que disponen las tarjetas sin contacto, su tecnología ofrece a los emisores y a los titulares de las tarjetas funcionalidades adicionales, como pueden ser el rango de funcionamiento o la forma de utilización, entre otras. Además de otorgar una gran simplicidad de uso.

Las tarjetas sin contacto ofrecen una ventaja de diseño ya que no tiene la necesidad de tener componentes técnicos a la vista para su uso como puede ser la banda magnética o los contactos, pero esto implica un mayor coste.

Hasta ahora las tarjetas sin contacto predominaban en sistemas locales por su uso como tiques electrónicos que empleaban una sola función, pero la demanda ha causado incorporar servicios. Por otro lado, están las tarjetas multifuncionales con microprocesadores integrados que se utilizan cada vez más con la función de pago

implementado basándose en la forma con contacto. Estas nuevas tarjetas tienen tanto elementos de las tarjetas con contacto como de las sin contacto, y se denominan “tarjetas de interfaz dual” o “combicards” [2].

Más adelante entraremos más en detalle en las tarjetas sin contacto.

2.2. Formatos

Los formatos más comunes de aspecto de tarjetas son:

1. ID-1: Es el más usado debido a las similitudes con una tarjeta de banda magnética. Además, es la más cómoda ya que tiene el tamaño justo para llevarla en la cartera. Las dimensiones de este formato de tarjeta son 54 mm de alto por 85’6 mm de ancho, siendo éste el formato típico de una tarjeta de crédito que conocemos [3].
2. ID-000: Solución para afrontar la necesidad de un formato más pequeño, como por ejemplo en la telefonía móvil, en el caso de la Suscriber Identity Module (SIM). Con unas dimensiones de 15 mm de alto por 25 mm de ancho [3].

En la Figura 2 [3] podemos ver una relación de aspecto entre los dos formatos de las tarjetas descritos.

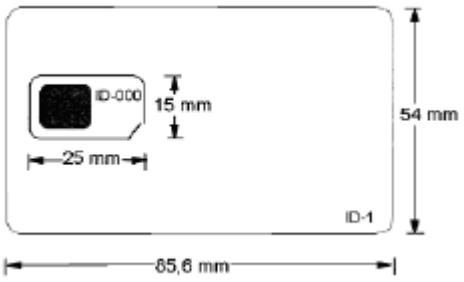

Figura 2: Relación entre ID-1 e ID-000 [3].

Incrustado en el plástico se encuentran unos contactos con forma cuadrada, que hacen de enlace entre el terminal/lector y la tarjeta, y a través de los cuales el microprocesador se alimenta y lleva a cabo la transmisión de datos. En la Figura 3 [3] vemos la correspondencia de cada contacto:

- VCC: Entrada de voltaje de alimentación.
- RST: Señal de reset controlada por el interfaz o hardware que empleemos.
- CLK: Señal de reloj.
- RFU: Reservado para un futuro uso.
- GND: Masa.
- VPP: Entrada de voltaje para programación.
- I/O: Entrada o salida de datos en serie al circuito integrado de la tarjeta.

Figura 3: Contactos del chip de una tarjeta Inteligente [3].

2.3. Protocolos de comunicación

La comunicación con una tarjeta inteligente siempre es iniciada por un dispositivo externo, quien hace la petición a la tarjeta, siguiendo un modelo de comunicación maestro-esclavo, donde el maestro es el dispositivo externo/terminal y la tarjeta el esclavo. Se emplea un modelo asíncrono half-duplex, como podemos ver en la Figura 4 [3].

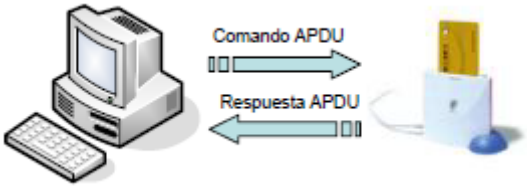

Figura 4: Modelo de comunicación [3].

Una vez conectados los terminales y activados, la tarjeta se enciende y envía una respuesta ATR al terminal, que es lo que contiene la información sobre las bases de la comunicación entre la tarjeta y el lector, la estructura de los datos intercambiados, el protocolo de transmisión, etc. El lector, tras recibir el ATR empieza a enviar instrucciones y la tarjeta las procesa y envía las respuestas asociadas a las mismas. El intercambio de comandos y respuestas termina una vez se desactiva la tarjeta.

La información de niveles superiores se intercambia mediante APDU. Definido su formato en el ISO 7816-4 [5], pero con un contenido y significado único para cada aplicación. Hay dos tipos de APDU: El comando, encargado de transferir datos externos a la tarjeta, y la respuesta, la cual es enviada por la tarjeta al lector en respuesta al comando recibido de este, como vemos en la Figura 4. Cada APDU tiene su propia estructura, como vemos en la Figura 5 [3].

Comando APDU

CLA	INS	P1	P2	Lc	Datos Opcionales	Le

Respuesta APDU

Datos Opcionales	SW1	SW2

Figura 5: Estructura de Comando y Respuesta APDU [3].

Donde:

- *CLA* es un byte el cual indica la clase de instrucción.
- *INS* es un byte que determina el comando enviado.
- *P1* y *P2* que son un byte cada uno y se utilizan como parámetros específicos del comando enviado.
- *Lc* es un byte que especifica la longitud de datos enviados a continuación de éste, si lo hubiera.
- Los *Datos* son opcionales y de longitud variable.
- *Le* es un byte que indica la longitud prevista de la respuesta.
- *SW1* y *SW2*, un byte cada uno, son "Status Word" e indican el resultado de la operación realizada.

Estas APDU se transmiten a través del protocolo de nivel inferior TPDU. Existen dos protocolos usados hoy en día regulados en la ISO 7816-3 [6]:

- T=0, que es de doble transmisión de byte asíncrona y half-duplex.
  Se usa un bit de paridad para detección de errores, con todo el byte erróneo retransmitido.
  Se trata de un protocolo que predomina por su sencillez y se emplea sobre todo para aplicaciones de monedero electrónico y GSM.
- T=1, que es doble transmisión de bloque asíncrona y half-duplex.
  Permite control de flujo en ambas direcciones de la comunicación, con lo que la tarjeta puede controlarla.

Los lectores suelen incorporar ambos protocolos, pero usan el que determine el ATR para mantener la compatibilidad con los sistemas que solamente puedan emplear el protocolo T=0. En la Tabla 1 [3] podemos ver la pila de protocolos propia de la comunicación tarjeta-dispositivo, y una comparativa con su equivalencia con la pila del modelo de capas OSI.

Capas OSI	ISO 7816
Capas Superiores	Protocolos Propietarios
Transporte Red	APDU
Enlace	T=0 ó T=1
Física	ISO 7816-3

Tabla 1: Comparativa del modelo OSI y los protocolos de la Tarjeta Inteligente [3].

2.4. Tarjetas Contactless

En este trabajo nos vamos a centrar en las tarjetas sin contacto, ya que es el tipo de tarjeta que podemos emular en el teléfono móvil.

Como ya hemos dicho anteriormente, las tarjetas sin contacto o contactless, son aquellas que no requieren de ninguna conexión eléctrica entre la tarjeta y el terminal, ya que disponen de una comunicación terminal-tarjeta usando Radio Frecuencia, aun así, estas siguen limitadas por las técnicas pasivas en las que la energía para alimentar la tarjeta se extrae del campo electromagnético del terminal de la tarjeta.

Estas tarjetas no tienen la necesidad de disponer de un chip. En comparación con los sistemas con contactos se tiene una prevención del deterioro de la tarjeta y un menor coste de mantenimiento. Estas tarjetas se caracterizan por su fácil uso y una mayor rapidez de comunicación.

Para que una tarjeta sin contacto se comunique con un terminal son necesarias tres funciones:

1. Transferencia de energía a la tarjeta para alimentar el circuito.
2. Transferencia de datos a la tarjeta.
3. Transferencia de datos desde la tarjeta al terminal.

2.4.1. ISO 14443

La norma ISO/IEC 14443 [13] es un estándar internacional que describe los modos de funcionamiento, y las propiedades de las tarjetas de proximidad con un alcance de aproximadamente 10 cm y una frecuencia de 13.56 MHZ.

El estándar ISO/IEC 14443 describe dos tipos de tarjetas: tipo A y tipo B. Las diferencias entre ambos están en su método de modulación y en su protocolo de inicialización de los procedimientos.

El estándar consta de cuatro partes:

- Parte 1: Define las características físicas de las tarjetas de proximidad.
- Parte 2: Especifica las características de la potencia de radio frecuencia y la señal de la interfaz de los tipos de tarjetas existentes.
- Parte 3: Se encarga de la comunicación entre el lector y los dos tipos de tarjetas existentes, A y B. Define desde el formato de las tramas, el contenido del comando inicial, hasta los métodos para detectar y comunicarse evitando la colisión.
- Parte 4: Especifica el protocolo de transmisión de bloque semidúplex. Define el formato de datos y elementos que se permiten durante la comunicación. Define además la secuencia de activación y desactivación del protocolo

2.5. Tecnología NFC, Near Field Communication

Se trata de una tecnología de comunicación inalámbrica de corto alcance (entre unos 5-10 cm) y alta frecuencia (13.56 MHZ) que permite el intercambio de información entre dispositivos y/o etiquetas/tarjetas sin contacto. Se puede usar con smartphones, tarjetas de crédito, etiquetas NFC, pegatinas con un chip NFC incrustado, permitiendo que el consumidor las configure. Esta tecnología en auge puede ser utilizada tanto para pagos, como para recibir/enviar información, o para identificar dispositivos [4].

2.5.1. Estandarización

Es una tecnología descrita por el NFCIP-1, y estandarizada en el ISO 18092 [7], ECMA 340 [8], y ETSI TS 102 190 [9], donde se especifica sus capacidades básicas, su velocidad de transferencia, los esquemas de codificación de bits, la modulación, su protocolo de transporte, y las precauciones para evitar colisiones en la fase de inicialización. Además, también se definen sus dos modos de operación:

- Activo: genera su propia señal de RF, para la transmisión de sus datos.
- Pasivo: Actúa como receptor sin generar campo magnético propio.

Hoy día, los dispositivos NFC implementan NFCIP-2, el cual se define en el ISO 21481 [10], ECMA 352 [11], y ETSI TS 102 312 [12], permitiendo elegir entre tres modos de operación, para asegurar la compatibilidad con los estándares internacionales de interoperabilidad en tarjetas inteligentes actuales, el ISO 14443 (para tarjetas de proximidad), el ISO 15692 (para las tarjetas en las inmediaciones), y el sistema para tarjetas inteligentes sin contacto FeliCa. Estos tres modos son:

- Transferencia de datos NFC.
- Dispositivo de acoplo cercano, definido en el ISO 14443.
- Dispositivo de acoplo en los alrededores, definido en el ISO 15692.

Con esto, NFC es compatible con la tecnología RFID, permitiendo la interacción con millones de tarjetas inteligentes, y otros dispositivos.

2.5.2. Interoperabilidad

Se necesitan distintos modos de operación para poder conseguir una interoperabilidad entre diferentes dispositivos NFC.

- Lectura/Escritura: El dispositivo NFC se usa en su modo activo para comportarse como un lector de chips NFC, permitiendo así la transmisión de información.
- Modo Peer-to-Peer (P2P): Se utiliza para intercambiar información entre dos dispositivos NFC.
- Emulación de tarjeta: Permite a un dispositivo emular a una tarjeta NFC en modo pasivo, permitiendo que otros dispositivos accedan a él y puedan realizar operaciones.

2.5.3. Aplicaciones

La cantidad de aplicaciones que se pueden realizar con la tecnología NFC crece continuamente, ofreciendo grandes posibilidades junto con la utilización con los smartphones:

- Pago sin contacto. El usuario realiza el pago sin necesidad de existir conexión entre el dispositivo empleado y la máquina.
- Control de acceso. Cuando el dispositivo NFC se aproxima al terminal, este se activa.
- Conectividad. NFC se puede emplear para transmitir información de forma rápida acercando los dispositivos.
- Aplicaciones médicas.
- Etc.

2.6. Emulación de tarjetas inteligentes

Según la RAE (Real Academia Española) emular significa imitar las acciones de otro procurando igualarlas e incluso excederlas. Más enfocado a la informática emular significa funcionar (un programa o dispositivo) de la misma manera que otro [14]. Partiendo de esto, nuestro objetivo en este trabajo es conseguir emular una tarjeta inteligente sin contactos en un smartphone Android. Los smartphones Android que ofrecen funcionalidades NFC, tienen los mismos tres modos de operación que NFC contempla.

Dentro de la posibilidad de emular una tarjeta en un dispositivo Android podemos encontrar dos tipos de emulación:

1. Emulación de tarjetas con elemento seguro (SE).
2. Emulación de tarjetas basado en Host.

2.6.1. Emulación de tarjetas con elemento seguro

Actualmente gran parte de los dispositivos Android que ofrecen la función NFC permiten emular tarjetas inteligentes. En la mayoría de los casos, el dispositivo Android contiene un chip conocido como elemento seguro el cual emula la tarjeta. Cómo podemos ver en la Figura 6 [15], cuando el usuario aproxima el dispositivo Android a un terminal NFC, el controlador NFC del smartphone enruta todos los datos del lector al SE.

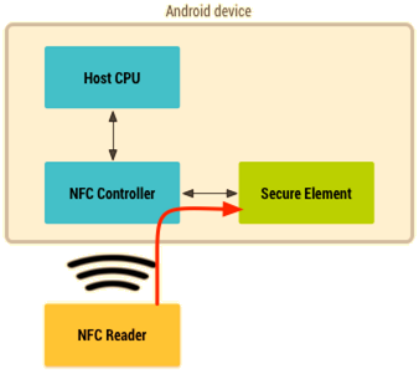

Figura 6: Emulación de tarjeta NFC con Elemento Seguro [15].

