Escuela Superior
de Ingeniería y Tecnología
Universidad de La Laguna

Trabajo de Fin de Grado

Sistema de triaje basado en tarjetas NFC con Criptografía Basada en Identidad

Triage system based on NFC cards with Identity Based Cryptography

Juan José Gregorio Díaz Marrero

La Laguna, 13 de septiembre de 2019

Dña. **María Candelaria Hernández Goya**, con N.I.F. 78858624-L profesora Titular de Universidad adscrito al Departamento de Ingeniería Informática y Sistemas de la Universidad de La Laguna, como tutora

Dña. **Alexandra Rivero García,** con N.I.F. 78646309V adscrito al Departamento de Ingeniería Informática y Sistemas de la Universidad de La Laguna, como cotutora

C E R T I F I C A (N)

Que la presente memoria titulada:

"Sistema de triaje basado en tarjetas NFC con Criptografía Basada en Identidad"

ha sido realizada bajo su dirección por **Juan José Gregorio Díaz Marrero**, con N.I.F. 78858624-L.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 13 de septiembre de 2019

Agradecimientos.

En primer lugar, quiero agradecer a la directora de este trabajo de fin de grado,

Dra. María Candelaria Hernández Goya, por la dedicación y el apoyo que me ha prestado en este trabajo, al gran respeto que ha mostrado en el planteamiento de mis sugerencias e ideas, así como la dirección y el rigor mostrado a la hora de tomar decisiones ante las adversidades encontradas en este trabajo. También, agradezco el apoyo prestado por los integrantes del Departamento de Ingeniería Informática y de Sistemas, especialmente a Alexandra Rivero García, quién con su conocimiento en la materia, me ha solventado las distintas dudas acaecidas durante el desarrollo del trabajo.

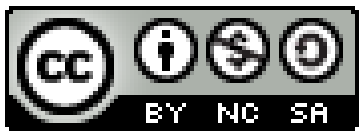
En general agradecer al grupo de Investigación CryptULL, de la Universidad de La Laguna por las pautas proporcionadas para encausar, de la mejor forma posible, este proyecto.

El entorno es muy importante para conseguir nuestras metas, por ello, quiero agradecerles a mis amigos y a mi familia todo el apoyo, tanto moral como humano, que me han brindado, siendo necesario en los momentos complicados, tanto en el transcurso de este trabajo como a lo largo de la carrera.

Para terminar, agradecer a la persona más importante en mi vida, mi madre, Elena Josefina Marrero Pérez, que ha sido mi gran motivación para realizar esta carrera.

A todos, muchas gracias.

Licencia



Resumen

Este trabajo se puede clasificar en dos puntos clave:

El primero, ha sido diseñar e implementar un sistema de clasificación basado en el algoritmo START, a través del desarrollo de una aplicación móvil en Android, cuyo propósito es ser utilizado para clasificar a las víctimas de manera automática, por medio de un sistema de paso de ventanas, con preguntas y respuestas acerca del estado de la víctima. Para hacer esto, se emplearán etiquetas NFC que representarán a las víctimas. Se genera un identificador con una función HMAC, cuyos parámetros de entrada son: el mensaje (estado de la víctima) y la clave secreta (identificador propio de la etiqueta NFC). Una vez que se genera el identificador de etiqueta, se agrega el estado de la víctima representado por un color en formato de texto y las coordenadas geográficas de su posición actual.

El segundo, ha sido desarrollar un sistema web que permita registrar el evento del triaje, almacenando los identificadores generados en la primera fase. Además, el sistema controla el acceso autorizado de personal cualificado, para asistir a la víctima y firmar en la etiqueta NFC. Para la creación de esta firma, el sistema utiliza un algoritmo de cifrado basado en identidad (IBE), que utiliza la clave privada asociada con el identificador de evento.

Palabras clave: START, Triaje, Android, HMAC, IBE, NFC.

Abstract

This work can be classified into two key points:

The first has been to design and implement a classification system based on the START algorithm, through the development of a mobile application on Android, whose purpose is to be used to classify victims automatically, through a step system of windows, with questions and answers about the state of the victim. To do this, NFC tags will be used to represent the victims. An identifier with an HMAC function is generated, whose input parameters are: the message (victim status) and the secret key (identifier of the NFC tag). Once the tag identifier is generated, the status of the victim represented by a color in text format and the geographic coordinates of its current position are added.

