

Rediseño y optimización de un sistema electrónico con tecnología NFC para facilitar su producción.

Christian Zorrilla Valencia

Ingeniería electrónica industrial y automática

Resumen

A día de hoy, se nos hace muy extraño el hecho de tener que pagar insertando la tarjeta de crédito en el datáfono o el no poder pagar con el teléfono móvil. Este tipo de aplicación de pago, está basado en la comunicación NFC, «Near Field Communication», comunicación que es sencilla, segura y que, a día de hoy, se usa en muchos dispositivos, ya sean tarjetas, móviles, pulseras, etc. Este proyecto trata sobre el rediseño y la optimización de un dispositivo de cronometraje deportivo que, previamente desarrollado por un estudiante de la UPC, y que usa esa la tecnología mencionada anteriormente. Lo que se pretende es rediseñar el sistema, tanto en hardware como en firmware, para que este sea más pequeño y consuma menos energía. Para llevar a cabo estas medidas, se reestructurará el hardware, escogiendo solamente los componentes que sean empleados, quitando así, todo aquel que no aporte nada al sistema. En cuanto a firmware, se modificará el código incluyendo un modo reposos, de tal manera que el usuario pueda, dentro de unos parámetros, poner durante un tiempo el microcontrolador a dormir y así evitar consumos innecesarios. Hay que añadir, que también se implementará una aplicación Android, que permita leer el contenido de las tarjetas NFC. Como resultado del sistema, se ha podido desarrollar una estructura más sólida que la anterior, cumpliendo los objetivos principales de reducción de tamaño y de disminuir el consumo. También se ha conseguido llevar a cabo la aplicación, mediante la cual podemos leer los datos de las tarjetas. Debido al nuevo código, la memoria del microcontrolador deja mucho espacio libre que no se usa. Esto podría llevar a utilizar un microcontrolador más pequeño del actual, cosa que podría hacer que el consumo y el espacio que ocupase fuese menor. También se le podría adjuntar una batería de más capacidad, aumentando la autonomía del sistema. Finalmente se implementará una aplicación que permita mostrar en el dispositivo móvil los datos que contienen las tarjetas NFC.

1. Introducción

Actualmente, el sector de la tecnología es un sector que se encuentra en una continua mejora. Este hecho conlleva la implementación de muchas tecnologías que simplifican la manera en la que vivimos, facilitando ciertas tareas cotidianas o transformando

aquellas que antes no lo eran, en tareas que se hacen con un simple clic. Uno de los tipos de tecnología que nos facilita el día a día, es la comunicación inalámbrica. Esta comunicación está presente en cualquier parte del mundo, ya sea para

encender la televisión (infrarrojos) o para enviar un documento a través del correo electrónico (WiFi). Dentro de este tipo de comunicación hay infinidad de ellas, pero hoy en día, la tecnología inalámbrica NFC es de las que más uso tiene. Además, permite, entre otras cosas, pagar con el teléfono o pagar sin introducir la tarjeta de crédito. Este tipo de comunicación únicamente necesita dos dispositivos para su funcionamiento, un emisor y un receptor. El emisor se encarga de establecer la comunicación con el receptor y maneja el intercambio de datos. Por otro lado, el receptor simplemente responde al primero.

Mediante esta tecnología, se desarrolla la comunicación entre nuestro emisor (Arduino) y nuestro receptor (tarjeta NFC). En este proyecto, se busca mejorar el sistema de cronometraje deportivo en cuanto a hardware y firmware, reduciendo el sistema y optimizando recursos, lo cual implica cambios en el emisor. Para finalizar, se implementará una app que utilice dicha tecnología con el fin de obtener los datos del cronometraje.

2. Objetivos

Como resultado de un previo Trabajo Final de Grado llevado a cabo por la Universidad Politécnica de Cataluña, se ha desarrollado un prototipo electrónico que puede utilizarse para el cronometraje de pruebas deportivas que utiliza tecnología NFC. Actualmente, el sistema se ha diseñado mediante placas de evaluación. El objetivo principal del proyecto será la integración y diseño de un prototipo pre-industrial, teniendo en cuenta las características necesarias para realizar la fabricación y la optimización del consumo: funcionamiento a baterías, cargador universal, caracterización del consumo en los diferentes modos de funcionamiento, documentación, etc.

