Facultad de Ciencias

DESARROLLO DE UN SISTEMA DE AUTENTIFICACIÓN MÓVIL (DEVELOPMENT OF A MOBILE AUTHENTIFICATION SYSTEM)

Trabajo de Fin de Grado
para acceder al título de

GRADUADO EN INGENIERÍA INFORMÁTICA

Autor: Guzmán Gutiérrez Fernández

Director: Enrique Vallejo

Co-Director: Jose Ángel Juárez

Septiembre – 2021

Índice:

1.Introducción 8
1.1. Motivación 8
1.2. Objetivos 9
1.3. Empresa 9
1.4. Apartados 9
2.Tecnología existente 11
2.1. Tarjetas de control de acceso 11
2.1.1. Definición de tarjeta de acceso 11
2.1.2. Tipos de tarjetas 11
2.1.3. Ventajas y desventajas 12
2.2. Estándares de comunicación ISO/IEC 13
2.2.1. ISO 14443 14
2.2.2. ISO/IEC 7816-4 17
2.3. NFC 18
2.3.1. Descripción de la tecnología NFC 18
2.3.3. Modos de operación en Android 19
2.3.4. Comparación con Bluetooth 19
2.3.5. Usos de la tecnología NFC 19
2.4. Lectores 21
3. Desarrollo 23
3.1. Hardware utilizado 23
3.1.1. Lectores 23
3.1.2. Móviles 24
3.1.3. Controlador de puerta 25
3.1.4. Controladora principal 25
3.1.5. Arquitectura del sistema 25
3.2. Software utilizado 26
3.2.1. CPRStart 26
3.2.2. Android Studio 27
3.2.3. Conacwin 27
3.2.4. CA1NWH 34

3.2.5. Embarcadero RAD Studio 10.2 Tokyo 36
3.3. Funcionamiento del servicio 36
3.4. Descripción de la emulación 37
3.4.1. Manifest 37
3.4.2. Servicio NFC 38
3.4.3. Permisos 39
3.4.4. Identificación única 39
3.4.5. Cifrado de datos 40
3.5. Metodología utilizada 41
3.5.1. Desarrollo del servicio NFC 41
3.5.2. Obtención de tramas 42
3.5.3. Programación de la controladora principal 42
3.5.4. Reconocimiento de smartphone en Conacwin 43
3.5.5. Cifrado de datos a transmitir 43
4. Pruebas 44
4.1. Pruebas del servicio 44
4.2. Pruebas de obtención de tramas 45
4.3. Pruebas de la controladora principal 45
4.4. Pruebas del reconocimiento de smartphone en Conacwin 45
4.5. Pruebas del cifrado de datos 46
5. Conclusiones y fases futuras 48
5.1. Conclusiones 48
5.2. Fases futuras 48

Índice de figuras:

Figura 1: Estándares ISO/IEC usados en Android HCE. Figura tomada de [12] ...13
Figura 2: Ejemplo de señales de comunicación. Figura tomada de [6].....15
Figura 3: Diagrama de estados de la comunicación. Figura tomada de [15]. 17
Figura 4: Identificación mediante DNI electrónico. Figura tomada de [11] 20
Figura 5: Etiquetas NFC. Figura obtenida de [16].....20
Figura 6: Vinculación mediante NFC entre un móvil y un altavoz. Figura obtenida de [11].....21
Figura 7: Lector de pared CPR02.10-B. Figura obtenida de [14].....23
Figura 8: Lector USB OMNIKEY 5022 CL. Figura obtenida de [19] 24
Figura 9: Samung Galaxy J7 (2016). Figura obtenida de [17].....24
Figura 10: Samsung Galaxy XCover 4s. Figura obtenida de [18] 24
Figura 11: Controlador de puerta CP100. Figura obtenida de [1].....25
Figura 12: Esquema del sistema de control de accesos 26
Figura 13: Interfaz principal del programa CPRStart. Figura obtenida de [14]....27
Figura 14: Interfaz principal del programa Conacwin 28
Figura 15: Ventana "Definición de recintos de lectores" 29
Figura 16: Ventana "Definición de lectoras", pestaña "Controladoras" 30
Figura 17: Ventana "Definición de lectoras", pestaña "Lectoras" 31
Figura 18: Ventana "Definición de grupos de accesos" 31
Figura 19: Ventana "Definición de márgenes horarios diarios".....32
Figura 20: Ventana "Asignación/Modificación Tarjetas de Personal" 33
Figura 21: Ventana "Reconocimiento de SmartID".....34
Figura 22: Accesos de un usuario en el programa Conacwin.....34
Figura 23: Pestaña "CPR02BUS" 35
Figura 24: Pestaña "LOG" 36
Figura 25: Declaración del servicio en el archivo manifest 37
Figura 26: Archivo censurado "apduservice.xml" 39
Figura 27: Declaración de una función nativa 40
Figura 28: Función que retorna la clave usada para generar el hash 41
Figura 29: Cifrador online. Figura obtenida de [20] 46

Lista de acrónimos

AID. - Application Identifier (Identificador de aplicación)

APDU. - Application Protocol Data Unit (Unidad de datos de protocolo, nivel aplicación)

APK. - Android Application Package

ATQA. - Answer To Request, Type A

ATS. - Answer To Select

CPU. - Central Processing Unit (Unidad central de procesamiento)

DNI. - Documento Nacional de Identificación

HCE. - Host Card Emulation (Emulación de tarjeta basada en el anfitrión)

HLTA. - HALT Command, Type A

IEC. - International Electrotechnical Commission (Comisión Electrotécnica Internacional)

ISO. - International Organization for Standardization (Organización Internacional de Normalización)

LED. - Ligth Emitting Diode (Diodo emisor de luz)

NFC. - Near Field Communication (Comunicación de campo cercano)

PCD. - Proximity Coupling Device (Dispositivo de acoplamiento de proximidad)

PICC. - Proximity Card (Tarjeta de proximidad)

PPS. - Protocol and Parameter Selection

RATS. - Request for Answer To Select

REQA. - Request Command, Type A

RFID. - Radio Frequency Identification (Identificación por radiofrecuencia)

UID. - User Identifier (Identificador de usuario)

WUPA. - Wake-Up Command, Type A

Resumen

En la actualidad, tener que llevar encima una tarjeta cuya única función es la autentificación presenta inconvenientes. Es fácil de extraviar, de olvidarse, o puede ser difícil de encontrar entre todas las tarjetas que se suelen llevar, haciendo que el usuario pierda tiempo buscando la tarjeta.

La mayoría de la gente suele tener a mano casi siempre su teléfono móvil, y cada vez está cobrando más importancia debido a las prestaciones y versatilidad que ofrece. Una de las mejoras de estos últimos años es la incorporación de la conectividad NFC, que permite conectarse a otros dispositivos para realizar funciones de autenticación y pagos mediante esta tecnología.

En este proyecto se ha desarrollado un sistema de autentificación móvil basado en NFC, de forma que el usuario pueda ahorrarse llevar una tarjeta adicional, con el fin de solucionar el problema planteado anteriormente.

La aplicación desarrollada consigue implementar la misma funcionalidad de autentificación que las tarjetas a las que reemplaza. Para ello se ha desarrollado, mediante el uso de Android HCE (Host Card Emulation), un sistema de autentificación móvil basado en NFC. Acercando el smartphone al lector como si fuera una tarjeta se garantiza el acceso al usuario.

Abstract

Nowadays, having to carry a card that only has the function of authentication is not optimal. It is easy to lose, to forget, or may be difficult to find between the big number of cards that people have to carry.

Most people usually have their mobile phone close at hand, and they are becoming more and more important due to the benefits and versatility they offer. One of the improvements in recent years is the incorporation of NFC connectivity, which allows you to connect to other devices to perform authentication and payment functions using this technology.

In this project a mobile authentication system based on NFC has been developed, so that the user can save carrying an additional card, in order to solve the problem raised above.

The developed application implements the same authentication functionality as the cards it replaces. For this purpose, an autentication system based on NFC has been developed using Android HCE (Host Card Emulation). Bringing the smartphone closer to the reader as if it were a card, guarantees access to the user.

1.Introducción

Este es el apartado introductorio del proyecto, se describe la motivación (Cap. 1.1), los objetivos (Cap. 1.2), la empresa en la que ha sido realizado (Cap. 1.3) y se presenta la organización del resto del documento (Cap. 1.4).

1.1. Motivación

El control de accesos hoy en día constituye un aspecto vital en cuanto a la seguridad de una empresa o recinto. Las funciones que implementan los sistemas de control de accesos se organizan en dos bloques. Por una parte, permite identificar al usuario que entra o sale de un recinto y el momento en el que lo hace. Por otra parte, se pueden establecer que usuarios concretos del servicio tengan acceso a diferentes lugares, en diferentes franjas horarias, etc. Gracias a esto se pueden diferenciar varios grupos de personas dentro de una organización. Los sistemas de control de acceso poseen un registro con los accesos que se producen en las instalaciones. De esta forma, los eventos que ocurran se controlarán de una manera más fácil, al contar con la posibilidad de recuperar el registro de fichajes de la instalación.

A nivel industrial y empresarial, están ampliamente extendidos los sistemas de control de accesos basados en la autenticación mediante algún tipo de tarjeta o datos biométricos, como la huella dactilar o el reconocimiento facial. Adicionalmente, desde la salida de la versión de Android 4.4 existe la posibilidad de emular el comportamiento de una tarjeta de acceso en un dispositivo Android haciendo uso de la tecnología NFC. Esta tecnología permite transmitir datos mediante radiofrecuencia a una corta distancia entre dispositivos compatibles con NFC. A partir de este momento, algunos desarrolladores de controles de accesos empezaron a estudiar la posibilidad de incluir la tecnología NFC en sus sistemas y no es hasta la actualidad, cuando unas pocas marcas a nivel mundial han integrado de forma efectiva esta forma de autentificación con éxito.

Debido a la gran cantidad de funciones que ofrece hoy en día un teléfono móvil, una gran parte de la población lo lleva consigo a lo largo del día. Por este motivo, el añadir todavía más funciones al móvil es conveniente, sobre todo si nos evita la necesidad de cargar con tarjetas adicionales.

En la actualidad existe un gran número de personas que en lugar de pagar con tarjeta de crédito optan por utilizar el móvil como forma de pago. Con el desarrollo de este sistema se ofrece la posibilidad que los usuarios puedan identificarse mediante un smartphone usando NFC, del mismo modo que se utiliza con los pagos.

1.2. Objetivos

El objetivo principal de este proyecto es el desarrollo de un sistema de autentificación móvil basado en NFC host, que permita emular tarjetas de acceso RFID físicas garantizando el acceso a los usuarios que dispongan de los permisos necesarios.

Otros objetivos secundarios del proyecto serán:

- Una interfaz de usuario sencilla cómoda y entendible para los usuarios, con las notificaciones correspondientes para que el usuario se maneje con soltura.
- Que la aplicación funcione en segundo plano, es decir que el usuario haya utilizado la aplicación una vez, no sea necesario volver a entrar salvo que quiera cambiar algún parámetro.

1.3. Empresa

La realización de este proyecto ha sido desempeñada en la empresa Setelsa Security en la cual he sido contratado después de realizar en ella un periodo de prácticas curriculares. El desarrollo del sistema ha sido realizado de forma individual, en el departamento de software de la empresa.

"*Setelsa Security es una empresa tecnológica perteneciente al Grupo Setelsa. Su principal misión es ofrecer soluciones tecnológicas dentro del ámbito de los Sistemas de Seguridad aplicados principalmente a edificios e infraestructuras*"[1].

1.4. Apartados

La presente memoria está estructurada de la siguiente forma:

- Tecnología existente (Cap. 2): En este apartado se describen las principales tecnologías con las que se trabaja en el control de accesos, su funcionamiento, protocolos existentes y otros usos que tienen aparte del control de accesos.
- Desarrollo (Cap. 3): Es el apartado principal del proyecto, aquí se detallan todas las tareas realizadas y se explican en profundidad los conceptos aplicados en el desarrollo del sistema de autentificación.

- Pruebas (Cap. 4): Aquí se encuentran los resultados de las pruebas que se han realizado a lo largo del desarrollo del proyecto junto con las correcciones que he aplicado en cada caso.
- Conclusiones (Cap. 5): Por último, se exponen las conclusiones obtenidas del proyecto y posibles cambios en el sistema de cara al futuro.

2.Tecnología existente

La tecnología utilizada en un sistema de control de accesos puede variar mucho dependiendo de cómo vaya a funcionar. En este apartado se tratarán sobre todo los sistemas que emplean tarjetas de acceso, ya que son los relacionados con el proyecto, sin entrar en detalle de los controles de accesos mediante datos biométricos.