The second has been to develop a web system that allows recording the event of the triage, storing the identifiers generated in the first phase. In addition, the system controls the authorized access of qualified personnel, to assist the victim and sign on the NFC label. For the creation of this signature, the system uses an identity-based encryption algorithm (IBE), which uses the private key associated with the event identifier..

Keywords: START, Triaje, Android, HMAC, IBE, NFC.

Índice del Contenido

Índice de Figuras

Índice de Tablas

Capítulo 1
Introducción

En la sociedad actual, el uso de los dispositivos móviles está cobrando cada vez más importancia, tanto en el ámbito social como empresarial. Por ello, para este trabajo de fin de grado, se ha propuesto la idea de implementar un sistema de clasificación de víctimas ante una situación de emergencia o desastre. Este sistema, hará uso de tecnología web y móvil, con el objetivo principal de mejorar el tiempo de respuesta a la hora de clasificar y asistir a las víctimas, tratando de salvar el mayor número de vidas posibles, sin la necesidad de recopilar datos sensibles de las mismas.

El sistema implementado está compuesto por una aplicación móvil y un sistema web.

La aplicación móvil, se encarga de automatizar el proceso de clasificación del estado de las víctimas, por medio de un sistema de paso de ventanas con preguntas y respuestas, usando el algoritmo START. Para almacenar este resultado, utiliza etiquetas NFC, en las que se emplea una función hash HMAC para generar un identificador único, usando como parámetros de entrada el estado de la víctima, (representado con un color en formato texto) y el identificador nativo de la etiqueta. Una vez generado, la aplicación envía las coordenadas geográficas y el identificador hacia el sistema web. Posteriormente, cuando un médico asista a la víctima, para marcarla como asistida, este deberá validarse contra el sistema web con su contraseña de acceso, permitiendo al médico acceder a la pantalla de firma. El proceso de firma, necesita una clave privada, que tendrá que estar almacenada previamente en el dispositivo móvil del médico. Se presupone en este proyecto que la clave viene dada y es generada por un sistema generador de claves privadas.

El sistema web, se encarga de gestionar la creación de médicos autorizados, del control de acceso de los mismos y de la creación de los eventos para el triaje. Además, este sistema contiene una vista donde se visualiza un mapa de Google, con los puntos de colores situados en las coordenadas de las etiquetas, transmitidas desde el dispositivo móvil en un lapso de 24 horas.

1.1 Motivación

Uno de los aspectos más llamativos, es la importancia que han ganado rápidamente las tecnologías en el mundo actual, debido a sus características. Las tecnologías nos permiten realizar labores complicadas en tiempos reducidos, proporcionar información a millones de usuarios en tiempo real, etc. Son tantas sus aplicaciones y sus beneficios, que se han expandido rápidamente a todas las ramas, incluyendo la sanidad. Por ello, resulta interesante plasmar los conocimientos adquiridos a lo largo del grado, implementando un sistema tecnológico para mejorar el tiempo de respuesta de la clasificación de las víctimas ante una catástrofe natural con el objetivo claro de facilitar la asistencia de las mismas.

1.2 Objetivos y competencias del proyecto

En este capítulo, se recogen los distintos objetivos a conseguir para la implementación completa del sistema de triaje, parte de clasificación y parte de asistencia.

La primera parte, consiste en desarrollar una aplicación móvil que permita realizar la clasificación de las víctimas por medio del algoritmo de triaje START, usando etiquetas NFC identificadas unívocamente por medio de un algoritmo de cifrado, sin necesidad de almacenar ningún dato sensible de la víctima.

La segunda parte del sistema, consiste en desarrollar una aplicación web que permita al personal médico asistir a las víctimas de manera óptima, por medio de una ruta, calculada a partir de un algoritmo de camino mínimo a través de un mapa de Google. Para finalizar, este sistema se apoyará en un algoritmo de encriptación basado en identidad para garantizar la identidad del médico que realiza la asistencia.

1.3 Planificación del desarrollo del proyecto

Planificación de la aplicación móvil:

1. Análisis de los sistemas de triaje actuales: comparativas y elección.
2. Implementar el protocolo o algoritmo de clasificación del triaje START en una aplicación Android.
3. Implementar escritura y lectura del resultado del algoritmo de triaje en una etiqueta NFC mediante Android.
4. Implementar función o servicio para obtener coordenadas de geolocalización del dispositivo móvil.
5. Implementar esquema de seguridad HMAC para identificar las etiquetas NFC.
6. Implementación de la comunicación con la aplicación web.