El SE se comunica con el terminal sin necesidad de ninguna aplicación. Cuando el proceso ha terminado la aplicación puede enviar una consulta al SE para conocer el estado de la transacción y si fuera necesario comunicarlo o mostrárselo al usuario.

2.6.2. Emulación de tarjetas basada en el Host

Desde la versión de Android 4.4 se incluye la funcionalidad de emulación de tarjetas basada en el host (HCE, Host Card Emulation) en el sistema operativo. En lugar de requerir de un SE para emular la tarjeta, este nuevo método permite que las aplicaciones Android puedan emular una tarjeta y esta pueda comunicarse directamente con el lector de NFC.

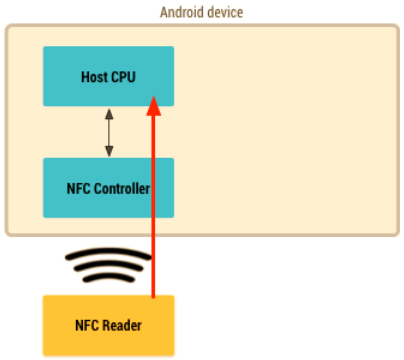

Figura 7: Emulación de tarjeta NFC sin Elemento Seguro [15].

En la Figura 7 [15] observamos cómo funciona la emulación de tarjetas basada en host, según la cual los datos se enrutan directamente a la CPU del host en la que se ejecutan las aplicaciones Android.

2.6.3. Tarjetas y protocolos NFC compatibles

Los estándares de NFC ofrecen compatibilidad con diferentes protocolos, y es posible emular diversos tipos de tarjetas.

En la versión 4.4 de Android se permiten diversos protocolos comunes en el mercado actual. Muchas de las tarjetas con tecnología sin contacto se basan en estos protocolos, como pueden ser las tarjetas de pago sin contacto. Estos protocolos además son compatibles con los lectores NFC actuales, así como con los lectores de dispositivos con NFC de Android. En concreto, Android 4.4 permite emular tarjetas basadas en la especificación ISO-DEP de NFC Forum, sustentada en ISO/IEC 14443-3, y procesar APDU definidas en la especificación ISO/IEC 7816-4; además Android exige que la emulación de ISO-DEP sea por encima de la tecnología NFC-A, de la ISO/IEC 14443-3 tipo A, en cambio la compatibilidad con la tecnología NFC-B, de la ISO/IEC 14443-4 tipo B, es opcional. Toda esta superposición de especificaciones y de protocolos la podemos ver en la Figura 8 [15].

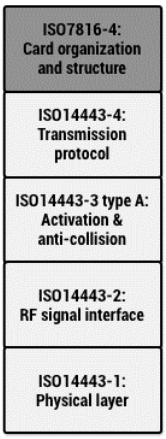

Figura 8: Pila de protocolos HCE de Android [15].

2.6.4. Servicios de HCE

La arquitectura HCE en Android se basa en los llamados "Servicios de HCE" de Android.

Estos servicios tienen la ventaja de que pueden ejecutarse en segundo plano sin una interfaz de usuario. Este es un comportamiento habitual en muchas aplicaciones de HCE, como puede ser el ejemplo de las tarjetas de transporte público o de fidelidad. En el caso de estas aplicaciones, el usuario no tendría que abrir la aplicación para usarla. En su lugar, cuando el usuario toca el lector de NFC con el dispositivo Android, el servicio se inicia, en caso de que no lo estuviera ya, y ejecuta la transacción en segundo plano. Pueden incorporarse interacciones adicionales con el usuario, como pueden ser las notificaciones si se considera necesario [15].

2.6.4.1. Selección de servicios

La especificación ISO/IEC 7816-4 define la forma de seleccionar aplicaciones centrada en un identificador de aplicación (AID), de forma para que cuando el usuario acerque el dispositivo a un lector NFC el sistema Android sepa qué servicio de HCE es el que se quiere comunicar con el lector de NFC.

Un AID puede componerse de hasta 16 bytes. Si se emula una tarjeta para infraestructuras de lectores de NFC ya existentes, se buscan AID que suelen ser conocidos, o están registrados públicamente. En cambio, si se quiere implementar una nueva infraestructura de lector para la aplicación de la tarjeta emulada se deben registrar los AID propios. Este procedimiento de registro se define en la especificación ISO/IEC 7816-5; es recomendable registrar los AID según dicha especificación para evitar colisiones con otras aplicaciones.

Los diferentes AID se gestionan en grupos de AID donde todos los del mismo grupo se enrutan al mismo servicio HCE.

Con esta premisa observamos que no es posible distinguir entre AID del mismo grupo.

Los grupos más relevantes, los cuales se encuentran definidos en la ISO/IEC 7816-5 son:

- AID registrados internacionalmente, los cuales empiezan con 'A'.
  Para este tipo, el AID se divide en dos partes, un RID obligatorio de 5 Bytes y un PIX opcional de hasta 11 Bytes.
- AID registrados a nivel nacional, los cuales empiezan con 'D'.
- AID patentados/sin registro, los cuales comienzan con 'F'.
  En el caso de este grupo la ISO/IEC 7816-5 no exige ningún registro de longitud mínima.

Cada grupo de AID puede asociarse con una categoría, lo que permite a Android agrupar los servicios de HCE por categorías y por lo tanto permite al usuario establecer valores predeterminados en el nivel de categoría en vez de en el nivel del AID.

Android distingue entre dos categorías:

- Categoría de pago, la cual cubre las aplicaciones de pagos estándares.
  Sólo se habilita un grupo de AID para esta categoría en el sistema.
  Se usa para aplicaciones que tengan los protocolos principales para el pago con tarjeta de crédito y posible de usar en cualquier comercio.
- Categoría de otras, para todas las demás aplicaciones de HCE.
  Los grupos de AID de esta categoría pueden estar siempre activos y si es necesario, se puede obtener la prioridad de los lectores de NFC durante la selección del AID.
  Para las aplicaciones de pagos de bucle cerrado las cuales solo funcionan en un comercio, como pueden ser las tarjetas de valor almacenado también se clasifican como "otras" y no como "de pago" [15].

3. Sistemas de Control de Acceso

Se entiende por control de acceso el mecanismo mediante el cual una función de identificación o autenticación permite acceder a datos, un recinto, o incluso a dispositivos. Podemos encontrar controles de acceso en recursos físicos (controles de acceso en puertas de edificios, salas donde se requiera un control del personal, etc.) como virtuales (como un control de acceso de una red de comunicaciones, control de acceso a una página Web, etc.). Los principales objetivos que se buscan en el control de acceso son: garantizar la seguridad tanto de los datos, recintos etc., y facilitar la organización empresarial, llevando un control de quien, y cuando accede a los mismos.

Un control de acceso tiene tres funciones básicas: La autenticación permite identificar al solicitante del acceso pudiendo ser de diferentes modos como clave por teclado, lector de tarjetas, biométrica, etc. La autorización comprueba los permisos o derechos de acceso. Y por último la trazabilidad que facilita los listados de los usuarios y accesos [16].

3.1. Clasificación de los controles de acceso

Los sistemas de control de acceso se pueden clasificar de tres formas diferentes:

1. Según el grado de automatización.
2. Según la forma de identificación.
3. Según el tipo de conexión.

3.1.1. Según el grado de automatización

3.1.1.1. Controles Manuales.

El permiso de acceso es concedido por vigilantes, guardias de seguridad, personal administrativo, etc. Este sistema requiere de una gran planificación para conocer a todo el personal, así como el acceso de cada uno. El inconveniente de este tipo de control es que no funciona para un grupo grande de usuarios, o cuando hay cambios constantes de personal y de acceso [16].

3.1.1.2. Controles Semimanuales.

Se utilizan equipos o elementos electromecánicos para apoyar a la evaluación del acceso. Estos toman la decisión de permitir o denegar la entrada. Uno de los elementos más utilizados en este tipo de control son las botoneras digitales, donde se ha de introducir un código para acceder [16].

3.1.1.3. Controles automáticos.

La verificación y el acceso son efectuados totalmente por sistemas electrónicos. Estos sistemas son los encargados de conceder o denegar el acceso. Este tipo de control otorga mayor seguridad y control, además de caracterizarse por un control más cómodo [16].

3.1.2. Según el sistema de identificación

3.1.2.1. Control de acceso con identificación biométrica.

La identificación se basa en leer datos físicos del usuario que quiere acceder, lo cual impide la suplantación de identidad, ya que estos datos son intransferibles. Esto hace de este tipo de control de acceso de los más seguros, además de cómodo y sencillo.

El tipo de control de acceso con identificación biométrica más extendido y conocido es el control de acceso mediante huella dactilar. El lector que lee la huella dactilar utiliza unos puntos biométricos de la huella para identificar al usuario autorizado. En la Figura 9 [16] se puede observar un lector utilizado para este caso.

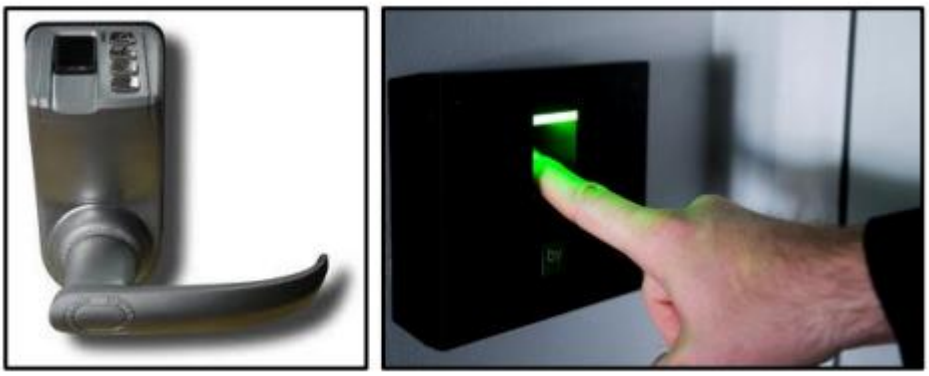

Figura 9: Lector por huella dactilar [16].

El control mediante reconocimiento facial y mediante lectura del iris son muy similares al del lector de huella descrito anteriormente. Utiliza una serie de puntos biométricos de la cara, para el reconocimiento facial, de cada individuo para identificar al usuario. En el caso del sistema de iris, el lector realiza un escáner ocular. Este tipo de control de acceso no necesita contacto físico entre la persona y el lector, lo que lo convierte en un control más higiénico y duradero. Esto supone una gran ventaja y hace que la tendencia futura, gracias al incremento en su fiabilidad, sea sustituir el control con huella dactilar [16]. En las Figura 10 y Figura 11 se pueden ver lectores de ambos tipos de control.

Figura 10: Lector por reconocimiento de iris [16].

Figura 11: Lector por reconocimiento facial [16].

3.1.2.2. Control de acceso con tarjetas.

Los controles de acceso mediante tarjetas son utilizados para controles de acceso en los que sea la opción de control más idónea, ya sea por motivos económicos, ambientales, etc.

Figura 12: Lector de banda magnética [16].

Figura 13: Lector de tarjeta inteligente [17].

Dentro de las tarjetas que se utilizan tenemos: las tarjetas magnéticas, que se introducen en el lector por el lado de la banda magnética que contiene los datos del usuario, y cuyo tipo de lector se ve en la Figura 12 [16]. Y las tarjetas inteligentes que no necesitan contacto con el lector ya que se conecta con él por radiofrecuencia, cuyo lector se tiene en la Figura 13 [17]. Este tipo de control cuenta con la ventaja de un bajo costo, con un gran periodo de vida, y en cuanto a las tarjetas inteligentes no son fáciles de duplicar, y proporcionan agilidad de acceso [16].

3.1.2.3. Control de acceso con lectura de código de barras.

Se trata de una tarjeta la cual lleva impreso un código de barras (el cual está protegido con una banda protectora para evitar fotocopias). Su principal ventaja es lo económico que puede resultar este sistema, y el poco deterioro de este, pero en su contra tiene una baja seguridad al ser fácilmente falsificables, y además ante un pequeño deterioro en la tarjeta puede alterar el código impidiendo el acceso [16]. El formato de los terminales que leen el código de barras se ve en la Figura 14 [18].

Figura 14: Lector de código de barras [18].

Los códigos QR dinámicos dotan de variabilidad y dan mayor robustez a este tipo de sistema ya que permiten un cambio de la URL de destino cada cierto tiempo o cada vez que le pase una persona garantizando un alto grado de seguridad contra copias.

3.1.2.4. Control de acceso con contraseña numérica.

Algunos controles de acceso permiten acceder, fichar o identificarse mediante una contraseña en el propio terminal de acceso (Figura 15). Consiste en introducir una clave en un teclado, alfanumérico o simplemente numérico, asignada previamente a cada usuario. El inconveniente de este sistema es que no es muy seguro, ya que el usuario es el que debe proteger la clave y a veces se puede olvidar, apuntar en algún lado perdiendo así toda la seguridad, o incluso diciendo la clave a otro usuario [16].

Figura 15: Control de acceso mediante clave [16].

Existen muchos sistemas de acceso los cuales utilizan varios métodos identificativos, o que contemplan la posibilidad de acceder mediante un método u otro de identificación para aumentar así la seguridad.

3.1.3. Según el tipo de conexión que se necesite

3.1.3.1. Sistemas de acceso autónomos.

Se trata de un sistema de inversión reducida, el cual aporta movilidad tanto de usuarios como de dispositivos, protección, precisión, control, gestión y trazabilidad del acceso. En este tipo de sistemas los datos están almacenados en el terminal, y es el propio terminal el que da o deniega los accesos, sin necesidad de software de control adicional.