Los subapartados de esta sección son:

- Tarjetas de control de acceso (Cap. 2.1)
- Estándares de comunicación (Cap. 2.2)
- Tecnología NFC (Cap. 2.3)
- Lectores (Cap. 2.4)

2.1. Tarjetas de control de acceso

En esta sección se describe lo que es una tarjeta de control de acceso (Cap. 2.1.1), su funcionamiento, qué tipos de tarjetas de acceso existen (Cap. 2.1.2), y por último, las ventajas y desventajas de cada tarjeta.

2.1.1. Definición de tarjeta de acceso

Es una tarjeta que se comunica con un lector con el fin de conseguir acceso a un recinto. La tarjeta transmite unos datos que sirven para identificar al usuario y el lector los procesa, concediéndole o denegándole el acceso a dicho usuario.

2.1.2. Tipos de tarjetas

En la actualidad existen varios tipos de tarjetas que se usan dependiendo de las necesidades del usuario. Se pueden clasificar de varias formas dependiendo de qué especificación de la tarjeta se use para diferenciarlas. En este caso se ha hecho distinción entre si necesitan entrar en contacto con el terminal (Caps. 2.1.2.1 y 2.1.2.2), y según su función (Caps. 2.1.2.3 y 2.1.2.4).

2.1.2.1. Tarjetas de acceso de contacto

Como su nombre indica este tipo de tarjetas requiere que haya contacto entre la tarjeta y el terminal.

Dentro de las tarjetas con contacto vamos a centrarnos en describir 2 tipos:

- De banda magnética. Podemos distinguir estas tarjetas fácilmente ya que cuentan con una banda oscura. Se clasifica como tarjeta de acceso con contacto debido a que los datos de lectura o escritura se transmiten mediante inducción magnética cuando la banda entra en contacto con el lector. Actualmente se le da poco uso a esta tecnología.
- Tarjetas chip con contacto. Estas tarjetas cuentan con un chip que se encuentra en la tarjeta de plástico. Su funcionamiento es similar al de las tarjetas con banda magnética ya que los datos se intercambian cuando el chip entra en contacto con el terminal de lectura, pero cuenta con mayor capacidad y tiene mayor durabilidad.

2.1.2.2. Tarjetas de acceso sin contacto

Este tipo de tarjetas de acceso no necesitan entrar en contacto con el lector para comunicarse con él debido a que la comunicación es por radiofrecuencia. Son las más usadas en el control de accesos porque necesitan estar menos tiempo comunicando para realizar la operación.

Las tarjetas sin contacto más extendidas a nivel mundial son las tarjetas de la marca Mifare debido a su alto nivel de seguridad ya que añade extensiones propietarias. Las tarjetas Mifare implementan el estándar ISO/IEC 14443 descrito en el Cap. 2.2.1.

2.1.2.3. De memoria

Son grabadas con dispositivos de lectura/escritura, el acceso se puede proteger con una contraseña. Su comportamiento es el mismo que el de una memoria, no ejecutan ningún código del usuario.

2.1.2.4. Microprocesadas

"*Tienen características similares a una CPU (con microprocesador) que contiene un sistema operativo, una estructura de archivos y una estructura de seguridad para la tarjeta"* [2]. En este tipo de tarjetas pueden implementarse funcionalidades adicionales que se ejecutan en el procesador.

2.1.3. Comparación entre tarjetas de contacto y tarjetas sin contacto

La principal ventaja de las tarjetas de contacto es que no pueden ser leídas a distancia, lo que aumenta la privacidad y seguridad. Sin embargo, cada vez se utilizan menos hasta que finalmente caigan en desuso.

Las principales ventajas de las tarjetas sin contacto son:

- Al comunicarse por radiofrecuencia con los lectores, sin necesidad de hacer contacto físico con ellos, son muy útiles en los procesos en que el tiempo de transacción debe ser corto.
- Su durabilidad es mayor que las tarjetas de contacto debido a que no se desgasta por rozamiento ni pierde la codificación como en el caso de las tarjetas de banda magnética [3].
- En cuanto al tema de higiene, también es una ventaja el que no haya que establecer contacto.

Además de lo anterior, las tarjetas Mifare siempre trabajan bajo los estándares ISO/IEC, lo cual es una ventaja respecto a las tarjetas propietarias que no sigan estos estándares debido a que *"Las tarjetas Mifare al trabajar bajo los estándares ISO/IEC 14443 y/o ISO/IEC 15693 permiten interoperabilidad"* [4].

2.2. Estándares de comunicación ISO/IEC

"*ISO e IEC forman el sistema especializado para la normalización mundial. Los organismos nacionales miembros de ISO e IEC participan en el desarrollo de las Normas Internacionales a través de comités técnicos establecidos por la organización respectiva, para atender campos particulares de la actividad técnica. Los comités técnicos de ISO e IEC colaboran en campos de interés mutuo*" [5].

ISO e IEC es una organización que establece comités técnicos a través de los cuales se desarrollan las Normas Internacionales

Este apartado trata sobre los principales estándares existentes relacionados con el control de accesos mediante el uso de HCE, se muestran en la Figura 1. Los apartados que se describen a continuación con mayor detalle son ISO14443-2 e ISO14443-3 debido a que tienen mayor relevancia en el proyecto.

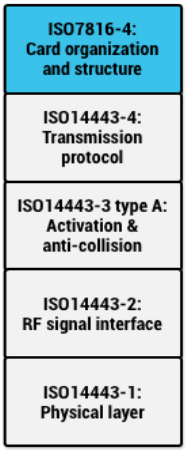

Figura 1: Estándares ISO/IEC usados en Android HCE. Figura tomada de [12]

2.2.1. ISO 14443

Este estándar consta de cuatro partes, está relacionado con las tarjetas y dispositivos de seguridad de identificación personal electrónicas. Se describe de forma general cada una de las partes, sin embargo, solo se detallarán los que tengan relación con la emulación. Existen dos tipos de tarjetas bajo estos estándares, las de tipo A y B, su principal diferencia son los métodos de modulación y codificación (Cap. 2.2.1.2).

2.2.1.1. ISO 14443-1

En esta parte del estándar se describen las características físicas de la tarjeta (material, dimensiones, etc.).

A la hora de la emulación no resulta relevante este apartado, ya que a pesar de que cada móvil posee unas características físicas diferentes, la emulación es posible siempre y cuando posean la tecnología NFC (Cap. 2.3).

2.2.1.2. ISO 14443-2

Esta parte del estándar se centra en la potencia de la señal de radiofrecuencia y la interfaz de señal.

Los siguientes apartados describen el diálogo inicial para las tarjetas de proximidad, la potencia de transferencia, la interfaz de señal y la Interfaz de señal de comunicación.

2.2.1.2.1. Diálogo inicial para las tarjetas de proximidad

Se establece que el diálogo inicial entre la tarjeta y el dispositivo emparejado deberá ser de la siguiente forma [6]:

- Activación de la tarjeta mediante radiofrecuencia por el dispositivo emparejado.
- La tarjeta deberá esperar un comando del dispositivo.
- Transmisión del comando por el dispositivo.
- Transmisión de la respuesta por la tarjeta.

2.2.1.2.2. Potencia de transferencia

Se crea un campo mediante radiofrecuencia entre los dispositivos que debe ser modulado para la comunicación [6].

- Frecuencia: La frecuencia del campo de radiofrecuencia será 13,56 MHz pudiendo oscilar 7 kHz respecto a este valor.
- Campo operativo: El valor mínimo será 1,5 A/m rms, el máximo 7,5 A/m rms. Operará continuamente entre estos valores.

2.2.1.2.3. Interfaz de señal

Se describen dos interfaces de señal, para los tipos A y B. Solo podrá haber una activa al mismo tiempo. En la siguiente imagen se puede ver un ejemplo de señales de comunicación para el tipo A y B Figura 2, arriba del dispositivo a la tarjeta y abajo de la tarjeta al dispositivo:

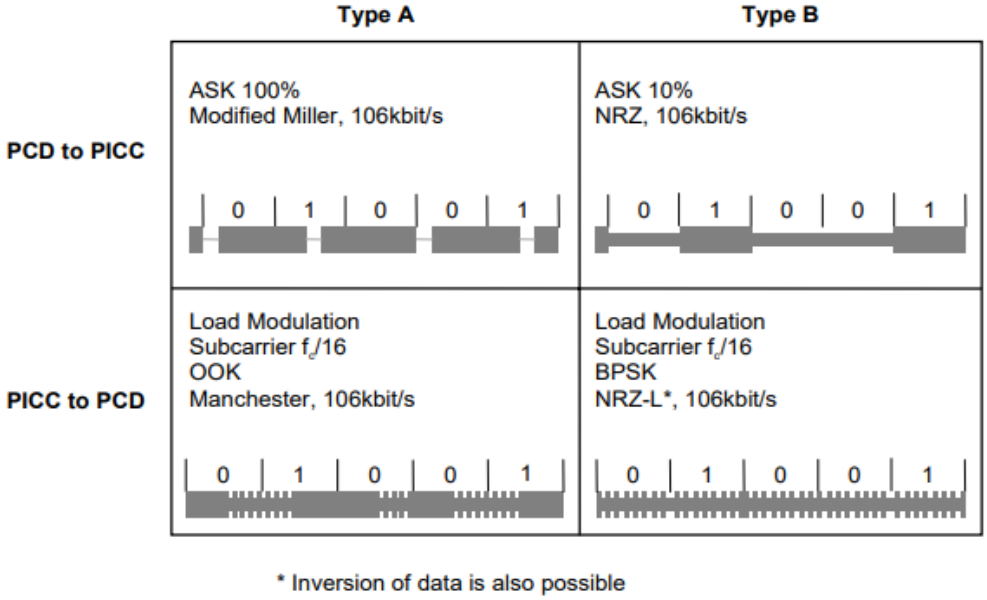

Figura 2: Ejemplo de señales de comunicación. Figura tomada de [6]

2.2.1.2.4. Interfaz de señal de comunicación

El estándar describe los siguientes campos respecto a la interfaz de señal de comunicación:

- Tasa de bits
- Modulación
- Representación y codificación de bits

De la tarjeta al dispositivo los campos que describe el estándar son:

- Tasa de bits
- Modulación de carga

- Subportadora
- Modulación de la subportadora
- Representación y codificación de bits

2.2.1.3. ISO 14443-3

Se refiere a la inicialización de la comunicación y al sistema de anticolisión. El sistema de anticolisión aumenta la eficiencia ya que su función es evitar que varios dispositivos interactúen a la vez con un mismo terminal, lo cual añadiría retardos a la comunicación debido a la colisión de señales.

Para detectar una tarjeta dentro del rango operativo, el dispositivo lector deberá enviar repetidamente comandos Request (Polling) [7].

Al no usar tarjetas en el sistema de autenticación desarrollado, a continuación, se describe cómo ocurren estos aspectos al emular una tarjeta con un móvil Android.

Al comenzar la comunicación el móvil presenta su identificador de usuario (UID), que es aleatorio (al contrario que con las tarjetas de acceso), motivo por el cual no se ha de utilizar como forma de autenticación. Después el lector puede seleccionar el dispositivo móvil utilizando un comando SELECT. La respuesta al comando SELECT que da el móvil tiene el sexto bit establecido a 0x20. Por tanto los dispositivos que se vayan a comunicar con un dispositivo móvil deberán buscar el sexto bit [12].