Planificación de la aplicación web:

7. Implementación del servidor para el manejo de datos.
8. Implementar comunicación con el servidor mediante una API RESTFUL para ver los triajes realizados.
9. Añadir esquema de seguridad IBE para identificar personal médico.

1.4 Estado del Arte

Con el fin de encontrar ideas para conseguir los objetivos principales de este proyecto, se investiga la existencia de aplicaciones tecnológicas actuales, desarrolladas para mejorar la calidad de la asistencia en un sistema de triaje, que concluye con el hallazgo de los siguientes proyectos:

- **SATS Mobile Triage:** Creado por el grupo de desarrolladores del proyecto "Open Medicine Project", basado en la escala de triaje de Sudáfrica, que clasifica automáticamente a los pacientes, evaluando los síntomas introducidos que presentan a través de una interfaz gráfica, que utiliza un sistema de paso de ventanas, con preguntas y respuestas para asignar con un color el estado de gravedad de las víctimas [1].

- **Quick Triage App:** Desarrollado por el grupo "JXT Applications, Inc", al igual que el proyecto citado en el punto anterior, se baja en un sistema de preguntas y respuestas para determinar el estado de las víctimas, además esta aplicación se encarga de contabilizar las víctimas [2].

- **CRIMA:** Desarrollado por la compañía "TreeLogic", que utiliza pulseras de distintos colores para clasificar a las víctimas según su estado de gravedad, para posteriormente realizar la asistencia en el hospital más próximo [3].

1.5 Estructura de la memoria

En este apartado se comentan brevemente los capítulos que conforman la estructura de la memoria.

Consta de cinco capítulos:

- Primer capítulo: Se hace una introducción a la solución tecnológica desarrollada para cumplir los objetivos propuestos para este trabajo de fin de grado, se analizan las distintas soluciones actuales que existentes, se explican las motivaciones para desarrollar este sistema, los objetivos propuestos y la planificación a seguir.
- Segundo capítulo: Se describen los antecedentes de los tópicos más importantes, relacionados con este proyecto.
- Tercer capítulo: Se describe la arquitectura hardware y software del sistema, separándolo en dos partes, la parte para la aplicación móvil y la parte para la aplicación web. Se describe la capa de seguridad para cada uno de ellos y la del proceso de desarrollo que se ha seguido en la implementación de cada una de las partes del sistema.
- Cuarto capítulo: Se calcula el presupuesto para realizar este proyecto.
- Quinto capítulo: Se recopilan las conclusiones obtenidas a lo largo del desarrollo de este trabajo, y se añaden posibles mejoras para el sistema implementado.

Capítulo 2
Antecedentes

En este capítulo, se detalla la evolución de las distintas tecnologías utilizadas para el desarrollo del sistema, que intentan resolver los objetivos comentados en el capítulo anterior.

2.1 Tipos de Triaje

Se puede definir triaje, como un proceso que nos permite una gestión del riesgo clínico para poder manejar adecuadamente y con seguridad los flujos de pacientes, donde la demanda y las necesidades clínicas superan a los recursos. Debe ser la llave de entrada a una asistencia eficaz, eficiente y, por tanto, una herramienta rápida y fácil de aplicar, que además posee un fuerte valor predictivo de gravedad, de evolución y de utilización de recursos [4].

En la actualidad, existe una clasificación que distingue entre primeros triajes, cuando se clasifican a las víctimas por gravedad de las heridas evaluando la probabilidad de sobrevivir en unos pocos segundos (START, SHORT o SIEVE), y los triajes con dos fases o más (denominados avanzados), en los que el personal médico evalúa al paciente (MAT, SET o MTS), una vez estos son clasificados, para posteriormente asistirlos [5].

Las comparativas reflejan que los resultados obtenidos entre los distintos tipos de triaje no son muy notables y, más o menos, todos arrojan los mismos resultados de éxito al aplicarlos [6,7]. Por ello, y a raíz de esta investigación, se decide implementar un triaje sencillo como lo es START.