Para aplicar todas estas medidas, se deben tener en cuenta las finalidades que busca este proyecto, que son:

- Mejorar el código y dividirlo para un microcontrolador más pequeño.
- Rediseño del sistema a nivel físico implementando el nuevo microcontrolador.
- Caracterización del consumo en los diferentes modos de funcionamiento.

Además, se propone como objetivo específico, el desarrollo de una aplicación de interfaz de usuario para dispositivos

móviles que permita la comunicación y transferencia de datos del prototipo electrónico, principalmente para la validación del funcionamiento del prototipo y, en segundo lugar, para realizar la descarga de datos.

3. ESTADO DEL ARTE

El sistema NFC es un sistema de comunicación inalámbrico que se base en la tecnología RFID. Esta tecnología tiene dos partes diferenciadas, la etiqueta y el lector (Montero, 2017):

- Etiqueta: se encargan de transmitir la información al lector. Existen 3 tipos de etiquetas, las activas, las pasivas y las semiactivas (Medina Delgado, 2017).
- Lector: es el dispositivo que se encarga, mediante una señal de radiofrecuencia, esperar a que una etiqueta entre en contacto y comience la transferencia de datos. Hay dos tipos de lector, los fijos y los móviles (Bin Abdul, 2017).

Todo sistema de comunicación NFC necesitará de un iniciador y un objetivo. Por un lado, el dispositivo que inicie y controle la transferencia de datos, será el iniciador. En el caso de la tecnología RFID, corresponde al lector (Medina Delgado, 2017). Por otro lado, el que responda a los requerimientos del iniciador, será el objetivo o etiqueta que en terminología del sistema RFID, corresponde a la etiqueta. Para llevar a cabo esta comunicación se usan los protocolos ISO/IEC 14443 y JISX6391-4. Actualmente hay 4 tipos de tarjetas (Ilustración 1) (Montero, 2017).

	Tipo 1	Tipo 2	Tipo 3	Tipo 4
Nombre del chip	Innovision Topaz	NXP MIFARE	Sony FeliCa	NXP DESFire, SmartMX-JCOP
ISO/IEC standard	14433 Type A	14433 Type A	JIS X 6319-4	14433 Type A, Type B
Velocidad de datos	106 kbps	106 kbps	212 kbps 424 kbps	106 kbps 212 kbps 424 kbps
Tamaño memoria	1Kb	2Kb	1Mb	64Kb
Lectura	Si	Si	Si	Si*
Escritura	Si	Si	Si	Si*
Soporte anticollisión	No	Si	Si	Si
Precio	Bajo	Bajo	Alto	Medio / Alto
Proveedor	Innovision, Research and technology	Philips / NXP	Sony	Several

*Solamente se pueden configurar de una manera para su proceso de vida, en cambio las otras pueden ser ambas cosas durante su vida útil.

Ilustración 1: Tipos de etiquetas.

4. Aplicación previa desarrollada

La idea principal del anterior estudio, fue crear un sistema robusto y seguro para poder cronometrar eventos deportivos. Este cronometraje se hizo mediante la tecnología NFC usando el modo de operación Lectura/Escritura. Para llevar a cabo esto, se dispuso de 4 tipos de bases, las cuales se encargaron de escribir la hora en la que se pasa por la base, leer los datos de la etiqueta, borrar los datos de la etiqueta y verificar que se borraron dichos datos de la etiqueta (Laaroussi, 2019). En cuanto al hardware, las bases se encuentran formadas por la unión de Arduino, una RTC, un lector de tarjetas NFC y una fuente de alimentación propia. En cambio, para el software se utilizó la programación de Arduino y se estructuró un solo programa para poder usar los 4 métodos. Cabe recalcar que la idea de este proyecto, es una

de las que se encuentra en el mercado de la mano de diversas empresas, como por ejemplo SPORTident (Laaroussi, 2019). Los 4 modos de operación son:

- Modo escritura: Para poder llevar a cabo la función de escribir en la tarjeta NFC, en primer lugar, se debe inicializar la RTC. Después, se asigna el número de base que será (ID_Base). Este número ID_Base asignará el bloque de la memoria en el que se guardará el dato. Esto es debido a que la memoria EEPROM de la etiqueta tiene ciertos bloques en los que es imposible escribir, por esta razón, mediante el programa obviamos estos bloques, para así, poder utilizar los huecos de la memoria que sí están disponibles. Una vez llegados a este punto, se pasa al bucle de escritura. En este bucle, primero se comprueba que haya una etiqueta NFC. Seguido, si se ha detectado una etiqueta, se escribe el dato. A continuación, se lee el dato escrito para verificar que se ha escrito correctamente, suena un pitido y se escribe el ID_Base más la fecha de la RTC. Toda esta explicación se puede observar en el flujograma de la Ilustración 14 (Laaroussi, 2019).
- Modo lectura: Este modo es el encargado de que al final de la carrera, se pueden ver la hora a la que se ha pasado por cada una de las bases de escrituras distribuidas a lo largo del evento. El flujograma de este modo, Ilustración 15, es más simple debido a que no es necesario que se introduzca ningún parámetro desde el monitor serie ni haya que inicializar ninguna RTC. Directamente se entra en el bucle de lectura debido a lo mencionado anteriormente. El programa espera pacientemente a que se acerque una tarjeta NFC. Una vez detectada, comienza a leer la memoria y a mostrarla por el monitor serie. Finalmente, suena un pitido y vuelve al inicio del bucle a esperar que pase una nueva tarjeta NFC (Laaroussi, 2019).
- Modo Borrar: La idea de este modo es que esté situado al inicio del evento, así se tendrá una tarjeta limpia sin datos en ella que lleven a la confusión. Una confusión podría ser dada a que en la carrera anterior hubiera 45 bases y en la actual solamente 15 y si no se llegasen a borrar todos los datos de esta, el corredor al ir a la base de lectura, se encontraría con sus marcas de la base 0 a las 14 pero de la 15 a las 44 encontraría las marcas de otro corredor, cosa que puede prestar a equívocos. La Ilustración 16, como en el modo de lectura, muestra cómo se entra directamente en el bucle. Cuando se detecta una etiqueta, el programa pasa a borrar la memoria de esta y, cuando haya terminado, sonará un pitido y volverá a esperar a que se acerque una tarjeta NFC (Laaroussi, 2019).
- Modo Verificar tarjeta vacía: Este último modo, al igual que los 2 anteriores, no necesita recibir ningún dato del monitor serie. Como se puede ver en la Ilustración 17, esto hace que también se entre directamente en bucle, esperando a detectar una etiqueta. Una vez detectada, este lee la memoria de dicha tarjeta y verifica que está vacía. Una vez acabada la comparación hace un tipo de pitido, dependiendo de si está vacía o no, y vuelve a

esperar a una tarjeta (Laaroussi, 2019).

Los componentes utilizados para esa aplicación son:

- Arduino Mega 2560 Rev V3.
- PN532 NFC module V3.
- RTC DS3132.
- Zumbador activo 5V.
- Batería 9V 650mAh.
- PCB
- Diversos materiales.

Todos estos elementos van conectados a una PCB que hace de unión de estos.

El presupuesto total de sistema es el siguiente:

PRODUCTO	CANTIDAD	COSTES (€)
Arduino Mega 2560 Rev V3	1	35,00
PN532 NFC module V3	1	3,55
RTC DS3132	1	2,25
Zumbador activo 5V	1	0,86
Batería 9V 650mAh	1	6,42
Conector clip de la batería de 9V	1	0,72
Fabricación placa PCB	1	1,82
Diversos materiales	1	1
Total		51,62

Ilustración 2: Presupuesto materiales.

Las conclusiones extraídas a cerca de este sistema son las siguientes:

- Optimización de la memoria del microcontrolador. Este tipo de sistema, tiene una memoria demasiado grande para el uso que se le da, dejando mucha vacía. Esto podría solucionarse utilizando un microcontrolador más pequeño.
- Tamaño grande con elementos que no se usan. Una mejora de reestructuración, eliminando los elementos que este no usan como los pines digitales y analógicos, podría ayudar a disminuir el tamaño.
- Una mejora de la implementación del Modo Verificar tarjeta vacía. Tras probar este modo de funcionamiento, me he percatado de que este siempre da error y, a pesar de que la tarjeta esté vacía, el sistema siempre muestra que está llena.
- Reducción del precio final. Mediante los cambios de los puntos anteriores, se puede llegar a implementar un sistema que sea más económico.
- Una nueva configuración de la RTC. Tras varias pruebas en varios días, se ha podido comprobar que, si se carga un programa y se ejecuta varios días después, la RTC marca la última fecha que tenía. Esto es debido a que se deja de alimentar, no sigue funcionando y no avanza su tiempo, llevando a errores de marca.
- Monitor serie. Si no hay monitor serie, no se puede configurar el modo. Esto implica que pese a grabar el programa en el sistema, cuando este deje de ser alimentado y se vuelva a alimentar, el programa volverá a pedir que se asigne el modo, con lo cual, si no hay monitor serie, no hay funcionamiento del programa.
- Por último, se ha medido el consumo del prototipo, cuando este está en pausa y cuando este está funcionando y los valores obtenidos han sido diferentes a los mencionados por Laaroussi, S. Cuando el sistema está esperando a una tarjeta, su consumo es de 0.118A. Mientras que el consumo del prototipo cuando se está tomando la lectura de la tarjeta es de 0.166A. Estos datos distan de los

obtenidos por Laaroussi, S.

5. Prototipo v1

Se ha constado un prototipo muy sólido en cuanto a las características de hardware y de software. En cuanto a software, se ha podido implementar una mejora con el Sleep Mode, cosa que ha hecho que se consiga un mejor consumo. También se ha mejorado y se ha podido ejecutar sin problemas en Modo Verificar tarjeta.

En cuanto a hardware, el haber suprimido los elementos innecesarios, nos ha llevado a una reducción del prototipo de Laaroussi, S. Esta reducción también se ha visto reflejada en el coste de los materiales, pasado de un prototipo de 51.62 € a 31.54 €. El consumo y la autonomía se han visto reflejados con una mejora bastante elevada, ya que instaurando los valores de consumo obtenidos de la medición del prototipo de Laaroussi, S. con su consumo real en pausa y en funcionamiento a un ciclo de 7 segundos, se obtiene el siguiente consumo y la siguiente autonomía:

De esta manera queda reflejado que el consumo pasa de 0.139 A de media a 0.096 A. mientras, su autonomía cambia de, aproximadamente, 5 horas a 27 horas.

En cuanto a la memoria, se ha optimizado en gran parte debido a que, Ilustración 40, se ha pasado de utilizar solamente un 5% de su memoria Flash y un 32% de la memoria SRAM a un 37% de la Flash y un 77% de la SRAM. A pesar de cumplir estas mejoras, han surgido nuevos problemas y no se han podido mejorar las siguientes:

- La PCB no entra en la caja. Debido a una mala medición por parte del fabricante, el interior de la caja no hace las medidas proporcionadas por él. Este hecho hace que la PCB no se pueda colocar en ella.
- Una nueva configuración de la RTC. A pesar de haber fraccionado en 4 programas los modos, la RTC sigue sin avanzar su hora cuando pierde la tensión, esto conlleva a errores cuando se usa el Modo Escritura.
- Monitor serie. Mediante el nuevo software, se ha conseguido que el sistema entre en el Sleep Mode y reduzca su consumo. A pesar de ello, se sigue necesitando el monitor serie para Asignar el ID_Base del Modo Escritura y la configuración del Sleep Mode en cada modo. Cabe recalcar que al no haber ninguna conexión con el puerto serie entre el sistema y el monitor serie, el Modo Lectura se vuelve inservible debido a que no se puede leer por ningún medio los valores que lee el programa.

Para los problemas mencionados anteriormente, se ha procedido a diseñar un prototipo v2 que pueda garantizar que se cumplen todos los puntos de los apartados. Por ello se va a proceder a lo siguiente:

- La PCB no entra en la caja. Implementar un nuevo diseño que quepa en la caja de protección.
- Una nueva configuración de la RTC. Averiguar el problema de por qué la RTC no avanza en el tiempo cuando esta pierde la tensión.
- Monitor serie. En el nuevo diseño, implementar

unos pines mediante los cuales se pueda ver por el monitor serie qué hace el programa en cada instante para así poder ver los resultados del Modo Lectura. También se procederá a una reestructuración del código de manera que no se tenga que pedir por el monitor serie que se escriba ningún dato.

- Finalmente, también se buscará que el consumo y el precio se no alejen de los valores obtenidos.