2.2.1.4. ISO 14443-4

En este apartado del estándar se establece el protocolo de transmisión. Es suficiente con tener claro un par de aspectos de este protocolo para el desarrollo del sistema de autentificación:

- La comunicación establecida funciona en modo half-dúplex, es decir, que la información solo puede ser transmitida en un sentido al mismo tiempo. El lector envía primero una señal, y espera una respuesta.
- El diagrama de estados de la comunicación está representado en la siguiente imagen Figura 3:

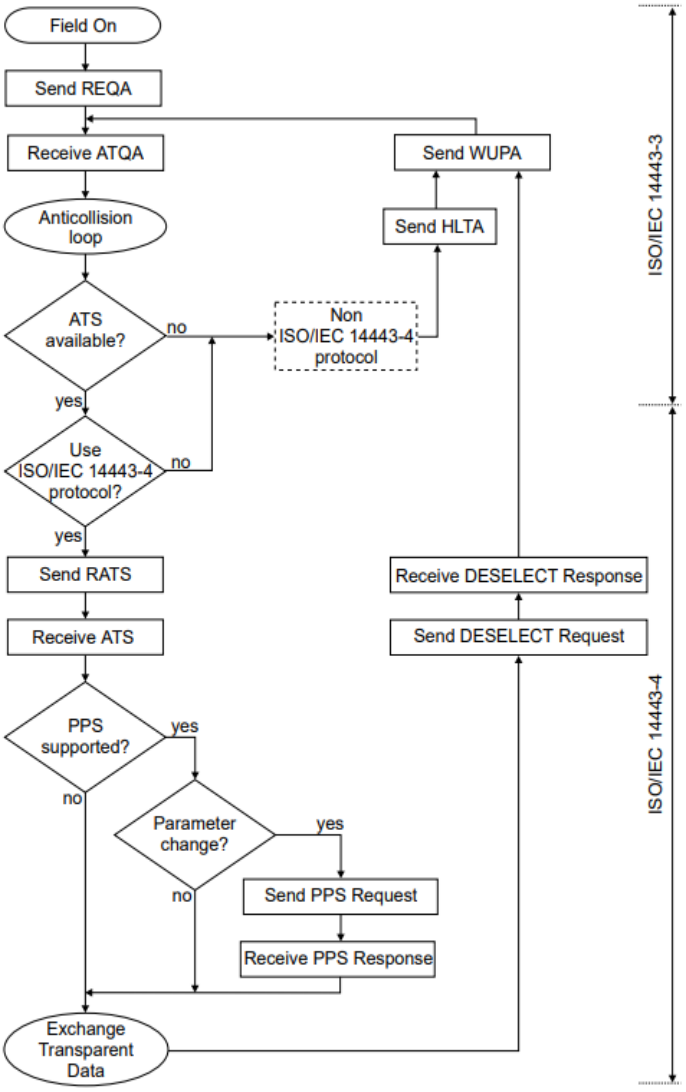

Figura 3: Diagrama de estados de la comunicación. Figura tomada de [15].

2.2.2. ISO/IEC 7816-4

Esta parte del estándar trata sobre la especificación del protocolo de transmisión y sus secuencias de activación y desactivación.

Dentro de los apartados que expone este estándar, lo útil relacionado con el desarrollo del sistema es lo referente a las APDU y el AID, que son las partes encargadas de accionar el servicio.

Una APDU (Unidad de Datos de Protocolo de Aplicación) es una trama con un formato definido que usaremos para activar el servicio.

El formato de la trama APDU consta de los siguientes campos:

- CLA: Byte de clase
- INS: Byte de instancia
- P1, P2: Bytes de parámetros
- LEN: Byte de longitud del campo de datos
- DATA: Bytes de datos

El AID es el Identificador de Aplicación de nuestro sistema, lo que hace es despertar el servicio cuando recibe una trama APDU que envíe el AID correspondiente al servicio.

Se explica en detalle cómo se han implementado en el desarrollo del sistema en el apartado "concepto de AID y APDU" (Cap. 3.4.2.3).

2.3. NFC

El sistema de control de acceso utiliza la tecnología NFC, por tanto, se va a explicar qué es esta tecnología, su origen y comparación con otras tecnologías inalámbricas, además de varios usos que se le pueden dar hoy en día.

2.3.1. Descripción de la tecnología NFC

Las siglas NFC vienen de Near-Field Communication. Es una tecnología de comunicación inalámbrica de corto alcance y alta frecuencia creada para el intercambio de datos entre dos dispositivos cercanos, principalmente teléfonos móviles. Los estándares están basados en ISO 14443 (Cap. 2.2.1) y la Identificación por radiofrecuencia. Están definidos por el Foro NFC (NFC Forum).

En teoría su radio de acción es de 10 centímetros como máximo, sin embargo, en la práctica este radio disminuye considerablemente, en algunos casos llegando hasta 2 centímetros como máximo.

Esta tecnología permite la comunicación de los teléfonos móviles que dispongan de ella con otros dispositivos compatibles con NFC.

"El NFC en los móviles se obtiene mediante un componente físico, con una bobina de inducción electromagnética" [8]. Por tanto para poder usar el NFC el dispositivo móvil debe tenerle incorporado de fábrica. Aun así, hoy en día la gran mayoría de los móviles cuenta con NFC.

2.3.3. Modos de operación en Android

Los tres modos de operación que describe la documentación de Android sobre NFC son:

- *"Modo de lectura/escritura, que permite que el dispositivo NFC lea o escriba etiquetas y calcomanías pasivas de NFC.*
- *Modo P2P, que permite que el dispositivo NFC intercambie datos con otros pares NFC; Android Beam utiliza este modo de operación.*
- *Modo de emulación de la tarjeta, que permite que el dispositivo NFC actúe como una tarjeta RFID. Un lector externo de NFC puede acceder a la tarjeta emulada, como una terminal de punto de venta NFC"* [9].

2.3.4. Comparación con Bluetooth

Tanto NFC como Bluetooth son tecnologías para la comunicación inalámbrica, por ello en este apartado se realiza una comparación entre ambas indicando sus ventajas y desventajas.

"El NFC tiene un tiempo de reacción realmente bajo, las transacciones con NFC no llegan a 1 segundo" [10]. Su uso de energía es muy bajo, prácticamente no consume cuando lo utilizamos. La distancia a la que funciona es muy cercana. Su velocidad para transferir archivos es muy lenta.

La velocidad para transferir archivos del Bluetooth es mucho más rápida, pero "*su tiempo para conectarse es relativamente alto, al menos 6 segundos. La distancia a la que se puede conectar es mayor que con NFC, llegando hasta los 10 metros"* [10].

Por estas características los usos que se le da a estas tecnologías son distintos. El Bluetooth se utiliza para conectar dispositivos entre ellos, mientras que el NFC principalmente se usa para recibir información o emitir pagos.

2.3.5. Usos de la tecnología NFC

A continuación, se muestran algunos de los usos más comunes que tiene hoy en día la tecnología NFC.

2.3.5.2. Identificación en cajeros

Si el móvil está correctamente configurado no es necesario utilizar la tarjeta para identificarse en el cajero. Con acercarlo al lector ya se procede con la identificación normal como si se hubiera introducido la tarjeta física en la ranura permitiendo sacar o ingresar dinero.

2.3.5.3. Pagos móviles

Este es el uso más conocido del NFC y el más usado. Es similar a llevar la tarjeta de crédito o débito. Se implementa mediante el modo de emulación de tarjeta.

2.3.5.4. Identificación personal

"Los móviles Android pueden ***usar el NFC de su móvil como un lector de DNI electrónico****, pudiendo identificarse con su DNI electrónico desde* ***algunas de las aplicaciones compatibles****"* [11]. En la imagen Figura 4 se puede ver un ejemplo de esta identificación.

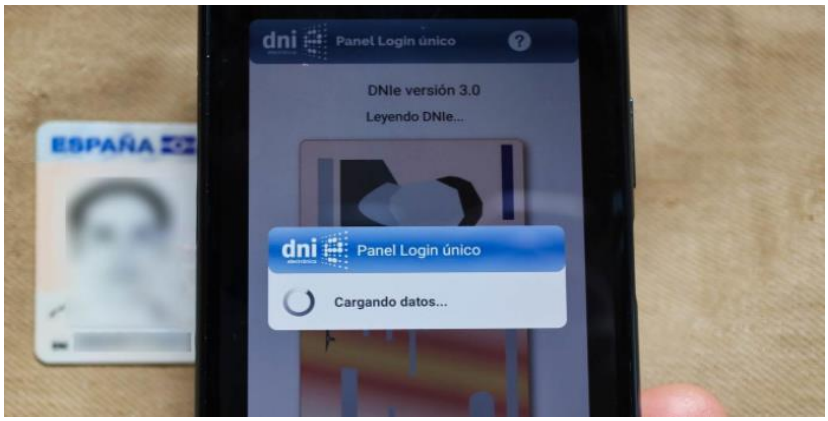

Figura 4: Identificación mediante DNI electrónico. Figura tomada de [11]

2.3.5.4. Automatización de acciones

Se pueden automatizar acciones gracias a las etiquetas NFC como las de la imagen Figura 5. "*Se puede configurar que el móvil realice una acción determinada al leer una etiqueta"* [10].

Figura 5: Etiquetas NFC. Figura obtenida de [16]

2.3.5.5. Sincronización de dispositivos

Actualmente existen algunos dispositivos como altavoces Figura 6, cascos o cámaras con NFC que al acercar el móvil al dispositivo se sincronizan en menos de un segundo vinculando el móvil por Wi-Fi o Bluetooth al dispositivo.

Figura 6: Vinculación mediante NFC entre un móvil y un altavoz. Figura obtenida de [11]

2.3.5.6. Emulación de tarjetas basada en Android

A partir de la versión 4.4 de Android, se introduce la emulación de tarjetas. Con esto se permite que una aplicación Android se comunique con un lector NFC de la misma forma que lo haría una tarjeta. Se detalla el funcionamiento más adelante en el apartado “Descripción de la emulación” (Cap. 3.4).

2.4. Lectores

Los lectores son dispositivos que se comunican con los dispositivos o tarjetas mediante el uso de ondas de radio. No todos los lectores son compatibles con los mismos estándares de las normas ISO/IEC o NFC. Normalmente se distinguen dos tipos de lectores:

- Fijos. Son los lectores que se montan en cualquier lugar estático, como paredes o portales.
- Móviles. Son los lectores portátiles que se pueden mover con facilidad, por ejemplo, un datáfono para cobrar.

Las antenas son un elemento necesario porque convierte la señal del lector en ondas. Sin antena el lector no puede enviar y recibir señales correctamente.

La antena puede estar integrada en el lector o ser externa a él.

Una vez obtenida la información, el lector transfiere dicha información a un sistema informático principal, donde puede ser almacenada en una base de datos y analizada posteriormente [13].

3. Desarrollo

En este apartado se encuentra toda la información correspondiente a la fase de desarrollo del sistema de control de accesos. Se explican las herramientas utilizadas en esta fase, tanto a nivel de hardware como de software y el funcionamiento detallado del sistema.

3.1. Hardware utilizado

El hardware utilizado en el desarrollo han sido lectores (Cap. 3.1.1), teléfonos móviles (Cap. 3.1.2), un controlador de puerta (Cap. 3.1.3) y una controladora (Cap. 3.1.4), se detallan sus especificaciones en los apartados a continuación junto con un esquema del sistema (Cap. 3.1.5).

3.1.1. Lectores

En el sistema se emplean 2 tipos de lectores. El lector de pared es el que se utiliza para obtener acceso al recinto. El lector USB lo usará el administrador del sistema para asignar los datos de acceso de la tarjeta de los usuarios a sus smartphones.

3.1.1.1. Lector de pared

Se ha empleado el CPR02.10-B Figura 7 de la empresa FQ Ingeniería Electrónica [14].

Figura 7: Lector de pared CPR02.10-B. Figura obtenida de [14]

Este lector puede realizar la lectura/grabación de chips basados en las normas ISO14443 A y B, soportando la parte 4 y ofreciendo compatibilidad con NFC. Además de poder leer y grabar los chips englobados en la norma ISO15693.

Revisando los modelos de lectores de los que disponía la empresa se escogió este por las características descritas anteriormente.

3.1.1.2. Lector USB

El lector USB utilizado en este proyecto es el modelo OMNIKEY 5022 CL Figura 8. Es empleado en la fase de reconocimiento de SmartID (Cap. 3.2.3.6) con acceso a los usuarios del sistema de control de accesos.

Figura 8: Lector USB OMNIKEY 5022 CL. Figura obtenida de [19]

3.1.2. Móviles

Los móviles utilizados para el desarrollo han sido un Samung Galaxy J7 (2016) Figura 9 y un Samsung Galaxy XCover 4s Figura 10.

Figura 9: Samung Galaxy J7 (2016). Figura obtenida de [17]

Figura 10: Samsung Galaxy XCover 4s. Figura obtenida de [18]

Ambos disponen de NFC y de una versión de Android superior a la 4.4. que es la necesaria para poder usar Android HCE.

3.1.3. Controlador de puerta

Un controlador de puerta es el encargado de gestionar la apertura de la puerta correspondiente dentro del sistema de control de accesos. Para el desarrollo del sistema se ha utilizado el controlador CP100, propietario de Setelsa Security Figura 11. Se ha elegido este controlador simplemente porque es con el que trabaja la empresa Setelsa Security.