2.2 Android

Es un sistema operativo para dispositivos móviles, creado por Andy Rubin en el año 2003, con el nombre de “Android Inc.”. Su núcleo, está basado en Linux, que es libre, gratuito y multiplataforma. Gracias a su máquina virtual, llamada Darvik, permite ejecutar código escrito en el lenguaje de programación Java, que es un lenguaje interpretado y basado en el paradigma de programación orientado a objetos. Este lenguaje es uno de los más usados en la actualidad, por lo tanto, a la hora de desarrollar aplicaciones para Android, existirá mucha información al respecto, lo que facilitará el desarrollo.

El sistema operativo Android se publica en el año 2005. Para ese entonces, era prácticamente desconocido hasta que Google lo compra en ese año. En el año 2017 llegó a ser el SO para dispositivos móviles más usado (“Figura 2.2”) y actualmente se mantiene como líder en el sector de SO para dispositivos móviles inteligentes, controlando más del ochenta por ciento del mercado [8]. Nos encontramos en la décima generación de dicho sistema operativo. Esta generación contiene numerosas bibliotecas que permiten interactuar con distintas interfaces y, especialmente, tecnologías como: la geolocalización y la comunicación inalámbrica NFC [9], imprescindibles para el desarrollo de la primera parte del trabajo. Cabe destacar que la mayoría de las bibliotecas de la capa de aplicación, en las que Android se apoya, están escritas en C o C++, lo que permite aprovechar los recursos al máximo dado que las llamadas al sistema se pueden controlar de manera directa “Figura 2.1”.

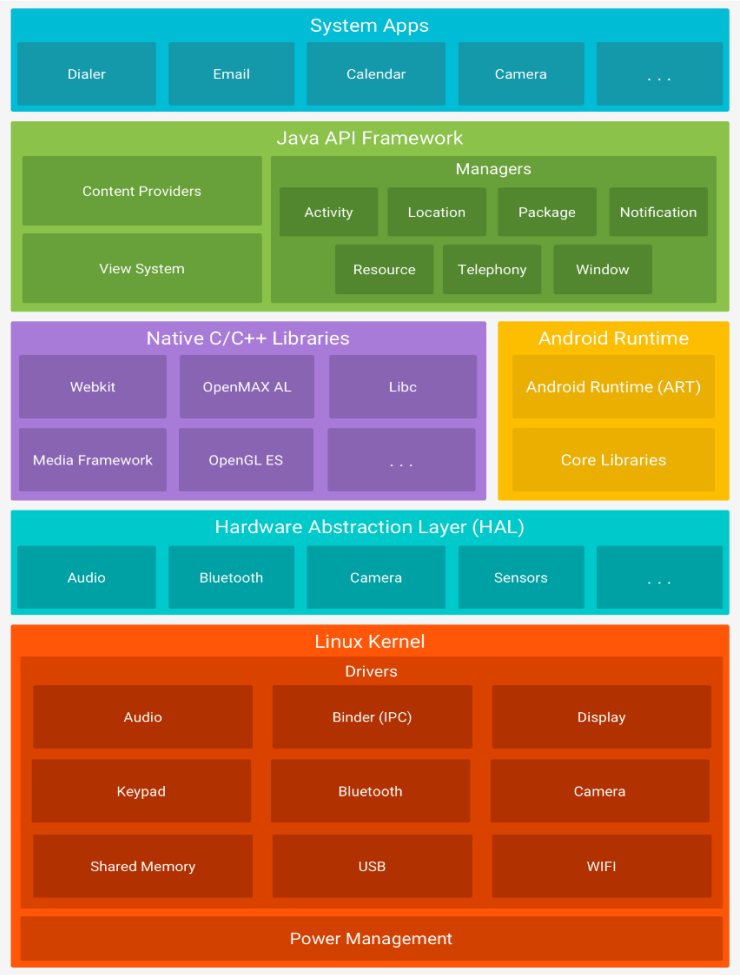


Figura 2.1: Arquitectura de Android.

Figura 2.2: Gráfica mercado de SO para dispositivos móviles año 2017.

Las ventajas de usar Android son:

- Es el sistema más usado en la actualidad para los dispositivos móviles. Por lo tanto, existen muchas bibliotecas implementadas, que permitirán acelerar el proceso de desarrollo de la aplicación móvil, además de proporcionar estabilidad y fiabilidad a la aplicación.
- Es un sistema donde las aplicaciones se desarrollan bajo la licencia de apache 2.0, por lo tanto, puede ser distribuido libremente, ser usado para producir aplicaciones o ser

modificado, sin tener que preocuparse por las regalías de terceros.