La información del acceso es registrada y se puede obtener bajo demanda. Esta información se puede transmitir de dos formas, yendo físicamente al terminal y transferir los datos a un USB, tarjeta de proximidad, etc., o de forma inalámbrica, introduciendo cambios que se envían al terminal inalámbricamente, siendo este el que los almacena. Esto provoca no tener acceso a la gestión de la información inmediatamente, sino que se tiene una gestión diferida [16].

3.1.3.2. Sistemas de acceso en red.

Son sistemas de control que se integran a través de un PC local o remoto, donde se usa un software que permite el registro de todas las operaciones realizadas sobre el sistema, permitiendo controlar diferentes zonas, acotar accesos por horarios y/o permisos, controlar horarios, etc.

Estos sistemas pueden ser tanto aplicaciones sencillas como sistemas complejos, según la necesidad [16].

3.1.3.3. Sistemas de control de accesos de presencia híbridos.

Tienen un comportamiento mezclado de los dos tipos anteriores, parte de la funcionalidad es realizada con conexión a un PC, y otra parte es realizada autónomamente. Su uso normal es conectado online a un servidor y funciona tal y como un sistema de acceso en red, pero en caso de fallo en la red y la perdida de comunicación con el servidor, el dispositivo tendría cargados los datos y funcionaría como un sistema autónomo [16].

3.1.4. Smartphone

También podemos encontrar el control de acceso en nuestros smartphones, ya que estos disponen de diferentes tecnologías para desbloquearlos, lo que no deja de ser un control de quien puede y quien no acceder a la información y aplicaciones que tenemos en el smartphone. Los sistemas que podemos encontrar hoy en día en los teléfonos móviles son básicamente los mismos que se han descrito anteriormente:

1. Desbloqueo mediante patrón.
   El usuario diseña un patrón uniendo una cantidad mínima de puntos distribuidos por la pantalla. Este tipo de bloqueo tiene una seguridad media, ya que puede ser fácil de copiar o memorizar.
2. Desbloqueo mediante PIN.
   Para el desbloqueo del smartphone se usa una serie de números de 4 dígitos. Este método tiene una seguridad media-alta.
3. Desbloqueo mediante contraseña.
   Este tipo es muy similar al desbloqueo mediante PIN, teniendo un mínimo de 4 dígitos y un máximo de 16. Sin embargo, en este caso puede ser una mezcla de diferentes tipos de caracteres, lo que confiere una alta seguridad, ya que se puede hacer la contraseña todo lo compleja que se quiera.
4. Desbloqueo mediante iris.
   Está basada en la forma y patrón únicos del iris de cada usuario, lo que garantiza una alta seguridad en el smartphone. Se escanea el iris desde la parte frontal de la pantalla, pero puede no funcionar correctamente si se lleva gafas, si se tiene un protector de pantalla o si se ha tenido una cirugía ocular recientemente.
5. Desbloqueo facial.
   Este método se considera menos seguro que otros tipos de desbloqueo, ya que existe la posibilidad de que alguien se acerque y con una foto pueda desbloquear el smartphone, para prevenir esto la mayoría de estos métodos requieren de prueba de vida, movimiento, etc. Además, para su correcto funcionamiento se debe tener la cara bien visible durante el registro y puede no funcionar de la forma adecuada si se producen cambios estéticos, como cambio de peinado, gafas, vello facial, maquillaje, etc.
6. Desbloqueo mediante huellas dactilares.
   Este método es igual al descrito de forma general anteriormente, los smartphones permiten registrar varias huellas (hasta 4) para mayor comodidad,

incluso se pueden registrar una misma huella varias veces para un mejor reconocimiento de esta.

7. Desbloqueo mediante escaneo inteligente.
   Utiliza tanto la cara como el iris para desbloquear el dispositivo, aumentando así la seguridad y facilidad de uso.

En los smartphones para mayor seguridad, si se elige un desbloqueo por datos biométricos se exige además un desbloque mediante un método convencional (patrón, PIN, contraseña) por si no reconoce correctamente los datos. En caso de olvidar el patrón, PIN o contraseña, se tiene que restablecer el teléfono por completo, borrándose todos los datos del dispositivo [23].

3.2. Componentes de un sistema de control de acceso

Los componentes de los que dispone un sistema de control de acceso dependen del tipo de sistema del que se disponga. En esta parte se van a mencionar los elementos principales, siendo:

- Controlador: Se trata del elemento que se encarga de determinar que usuario tiene acceso y cual no. Es consultado en cada intento de ingreso.
- Servidor: Es un dispositivo, normalmente un PC, encargado de almacenar la información de cada acceso para mantener un registro de dichos intentos. En el servidor es donde se ejecutan las instrucciones de los programas. En los sistemas autónomos este componente no existe, sino que la información es almacenada en el terminal directamente.
- Lector/Terminal: Dispositivo con la información necesaria para identificar al usuario que solicita el acceso. El lector se comunica con una credencial y envía información de esta al controlador para determinar los permisos.
- Credencial: Es aquello que identifica a la persona y de la que se necesita la información para obtener el acceso. Se puede definir como algo que le usuario sabe, es, o tiene.
- Mecanismo de apertura: Se tratan de los mecanismos que permitirán el acceso al usuario, en el caso de que lo tenga, abriendo una puerta, un torno, etc.

Según el tipo de sistema de control de acceso que se tenga estos componentes pueden variar, desde los elementos de alimentación, hasta el tipo de terminal [16].

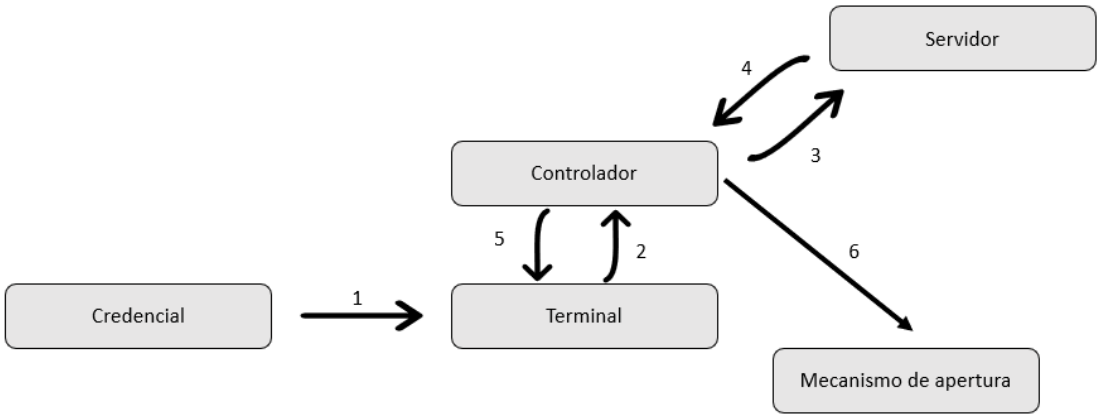

Figura 16: Interacción de los componentes de un sistema de control de acceso.

En la Figura 16 se tiene la arquitectura típica de un sistema de control de acceso. En primer lugar, la credencial se identifica con el terminal. Después este para consultar si dicho credencial tiene o no acceso se comunica con el controlador, que a su vez se comunica con el servidor. El controlador, tras determinar si tiene o no acceso, y la acción a realizar, manda una instrucción al mecanismo de apertura si se concede el acceso. En caso de que el usuario no disponga de acceso le mandará una instrucción al terminal para indicarlo.

3.3. Soluciones comerciales

En este punto se hablará sobre las soluciones que otorgan algunas empresas en sistemas de control de acceso, pudiendo aportar una experiencia más profesional sobre los sistemas más empleados, sus ventajas, inconvenientes, etc.

3.3.1. Salto Systems.

Se trata de una empresa dedicada a los controles de acceso la cual ha tenido un gran impacto en este ámbito debido a sus innovaciones.

Su plataforma XS4 basada en tecnología online y en tiempo real permite la monitorización, el control y la seguridad de cualquier puerta, edificio y usuario. Los hardware y software SALTO tienen la posibilidad de estar conectados en red sin necesidad de cables para un control instantáneo e inteligente a tiempo real, permitiendo la integración de sistemas ya existentes para mejorar la manejabilidad y la experiencia de los usuarios.

SALTO dispone de soluciones en el ámbito comercial, en la educación, en el sector gubernamental, en sanidad, hoteles, en entornos de ocio y entretenimiento, y en el ámbito del transporte.

Ofrece una amplia gama de opciones que se adecuan a las necesidades de seguridad del usuario, tanto de forma presente como en vista a un futuro. Van desde un simple sistema autoprogramable hasta un sistema data-on-card de la SVN [19].

Dentro de todas sus opciones disponen de un sistema de control llamado “SALTO Neo Cylinder” (Figura 17) que se trata de un cilindro de cerradura de puerta inteligente con tecnología inalámbrica de última generación que se adapta a todo tipo de puertas.

Figura 17: SALTO Neo Cylinder [19].

Su principal característica es que no se usan llaves mecánicas convencionales, sino que recopilan todos los derechos de acceso en una etiqueta inteligente (RFID, NFC o BLE).

Permite la conexión a una red local a través de la tecnología SVN, o conexión a la nube con la plataforma de gestión de acceso alojada en la nube SALTO KS. Su tecnología en la nube proporciona una funcionalidad y rendimiento superior al posible en una solución mecánica tradicional. Permite tener el control de las funciones y características de la aplicación web además de permitir usar la aplicación móvil para realizar seguimientos, desbloquear el cilindro de forma remota y bloquear el acceso de usuarios.

Esta solución ofrece una plataforma de cerradura electrónica fácil de usar, instalar y reequipar con todas las necesidades de seguridad haciendo uso de las prácticas de seguridad líderes en la industria para brindar una experiencia segura y sin llave [19].

3.3.2. HID global.

Es el líder mundial en control de acceso brindando de seguridad a instalaciones, activos, redes y recursos. Las soluciones de control de acceso de HID permiten un acceso confiable, entornos seguros, comodidad para el usuario y optimización del flujo de trabajo, mayor capacidad de visualización debido a la tecnología en tiempo real, protección de la inversión con soluciones escalables, y soporte técnico experto y personalizado. Consta de diversas soluciones como el control de acceso dinámico, acceso mediante smartphone, servicios de localización, autenticación biométrica, gestor de visitantes, etc. [20].

HID global ha desarrollado una solución para un sistema de control de acceso llamada "HID Mobile Access" (Figura 18) que satisface la necesidad del acceso móvil y el deseo de la funcionalidad, respondiendo a las demandas de comodidad, facilidad de los usuarios finales y la seguridad. Se basa en el control de acceso mediante el uso de un dispositivo móvil a corta distancia con tecnología NFC, o a mayor distancia usando la tecnología twist and go, y un terminal HID. HID Global ofrece herramientas que permiten integrar esta solución en cualquier entorno laboral o aplicación organizacional. Ofrece un gran nivel de seguridad y protección de privacidad debido a la tecnología Seos utilizada, ajustándose a las normas ISO. Los lectores de HID tienen una seguridad por niveles, lo que significa que combina varios controles de seguridad para proteger los recursos y la información.

Figura 18: Ejemplo de un sistema HID Mobile Access [20].

Este sistema consta de un interfaz fácil de usar para los administradores, y de inscripción sencilla para el usuario final a través de la aplicación. Es un sistema con transacciones únicas que garantizan la privacidad, y dispone de una fácil gestión de los derechos de acceso que son controlados por el administrador en cualquier momento [20].

3.3.3. Assa Abloy.

Distribuido en más de 100 países proporcionan una mejora de flujo, y comodidad en la empresa proporcionando soluciones de entradas automatizadas, ayudando a optimizar entradas en función de los requisitos específicos de cada empresa.

Assa Abloy consta de una gran experiencia en soluciones para conseguir entradas óptimas, como pueden ser soluciones completas de puertas para fábricas, puertas de acceso para hospitales, puertas de acceso para el transporte, para minería, astilleros, aviación, etc. [21].

Assa Abloy dispone de la Puerta plegable “Crawford FD2050P”, que se puede ver en la Figura 19, la cual ofrece una amplia gama de funciones que permiten el control de apertura y la seguridad de una forma avanzada.

Figura 19: Crawford FD2050P [21].

La puerta dispone de funciones de control básicas, automáticas, externas, de seguridad, etc. mediante el sistema de control de puerta 950 (Figura 20) para accionar esta. El sistema 950 FD controla desde la apertura, a el indicador de servicio, la función de apertura, la banda de seguridad, etc.

Figura 20: Sistema 950 FD [21].

3.3.4. Zitelia

Se trata de una empresa de Cantabria dedicada a la transformación digital que acerca las tendencias tecnológicas a cualquier sector. Resuelven las necesidades de los clientes apoyándose en las nuevas tecnologías para adaptarlas a su entorno. La empresa se dirige principalmente a empresas y asociaciones, proyectos emprendedores, y a organismos e instituciones. Se centran en crear soluciones concretas para problemas específicos. Dentro del ámbito de sistemas de control de acceso se basan en la distribución de sistemas de control de acceso e identificación.

Desarrollan proyectos a medida, desde la configuración y elección de la tipología de los dispositivos, pasando por el suministro de identificadores (como tarjetas, pulseras, llaveros de proximidad RFID, etc.) y todo tipo de terminales y mecanismos de apertura, la programación e instalación del sistema, y la puesta en marcha del sistema completo.

Zitelia aporta una sencilla implementación en múltiples entornos, integran las infraestructuras y dispositivos ya existentes, dispone de sistemas flexibles y adaptables a cualquier instalación, además dispone de múltiples posibilidades en función del terminal, credencial y mecanismos deseados, etc. [22].

Al tratarse de una empresa distribuidora de diferentes dispositivos de sistemas de control de acceso, no tiene productos propios como tal, sino que implementa sistemas a partir de productos de diferentes marcas o fabricantes en función de las diferentes necesidades de cada sistema.