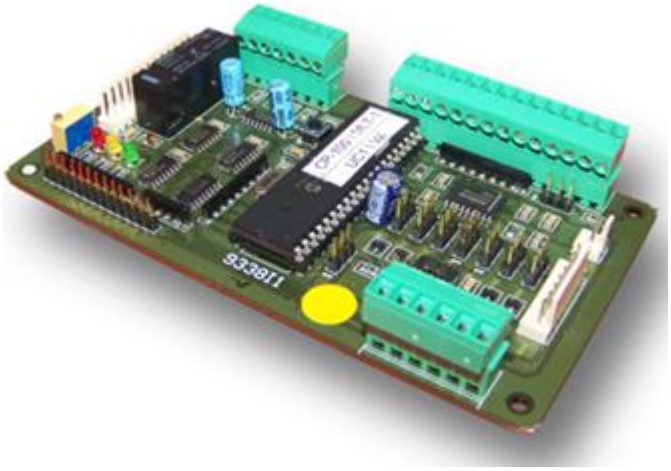

Figura 11: Controlador de puerta CP100. Figura obtenida de [1]

3.1.4. Controladora principal

La controladora principal está conectada a los dispositivos del sistema de control de accesos, es la encargada de gestionar el sistema. En este proyecto, se ha utilizado el propio PC ejecutando el programa de la controladora a modo de pruebas de laboratorio para el desarrollo. A la hora de implementar el sistema en un entorno real el programa que se ejecuta en el PC se ejecutará en una controladora L512.

3.1.5. Arquitectura del sistema

Con el fin de aclarar las conexiones del sistema y las relaciones entre sus componentes, en la Figura 12 se muestra un esquema del sistema completo de control de accesos.

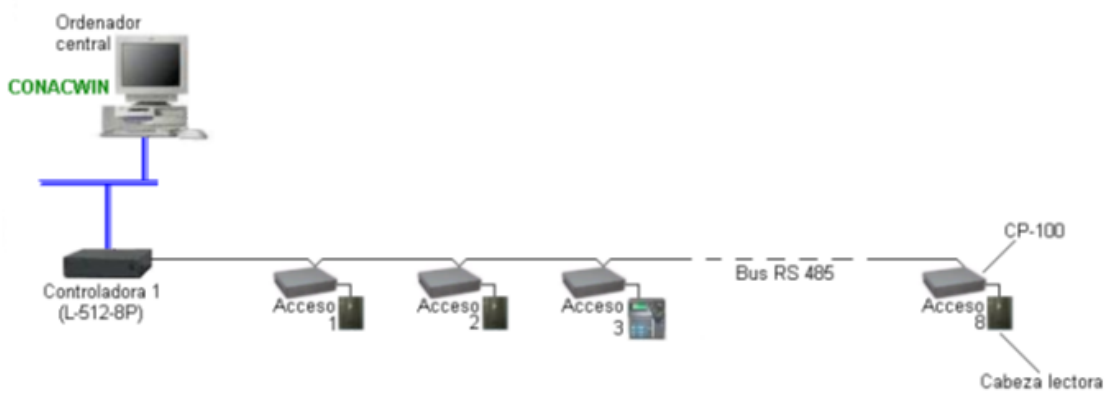

Figura 12: Esquema del sistema de control de accesos

3.2. Software utilizado

A continuación, se describirán los programas utilizados en el desarrollo del sistema, mencionando sus principales características y funcionamiento.

3.2.1. CPRStart

Es el programa del fabricante del lector. Este programa permite:

- Comprobar la comunicación entre el lector y los dispositivos.
- Ver y modificar la configuración del lector, cambiando parámetros como el color del LED según el estado (conectado, desconectado, transmitiendo...), si emite un sonido cuando recibe alguna trama, etc.
- Cambiar la versión del firmware ya que no todas usan el mismo estándar de comunicación
- Enviar comandos de comunicación tanto manualmente como en un script automatizado y ver las tramas enviadas en el proceso

En la imagen Figura 13 se puede apreciar la interfaz principal del programa. Lo relevante en el desarrollo del sistema fue la ventana “Commands”. Esta ventana es la utilizada para enviar comandos manualmente, en la ventana inferior aparecen en formato hexadecimal las tramas que envía y recibe el lector.

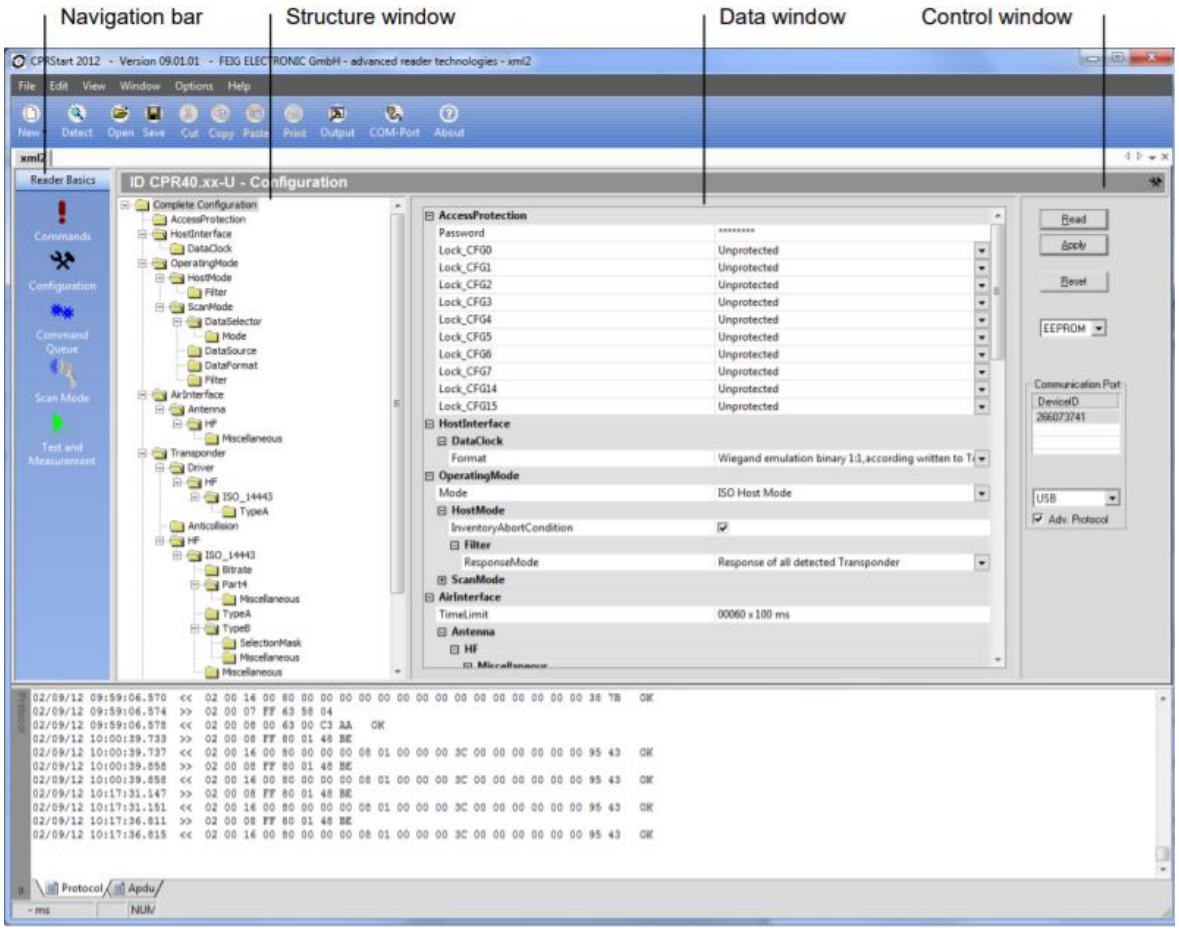

Figura 13: Interfaz principal del programa CPRStart. Figura obtenida de [14]

3.2.2. Android Studio

En este programa se ha realizado la implementación del servicio Android HCE. Con él se pueden desarrollar APKs (Android Application Package), el paquete que usa el sistema operativo Android para sus aplicaciones. Estas APKs se pueden desplegar en un emulador integrado en Android Studio, o en un dispositivo físico. En este caso, se ha realizado en un dispositivo físico ya que es necesario usar el NFC del smartphone para las pruebas correspondientes.

Android Studio permite usar los lenguajes de programación Kotlin y Java para programar las aplicaciones; para el desarrollo se ha utilizado Java.

3.2.3. Conacwin

Es una solución de Control de Accesos propietaria de la empresa Setelsa Security, se ejecuta en el pc del administrador del sistema y cuenta con una gran variedad de funciones. Las utilizadas en este proyecto son definición de recintos de lectores (Cap. 3.2.3.2), definición de cabezas lectoras (Cap. 3.2.3.3), definición de grupos de accesos (Cap. 3.2.3.4), definición de márgenes horarios diarios (Cap. 3.2.3.5), asignación/modificación tarjetas de personal (Cap. 3.2.3.6).

Adicionalmente, se ha integrado el reconocimiento de smartphones y su asociación con usuarios del control de accesos ya dados de alta en el sistema en la función reconocimiento de SmartID (Cap. 3.2.3.7). Se muestra también su interfaz principal (Cap. 3.2.3.1).

La descripción de los siguientes apartados sigue el orden en el que se han utilizado las funciones a la hora de crear este proyecto.

3.2.3.1. Interfaz principal

La interfaz principal del programa Figura 14 cuenta con pestañas para seleccionar los distintos recintos asignados, aparece el estado de los lectores y las controladoras del recinto seleccionado, además de un registro de los eventos que ocurren en el sistema.

En la parte de arriba cuenta con menús desplegables que permiten acceder a todas las funcionalidades del programa. Justo debajo de los menús desplegables se encuentran algunas de las funcionalidades más usadas junto con un icono, esto es útil para acceder de forma más rápida que desde el menú desplegable.

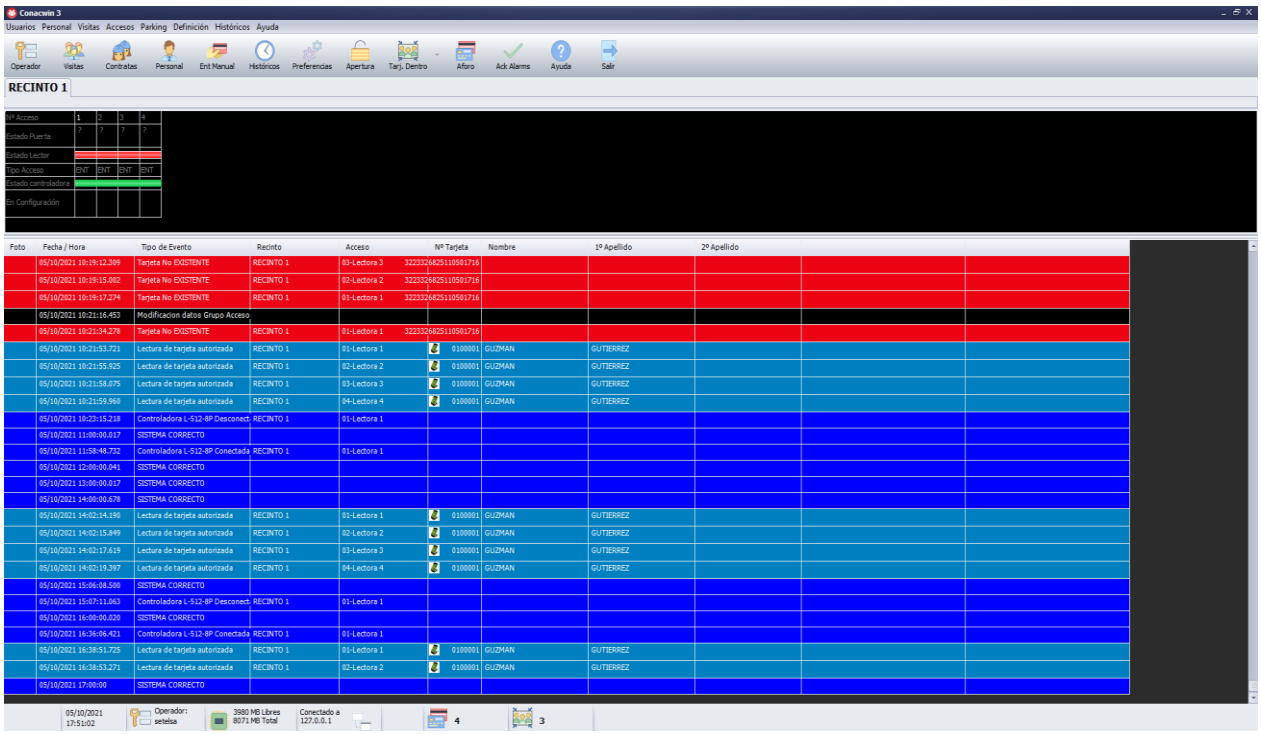

Figura 14: Interfaz principal del programa Conacwin

3.2.3.2. Definición de recintos de lectores

Esta ventana se utiliza para gestionar los recintos, donde cada recinto es una agrupación de lectores. Cada recinto se diferencia del resto por su código de recinto. Tiene asociado un texto que permite añadir una descripción breve para facilitar su identificación. Opcionalmente se puede asignar un recinto padre para establecer una jerarquía (una planta dentro de un edificio), y el número máximo de lectores que puede contener el recinto. Es necesario que exista al menos un recinto antes de asignar los lectores. Esto se debe a que en la asignación del lector se ha de indicar a qué recinto pertenece, como se describe en el Cap. 3.2.3.2.