- El software desarrollado se puede distribuir por la plataforma de distribución digital propia de Google.
- Tiene una comunidad propia para desarrolladores.

Por las razones comentadas anteriormente, se decide implementar la aplicación móvil encargada de realizar el proceso de clasificación de víctimas, bajo este sistema operativo.

2.3 Tecnología NFC (Near Field Communication)

Near Field Communication (NFC) es una tecnología de comunicación inalámbrica de corta distancia y alta frecuencia. Esta tecnología se encuentra estandarizada bajo ISO 14443 (RFID, estándar para identificación de radio frecuencia) y FeliCa. Específicamente, no es más que una extensión de RFID, con la diferencia, de que esta limita el uso de la misma a una distancia máxima de diez centímetros. Su mecanismo de comunicación se basa en un campo magnético inducido que le permite trabajar en la banda de los 13,56 MHz. Esto implica, que se pueda emplear sin ninguna restricción bajo alguna normativa u obtener alguna licencia para usarla [10,11].

La especificación de la capa física se define en el estándar NFCIP-1, este estándar implica que los dispositivos que se apoyen en esta tecnología soporten básicamente dos tipos de funcionamiento: modo activo y modo pasivo.

- **Modo activo**: El dispositivo emisor y el dispositivo receptor, emiten un campo magnético para transferir los datos entre ambos.
- **Modo pasivo**: Solo uno de los dispositivos emite una fuente electromagnética y el otro dispositivo, aprovecha dicho campo para intercambiar información en él. Este modo de funcionamiento es el empleado por las etiquetas que se usarán para identificar a las víctimas del triaje.

Por otro lado, los modos de comunicación que permite esta tecnología son:

- **Punto a Punto**: Se crea una comunicación entre dos dispositivos para intercambiar datos.
- **Lectura y escritura**: Un dispositivo se encarga de leer la información de una etiqueta o de escribir en ella.
- **Emulación de tarjeta**: El dispositivo que use NFC puede actuar como una tarjeta para que el lector lea su información.

Es en el año 2006 cuando NFC-Forum (la organización encargada de regular los estándares y características de esta tecnología), realiza la primera publicación acerca de NFC. Esta organización empieza a incluir a grandes empresas (Google, Visa, Dell, Intel, Microsoft, Samsung, Sony, At&t, Paypal, Nokia) para extender el uso de NFC, que ven en ella, un potencial para incorporarlos a proyectos en desarrollo o innovar con ella. Actualmente, se siguen descubriendo múltiples usos de la misma y ya son más los dispositivos que son compatibles con ella, como se puede observar en la “Figura 3.1”.

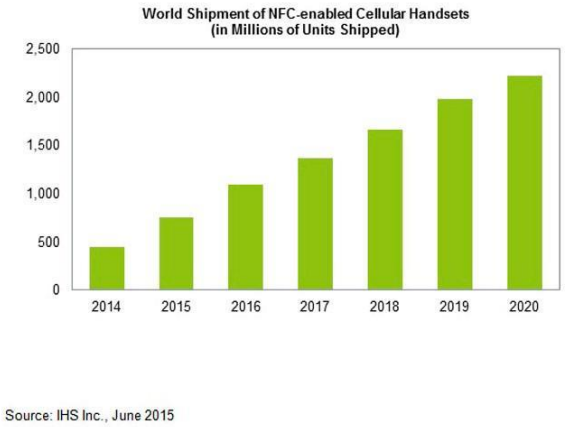


Figura 2.3: Tendencia del uso NFC en dispositivos móviles.

Actualmente, se pueden encontrar infinidades de usos para esta tecnología como pueden ser:

- Almacenar datos de manera cómoda y segura.
- Realizar pequeños intercambios de información por proximidad.
- Guardar código ejecutable para realizar alguna automatización una vez se detecte la etiqueta, por ejemplo, buscar un enlace web automáticamente una vez se lea la información contenida en el dispositivo con NFC.
- Realizar pagos mediante el almacenamiento seguro de la información de identificación del usuario.

Por los efectos prácticos citados anteriormente y la manera en la que trabaja esta tecnología, se decide integrarla en el sistema de la primera parte desarrollado para este trabajo, usando una etiqueta NFC y un dispositivo que sea compatible con dicha tecnología.