Entre los diferentes productos que distribuye se puede destacar el sistema "BioStation" (Figura 21) de la empresa Suprema.

Figura 21: BioStation de Suprema [22].

El terminal BioStation permite la identificación de hasta 200.000 usuarios mediante PIN, huella dactilar, y tarjetas RFID. Además, permite tener un sistema autónomo, o conectarse a una red mediante Wi-Fi dependiendo de las necesidades de cada lugar. Debido a su algoritmo de comunicación encriptada y a su tecnología este sistema aporta una mayor seguridad frente a otros terminales de huella dactilar.

3.4. Factores de elección para despliegues

Como se ha visto se tiene diversos sistemas de control de acceso. A la hora de saber qué tipo de sistema implantar es esencial tener en cuenta diferentes factores, tanto internos como externos. Algunos de los factores a considerar antes de implantar un sistema de control de acceso son:

1. **Localización.** Dependiendo del lugar en el que se va a implantar se necesita más o menos seguridad y control. Un local dentro de un centro urbano no necesita la misma cantidad de seguridad y control que si dicho local se encuentra en una zona de paso o dentro de algún edificio.
2. **Flujo de tráfico.** Se ha de tener en cuenta la cantidad de usuarios que entran y salen del recinto, siendo así importante buscar sistemas más rápidos para aquellos recintos con un mayor flujo.
3. **Necesidad de la empresa/negocio.** En función del tipo de negocio, empresa, entidad o recinto al que se quiera acceder se ha de tener en cuenta un control de acceso u otro. Empresas como bancos, industrias químicas, embajadas, etc. necesitan un control seguro debido a la importancia de la entidad, incluso requiriendo de seguridad reforzada.

4. **Coste.** Los diferentes sistemas de seguridad, como el acceso biométrico, las tarjetas inteligentes, etc. tienen costes variados. Se ha de tener en cuenta la necesidad de seguridad del lugar para comparar el coste entre un sistema y otro, así como las versiones de cada uno y su reparación en caso de fallo.
5. **Vida del sistema.** Al elegir un sistema de control de acceso, se ha de pensar en el tiempo que los sistemas podrán estar en uso útil, y pensar en sistemas que permitan adaptarse a futuras actualizaciones en cuanto al recinto o a desarrollo tecnológico.
6. **Interoperabilidad.** Es importante saber si el sistema de acceso se va a necesitar integrar en un software externo como puede ser un control de horario, sistemas de seguridad, etc. o no a la hora de elegir la mejor tecnología.
7. **Facilidad de uso.** Se ha de pensar en los usuarios y asegurar un sistema de seguridad fácil para ellos. Además, ha de ser fácil de administrar y de mantener.

Una vez se tiene en cuenta todos los aspectos citados a la hora de elegir el sistema de control de acceso más apropiado para cada caso se presenta un ejemplo en el que plantear diferentes controles de acceso.

Considérese un laboratorio dentro de una empresa privada. La empresa ya de por sí tendrá su propia seguridad, y su control de acceso, luego en el laboratorio solo haría falta tener un control de los usuarios que tienen acceso al edifico. Si se busca tener un acceso más seguro se tendría que emplear alguna de las tecnologías de medida biométrica, en cambio si lo único que se busca es tener cierto control, pero de una forma económica la mejor opción sería desarrollar un control de acceso mediante código.

4. Equipo embebido de control de acceso

En este capítulo se detalla el desarrollo de la configuración, interconexión y programación del equipo embebido que permitirá el acceso con la lectura de la tarjeta. Se partirá de un ejemplo de comunicación básica con una tarjeta, que se extenderá a necesidad para lograr una comunicación entre el terminal de acceso y el dispositivo NFC pasivo. En este capítulo el dispositivo NFC será una tarjeta inteligente, y en capítulo 6 se modificará para que la comunicación sea con la aplicación móvil creada.

4.1. Escenario

El objetivo del trabajo es tener un control de acceso donde las credenciales de acceso se obtengan mediante la emulación de una tarjeta inteligente a través de un dispositivo Android. Se plantea que el equipo embebido y que habilitará el acceso a la instalación no tenga acceso a internet, siendo la aplicación de emulación móvil la que proporcionará dicho acceso para adquirir las credenciales y los pertinentes permisos.

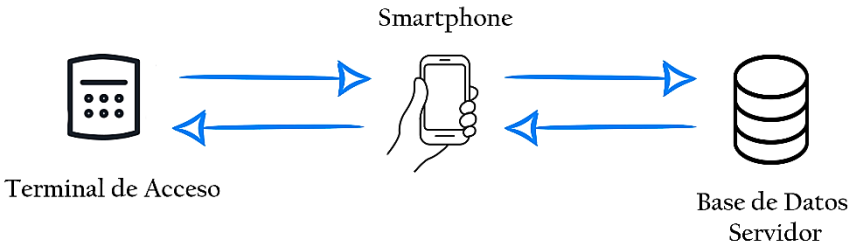

Figura 22: Esquema del funcionamiento del proyecto.

En el esquema de la Figura 22 se muestra una visión general de la solución planteada en el ámbito de este proyecto. El terminal de acceso se comunica con el smartphone que emula una tarjeta inteligente para validar si el usuario tiene acceso o no. Este, basándose en la información alojada en la base de datos remota en la que se tienen los usuarios, remitirá la información necesaria como respuesta para que en función de los permisos el terminal de acceso tome las decisiones pertinentes.

Por tanto, de una parte, se tiene que configurar un equipo embebido y de bajo coste que empleando un lector de tarjeta inteligente sea capaz de comunicarse con una tarjeta y gestionar el acceso a la instalación. De otra, se deberá desarrollar la correspondiente aplicación móvil que emule la tarjeta y desplegar el servidor de base de datos.

En este sentido, en este capítulo nos centraremos en los desarrollos del equipo embebido, para en los siguientes capítulos acometer el desarrollo de la aplicación móvil y el proceso de validación de la solución completa.

El equipo embebido empleará un lector conectado a él para leer la tarjeta siguiendo la metodología descrita en la Figura 4 por la cual mandará un comando APDU, recibiendo la respuesta encapsulada en una respuesta APDU. Este equipo además deberá detectar

la existencia de tarjetas en su proximidad para iniciar la comunicación en la detección de la tarjeta, y en envío de APDU desde el terminal de acceso. Se configurará la palca de forma que pueda leer la tarjeta y el smartphone, y ya en el capítulo 6 se explicará la programación una vez se tiene la comunicación con la aplicación deseada.

4.2. Componentes hardware

Para desarrollar el sistema embebido a desplegar en la instalación a controlar, se ha considerado emplear una placa Arduino Mega 2560 a la que se conectará un lector de tarjetas inteligentes sin contacto basado en el chip PN532 (Breakout Board).

En las Figura 23 [24] y Figura 24 [27] se puede ver la placa Arduino Mega 2560 y el lector PN532 Breakout Board, respectivamente con los que se va a trabajar.

Figura 23: Arduino Mega 2560. [24]

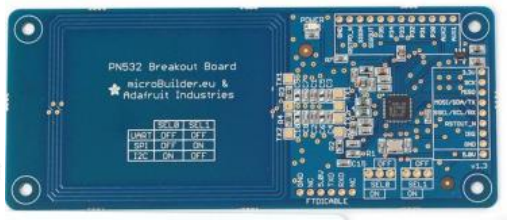

Figura 24: Circuito integrado PN532 Breakout Board. [27]

4.2.1. Arduino Mega 2560

Arduino Mega 2560 (Figura 23) es una placa para desarrollo que se basa en el microcontrolador ATmega 2560, y posee 54 entradas y salidas digitales, de las que entre ellas 15 se pueden usar como salida PWM, 16 entradas analógicas, 4 puertos serie por hardware (UART), un cristal de 16MHz, conexión USB, un conector de alimentación, un conector ICSP para programación, que tiene acceso a la memoria de programa del AVR (flash) y un botón de reset [25].

La placa dispone de todo lo necesario para dar soporte al microcontrolador, y es compatible con la mayoría de las placas de expansión (shields) que se han diseñado para la Arduino UNO y modelos anteriores a esta (como pueden ser la Arduino Deumilanove, la Arduino Decimila, etc.), gracias a su microcontrolador, posee más y mejor poder de cómputo, capacidad de memoria, y líneas de expansión que las placas citadas antes.

La Arduino Mega 2560 es ideal para proyectos grandes, como pueden ser proyectos domóticos, en los que se haya que controlar gran cantidad de sensores, control de robots o para impresoras 3D, etc.

La placa Arduino Mega 2560 se puede alimentar mediante conexión USB o mediante una fuente de alimentación externa, siendo seleccionada automáticamente por la placa [25].

4.2.2. PN532 Breakout Board

El PN532 (Figura 24) es el chip NFC más popular y el que está integrado en todos los dispositivos que utilizan NFC.

Dicho chip incorpora las funcionalidades para llevar a cabo operaciones de lectura y escritura en etiquetas y tarjetas, comunicarse con smartphones, y comportarse como una etiqueta NFC.

Se trata de un chip muy flexible. Se puede usar 3.3V TTL UART a cualquier velocidad, I2C, o SPI para comunicarse con él.

4.3. Desarrollo

Tras configurar el entorno de desarrollo Arduino IDE [26], y realizar algún código simple de prueba para familiarizarse y comprobar el correcto funcionamiento de la placa, se procede a conectar la placa Arduino con el lector PN532. En Tabla 2, y en la Figura 25 podemos ver las conexiones entre ambos.

Arduino MEGA 2560		**PN532**
3.3V		*3.3V*
GND		*GND*
Conector de Entrada/Salida Digital 2		*SCK*
Conector de Entrada/Salida Digital 5		*MISO*
Conector de Entrada/Salida Digital 3		*MOSI/SDA/TX*
Conector de Entrada/Salida Digital 4		*SSEL/SCL/RX*

Tabla 2: Conexiones entre Arduino MEGA 2560 y lector PN532 Breakout Board, con correspondencia de colores para la figura 25.

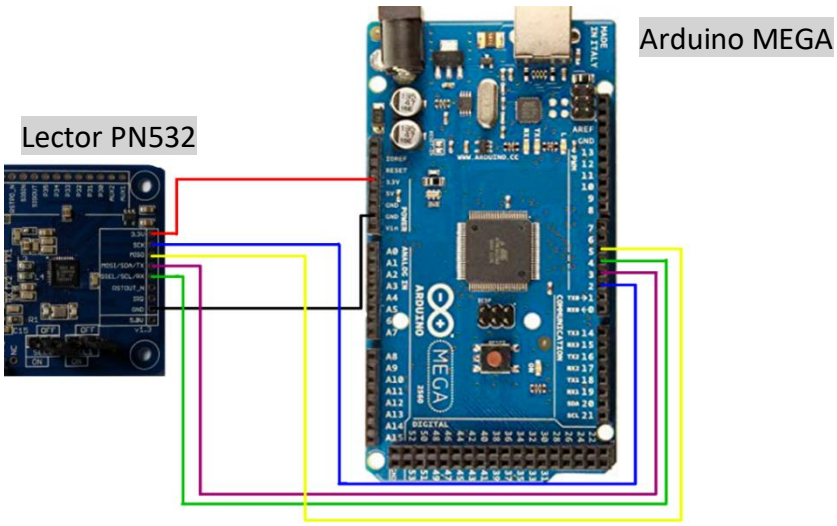

Figura 25: Conexión entre Arduino MEGA 2560 y lector PN532 Breakout Board.

Donde tenemos que:

- GND es la masa/tierra.
- SCK es señal de reloj de bus, el cual rige la velocidad que toma cada bit.
- MISO es la señal de entrada al lector, por aquí se reciben los datos.
- MOSI: es la señal de salida al lector y sirve para la transmisión de datos.
- SCL habilita hacia el que se envían los datos.

Se valida a continuación la correcta conexión entre ambos dispositivos con un pequeño código para poder detectar tarjetas físicas. En este caso se probará con la TUI de la Universidad de Cantabria (Figura 26 [28]).

Figura 26: Tarjeta TUI Unican [28].

Además de la detección, se programará la funcionalidad que permita identificarla obteniendo el UID de la misma. En la Figura 27 se tiene un diagrama del funcionamiento del programa, consistente en un ciclo infinito, es decir, una vez se ha terminado el proceso se vuelve a ejecutar para detectar más tarjetas.

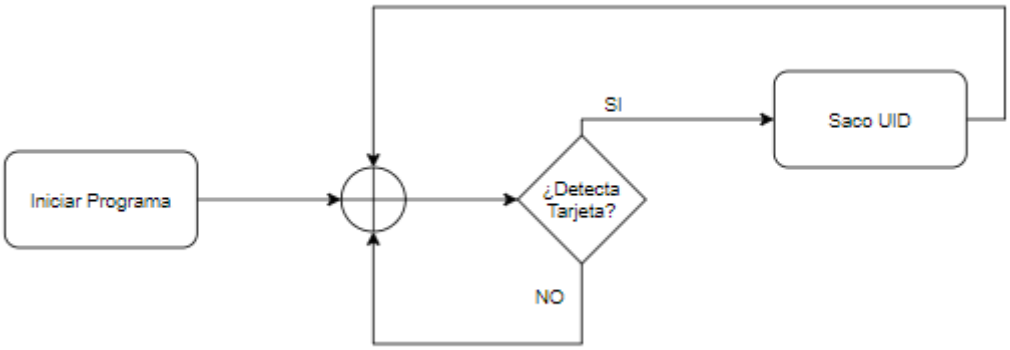

Figura 27: Diagrama del funcionamiento del código de prueba.

Al iniciar el programa se comprueba que el lector de tarjetas esté correctamente conectado, y si es así muestra un mensaje corroborándolo además de mostrar la versión de este.