En la siguiente imagen Figura 15 se muestra la interfaz de la ventana “Definición de recintos de lectores”.

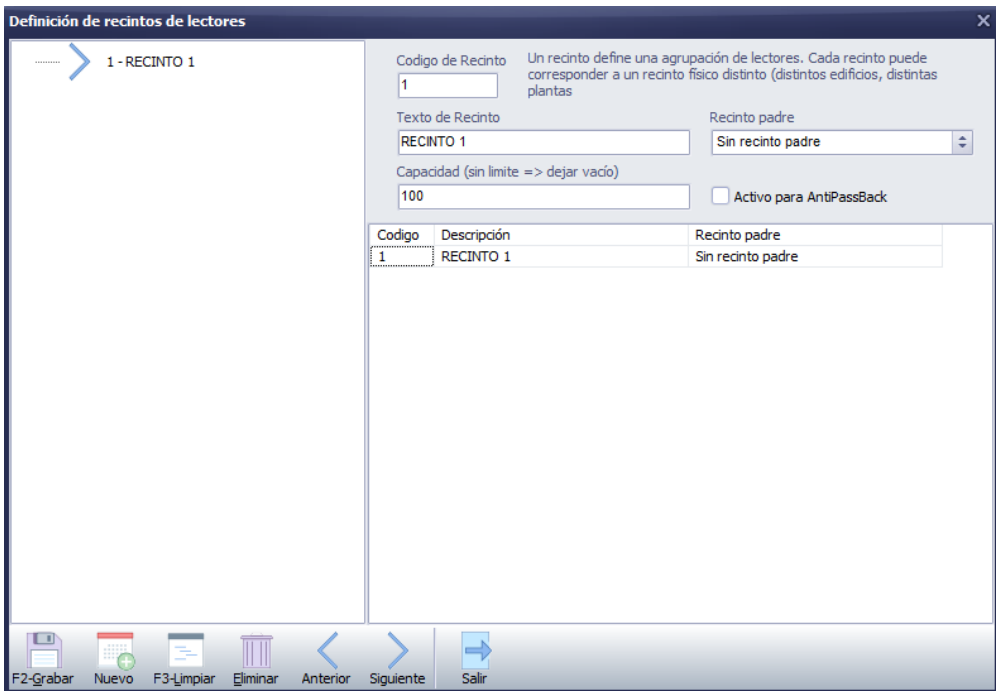

Figura 15: Ventana "Definición de recintos de lectores"

3.2.3.3. Definición de cabezas lectoras

Esta función sirve para asignar las controladoras y las cabezas lectoras del sistema.

En este sistema una sola controladora es capaz de gestionar hasta 8 lectores en bus, por eso es necesario definir al menos 1 controladora para cada 8 lectores. Para ello se usa la pestaña llamada “Controladoras”, que se encuentra dentro de la función “Definición de lectoras”. Se asigna un número y un texto

descriptivo a la controladora, además de indicar otros parámetros como el modelo de la controladora, su dirección IP, etcétera.

La interfaz de esta pestaña es la mostrada en la imagen Figura 16.

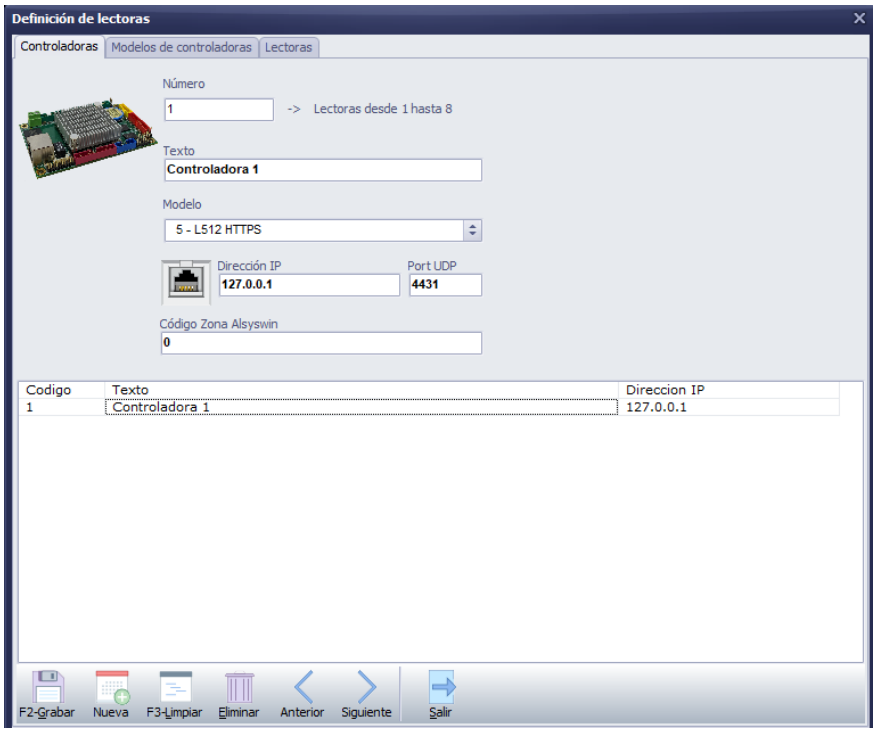

Figura 16: Ventana "Definición de lectoras", pestaña "Controladoras"

Con las controladoras necesarias asignadas, se pueden asignar los lectores correspondientes. Para asignar los lectores se utiliza la pestaña “Lectoras” dentro de “Definición de lectoras”. En esta pestaña se asigna un número y un texto descriptivo al lector. Se ha de indicar a qué controladora está conectado el lector y el recinto al que pertenece. También se gestionan desde esta pestaña otras propiedades como el tipo de lector, el tiempo de apertura, si hay dos lectores en una misma puerta, etcétera.

Se muestra la interfaz de esta pestaña en la siguiente imagen Figura 17.

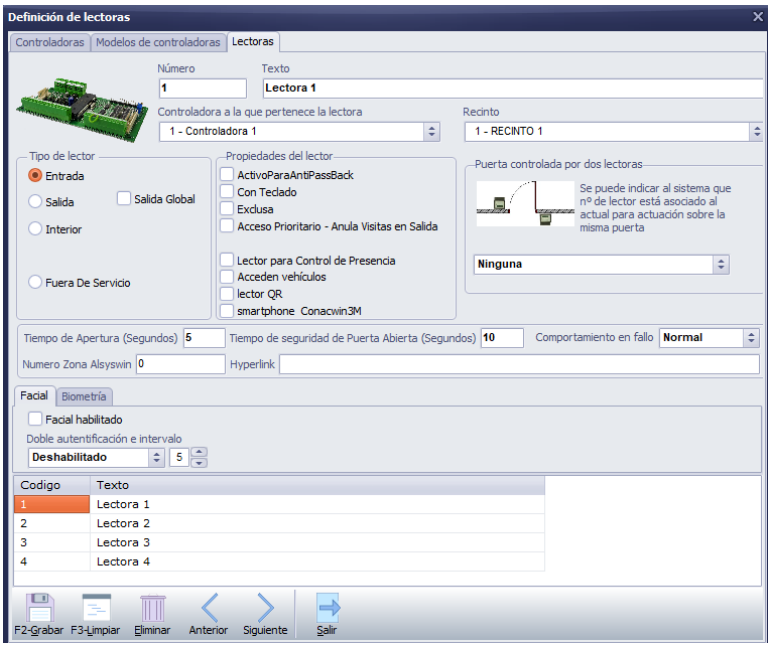

Figura 17: Ventana "Definición de lectoras", pestaña "Lectoras"

3.2.3.4. Definición de grupos de accesos

Un grupo de accesos sirve para determinar a qué lugares tienen el acceso garantizado los usuarios pertenecientes a ese grupo. Se indica un número de grupo con un texto descriptivo y se marcan las cabezas lectoras que pueden usar los usuarios del grupo para acceder. Posteriormente en la asignación del usuario se le asignará un grupo de accesos (Cap. 3.2.3.5).

En la siguiente imagen Figura 18, se aprecia la interfaz de la ventana “Definición de grupos de accesos”.

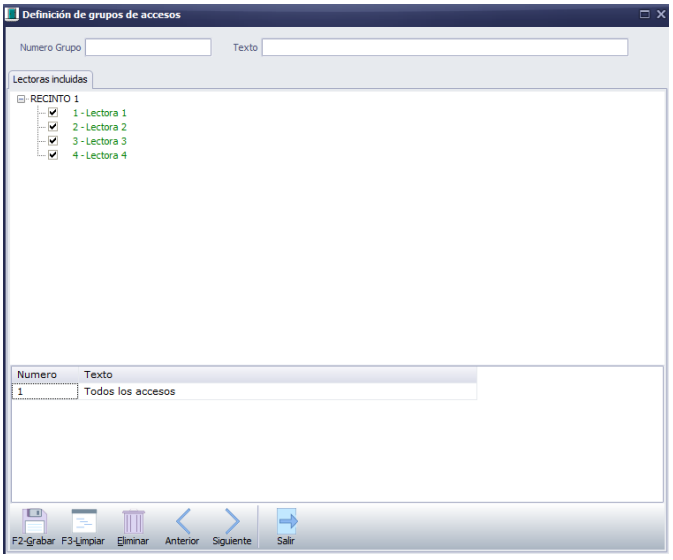

Figura 18: Ventana "Definición de grupos de accesos"

3.2.3.5. Definición de márgenes horarios diarios

Esta ventana sirve para establecer las horas en las que un usuario tendrá disponible el acceso en las cabezas lectoras que le corresponda a su grupo de accesos, de forma que solo pueda acceder en las horas que el administrador deseé. A la hora de asignar usuarios (Cap. 3.2.3.5) es necesario asignar un margen horario al usuario.

A continuación Figura 19, se muestra la interfaz de la ventana “Definición de márgenes horarios diarios”.

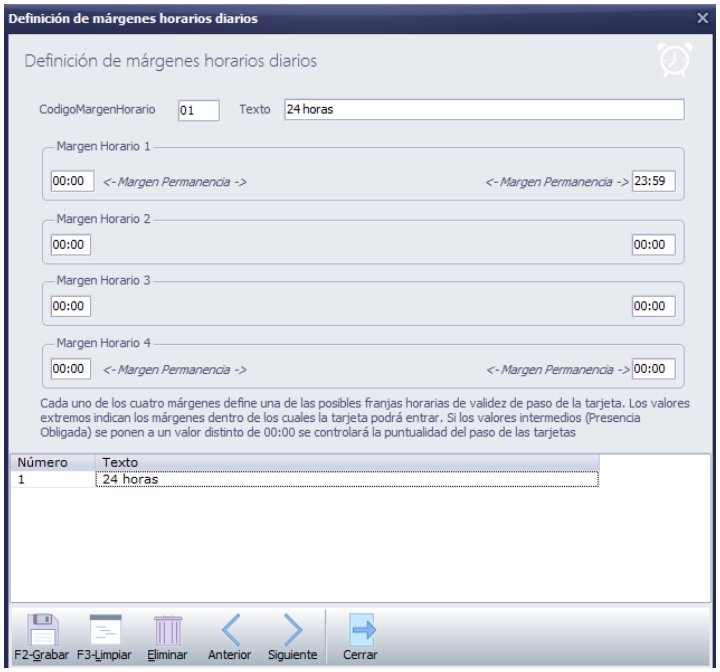

Figura 19: Ventana "Definición de márgenes horarios diarios"

3.2.3.6. Asignación/modificación tarjetas de personal

Esta ventana Figura 20 es la utilizada para dar de alta a los usuarios. Para ello se introducen sus datos personales (nombre, apellidos, DNI, etc.) junto con un número de tarjeta y el código de proximidad (UID) de la tarjeta. Se seleccionan la fecha inicial de validez y la fecha final de validez de la tarjeta, los grupos de acceso, y los grupos horarios del usuario. Adicionalmente se pueden añadir campos como la empresa, el departamento, el teléfono, etc.