2.4 Node.js

Si se comparan los lenguajes más utilizados en el mercado del desarrollo de aplicaciones web, el lenguaje preponderante es JavaScript (estándar ECMAScrip 6) [12]. El problema, es que este lenguaje en su origen fue diseñado para ejecutarse en el navegador del cliente, pero a medida que ha ido evolucionando, se ha creado un entorno de ejecución para interpretarlo en el lado del servidor. Este entorno es Node.js. Node.js fue creado por Ryan Lienhart Dahl y lanzando a mediados del año 2009. No es solo un entorno de ejecución en el lado del servidor para código JavaScript, sino también, una librería escrita en el mismo lenguaje (tiene partes implementadas en C++). Es multiplataforma, su arquitectura se basa en eventos, permite la ejecución de funciones asíncronas, se basa en el motor de Google V8 que permite la construcción de aplicaciones altamente escalables. Los ejemplos de aplicaciones más significativas basadas en este entorno son: Netflix, Linkeln, Ebay y Uber [13].

Sus principales ventajes:

- Permite la ejecución de código de manera asíncrona, sin esperar los procesos de entrada y salida de datos en curso.
- Ejecuta el código muy rápidamente y de manera óptima.
- Tiene mucha documentación asociada.
- Tiene una amplia variedad de bibliotecas implementadas.
- Tiene su propia comunidad de desarrolladores.

- Proporciona su propio gestor de paquetes (npm).

Por las características citadas se decide usar este entorno.

2.4.1 Express

Es un framework propio de Node.js que permite la creación de sistemas web minimalistas, de manera muy fácil y que además son flexibles ya que contiene un abanico de bibliotecas tanto para aplicaciones web como para móvil. Al ser minimalista, este framework no sobrecarga el sistema con funcionalidades extras que se incorporan en otros framework existentes. Al ser creado por la fundación de Node.js, se garantiza el mantenimiento de sus bibliotecas, lo que añade estabilidad y compatibilidad con las nuevas tecnologías [14]. Se decide optar usar este framework por la característica minimalista en la que se basa.

Capítulo 3
Descripción del Sistema

En este capítulo se detalla la composición del sistema, las tecnologías que lo conforman, el desarrollo y el funcionamiento del mismo.

3.1 Hardware

Se detallan los componentes hardware usados para realizar las pruebas con el sistema implementado:

- **Dispositivo móvil**: Es un dispositivo de la marca Sony, modelo Xperia XZ, con una capacidad de 32 GB de eMMC de memoria interna y 3 GB de memoria RAM. Tiene un procesador de 64 bits Qualcomm® Snapdragon™ 820. Compatible con conexiones de cuarta generación. Con el sistema operativo actualizado a la versión 8.0 Oreo [15].

- **Etiqueta NFC**: Es una etiqueta conformada por una EEPROM 924 bytes organizados en 231 páginas, 4 bytes por páginas de los cuales solo 888 bytes son programables para la lectura y escritura. El almacenaje temporal de cada dato tiene una durabilidad de 10 años con 100000 ciclos de escritura. Opera en un rango de frecuencia de 13.56 MHZ con una tasa de transferencia de 106 kbit/s. Soporta el formato NDEF [16].

- **Servidor para el sistema web:** Para la implantación de este sistema web, se utiliza un servicio en la nube, proporcionado por Google, que permite crear una máquina virtual. Esta instancia cuenta con: una memoria RAM (memoria de acceso aleatorio) de 1.7 gigabytes, un almacenamiento de 10 gigabytes, un núcleo virtual de CPU dedicado y un sistema operativo Debian GNU 9.0.

3.2 Aplicación móvil en Android

3.2.1 Descripción de la Arquitectura software

Como se comentó en el capítulo uno, para la realización de este trabajo, se decide desarrollar una aplicación móvil en Android, cuya función es ser utilizada para grabar el estado de las víctimas en etiquetas NFC, y enviar el identificador generado por un algoritmo HMAC junto a las coordenadas geográficas de las víctimas hacia el sistema web. Para esto, se utiliza el entorno de desarrollo integrado “Android Studio”, proporcionado por Google, a través del uso de su herramienta de desarrollo (SDK). Con el objetivo de que esta aplicación sea compatible con el mayor número dispositivos del mercado, se decide optar por Android API 6.0.