Tras esta etapa inicial de configuración, se entra en el ciclo de búsqueda de tarjetas.

El programa espera una tarjeta del tipo ISO14443-A, y cuando detecta una recogerá de esta el valor de su UID, y la longitud de este. Este procedimiento se produce mediante el comando mostrado en la Figura 28.

```
success = nfc.readPassiveTargetID(PN532_MIFARE_ISO14443A, &uid[0], &uidLength);
```

Figura 28: Comando para sacar la UID de una tarjeta ISO 14443A [29].

Una vez que el equipo embebido ha detectado la tarjeta, y tiene la información deseada, se imprime por pantalla longitud del UID de la tarjeta detectada y su valor en hexadecimal. En caso contrario, si tras un tiempo de espera no se detecta una tarjeta, se informa de ello con un mensaje de 'Timed out waiting for a card'. Tras cualquiera de los dos casos vuelve a ejecutarse la función para detectar la tarjeta nuevamente.

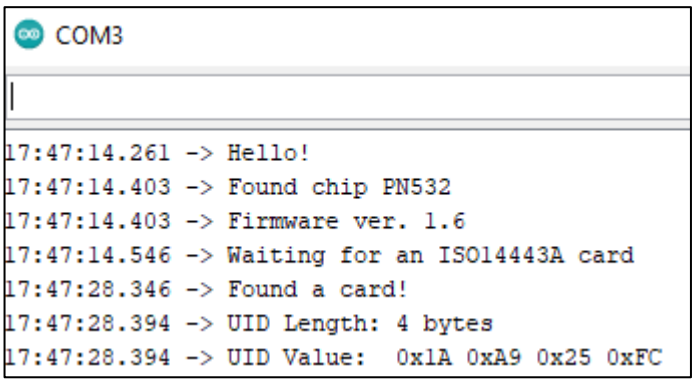

Figura 29: Ejemplo tras ejecutar el programa ISO14443a_uid y leer una tarjeta TUI.

En la Figura 29 se muestra un ejemplo de la salida por pantalla al ejecutar el programa, acercar la TUI y tener una lectura correcta de esta.

Comprendido y validado el programa inicial, se procede a evolucionar el código del mismo. Este nuevo programa busca tener cierta comunicación entre el equipo embebido y la tarjeta que se le aproxime.

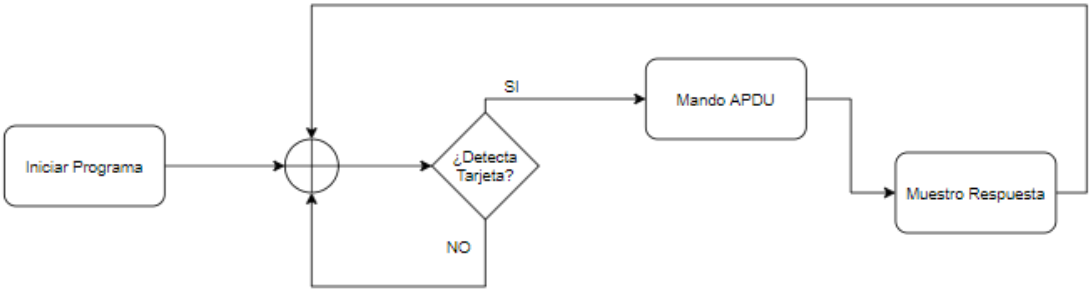

Figura 30: Diagrama de funcionamiento del programa creado a partir del ejemplo dado.

El inicio de este programa es prácticamente igual al ejemplo dado anterior como se puede ver en el diagrama de la Figura 30. Primero se comprueba que el lector este correctamente conectado, y si es así muestra un mensaje y su versión. Después se pasa a detectar la tarjeta inteligente. Además, en este caso el comando detectará un dispositivo NFC pasivo, pudiendo ser una tarjeta inteligente o un smartphone (Figura 31).

```
success = nfc.inListPassiveTarget();
```

Figura 31: Comando para detectar dispositivos NFC.

Si el programa detecta algún dispositivo se procede a enviar un comando APDU. Para esto se ha de definir un parámetro en el programa que ha de tener la estructura típica de un comando APDU como se puede ver en la Figura 32.

```
uint8_t selectApdu[] = {  0x00, /* CLA */
                          0xA4, /* INS */
                          0x04, /* P1 ) */
                          0x00, /* P2  */
                          0x08, /* Lc  */
                          0xA0, 0x00, 0x00, 0x00, 0x04, 0x00, 0x00, /* Data */
                          0x00  /* Le */
                        };

success = nfc.inDataExchange(selectApdu, sizeof(selectApdu), response, &responseLength);
```

Figura 32: Ejemplo de comando APDU a enviar, y comando para el intercambio de APDU.

Una vez definida la APDU deseada se procede a mandar el comando al dispositivo que se ha aproximado al lector. La función que desencadena el intercambio de datos se puede ver también en la Figura 32, donde tras pasar el comando APDU definido se obtiene la respuesta y su longitud como salida de la llamada.

Si el intercambio de datos se produce con éxito, se obtendrá una respuesta al comando APDU enviado, mostrando dicha respuesta por pantalla. En el caso del comando de la Figura 32, aproximando una tarjeta “Optelio Contactless”, o a la tarjeta TUI se recibiría una respuesta de 0x90 0x00, lo cual se corresponde con: "Command successfully executed, OK".

Tras todo lo anterior, el estado de ejecución del programa volverá a la fase en la que espera detectar un dispositivo nuevamente.

A partir de este intercambio de información se modificará el código para poder comunicarse con la aplicación desarrollada. Estos cambios pertinentes se explicarán en el capítulo 6, una vez ya se haya explicado el desarrollo de la aplicación.

5. Aplicación Móvil

Este capítulo del trabajo se centra en detallar el desarrollo de la aplicación Android para el smartphone. La creación de esta se dividirá en tres partes principales: la primera parte será la creación de la base de datos a la que se asociará el contenido gestionado por la misma, la segunda parte será la relacionada con el registro y login de los usuarios, y por último se adecuará la aplicación para incluir la emulación de tarjeta inteligente y poder extraer la información siguiendo este canal. Esta última parte se tratará en el siguiente capítulo, en el que se relaciona el equipo embebido con la aplicación móvil.

5.1. Escenario

Como ya se explicó en el capítulo anterior, el objetivo del trabajo es conseguir un control de acceso.

En la Figura 22 se puede ver el esquema básico del trabajo y los componentes funcionales que lo conforman.

En lo que respecta a la aplicación móvil deberá identificar al usuario, acceder a la base de datos para gestionar el perfil del mismo, y habilitar la emulación de la tarjeta para comunicarse con el equipo embebido de control de acceso.

En este sentido, el desarrollo de la operativa de la aplicación queda dividida en 4 etapas que en cierto modo coinciden con las funcionalidades y objetivos planteados para la aplicación:

- Fase 1: Registro.

  En esta fase se pretende desarrollar una ventana de registro a través de la cual los usuarios puedan registrarse e incluir su perfil en la base de datos de gestión de accesos. En el registro se pedirán datos de identificación, el número de tarjeta y, como es habitual, un usuario y una contraseña.

- Fase 2: Login.

  Para acceder a las funcionalidades de la aplicación, se deberá introducir el usuario y la contraseña indicados en el registro. Esta información se empleará para identificar y autenticar al usuario y garantizar el acceso a la información necesaria alojada en la base de datos.

- Fase 3: Correcto funcionamiento de la aplicación.

  Incluye el flujo de ventanas que se encuentran tras el login en los que se muestra la información de la cuenta y la opción de emular la tarjeta. Se comprueban que los datos son correctos y se crea una secuencia de transición entre ventanas para poder hacer la aplicación lo más amigable al usuario.

- Fase 4: Emular la Tarjeta.

  Entorno que habilita la configuración para que la aplicación emula la tarjeta del usuario. Mediante APDU presentará la información del usuario al equipo lector.

5.2. Desarrollo

La realización de esta aplicación móvil Android se lleva a cabo empleando el entorno "Android Studio" [30], que provee también un emulador de terminal móvil. Para depurar las soluciones durante su desarrollo se puede emplear el emulador o el propio teléfono físico. No obstante, debido a la lentitud y limitaciones de dicho emulador se ha optado principalmente por emplear el smartphone físico.

Para el entorno del servidor remoto, se despliegan las diferentes funcionalidades de base de datos y servidor web mediante XAMPP. Si bien se disponen de más opciones a la hora de desplegar una base de datos, se ha decidido utilizar XAMPP ya que se trata de un entorno completo y fácil de instalar y utilizar pensada para desarrolladores que están iniciando en el mundo de Apache.

5.2.1. Base de Datos

XAMPP es un entorno de despliegue de servidor que facilita la instalación de un entorno de gestión de base de datos MySQL, de un servidor web Apache, y PHP y Perl.

A través de Apache y haciendo uso de páginas PHP se habilita la conexión y el acceso HTTP al servidor de base de datos MySQL.

Como primer paso se acomete la configuración de la base de datos a través del panel de administración habilitado con phpAdmin en la dirección "http://localhost/phpmyadmin/" y mostrará en la ventana lo que se puede ver en la Figura 33.

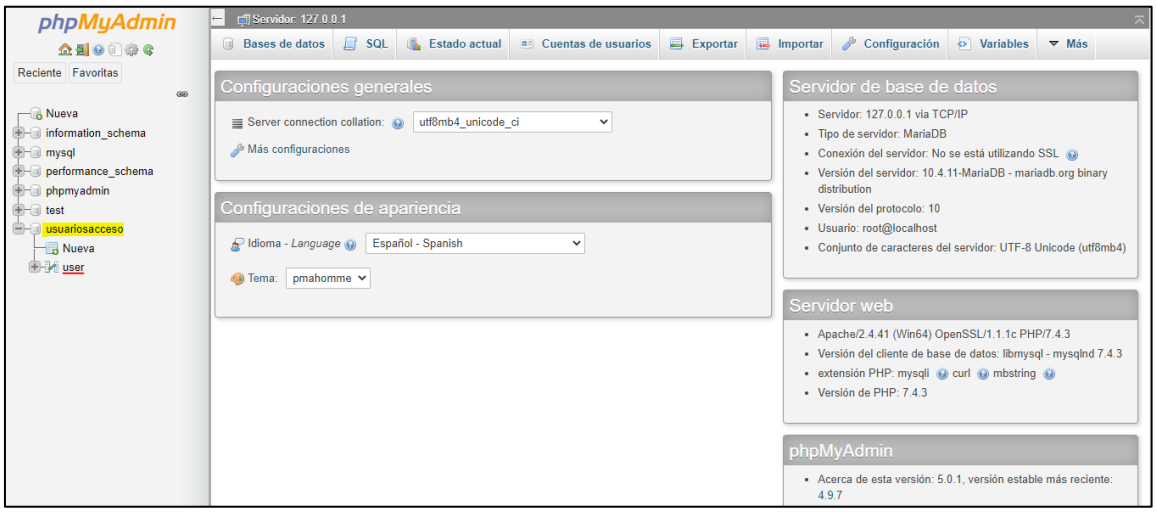

Figura 33: Ventana phpMyAdmin.

A la derecha de la imagen vemos las especificaciones del servidor de bases de datos, MySQL y del servidor web, Apache. A la izquierda de la figura observamos las bases de datos, donde se ha creado una nueva llamada “usuariosacceso” la cual se puede ver en la figura sombreada de amarillo. En dicha base de datos, se tiene una tabla llamada “user” en la cual se definen los parámetros deseados para este trabajo:

1. user_id: número de usuario registrado.
2. name: nombre del usuario que se registra.
3. lastname: apellido del usuario que se registra.
4. username: valor del nombre de usuario con el que se registra y con el que posteriormente hace el login.
5. password: contraseña que pone el usuario para acceder a su cuenta de la aplicación.
6. numTI: número de tarjeta que está emulando la aplicación, sobre el cual el servidor cotejará si el usuario tiene o no acceso.
7. date: es la fecha de caducidad de la tarjeta vinculada, para que el gestor compruebe su validez temporal.
8. access: determina si el usuario tiene acceso o no, es el gestor de la base de datos quien determina el acceso y no por el usuario. Tiene un valor por defecto de 0 significando que no tiene acceso. En caso de que tenga acceso se cambia el valor a 1.

Mediante el panel de administración phpAdmin se pueden introducir o pre-registrar usuarios. En la Figura 34 puede verse un usuario ejemplo creado en la base de datos descrita cuyo nombre es Paula, con identificador 3, y con permisos de acceso habilitados.

	user_id ▲ 1	name	lastname	username	password	numTI	date	access
☐ Editar Copiar Borrar	3	Paula	Perez	pperez1	0022	1234567890098765	03/06/2023	1

Figura 34: Usuario de ejemplo de la tabla "user" Base de Datos "usuariosacceso".

5.2.2. Aplicación de Login y Registro

Finalizada la definición de la base de datos, se crea la aplicación que accederá a ella, y los archivos PHP en el servidor que gestionan la comunicación entre ambos.

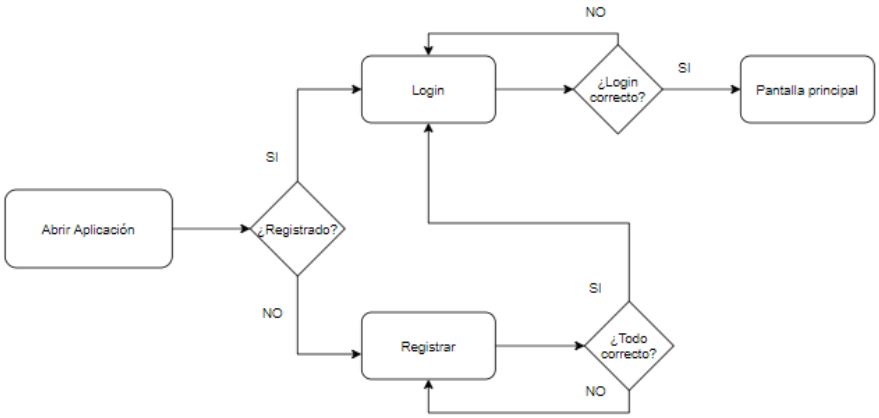

Figura 35: Diagrama de funcionamiento de la aplicación "AccessControl" para el login y el registro.