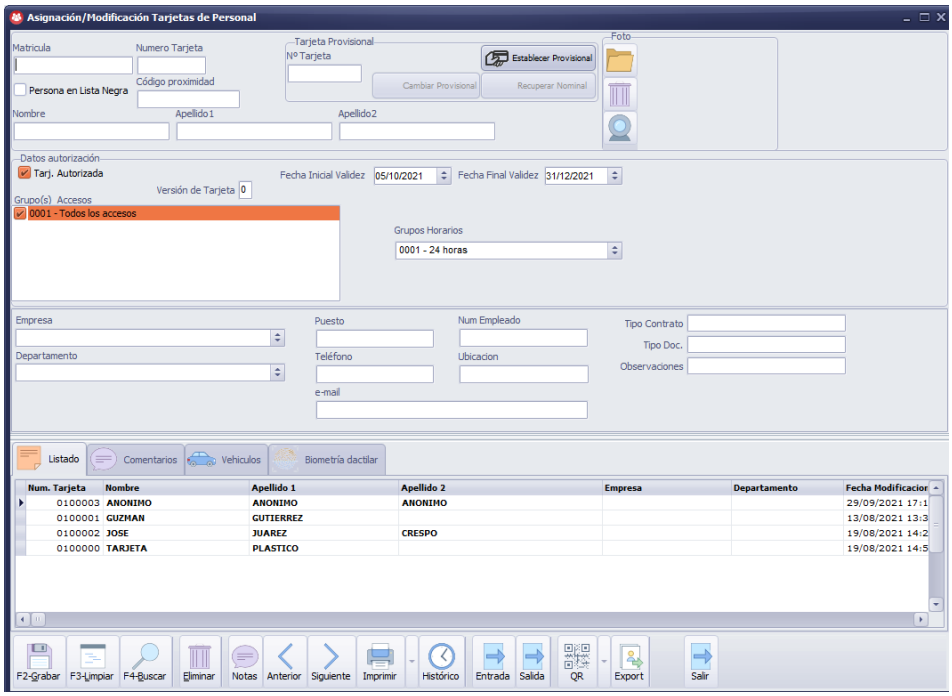

Figura 20: Ventana "Asignación/Modificación Tarjetas de Personal"

3.2.3.7. Reconocimiento de SmartID

Sobre esta ventana se ha añadido la funcionalidad de reconocimiento de smartphones. Antes de la integración, funcionaba de la siguiente forma:

- Se pulsa el botón “Buscar Lector USB”
- Una vez se encuentra el lector, se acerca la tarjeta que posea los datos de un usuario, los campos se rellenan automáticamente con los datos del usuario
- Por último se acerca otra tarjeta a la que se le quieran asignar los datos del usuario y se pulsa el botón enlazar

De esta forma los datos quedan asignados a la nueva tarjeta.

Después de integrar la funcionalidad del reconocimiento de smartphones el funcionamiento es prácticamente el mismo. La diferencia es que se ha de seleccionar la opción smartphone en el menú desplegable de la interfaz antes de usar el botón “Enlazar”, una vez hecho esto el usuario tendrá asignado su smartphone en el sistema de control de accesos.

La interfaz de la ventana “Reconocimiento de SmartID” es la mostrada en la siguiente imagen Figura 21.

Figura 21: Ventana "Reconocimiento de SmartID"

Cuando el usuario acceda con el smartphone, en la interfaz principal (Cap.3.2.3.1) aparecerá el icono de un teléfono móvil junto al número de tarjeta como se muestra en la imagen a continuación Figura 22.

	05/10/2021 10:21:53.721	Lectura de tarjeta autorizada	RECINTO 1	01-Lectora 1		0100001	GUZMAN	GUTIERREZ
	05/10/2021 10:21:55.925	Lectura de tarjeta autorizada	RECINTO 1	02-Lectora 2		0100001	GUZMAN	GUTIERREZ
	05/10/2021 10:21:58.075	Lectura de tarjeta autorizada	RECINTO 1	03-Lectora 3		0100001	GUZMAN	GUTIERREZ
	05/10/2021 10:21:59.960	Lectura de tarjeta autorizada	RECINTO 1	04-Lectora 4		0100001	GUZMAN	GUTIERREZ

Figura 22: Accesos de un usuario en el programa Conacwin

3.2.4. CA1NWH

Es el programa que se implementa en la controladora. En el desarrollo de este proyecto es el encargado de gestionar las tramas que envían y reciben los lectores. Cuenta con varias pestañas, cada una con una funcionalidad propia. A continuación, se describen las relevantes a la hora del desarrollo.

3.2.4.1. Ventana CPR02BUS

Esta ventana del programa muestra los milisegundos que han transcurrido desde el último intento de polling y los milisegundos que han transcurrido desde el último polling correcto. Esto permite detectar si existe algún fallo en la

conectividad de las cabezas lectoras con la controladora.

En la siguiente imagen Figura 23 se muestra un ejemplo de esta interfaz cuando 4 de los 8 lectores del bus se encuentran desconectados.

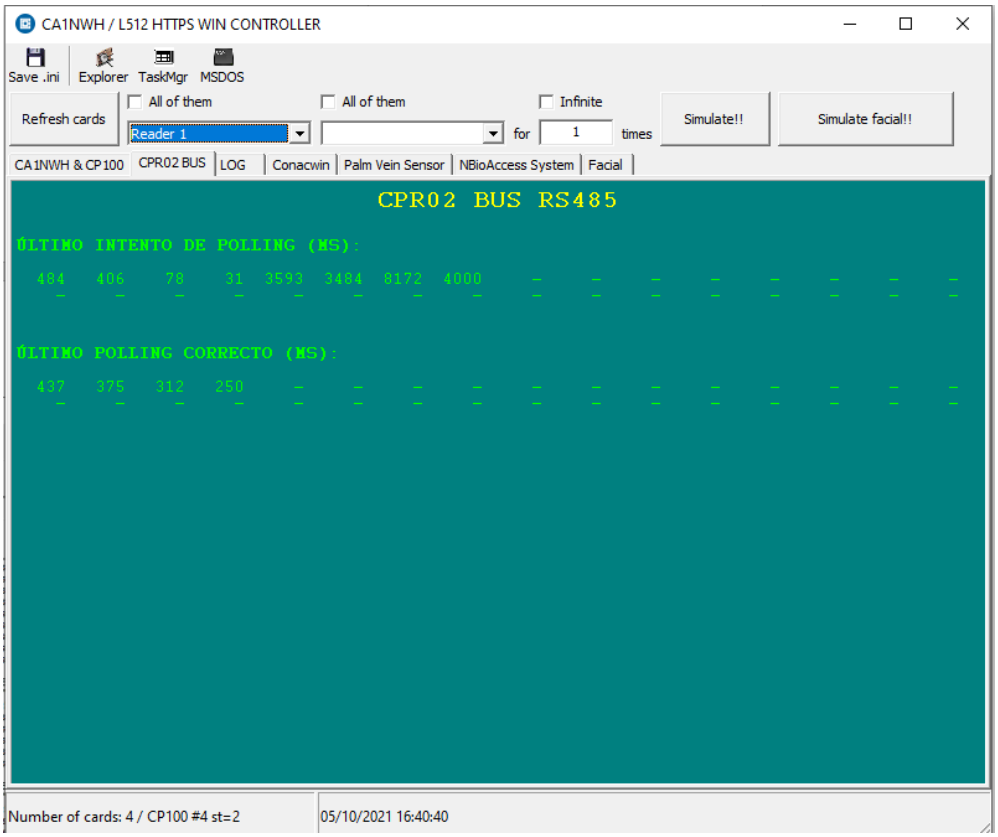

Figura 23: Pestaña "CPR02BUS"

3.2.4.2. Ventana LOG

Esta ventana Figura 24 se ha utilizado durante la implementación de la funcionalidad HCE. Cuenta con un log que ayuda durante la fase de desarrollo debido a que se pueden mostrar mensajes para detectar errores.

La siguiente Figura es un ejemplo de cómo se ve la ventana LOG cuando se acercan smartphones al lector. En los casos que el mensaje es “APDU NOK!” se debe a que se ha acercado un smartphone sin el servicio correspondiente instalado, por tanto la respuesta no es la esperada. En los casos que el smartphone cuenta con el servicio se muestra el mensaje “APDU OK!” junto con los datos del usuario.

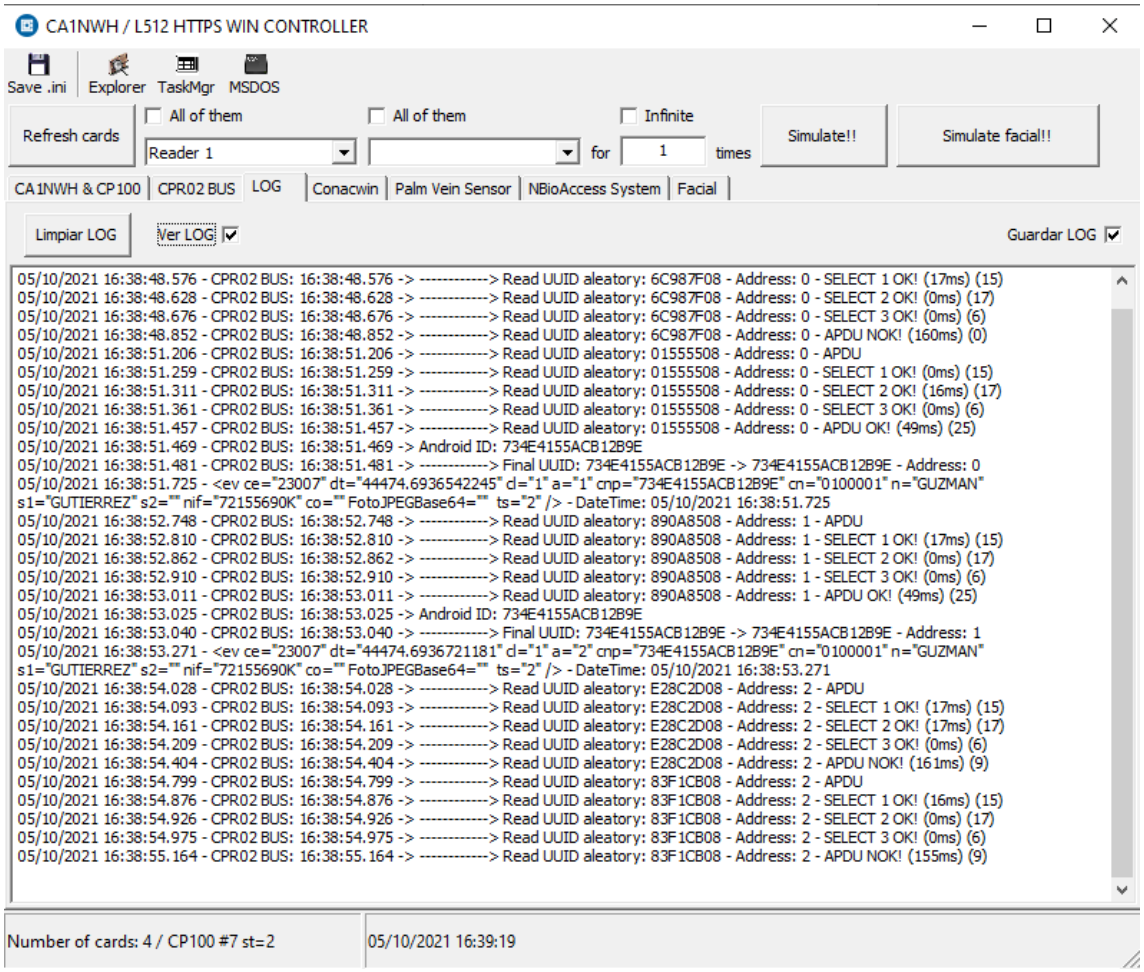

Figura 24: Pestaña "LOG"

3.2.5. Embarcadero RAD Studio 10.2 Tokyo

Es un entorno de programación en lenguaje c y c++. Este programa se ha utilizado a la hora de extender la funcionalidad del programa de la controladora (CA1NWH) para soportar la emulación de tarjetas en Android. También se ha usado para integrar la función de reconocimiento de smartphones en el sistema Conacwin.