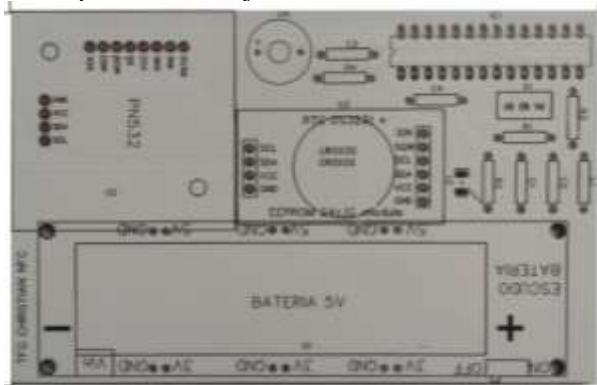


Ilustración 3: PCB con serigrafía de los componentes prototipo v1.

6. Prototipo v2

Tras finalizar todas las pruebas con este prototipo, se puede asegurar su fiabilidad en todos los modos de funcionamiento. Se ha conseguido mejorar el prototipo v1, arreglando los problemas que este presentaba tanto de software como de hardware. Los problemas de software se han solucionado agregando un programa solamente para la inicialización de la RTC, para así garantizar su perfecto funcionamiento. También se ha cambiado la manera de introducir los valores del ID_Base y del Sleep Mode, convirtiéndolo en una manera más sencilla y que no reporta problemas.

En cuanto a hardware, se ha diseñado una PCB que encaja a la perfección en la caja mientras que su predecesora no entraba. También se ha añadido un espacio para los condensadores del cristal, se ha suprimido la resistencia que impedía funcionar al zumbador y se han añadido 3 nuevos pines para poder monitorizar, en caso de que eso sea requerido, los modos de funcionamiento.

Estas medidas, no han hecho cambiar el consumo de la batería de un modelo respecto del otro. Esto se debe a que las medidas no han sido de gran influencia a la hora del este, sino más bien en la distribución de los componentes, teniendo un consumo medio por segundo de 0.096 A y una autonomía de 27 horas.

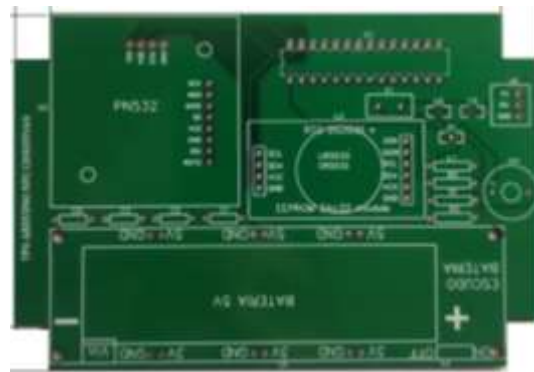


Ilustración 4: PCB con serigrafía de los componentes prototipo v2.

7. Aplicación

Actualmente, como se mencionó en los primeros puntos del proyecto, la tecnología NFC está en auge y, como toda tecnología, esta se expresa para poder sacar su mejor versión. De este modo, en el mercado existen muchos tipos de aplicación, por ejemplo, NFC Tools, para la lectura y escritura de tarjetas NFC. Además, una aplicación puede ser también otro modo de marketing del que se pueden obtener beneficios. En este proyecto se dispone a la creación de una app que se capaz de leer los datos que se escriben en el Modo Escritura.

Para esta programación, se utiliza el lenguaje por excelencia de las aplicaciones Android, Java. Este lenguaje dentro del programa Android Studio, permite que se puedan crear aplicaciones Android gracias a su versatilidad y su fácil interfaz.

Primero de todo, se va a pedir el acceso al sistema NFC del teléfono para poder empezar a ejecutar la aplicación. Una vez permitido, se iniciará la app y se mostrará por pantalla, en caso de que no esté activa, que se active el NFC. Cuando este lo esté, ya podremos pasar la tarjeta por el lector NFC del móvil. Si se mira la interfaz, se puede ver que arriba hay un recuadro llamado Nombre, donde se debe introducir el nombre de la persona o dispositivo. Si se lee una etiqueta y no tiene nombre, la aplicación interactuará diciendo que se añada uno. Una vez pasada la tarjeta, el móvil hará un ruido conforme se ha leído la tarjeta y se mostrará por pantalla que se ha leído y su contenido, ya sea textos, webs, etc.