Para desarrollar la aplicación se parte del diagrama de la Figura 35, que consiste en un login básico, y se irá modificando hasta tener la aplicación con las funcionalidades deseadas.

Lo primero que hay que tener en cuenta a la hora de desarrollar la aplicación es que para registrarse y hacer el login hay que conectarse a la base de datos, por lo que se debe otorgar a la aplicación acceso a Internet. Este acceso se hace dentro del archivo "AndroidManifest.xml", declarando en él la línea de código que se tiene en la Figura 36.

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

Figura 36: Comando para permitir acceso a internet a la aplicación.

Una vez se concede el permiso para acceder a Internet, se procede a programar la aplicación en sí misma.

Partiendo de una ventana en blanco al crear la aplicación, se realizan las modificaciones necesarias para habilitar tanto el login como el registro. En cuanto a la apariencia de esta ventana se ha optado por una solución sencilla compuesta por los campos de usuario y contraseña, y dos botones, uno para hacer el login y otro para registrarse. En la Figura 37 se muestra la apariencia de la aplicación "AccessControl", donde se tienen los elementos descritos anteriormente, así como algún elemento para hacer la ventana más visual al usuario. Además, en la parte inferior de la pantalla se tiene a mano derecha dos botones, los cuales se explicarán en la fase 3 del desarrollo de la aplicación.

Figura 37: Ventana principal de la aplicación "AccessControl".

Una vez se tiene esta ventana, se pasa a crear otra nueva la cual corresponderá al registro. En la ventana de registro se definirán los campos que se necesite que rellene el usuario para el registro, en este caso, se necesitan los mismos campos que se han definido en la base de datos, sin contar el identificador del usuario ('user_id'), que se genera automáticamente al registrar un usuario en la base de datos, ni el parámetro que determina el acceso ('access'), ya que el usuario no puede determinar el mismo si puede o no tener acceso sino que esta parte se gestiona directamente en la base de datos. Por último, la ventana también necesita un botón para registrarse.

En la Figura 38 se muestra el código que habilita la transición entre dos ventanas de la aplicación, en este caso la ventana principal y la ventana de registro. Este procedimiento se replicará para en todos los casos que así se requiera en la aplicación. El formato de esta conexión se usará para la conexión entre todas las ventanas de la aplicación.

```
tv_registrar.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick (View view) {
        Intent intentRegistro = new Intent( packageContext: MainActivity.this, Regist.class);
        MainActivity.this.startActivity(intentRegistro);
    }

} );
```

Figura 38: Asociación del botón para el registro de la aplicación "AccessControl".

Una vez definida la comunicación entre ventanas, y con la base de datos creada, se procede a programar la funcionalidad de las ventanas creadas.

Como ya se comentó anteriormente, se empezará configurando el registro de los usuarios en la base de datos para poder hacer posteriormente el login. La apariencia de la ventana de registro se puede ver en la Figura 39.

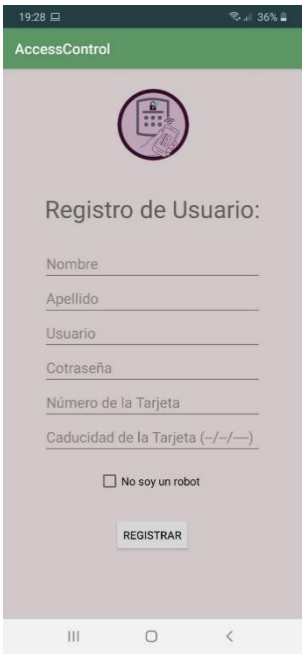

Figura 39: Ventana de Registro de la aplicación "AccessControl".

Para programar el registro se crea primero un archivo php en el servidor de base de datos que se encarga de comunicar la aplicación con la base de datos. En el archivo php creado se determina el acceso a la base de datos creada anteriormente. Tras esto se definen los parámetros de la base de datos que se encuentran en la ventana de registro y, una vez establecida la conexión a la base de datos, se define el almacenamiento de los valores introducidos por el usuario a la base de datos.

De vuelta en la aplicación, para que esta tenga acceso al archivo php creado se ha de crear una clase en la que se definirán las consultas para registrar y la comunicación con el archivo php creado, lo que va a permitir hacer toda la operación de registro. Para que la comunicación se produzca se ha de definir la URL en la que se encuentra nuestro archivo php de la base de datos local.

Una vez definida la URL, y creados todos los parámetros, peticiones y métodos, se le pasan al archivo php los valores que ha introducido el usuario para registrarse.

Con la comunicación que se tiene entre la aplicación con el archivo php, y de este con la base de datos, se tiene disponible la fase de registro. Para un registro más fiable, se programan diferentes excepciones a la hora de registrar, como número de tarjeta o usuario inválido, necesidad de rellenar todos los campos de registro, etc., para que

evitar un registro incompleto o invalido. Si todo el registro es correcto, al registrarse se informa al usuario del éxito de la operación y enlaza con la ventana principal (Figura 37)

Como paso previo a la puesta en marcha del sistema, se habilitará en el firewall del servidor las comunicaciones de entrada para poder acceder a Apache y MySQL. En el smartphone que se va a instalar la aplicación se habilita el modo desarrollador para que se le permita instalar la aplicación desde el programa. Posteriormente se instala la aplicación en el dispositivo (con un sistema operativo Android) cuyo proceso de instalación puede tardar unos minutos. Al concluir se abrirá la aplicación automáticamente y se puede utilizar con total normalidad. Si hay algún error grave en cualquier parte del código saldrá un mensaje de error y la aplicación se cerrará.

Para comprobar el registro se pincha en el botón de registrar colocado en la ventana principal y si el enlace es correcto llevará a la ventana para registrarse, una vez ahí se prueba a registrar un usuario, y se observa el correcto funcionamiento. Se prueban también todas las excepciones que pueden impedir el registro, como un usuario ya registrado, un número de tarjeta ya registrado, o que no se han rellenado todos los campos y así comprobar que es funcional en todos los posibles casos de registro.

Tras un registro operativo, se procede con la fase de login de la aplicación. Al igual que en el registro se programa un archivo php en el servidor de base de datos. En este caso, el archivo php no se utiliza para almacenar usuarios en la base de datos, sino que se hace una consulta la base de datos donde compara los valores introducidos de usuario y contraseña con los nombres de usuario y contraseñas que se encuentran ya registrados en la base de datos. Si el usuario y la contraseña son correctos el archivo devuelve a la aplicación todos los datos almacenados en la consulta, los cuales son el nombre, el apellido, usuario, contraseña, número de tarjeta, fecha de caducidad de esta, y si tiene o no acceso.

En la aplicación se programa una clase, como en el registro, donde se hace la conexión al archivo php que se ha creado para el login.La programación de esta clase es muy parecida a la del registro, definiéndose igualmente la URL de donde se encuentra el archivo php, en este caso el de login, y actualizando los parámetros para incluir los de login, es decir: usuario y contraseña. Se define el método de envío POST, y por último se recoge la respuesta al igual que con el registro.

En la ventana principal de la aplicación se configura que, si el login es correcto se recoja los datos obtenidos de la base de datos y los transfiera a la siguiente ventana en el flujo de la aplicación. En cambio, si los parámetros introducidos son erróneos no se produce el login y muestra un mensaje de error.

Tras tener la ventana asociada después de hacer el login se vuelve a cargar la aplicación en el smartphone y probar que funciona correctamente con alguno de los usuarios registrados en la base de datos.

Una vez se tiene un correcto registro y login se puede pasar a la fase 3, donde se programarán detalles para hacer a la aplicación más funcional y visual para el usuario,

además de preparar la ventana en la que se va a emular la tarjeta, y que se explicará en el próximo capítulo.

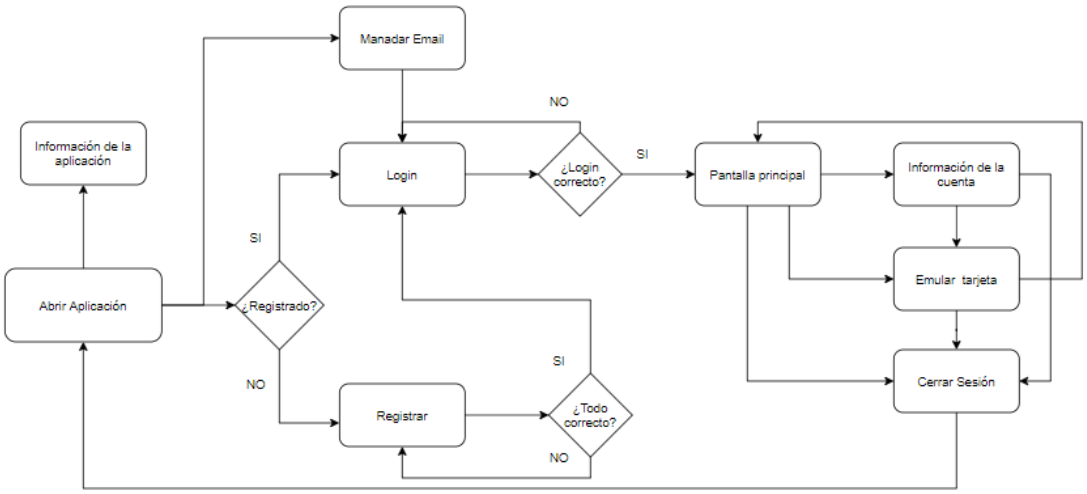

Figura 40: Diagrama de funcionamiento de la aplicación "AccessControl".

Tras conseguir la funcionalidad que se planteó en el diagrama de la Figura 35, respecto al registro y el login, se plantea un nuevo diagrama, el cual está representado en la Figura 40, donde se le dan más opciones al usuario dentro de la aplicación. Estas nuevas configuraciones se pueden diferenciar en dos grupos: el primer grupo son funcionales sin necesidad de hacer el login, y el segundo grupo son ventanas y funcionalidades que solo se puede acceder si se ha hecho un login previo.

Figura 41: Pantalla para enviar un email al gestor de la base de datos de la aplicación "AccessControl".

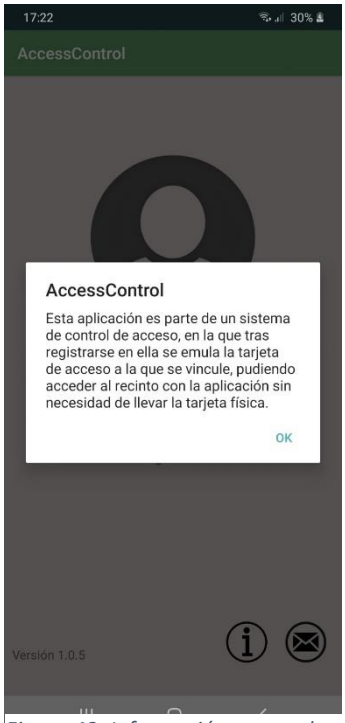

Figura 42: Información mostrada tras pulsar el botón de información de la aplicación "AccessControl".

Lo primero que se va a configurar es en la pantalla principal de la aplicación que se vio en la Figura 37. A parte de tener las opciones de login y de registrarse, se tienen dos botones en la parte inferior. Estos botones simplemente sirven como complemento a la aplicación. Por un lado, se tiene un botón de información, que simplemente al pulsar muestra un mensaje de texto en el que explica para qué es la aplicación (Figura 42).

Por otro lado, se tiene un botón que permite mandar un email al gestor del sistema en caso de que haya algún error en el registro, login, o en los datos dentro de la aplicación. El botón del email enlaza la ventana principal con una ventana creada para redactar el email, tal como se muestra en la Figura 41. Al pulsar en el botón de enviar, se abre la opción para enviar el correo y que sea el usuario el que elija desde la cuenta que quiere enviar el email, con un destinatario predefinido en la aplicación que es el gestor de la base de datos. Tras enviar el correo se vuelve a la ventana principal.

Figura 44: Pantalla principal tras hacer el login en la aplicación "AccessControl".

Figura 43: Pantalla de información de la cuenta para un usuario ejemplo de la aplicación "AccessControl".

La ventana creada para enlazar tras hacer el login dispone de un diseño básico, en el cual simplemente se muestra cierta información del usuario, y diferentes opciones a realizar mediante botones. Como se puede ver en la Figura 44 esta ventana es puramente de tránsito, sin ninguna funcionalidad. Simplemente recoge los datos de la base de datos tras el login y muestra alguno de ellos para una apariencia más vistosa. En la ventana, el usuario tiene tres opciones a realizar en función del botón que pulse: acceder a una ventana de información en donde se muestra toda la información de la cuenta, acceder a una ventana en la que se emulará la tarjeta, o bien cerrar sesión volviendo a la ventana principal.

Si el usuario decide consultar la información, la ventana que se mostrará será la que se observa en la Figura 43. Esta ventana presenta todos los parámetros que introdujo el usuario a la hora de hacer el registro en la base de datos, además de mostrar si dispone de acceso o no en ese momento. Además consta de dos botones que permiten o bien cerrar sesión o ir a la ventana en la que se emula la tarjeta directamente.

Por otra parte, si el usuario quiere emular la tarjeta y se dirige a la ventana correspondiente lo que vería sería del formato de la Figura 45.

Figura 45: Pantalla para emular la tarjeta para un usuario ejemplo de la aplicación "AccessControl".