3.3. Funcionamiento del servicio

El lector está haciendo polling constantemente hasta que encuentre un dispositivo móvil. Cuando un móvil se acerca al lector comienza la comunicación entre ambos. Primero se transmiten unas tramas iniciales que sirven para diferenciar si es un móvil o una tarjeta. Posteriormente se envía la trama APDU concreta que despierta el servicio Android HCE. El servicio obtiene un identificador único (Android ID) del teléfono, y transmite los datos cifrados con una clave privada. Después de verificar que todas las tramas son correctas, la controladora descifra los datos recibidos y envía el Android ID a Conacwin como

si se tratara del UID de una tarjeta física. Una vez hecho esto, el funcionamiento es el mismo que con una tarjeta, si el usuario está dado de alta con esa clave obtiene acceso y si no se deniega. No es necesario desbloquear el dispositivo móvil ni iniciar ninguna aplicación ya que el servicio está activo continuamente en segundo plano.

3.4. Descripción de la emulación

A continuación, se describen cómo funciona la emulación de la tarjeta en la versión final del sistema de autentificación.

Se tratarán aspectos como el archivo "Manifest" (Cap. 3.4.1), el servicio NFC (Cap. 3.4.2), los permisos necesarios (Cap. 3.4.3) y la identificación única (Cap. 3.4.4) y el cifrado de los datos transmitidos (Cap. 3.4.5).

3.4.1. Manifest

Los proyectos realizados en Android contienen un archivo XML llamado "AndroidManifest". La función de este archivo principalmente definir los elementos que forman la aplicación.

En este caso lo necesario a incluir en el archivo es lo siguiente:

- El nombre del paquete es necesario en todas las aplicaciones, Android Studio lo hace automáticamente,
- El permiso para usar el NFC del dispositivo,
- Las funciones NFC que usa,
- El servicio NFC Figura 25

```
<service android:name=".ServicioAcceso" android:exported= "true"
    android:permission="android.permission.BIND_NFC_SERVICE">
    <intent-filter>
        <action android:name="android.nfc.cardemulation.action.HOST_APDU_SERVICE"/>
    </intent-filter>
    <meta-data android:name="android.nfc.cardemulation.host_apdu_service"
        android:resource="@xml/apduservice"/>
</service>
```

Figura 25: Declaración del servicio en el archivo manifest

3.4.2. Servicio NFC

Este servicio NFC es el encargado de realizar la comunicación entre el dispositivo móvil y el lector. Está siempre activo en segundo plano para que se active cada vez que se acerque el móvil al lector.

3.4.2.1. Métodos

La clase llamada "ServicioAcceso" extiende la clase base de Android "HostApduService", es un requisito necesario para poder usar Android HCE, e implementa sus dos métodos:

- *processCommandApdu*, es llamado cada vez que el servicio recibe una APDU. Lo que retorna es una APDU de respuesta, de modo que la comunicación es half-dúplex.
- *onDeactivated*, se llama cuando deja de funcionar el vínculo entre el lector y el móvil con un argumento que indica el motivo de la desconexión.

3.4.2.2. Transmisión

La comunicación la inicia el lector, haciendo polling enviando un comando Inventory. Cuando este comando llega al móvil, éste le responde con otro comando en el que se encuentra incluido el UID aleatorio que devuelve Android como se explica en el apartado sobre ISO14443-3 (Cap.2.2.1.3).

Después de eso el lector envía una serie de tramas a las que va recibiendo respuestas que sirven para identificar si se está intentando establecer la comunicación una tarjeta física o un móvil. Esta fase de la transmisión es conocida como Select.

Por último, el lector envía una trama APDU que despierta el servicio y recibe la respuesta. A continuación, se explica el funcionamiento interno de este comando.

3.4.2.3. Concepto de APDU y AID

Como ya se mencionó en el apartado ISO/IEC 7816-4 (Cap. 2.2.2) una APDU es una trama con un formato definido.

En este sistema se envía una APDU en cada sentido después de haberse establecido la conexión (selección de la tarjeta/móvil).

El servicio se despierta cuando recibe el AID correspondiente, entonces para despertarlo, en el caso de esta aplicación el campo DATA de la APDU será el AID.

Cuando se envía la APDU con el AID correcto en el campo de datos, el servicio se activa y envía la respuesta correspondiente (el Android ID) para determinar si el usuario tiene acceso o no.

Como se puede ver en el apartado del Android Manifest (Cap. 3.4.1) existe un XML llamado apduservice Figura 26. En este archivo se indican características del servicio. En el caso de este sistema se establece que no es necesario desbloquear el dispositivo para que se active, así como el AID que despierta el servicio. Nótese que la imagen ha sido censurada mostrando espacios en blanco por la política de privacidad de Setelsa Security.

```
<host-apdu-service xmlns:android="http://schemas.android.com/apk/res/android"
    android:description="@string/servicedesc"
    android:requireDeviceUnlock="false">
    <aid-group android:description="@string/aiddescription"
        android:category="other">
<!--        AID generico-->
<!--        <aid-filter android:name=                />-->
<!--        AID                 ->
        <aid-filter android:name=                /> 
    </aid-group>
</host-apdu-service>
```

Figura 26: Archivo censurado "apduservice.xml"

3.4.3. Permisos

El sistema solamente necesita los servicios correspondientes al uso del NFC y la función de emular una tarjeta física.

3.4.4. Identificación única

Para que varios usuarios utilicen este sistema de autentificación es necesario que cada uno disponga de una identificación única que pueda diferenciarlo del resto de usuarios.

Se valoró la posibilidad de usar como identificación única el IMEI de la tarjeta SIM del teléfono móvil, o el propio número de teléfono. Pero estas alternativas fueron descartadas debido a que Android tiene restringido el acceso a las mismas.

La decisión final ha sido utilizar el Android ID, es un identificador único para los dispositivos Android. Tiene la ventaja de que no necesita ningún tipo de permiso para ser utilizado. Además, si se restablece el teléfono de fábrica el ID cambia. Esto puede resultar útil en caso de que cambie algún empleado ya que al

cambiar el ID ese nuevo empleado puede utilizar el mismo smartphone pero no tendrá los mismos accesos que el anterior.

3.4.5. Cifrado de datos

Se ha decidido cifrar los datos para que en caso de que alguien no deseado capture el tráfico no obtenga el identificador utilizado para acceder al recinto.

Los datos que transmite el smartphone están cifrados con un algoritmo de 256 bits usando como clave de cifrado un hash generado con una clave privada.

El cifrado es simétrico, por tanto los datos que cifra el smartphone son descifrados en el otro extremo usando la misma clave y algoritmo.

Al ser un programa en lenguaje Java, el archivo que se genera al compilar se puede descomprimir para ver su contenido. Para evitar que esto ocurra se ha utilizado el framework JNI (Java Native Interface). Haciendo uso de JNI es posible interactuar entre programas escritos en lenguaje nativo C o C++ y programas escritos en Java. La parte programada en lenguaje nativo no será accesible descomprimiendo el archivo APK, de modo que ese código quedará oculto. Por este motivo se han programado en C las funciones que obtienen la clave de cifrado y las que realizan el cifrado de los datos, para que las tanto las claves como el algoritmo utilizado sean conocidos solamente por el programador.

Para realizar esto se han de declarar las funciones desarrolladas en C como nativas en el código Java utilizando “native” como se ve en la Figura 27.

```
//Obtiene la key para generar el hash
public native String getClaveHash();
```

Figura 27: Declaración de una función nativa

La función se declara en la parte en Java pero se implementa en la parte en C. El nombre del método implementado en la parte en C debe ser la ruta al archivo Java que ha declarado el método. Debe estar precedido por:

“extern” + lenguaje
“JNIEXPORT” + tipo de dato + “JNICALL”.

Siendo lenguaje el lenguaje en el que está programada la función (en el caso de la imagen “C”), y tipo de dato el tipo de dato que retorna la función programada (en el caso de la imagen jstring). En la Figura 28 se encuentra una captura de pantalla de la función que devuelve la clave utilizada para generar el hash ocultando las claves mediante espacios en blanco.

```
/*
 * Devuelve la clave que se emplea para generar el hash sha1
 */
extern "C"
JNIEXPORT jstring JNICALL
Java_com_example_emulatenfc_CardService_getClaveHash(JNIEnv *env, jobject CardService thiz) {
    char keyHash[  ]=
    int z=0;
    for (int u=7;u<17;u++){
        keyHash[u]
        z=z+1;
    }
    return env->NewStringUTF(keyHash);
}
```

Figura 28: Función que retorna la clave usada para generar el hash

3.5. Metodología utilizada

A continuación, se explica cuál ha sido la metodología utilizada hasta llegar a la versión final descrita en el apartado anterior “Descripción de la emulación” (Cap. 3.4).

Para el desarrollo del proyecto se ha utilizado una metodología incremental. De forma que en cada fase del desarrollo se han ido añadiendo funcionalidades hasta conseguir cumplir los objetivos propuestos.

Se describirán las fases del servicio NFC (Cap. 3.5.1), la parte de la obtención de tramas (Cap. 3.5.2), la programación de la controladora (Cap. 3.5.3), el reconocimiento de smartphone en Conacwin (Cap. 3.5.4) y el cifrado de los datos a transmitir (Cap. 3.5.5).

3.5.1. Desarrollo del servicio NFC

Las fases 1,2,3 y 4 descritas en este apartado se han realizado con un móvil realizando la función de lector en lugar de con un lector físico, debido a que la empresa no disponía del lector necesario en el momento que se realizó.

- Fase 1: Comprobar que el móvil envío de datos por NFC, sin importar el dato funciona correctamente. Se hizo una versión básica del servicio que respondiera un valor fijo a modo de ejemplo para comprender mejor el funcionamiento del sistema.
- Fase 2: Cambiar la respuesta para que no fuera un valor fijo, sino que el móvil devolviera el valor introducido en un campo de texto.
- Fase 3: Conseguir que el valor se mantenga al cerrar la aplicación para que el usuario pueda utilizarlo sin necesidad de entrar a la aplicación a obtener el valor de nuevo. Hasta este momento la aplicación contaba con una interfaz básica de un campo de texto con un botón que guardaba la clave.
- Fase 4: A pesar de no estar planeado al comienzo, esta fase consistió en eliminar la interfaz de usuario porque al finalmente usar el Android ID como identificación no se consideró necesaria una interfaz. La interfaz fue útil en el desarrollo a modo de debug pero no se incluye en el resultado final.

3.5.2. Obtención de tramas

Con las fases del servicio completadas la siguiente fase fue pasar de utilizar un móvil funcionando como lector, a un lector físico real.

- Fase 5: Obtener las tramas enviadas durante la comunicación usando el programa CPRStart. Se enviaron manualmente los comandos Inventory, Select, y APDU en ese orden, observando en la ventana de protocolo las tramas que se envían para que se ejecute cada comando y las recibidas desde el móvil.

3.5.3. Programación de la controladora principal

Como se ha visto en el Cap. 3.2.4, la empresa Setelsa Security cuenta con su software propietario para la controladora principal. Este programa es el que se encarga de que el lector envíe y reciba las tramas correspondientes, además de realizar labores de autenticación de los datos recibidos y activar el relé encargado de abrir la puerta en caso de que el usuario tenga acceso. Antes de realizar este proyecto, el programa estaba diseñado para funcionar con tarjetas físicas. Por tanto, para realizar este proyecto ha sido necesario hacer una extensión a este programa para que acepte móviles.

- Fase 6: Programar el polling que realiza la controladora con las tramas obtenidas en la Fase 5.

3.5.4. Reconocimiento de smartphone en Conacwin

El programa Conacwin ya contaba con esta interfaz, pero era utilizada para asignar datos de una tarjeta física a otra. Al ser tarjetas se utilizaba su UID como clave para asignar los datos, de forma que para implementar la funcionalidad de reconocimiento de smartphone ha sido necesario utilizar el Android ID como clave en caso de que se seleccione la opción smartphone al asignar datos.

- Fase 7: Conseguir utilizar el Android ID como clave en esta ventana. Para esto se ha hecho que el lector USB envíe la trama APDU que despierta el servicio NFC. En caso de que alguna trama no sea correcta no se autentica la asignación.

3.5.5. Cifrado de datos a transmitir

Con la finalidad de añadir seguridad a la comunicación entre dispositivos, se cifraron los datos transmitidos usando un algoritmo con clave de 256 bits usando como clave de cifrado un hash generado con una clave privada. Estos datos se descifran en el otro lado de la comunicación tanto al reconocer el smartphone para asignar los datos de un usuario con el lector USB, como al acceder al recinto con el lector de pared.

Como se indica en el Cap. 3.4.5. esto se ha realizado en lenguaje C para que no se puedan obtener las claves al descomprimir el archivo APK. Al no haber trabajado nunca con JNI, se realizaron funciones básicas para comprender el funcionamiento, incrementando la complejidad de las funciones gradualmente.