Mediante los conocimientos adquiridos durante la programación del modo de lectura, se ha implementado una aplicación que hace los mismo, pero para el teléfono móvil. Tras acabar la programación y ejecutarla se ha observado que tiene dos problemas a resolver:

- No muestra el contenido de tarjetas escritas por Arduino. Tras varias comprobaciones y pruebas, se ha verificado que la aplicación no es capaz de mostrar ningún dato escrito por Arduino, ya sea el del Modo Escritura, un simple número o una simple letra. Esto puede ser debido a que el protocolo de lectura de Arduino y de Android sea diferente y por eso mismo no se puede implementar la misma forma de lectura.
- Muestra solamente parte del contenido. Tras probar de grabar una tarjeta mediante Android y pasarla

por la app de lectura, se muestra parte del contenido. Esto puede ser debido a lo comentado en el punto anterior que tal vez se una implementación errónea plantear la lectura como se plantea en Arduino.

6. Conclusiones

Este proyecto empezó siendo la continuación de otro que había logrado una muy buena implementación de un primer prototipo. A día de hoy, mediante este proyecto, se ha podido obtener un prototipo que cumple las expectativas creadas al inicio de él superándolas en algunos casos.

En cuanto al sistema de Arduino, se conseguido un modelo de programación muy simple de utilizar en el que se declara un sistema muy fiable que no permite errores y de utilización sencilla. Se ha pasado de un programa principal a 5 subprogramas (Configuración RTC, Modo Escritura, Modo Lectura, Modo Borrar, Modo Verificar tarjeta vacía). De esta manera se ha podido cambiar de un sistema rocoso y grande como el del Laaroussi, S. a uno más reducido y que en cuanto a hardware aprovecha mejor su memoria. También se ha introducido un método de reposos llamado Sleep Mode mediante el cual duerme el sistema cuando no se es utilizado, de esta manera el sistema consumirá menos.

Se ha cambiado el microcontrolador (Arduino Mega 2560) a uno de menor memoria, (Arduino UNO), dejando memoria libre por futuras mejoras de código, pero sin ser excesiva. También se ha eliminado todo aquello que no se usaba y que ocupaba espacio como el microcontrolador encargado de la comunicación entre el micro y el ordenador y las entradas y salidas digitales y analógicas.

Se han podido solucionar todos los errores que han surgido en cuanto al prototipo v1 en su versión mejorada v2 implementando una nueva PCB. Esta PCB se ha diseñado quitando la resistencia que hacía fallar al zumbador, añadiendo los condensadores para el cristal y los pines para comunicar con el monitor serie.

Se ha podido comprobar que los dos prototipos implementados han sido más eficaces en cuanto a consumo, teniendo así una autonomía mucho mayor a la del prototipo inicial. Estas mejoras también se han visto reflejadas en el precio ya que los prototipos v1 y v2 son mucho más económicos que el inicial.

En cuanto a la aplicación, pese a no poder llegar a leer los datos obtenidos por los prototipos v1 o v2, se ha podido obtener parte la información de las tarjetas grabadas por la aplicación NFC Tools. Se ha obtenido que la implementación del método de lectura de Arduino no ha sido funcional en el de Android debido a que no se ha conseguido mostrar la información de las tarjetas. Cabe decir que a pesar de la existencia de la aplicación NFC Tools y demás, ninguna es capaz de mostrar ningún dato por pantalla mientras sí lo hace de cualquier otro tipo de tarjeta. Es posible que el parte del problema esté en el tipo de tarjeta usado y no la estructura del código.

8. Agradecimientos

Me gustaría agradecer a mi tutor del proyecto, Joaquin Del Rio, toda la ayuda aportada a lo largo de este proyecto. También me gustaría agradecer la ayuda recibida por los compañeros del centro tecnológico.

Referencias

- Bin Abdul, M. (2017). Development of an NFC-based system for control and remote communication. *UPCommons*, 66.
- Laaroussi, S. (2019). Sistema para cronometrarse deportivo basado en tecnología NFC. *UPCommons*, 94.
- Medina Delgado, J. (2017). Diseño e implementación de un prototipo para la identificación de ganado bovino mediante la lectura y escritura de etiquetas con tecnología NFC. *Semantic Scholar*, 56.
- Montero, R. (2017). Estudio de tecnología de comunicación de campo cercano, NFC. *Archivo Digital UPM*, 155.