En este capítulo no se explicará la forma de emular la tarjeta, por lo que hasta el momento esta ventana no consta de funcionalidad más allá de mostrar ciertos datos informativos. La ventana además incluye dos botones: uno para volver a la ventana de inicio tras el login de la Figura 44, y otro botón para cerrar sesión directamente.

Para comprobar que todo esté correcto en lo referido a esta fase se ha de verificar que la información de la cuenta que se recoge de la base de datos tras el login es correcta. Además, se ha de tener en cuenta que la información se transmita entre las ventanas de la aplicación de forma correcta y completa para así no perder datos en el cambio de ventanas, y que no se produzcan fallos en la aplicación.

Para corroborar que esta parte se ha configurado adecuadamente y no hay fallos, errores de datos, o algún dato descolocado, se vuelve a instalar la aplicación en el smartphone. Se realiza un login con un usuario ya creado y se navega por todas las ventanas de la aplicación comprobando que todos los datos mostrados y el flujo de ventanas son correctos.

En la Figura 46 se tiene de una forma más visual las opciones de las que dispone el usuario en el caso de la aplicación creada en este proyecto donde realiza el login (habiéndose registrado en la base de datos previamente). Si este es correcto se le llevará a una pantalla de bienvenida, donde se le da al usuario dos opciones a realizar en la aplicación: acceder a la información de la cuenta o emular la tarjeta.

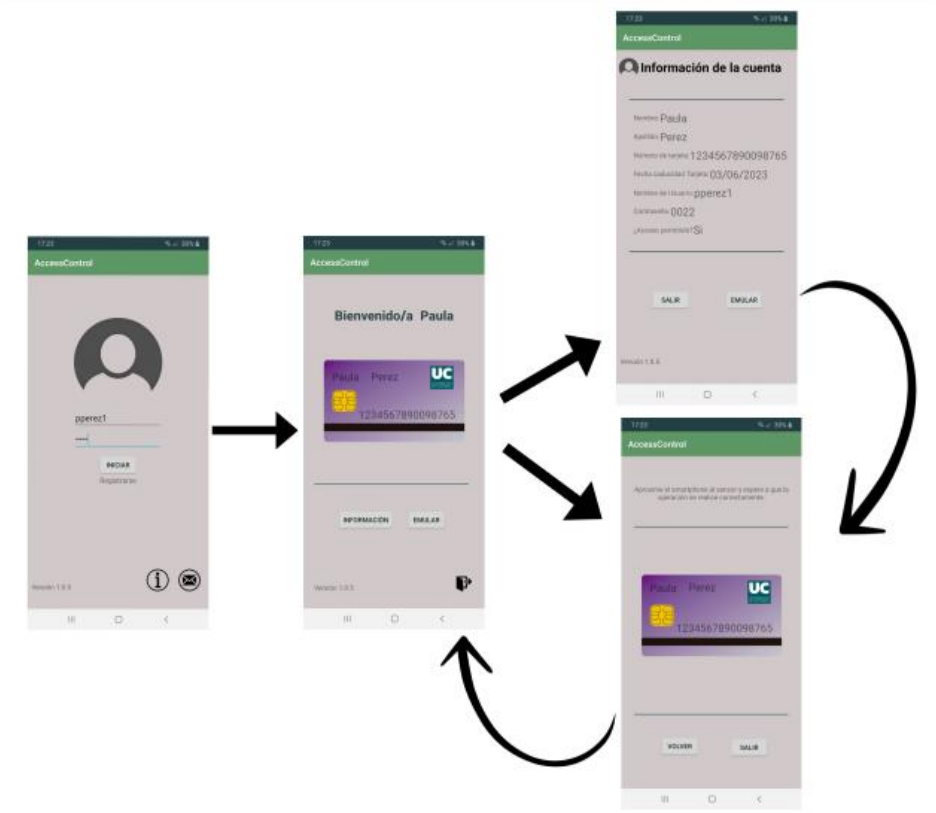

Figura 46: Esquema de funcionamiento de la aplicación "AccessControl" con un usuario ejemplo.

Una vez se tiene esta estructura de aplicación, queda modificar la ventana en la que se emulará la tarjeta para que se comporte como tal, y para que se comunique con el lector.

6. Integración

Esta parte del trabajo se va a dedicar a la programación restante para alcanzar el objetivo inicialmente previsto. Una vez se tiene una programación tanto del equipo embebido y de la aplicación móvil, se procede, a partir de dichos códigos, a completarlos con las funcionalidades previstas en el proyecto. Una representación gráfica y sencilla de lo que se quiere conseguir, y lo que se va a desarrollar en este capítulo, se puede apreciar en el icono de la aplicación diseñada (Figura 47).

Figura 47: Icono de la aplicación "AccessControl".

En la Figura 47 se puede ver como la mano del usuario aproxima el Smartphone, que está emulando la tarjeta, a un terminal de acceso y este le concede el acceso. Esto precisamente es lo que se busca en este capítulo, conseguir una comunicación entre el terminal de acceso y la tarjeta emulada en la aplicación creada.

6.1. Escenario

Como ya se ha mencionado, y se vio en la Figura 22, el objetivo del trabajo es conseguir emular una tarjeta inteligente en una aplicación móvil creada para poder hacer un control de acceso mediante el smartphone sin depender de la tarjeta física.

Hasta ahora se ha explicado el equipo embebido programado para detectar dispositivos NFC, y comunicarse con ellos, y una aplicación Android que hasta el momento incluye el registro y login.

En lo que respecta a este capítulo se va a centrar en lo que es la comunicación entre el terminal de acceso y el smartphone. Para conseguir la comunicación se han de configurar tanto la aplicación creada, como el equipo embebido programado.

6.2. Desarrollo

Para completar el desarrollo de la aplicación se procede a cumplir con la fase 4 planteada en el capítulo 5, donde se va a modificar la aplicación para conseguir emular una tarjeta NFC en ella. Estos cambios se basan en la explicación sobre este tema en el apartado teórico, y se seguirán las indicaciones aportadas por los desarrolladores de Android [15].

El primer paso que hacer es otorgar permisos a la aplicación para que pueda usar NFC, además de especificar la tecnología necesaria, como es NFC y HCE, y permitir el verificar al dispositivo si es compatible con dicha tecnología HCE. Estos permisos y declaraciones se realizan en el "ActivityManifest.xml" y en la Figura 48 se pueden observar la forma en la que se definen.

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

<uses-feature
    android:name="android.hardware.nfc.hce"
    android:required="true" />
<uses-feature
    android:name="android.hardware.nfc"
    android:required="true" />
<uses-feature
    android:name="FEATURE_NFC_HOST_CARD_EMULATION"
    android:required="true" />
```

Figura 48: Permisos y declaraciones en el Manifest de la aplicación "AccessControl".

Tras esto se sigue el diagrama de Figura 49, donde se tiene lo que se quiere conseguir con la aplicación.

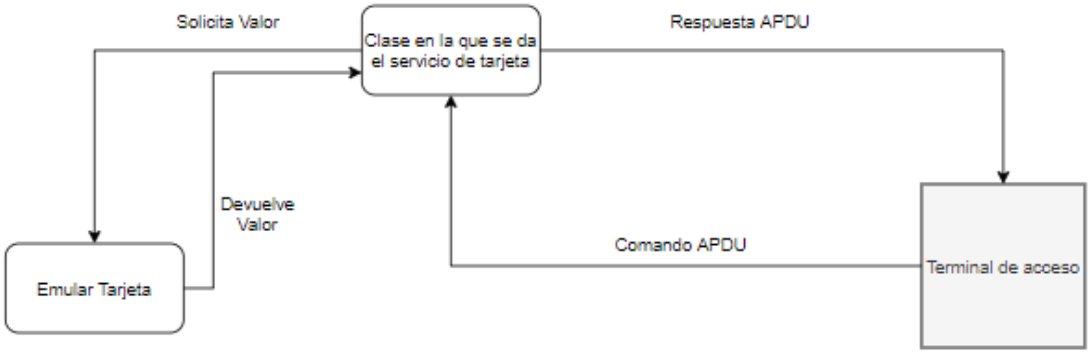

Figura 49: Diagrama de la parte de emulación de una tarjeta en la aplicación "AccessControl".

En el diagrama se observa la ventana programada en el capítulo anterior llamada "Emular Tarjeta" que se mostró en la Figura 45. Se observa también el terminal de acceso con el que se quiere conseguir la comunicación. Para establecer la comunicación entre ambos, se ha creado una clase que es la encargada de comunicarse con el terminal y conseguir la información necesaria.

A continuación, se va a explicar la clase en la que se emula la tarjeta. Esta clase se va a apoyar en una clase que Android 4.4 incluye, y que se usa como base para implementar un servicio de HCE. Esta clase crea dos métodos:

- ‘processComandApdu()’: Dicho método es llamado cada vez que la placa lectora envía una APDU hacia la aplicación del smartphone.
- ‘onDeactivated()’: Este método se llamará si el enlace NFC se ha desactivado/perdido, o bien si la comunicación requerida no se corresponde con el AID de la aplicación.

Al crear la clase que se usa como base para implementar un servicio de HCE se ha de definir en el “ActivityManifiest.xml” el permiso para implementar dicho servicio, así como el tipo de tarjeta que se va a emular en función de su grupo AID. Por último, se configuran atributos para conceder y solicitar permisos.

A la hora de especificar el grupo de AID se le determina como de la categoría “otros”, ya que como se vio en el capítulo 2 referente a la emulación de tarjetas los AID solo se dividen en dos grupos: de pago y otros. También se define el AID de la aplicación. Siguiendo los estándares descritos por los desarrolladores de Android [15] al no ser un AID registrado este ha de empezar por ‘F’. Para la aplicación se ha decidido poner el AID “F200815040”, el cual se ha de tener en cuenta posteriormente.

En la clase creada se definen los parámetros necesarios para la comunicación con el terminal de acceso. Entre estos se define de nuevo el valor del AID y la cabecera del comando APDU de selección que se espera con valores:

- Class/CLA = 00
- Instruction/INS = A4
- Parameter 1/P1 = 04
- Parameter 2/P2 = 00

En la clase se crea la estructura de comando de selección que ha de recibir la aplicación, siguiendo la estructura de comando que se ven la Figura 50. Los parámetros de la cabecera de la APDU son los definidos en un comando normal, y seguido de estos se tiene la longitud del AID de la aplicación y por último se tiene el valor del AID.

```
Formato: [CLASS | INSTRUCTION | PARAMETER 1 | PARAMETER 2 | AID LENGTH | AID]
```

Figura 50: Formato del comando APDU para comunicar la placa con la aplicación Android.

En el momento que la aplicación recibe un comando como el de la figura responde. La respuesta se compone del valor de acceso del usuario y de un ‘OK’. En cambio, si el comando no es el deseado responderá simplemente con un ‘UNKNOWN’.

El proceso en el que se recoge el valor de acceso se produce en la ventana mostrada en la Figura 45, siendo esta ventana donde se quiere emular la tarjeta.

Para corroborar que la aplicación emula una tarjeta se puede aproximar el teléfono móvil al terminal programado con anterioridad para ver si este detecta la tarjeta emulada. En el caso del primer código de ejemplo del terminal se captura un valor de UID aleatorio. En el código programado se captura la respuesta programada de la aplicación.

En este punto se tiene una aplicación que emula una tarjeta, pero no se puede comunicar con el terminal de acceso debido a que este no envía el formato de APDU que la aplicación espera.

Para conseguir la comunicación entre ambos se parte del programa creado anteriormente para el terminal de acceso.

Según [15] dentro de lo posible si se trabaja con una infraestructura de lector que se controla y se puede definir un protocolo y una secuencia de APDU propios, se ha de intentar limitar la cantidad de APDU y el tamaño de datos de estos que se deben intercambiar. De esta forma se asegura que los usuarios tengan el dispositivo sobre el lector NFC durante un periodo corto de tiempo. En nuestro caso, la configuración de la comunicación entre ambos dispositivos (equipo embebido y terminal móvil) va a ser lo más sencilla posible para una mayor velocidad de proceso.

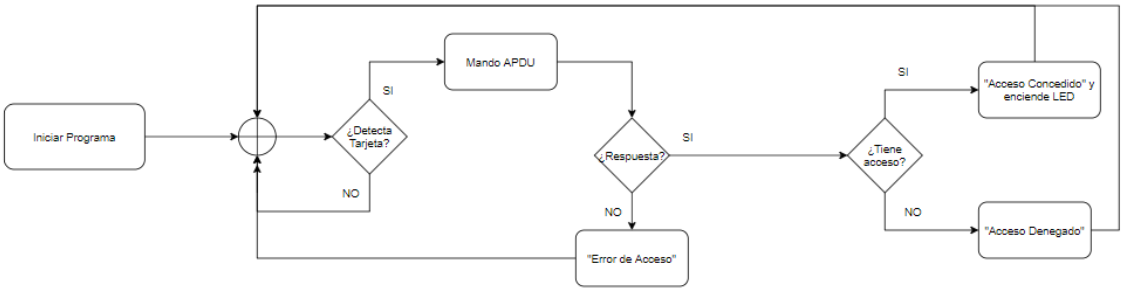

Figura 51: Diagrama de funcionamiento final del terminal de acceso programado.

Como se ve en la Figura 51, tras detectar un dispositivo NFC se procede a entablar la comunicación. Para seguir con la premisa de limitar la cantidad de APDU para que sea una conexión breve, se define que sea el propio proceso de selección de la aplicación la que devuelva la respuesta requerida. La estructura de la APDU que se transfiere es la que se ha mostrado en la Figura 50, siendo este el comando esperado por la aplicación. Este comando de selección de aplicación dispone del AID de la aplicación creada "AccessControl", y por consiguiente también de su longitud. Dicho comando queda de la forma de la Figura 52.