- Fase 8: Se cifraron unos valores fijos con unas claves sencillas de los que se conocía el resultado.
- Fase 9: Pasar el Android ID como parámetro y retornar los datos cifrados para transmitirlos en la APDU de respuesta del servicio.

4. Pruebas

Se realizaron pruebas hasta cumplir el objetivo de cada fase del desarrollo (indicadas en el capítulo 3.5). En algunos casos en las pruebas realizadas se obtuvo el resultado esperado cuando se implementaron cambios menores, pero para cambios significativos de funcionalidad permitieron identificar problemas de implementación y exigieron varias iteraciones de desarrollo y verificación.

A continuación, se exponen las pruebas del servicio (Cap.4.1), las pruebas de obtención de tramas (Cap. 4.2), las pruebas de la programación del lector (Cap. 4.3), del reconocimiento del smartphone en Conacwin (Cap. 4.4) y las pruebas del cifrado de datos (Cap. 4.5).

4.1. Pruebas del servicio

En la Fase 1 descrita en el Cap. 3.5.1. se buscaba obtener cualquier respuesta del móvil. Se cometió el error de probarlo en un lector sin saber cómo era la comunicación, de modo que no funcionó debido a que el lector no era compatible con el protocolo de transmisión. Para seguir con la programación del servicio, se utilizó otro móvil a modo de lector hasta disponer de un lector compatible.

Utilizando el móvil como lector, se realizó la prueba de transmitir un valor y mostrarlo por pantalla. Estas pruebas permitieron identificar un error en la definición del formato de la APDU enviada.

En la Fase 3 el objetivo fue que el valor de retorno no se perdiera al cerrar la aplicación, para esto se almacenó el valor a devolver en el archivo de preferencias de Android. El nombre preferencias puede dar lugar a confusiones, simplemente es un archivo almacenado dentro de la APK que contiene pares clave-valor, de modo que con la clave se obtiene el valor a retornar. Funcionó como se esperaba.

Respecto a conseguir una identificación única.

- Se probó a utilizar el IMEI de la SIM del teléfono, al intentar obtenerlo, ocurrían varias excepciones de seguridad. Esto se debía a que en la última actualización de seguridad de Android los accesos a los componentes hardware del teléfono estaban restringidos, siendo necesario un permiso que no puede ser utilizado en aplicaciones de terceros (solamente en las que vienen predefinidas en el teléfono). Se escogió finalmente el Android ID

como alternativa de identificación ya que no necesita ningún tipo de permiso. Al probar con el Android ID no ocurrió ningún error.

Por último, se probó a instalar la aplicación sin la interfaz de usuario, el proyecto no se instaló correctamente. Para corregir esto hubo que modificar las opciones de ejecución, ya que buscaba la actividad principal.

4.2. Pruebas de obtención de tramas

En la fase de obtención de tramas (Cap. 3.5.2), surgieron algunos problemas con respecto al envío de APDU. Al tratar de enviar un comando APDU, la ventana de protocolo del CPRStart mostraba el siguiente mensaje "Reader error: Unknown Command". La empresa suministradora ofreció soporte sobre esta incidencia y se consiguió la comunicación correctamente. Fue necesario instalar una actualización del firmware que permite al lector el uso de APDU, con el firmware de serie no era posible enviar estos comandos.

4.3. Pruebas de la controladora principal

Las fases de la programación de la controladora se encuentran en el Cap. 3.5.3. Se verificó que el comportamiento del móvil era correcto ya que se obtenía un UID diferente cada vez que se acercaba el móvil, como se ve en el Cap. 2.2.1.3 (motivo por el que no se usa como identificador).

Se detectó un error en la comunicación. El programa recibía el mensaje de error en la trama APDU (mensaje de depuración introducido en caso de que el formato fuera incorrecto o no llegara la trama completa). Al depurar el programa ejecutando paso a paso el funcionamiento era el esperado, pero en tiempo real de ejecución no desapareció el mensaje de error. Las tramas APDU tienen mayor longitud que las tramas que usa el programa para las tarjetas, el tiempo durante el que se reciben datos que estaba establecido en el programa no era el suficiente, por tanto la APDU no llegaba completa, lo que ocasionaba el mensaje de error.

4.4. Pruebas del reconocimiento de smartphone en Conacwin

Estas pruebas han resultado similares a las del apartado anterior (Cap. 4.3) ya que se busca el mismo objetivo, que el lector además de las tarjetas reconozca el móvil.

Se extendió el programa para funcionar con móviles enviando la trama APDU que despierta al servicio. Al probarlo, cada vez que se acercaba el móvil

aparecía una ventana con el siguiente mensaje de error: “Error, it’s not a valid value”. Aparecía porque el programa cuenta con una variable que indica el tipo de tarjeta que se ha acercado, no existía ningún tipo de tarjeta que su UID tuviera la longitud del Android ID, lo que hizo que apareciera el error. Se corrigió añadiendo el tipo “Smartphone”.

4.5. Pruebas del cifrado de datos

Respecto al cifrado de los datos, se implementó el cifrador en lenguaje Java usando valores fijos, tanto para el texto a cifrar como para la clave de cifrado. Para comprobar si la implementación era correcta se empleó un cifrador online, mostrado en la Figura 29.

Figura 29: Cifrador online. Figura obtenida de [20]

Se cifró el contenido de la APDU de respuesta, también en lenguaje Java, descifrándolo en el otro extremo de la comunicación (la controladora principal), en lenguaje C. El resultado de esta prueba fue el esperado, por tanto se implementó esta función en lenguaje C. Esta parte resultó ser la más problemática debido al desconocimiento en el uso del framework JNI, por tanto surgieron varios errores.

Se realizaron programas básicos aparte que usaran este framework para comprender su funcionamiento. Se trató de verificar que la función en C funcionaba correctamente ejecutando el programa, al llamar a la función el programa detenía su ejecución. Al depurar el programa no es posible debuggear

las partes del archivo en C que usan JNI, lo cual dificulta mucho el proceso de corrección. El error era ocasionado por el tipo de datos que retorna la función de cifrado. La función retornaba un jstring (equivalente a un String en Java) con los datos cifrados, y posteriormente se convertía a bytes para devolverlo en la APDU de respuesta. Lo correcto era devolver un jbyteArray (equivalente a un array de bytes en Java), ya que hay caracteres cifrados que no son representables en un String.

5. Conclusiones y fases futuras

Con el proyecto finalizado, se pueden exponer tanto las conclusiones obtenidas (Cap. 5.1) como las fases futuras y posibles funcionalidades a añadir al sistema (Cap. 5.2).

5.1. Conclusiones

Se ha realizado una integración que permite el reconocimiento de smartphones y su asociación con usuarios del sistema de control de accesos ya dados de alta, la asociación se realiza mediante un lector USB portátil. Durante la realización del sistema se han utilizado los lenguajes de programación Java y C.

Se ha realizado una aplicación para móviles Android que permite emular el comportamiento de una tarjeta RFID tradicional.

Se ha obtenido conocimientos y experiencia en temas con los que no se contaba al plantear los objetivos al inicio del proyecto que han resultado ser realmente útiles, por ejemplo, el cifrado de los datos a transmitir y el uso de código C en Android Studio para dificultar en gran medida la obtención del código mediante ingeniería inversa.

Se han desechado los objetivos secundarios planteados al principio del proyecto. Se ha considerado que la opción de que sea un servicio corriendo continuamente en segundo plano ofrece una mayor comodidad para el usuario al no necesitar una interfaz para simplemente obtener la clave que lo identificará.

5.2. Fases futuras

Aunque el sistema de control de accesos final funciona correctamente y puede ser implementado en un entorno real, sigue existiendo la posibilidad de añadir funcionalidades útiles. A continuación, se exponen las próximas fases futuras del sistema:

- Reducción del tiempo de respuesta del teléfono móvil. El tiempo medio de respuesta actual del smartphone es 80ms, a pesar de ser un tiempo de respuesta bajo se está estudiando la forma de reducirle aún más para futuras versiones de la aplicación.
- Implementación de un sistema similar que en lugar de por NFC, funcione por bluetooth, ya que iOS no permite a aplicaciones de terceros el uso del NFC para emular una tarjeta.

Referencias

[1] http://setelsa-security.es/ (Julio 2021), Autor: Setelsa Security.

[2] https://www.biosys.es/sistemas-de-tarjetas/sistemas-con-banda-magnetica-chip-o-codigo-de-barras/ (Julio 2021), Autor: Biosys, Título: “Sistemas con tarjetas de banda magnética, chip o código de barras.”

[3] https://www.actum.es/preguntas-frecuentes/que-es-una-tarjeta-de-proximidad (Julio 2021), Autor: Actum, Título: “Tarjeta de proximidad. Tipos de tarjetas.”

[4] https://veriddica.com/tipos-de-tarjetas-sin-contacto (Julio 2021), Autor: Veriddica, Título: “Tipos de tarjetas sin contacto.”

[5] https://www.iso.org/obp/ui/#iso:std:iso-iec:guide:60:ed-2:v1:es (Julio 2021), Autor: ISO, “Título: ISO/IEC Guide 60:2004(es).”

[6] http://www.emutag.com/iso/14443-2.pdf (Agosto 2021), Autor: ISO, Título: “Radio frequency power and signal interface.”

[7] http://www.emutag.com/iso/14443-3.pdf (Agosto 2021), Autor: ISO, Título: “Initialization and anticollision.”

[8] https://www.xatakamovil.com/conectividad/origen-nfc-variante-rfid-que-se-ha-convertido-llave-para-pagos-moviles (Agosto 2021), Autor: Samuel Fernández, Título: “El origen del NFC, la variante del RFID que se ha convertido en la llave para los pagos móviles.”

[9] https://developer.android.com/guide/topics/connectivity/nfc (Agosto 2021), Autor: Desarrolladores de Android, Título: “Descripción general sobre la Comunicación de campo cercano.”

[10] https://www.redeszone.net/2019/03/21/diferencias-nfc-bluetooth/ (Agosto 2021), Autor: Javier Jiménez, Título: “NFC vs Bluetooth: diferencias y puntos positivos y negativos de cada tecnología.”

[11] https://www.xatakamovil.com/tutoriales/nfc-movil-que-sirve-siete-usos-para-sacarle-todo-partido (Agosto 2021), Autor: Cosmos, Título: “NFC en el móvil: qué es, para qué sirve y siete usos para sacarle todo el partido.”

[12] https://developer.android.com/guide/topics/connectivity/nfc/hce?hl=es-419 (Agosto 2021), Autor: Desarrolladores de Android, Título: “Descripción general de la emulación de tarjetas basada en el host.”

[13] https://www.cursosaula21.com/que-es-el-rfid/ (Agosto 2021), Autor: Aula21, Título: “RFID: todo lo que necesitas saber.”

[14] https://www.fqingenieria.com/productos/lector-grabador-de-pared-cpr02-de-13-56-mhz-para-tags-iso15693-mifare-iso14443-a-b-y-nfc-con-rele-integrado-y-bus-rs485-40-40 (Consulta Agosto 2021), Autor: FQ Ingeniería Electrónica, Título: Lector/grabador de pared CPR02 de 13,56 Mhz para tags ISO15693, MIFARE, ISO14443-A-B Y NFC, con relé integrado y bus rs485.

[15] http://www.emutag.com/iso/14443-4.pdf (Agosto 2021), Autor: ISO, Título: Transmission protocol.

[16] https://andro4all.com/guias/tecnologia/etiquetas-nfc-que-son-como-funcionan-usos (Septiembre 2021), Autora: Beatriz Alcántara, Título: Etiquetas NFC: qué son, cómo funcionan y 21 usos increíbles.

[17] https://www.game.es/samsung-galaxy-j7-2016-blanco-libre-smartphone-179609) (Septiembre 2021), Autor: Game, Título: Samsung galaxy j7 2016 blanco.

[18] https://www.xatakandroid.com/moviles-android/samsung-galaxy-xcover-4s-caracteristicas-precio-ficha-tecnica (Septiembre 2021), Autora: Laura Sacristán, Título: Samsung Galaxy XCover 4s: la línea robusta de Samsung se actualiza con mejores prestaciones y un diseño más resistente.

[19] https://www.holydogwater.com/product/B079T2FKN1/ (Octubre 2021), Autor: Holydogwater, Título: Lector USB OMNIKEY 5022 CL.

[20] http://symmetric-ciphers.online-domain-tools.com/ (Octubre 2021), Autor: OnlineDomainTools, Título: Symmetric Ciphers Online.