```
uint8_t selectApdu[] ={0x00, 0xA4, 0x04, 0x00, 0x05, 0xF2, 0x00, 0x81, 0x50, 0x40};
```

Figura 52: APDU para comunicarse directamente con la aplicación "AccessControl".

En dicha Figura 52, se puede ver que se compone igual que el APDU definido en la aplicación, con:

- CLASS = 0x00
- INS = 0xA4
- P1 = 0x04
- P2 = 0x00
- AID Length = 0x05
- AID = 0Xf2 0x00 0x81 0x50 0x40

Con el comando APDU definido se procede a realizar el intercambio de APDU entre los dispositivos al igual que se hizo en el capítulo 4.

Como la idea es que la comunicación sea lo más breve posible, la parte de la aplicación que emula la tarjeta que se comunica con la placa se ha programado de forma que si le lleva el APDU a la aplicación esta le responde con el tipo de acceso del usuario con el que se haya abierto la sesión seguido de un 'OK'.

Una vez recibida la respuesta se compara el primer valor de esta, que corresponde con el valor del acceso. Si la respuesta determina que el usuario tiene acceso, entonces se enciende el led de la placa, y se muestra un mensaje de "Acceso Concedido", en cambio, si el usuario no tiene acceso se muestra por pantalla un mensaje de "Acceso Denegado". Si la respuesta recibida no es la esperada, o bien se ha producido algún fallo a la hora de autenticar, o no se trata de un usuario de la aplicación creada se muestra por pantalla un mensaje de "Error de Acceso".

```
18:03:58.467 -> Hello!
18:03:58.612 -> Found chip PN532
18:03:58.612 -> Firmware ver. 1.6
18:03:58.654 -> Waiting for an ISO14443A Card ...
18:04:00.713 -> Tag number: 1  Acceso 1
18:04:00.902 -> Acceso Concedido
18:04:00.902 ->
18:04:02.406 -> Waiting for an ISO14443A Card ...
18:04:05.983 -> Waiting for an ISO14443A Card ...
18:04:09.605 -> Waiting for an ISO14443A Card ...
18:04:09.934 -> Tag number: 1  Acceso 2
18:04:10.075 -> Acceso Denegado
18:04:10.075 ->
18:04:11.582 -> Waiting for an ISO14443A Card ...
18:04:15.161 -> Waiting for an ISO14443A Card ...
18:04:16.100 -> Tag number: 1  Acceso 3
18:04:16.193 -> Error de Acceso
```

Figura 53: Diferentes ejemplos de Accesos programados.

Para concluir, en la Figura 53 se puede ver un ejemplo de cada posible caso. Lo primero que se ve, es el proceso inicial en el que detecta el sensor, muestra su versión, y después de esto espera a una tarjeta ISO 14443-A. Cuando detecta un dispositivo NFC pasivo, ya sea tarjeta o, en este caso, el smartphone, muestra un mensaje. En el caso de la figura se han intentado acceder de tres formas diferentes, para obtener las tres respuestas programadas:

- Acceso 1. Se ha producido acercando el smartphone con la aplicación en la ventana correspondiente y con un usuario que dispone de acceso, luego en este caso muestra el mensaje de "Acceso Concedido" y se enciende un led de la placa.
- Acceso 2. Se ha seguido el mismo procedimiento que en el acceso 1, pero en este caso se ha intentado acceder con un usuario diferente, el cual no dispone de acceso.

- Acceso 3. Se ha producido intentado acceder con una tarjeta inteligente cualquiera, denegando el acceso ya que esa tarjeta no dispone de ninguna aplicación con el AID asignado para el control de acceso. En este caso la placa detecta la tarjeta, pero no se produce la comunicación entre ambos.

7. Conclusiones y Líneas Futuras

Para finalizar la memoria de este proyecto, una vez se ha comprobado que se ha logrado los objetivos planteados al inicio, se van a exponer ciertos aspectos importantes del proyecto a modo de resumen, extrayendo las conclusiones del mismo, además, de identificar posibles líneas futuras.

7.1. Conclusiones

Se puede ubicar este proyecto dentro del área de los Sistemas de Comunicación, ya que el objetivo del proyecto se basaba en el diseño de un sistema de comunicación, y el desarrollo de cada uno de los componentes que lo componen transmitiendo datos de extremo a extremo.

Se puede decir que se han cumplido los objetivos, y lo esperado a lo largo de su desarrollo.

Con este trabajo se ha conseguido crear un sustituto de las tarjetas inteligentes de acceso convencionales, promoviendo, a la cada vez más común, ideología de tener todo en el smartphone. Además, se ha desarrollado y demostrado la viabilidad de un control de acceso que, a la seguridad de una tarjeta inteligente sin contacto convencional, sume la seguridad del login de la aplicación y el desbloqueo del smartphone en el que se instale dicha aplicación. Por otra parte, mediante la gestión en tiempo real de la base de datos, se tiene la posibilidad de cancelar u otorgar el acceso a un usuario registrado de una forma muy sencilla.

Durante este proyecto, se han programado en diferentes lenguajes, como puede ser una adaptación de C/C++ para adecuarlo a los equipos embebidos Arduino, o Java a la hora de crear la aplicación Android, los cuales no han presentado gran dificultad a la hora de empezar a programar tanto en Arduino IDE como en Android Studio. Al ser programas y lenguajes conocidos y de código abierto, ante cualquier complicación, error o duda, resulta fácil de aclararlas con ayuda tanto de la comunidad como sitios oficiales.

7.2. Líneas futuras

Este proyecto es una forma sencilla para poder sustituir la tarjeta/etiqueta que se use para el control de acceso en un recinto en una zona local. Sin embargo, este trabajo puede mejorarse hasta el punto de tener una mayor seguridad y funcionalidad.

- Una posible mejora sería modificar el código de la placa Arduino, para permitir acceso también a los usuarios que no disponen de la aplicación creada y utilizan la tarjeta/etiqueta física. Esto se podría realizar añadiendo el número de identificación de las tarjetas que se quieren permitir o programando una comunicación formal entre el terminal y los dispositivos NFC.
- Considerando modificaciones en el código de la placa Arduino, se podría también configurar acciones ante el acceso, ya que en el caso del proyecto es una indicación del funcionamiento, pero la configuración del terminal sería diferente

ya que no sería suficiente con mostrar un mensaje de acceso concedido, sino que debe despertarse una acción como la apertura de una puerta, el desbloqueo de una instalación, etc. En este caso lo que habría que programar es la comunicación con esos elementos para darle un funcionamiento practico al proyecto en la vida real.

- Respecto a la seguridad, principalmente de la aplicación, se podrían modificar varios aspectos. Por ejemplo, modificar el campo de la contraseña, en el cual no se permita cualquier tipo de contraseña, como es el caso, sino que se requiera una contraseña más segura, obligando a meter todo tipo de caracteres, una longitud mínima, etc. También se podría programar un inicio de sesión más seguro, como el que se planteó a la hora de idear el proyecto, mediante la huella dactilar del usuario registrado, el cual enlazaría su huella con el login.
- Además, se podría modificar en cierta parte la comunicación entre la aplicación y la placa, ya que en el trabajo se ha realizado una conexión directa en la que la placa solo se comunica con la aplicación, y la aplicación directamente le envía los datos para una comunicación más rápida. La modificación podría estar relacionada con el establecimiento de un proceso de autenticación mutua entre las entidades, y extender el intercambio con información cifrada.
- En este sentido, otra posible línea futura sería modificar los datos que la aplicación envía a la placa, pudiendo mandar un identificador de usuario, para tener un control de quien y cuando se accede (o intenta acceder), o indicar los recintos a los que el usuario puede acceder.
- Una posible opción de registro diferente a la programada podría ser leer el número de tarjeta y la información del usuario directamente de la tarjeta, programando de esta forma la aplicación para que funcione como lector para leer los datos de la tarjeta para registrar, y como tarjeta para comunicarse con el sistema de acceso.

Referencias

[1]. “Tarjetas Inteligentes”, Fábrica Nacional de Moneda y Timbre, [En Línea]. https://www.fnmt.es/es/productos-y-servicios/tarjetas-electronicas/tarjetas-inteligentes [08 de Diciembre del 2020].

[2]. Rankl Wolfgang, Effing Wolfgang,“Smart Card HandBook Third Edition, 2003.

[3]. Jorge Lanza, “SmartCards sin SIM”.

[4]. Javier de Gregorio Menezo, Trabajo fin de Grado, “Sistema de autenticación de dos factores basado en la tarjeta inteligente y tecnologías NFC”, 2017, [En Linea]. https://repositorio.unican.es/xmlui/bitstream/handle/10902/10509/391522.pdf?sequence=1&isAllowed=y [20 de Noviembre del 2020]

[5]. International Organization for Standardization and International Electrotechnical Commission, ISO/IEC 7816-4 – Identification Cards – Integrated circuit(s) cards with contacts – Part 4: Organization, security and commands for interchange, 2013.

[6]. International Organization for Standardization and International Electrotechnical Commission, ISO/IEC 7816-3 – Identification Cards – Integrated circuit(s) cards with contacts – Part 3: Electrical interface and transmission protocols, 2006.

[7]. International Organization for Standardization and International Electrotechnical Commission, ISO/IEC 18092 – Information technology – Telecommunications and information exchange between systems – Near Field Communication – Interface and Protocol (NFCIP-1), 2013.

[8]. European Computer Manufacturers Association, ECMA 340 – Near Field Communication Interface and Protocol (NFCIP-1), 2013.

[9]. ETSI - European Telecommunications Standards Institute, «ETSI TS 102 190 - Near Field Communication Interface and Protocol (NFCIP-1), 2004.

[10]. International Organization for Standardization, «ISO 21481 - Information technology - Telecommunications and information exchange between systems - Near Field Communication Interface and Protocol-2 (NFCIP-2), 2005.

[11]. European Computer Manufacturers Association, ECMA 352 – Near Field Communication Interface and Protocol (NFCIP-2), 2013.

[12]. European Telecommunications Standards Institute, «ETSI TS 102 312 - Near Field Communication Interface and Protocol-2 (NFCIP-2), 2004

[13]. International Organization for Standardization and International Electrotechnical Commission, ISO/IEC 14443 – Identification Cards – Contactless circuit cards – Proximity cards, 2011.

[14]. REAL ACADEMIA ESPAÑOLA: *Diccionario de la lengua española*, 23.ª ed., [versión 23.4 en línea]. <https://dle.rae.es> [13 de Mayo del 2020].

[15]. Android Developer, "Descripción general de la emulación de tarjetas basada en el host", [En Linea]. https://developer.android.com/guide/topics/connectivity/nfc/hce?hl=es-419.es [08 de Diciembre del 2020].

[16]. Arantxa Mora Perez, Trabajo fin de Máster, "Gestión de la prevención. Control de Accesos", 2016, [En Línea]. https://repositorio.upct.es/bitstream/handle/10317/5636/tfm-mor-ges.pdf?sequence=3&isAllowed=y, [08 de Diciembre del 2020]

[17]. Casas Digitales, "Control de Accesos. – Permiso de acceso mediante tarjetas o huella dactilar", [En Linea]. https://www.casasdigitales.com/control-accesos-permiso-acceso-mediante-tarjetas-huella-dactilar/ [08 de Diciembre del 2020]

[18]. Axess Iberia sl, "Lector de códigos de barras para control de acceso Smart Scanner 600", [En Linea]. https://www.archiexpo.es/prod/axess/product-154125-2013831.html [08 de Diciembre del 2020]

[19]. Salto Systems, [En Línea]. https://www.saltosystems.com/es/ [08 de Diciembre del 2020]

[20]. HID, [En Linea]. https://www.hidglobal.mx/ [08 de Diciembre del 2020]

[21]. Assa Abloy Entrance Systems, [En Linea]. https://www.assaabloyentrance.es/es/ [08 de Diciembre de 2020]

[22]. Zitelia, [En Linea]. https://www.zitelia.com/ [08 de Diciembre del 2020]

[23]. Samsumg, “Métodos de bloqueo de un dispositivo (Galaxy)”. [En Línea], https://www.samsung.com/es/support/mobile-devices/que-metodos-de-bloqueo-puedo-utilizar-en-mi-dispositivo-galaxy/ [08 de Diciembre del 2020]

[24]. Arduino, “Arduino Mega 2560”, [En linea]. https://arduino.cl/arduino-mega-2560/ [08 de Diciembre del 2020]

[25]. Daniel Lastra Lamarca, Trabajo fin de Grado, “Modelo analógico y digital en SystemC-AMS de la placa Arduino Mega 2560” https://repositorio.unican.es/xmlui/bitstream/handle/10902/7348/378290.pdf?sequence=1ç

[26]. Arduino, [En línea]. https://www.arduino.cc/ [08 de Diciembre del 2020]

[27]. Lady Ada, “Adafruit Learning System, Adafruit PN532 RFID/NFC Breakout and Shield”, [En Linea]. https://cdn-learn.adafruit.com/downloads/pdf/adafruit-pn532-rfid-nfc.pdf

[28]. Unican, “Solicitud tarjeta de estudiante TUI Universidad de Cantabria”, [En Linea]. https://web.unican.es/unidades/oficina-relaciones-internacionales/Documents/SOLICITUD%20TARJETA%20DE%20ESTUDIANTE%20TUI%20octubre%202017%20ingl%C3%A9s.pdf [08 de Diciembre del 2020]

[29]. Github, “Librería Adafruit_NFCShield_I2C, y ejemplo iso14443a_uid”, [En Linea]. https://github.com/pkourany/AdafruitPN532_Library [20 de Noviembre del 2020]

[30]. Android Developer, “Android Studio”, [En Linea]. https://developer.android.com/studio [08 de Diciembre del 2020]