Otro de los objetivos de este proyecto es el uso de etiquetas NFC, cabe destacar que esta aplicación sólo podrá escribir y leer en etiquetas compatibles con el formato estándar de datos NDEF, definido por NFC fórum. En este estándar se definen las distintos formatos o encapsulamiento de los datos en el interior de un mensaje, para realizar el intercambio de los datos entre el dispositivo y la etiqueta. El formato estándar, se denomina NDEF, este está compuesto por cargas de un tamaño arbitrario y el tipo asociado al mensaje (URI, MIME, media type, etc). Cada carga puede estar definido por un identificador [17].

Para implementar la aplicación, se utilizan las siguientes bibliotecas:

- **android.Volley:** Se encargará de encapsular el mensaje y gestionar las peticiones por medio del protocolo HTTP para enviar el contenido de las etiquetas al servidor web.
- **android.NFC**: Se encargará de permitir el uso de la tecnología NFC, generar los registros NDEF para almacenar la identificación de la etiqueta encriptada.
- **android.gms**: Es un servicio de Google, que permite obtener las coordenadas geográficas del dispositivo móvil.
- **org.json**: Es la biblioteca usada para gestionar los datos que se enviarán al servidor en objetos JSON. Es un formato de texto estándar y representa datos estructurados en la sintaxis de objetos de JavaScript.
- **javax.crypto**: Es la biblioteca que contiene funciones matemáticas, en las cuáles se encuentra el algoritmo de cifrado HMAC, utilizado para garantizar la unicidad de los identificadores para las etiquetas de las víctimas.

3.2.2 Seguridad

Para la capa de seguridad de esta aplicación se han integrado dos algoritmos de cifrado, HMAC y IBE. El primero, se encarga de garantizar que la identidad de las etiquetas sea única. El segundo, utiliza una clave privada, asociada al identificador del evento (el triaje que se está llevando a cabo), para generar la firma, que más tarde será empleada por el médico en la asistencia de la víctima.

- **HMAC**

En la criptografía, es un algoritmo de construcción específica que es usado para calcular un código de autentificación del mensaje (MAC), que implica una combinación entre una función hash y una clave secreta. Esta función va rompiendo el mensaje en bloques de tamaño fijo y le aplica a su vez función de compresión. La calidad del código generado, depende de las propiedades subyacentes de la función hash utilizada, así como su tamaño, siendo este, igual al de la función hash utilizada [18]. Se pueden usar diferentes tipos de funciones hash iterativas como son: MD5, SHA-1, SHA-256, etcétera.

Su uso comúnmente, se emplea para garantizar la autenticidad de un mensaje transmitido por in canal inseguro, ya que, se envía el hash generado usando una clave secreta, junto con el mensaje, la clave es conocida por el emisor y por el receptor, si el receptor del mensaje utiliza el mismo algoritmo y la misma clave, pero el resultado es distinto, al hash enviado por el emisor, entonces el mensaje ha sido alterado.

En este sistema se utilizó para dos puntos claves: el primero garantizar que la etiqueta es genuina y no se ha visto alterada por algún agente externo del sistema (aunque no se almacenan datos sensibles de la víctima) y para generar identificadores aleatorios que no entraran en colisión, ya que se podría dar el caso de que dos etiquetas tuviesen el mismo identificador, siendo imposible identificar a las víctimas afectadas.

- **IBE**

Es un algoritmo que nace de la idea de Adi Shamir en 1984, en la cual plantea un nuevo sistema de criptografía donde las claves públicas sean cadenas de textos, identificadores de los usuarios (por ejemplo el número de colegiado de un médico, el correo electrónico, etcétera). Pero nadie pudo desarrollar esta idea hasta el año 2001, cuando Dan Boneh y Matthew K. Franklin presentan la primera estructura matemática (que hace uso de emparejamientos bilineales) compatible con esta idea.

Es un algoritmo, que consiste en un cifrado de clave pública, en el que cualquier cadena de caracteres puede servir como clave pública válida. Para ello, es necesario que un servidor generador de claves (PKG, Private Key Generator) genera las claves correspondientes para los usuarios cura identidad es válida para ese servidor. Si un usuario "A" quiere enviar un mensaje a un usuario "B", usando su identidad (pública), primero tiene que contactar con el sistema generador para que este le envíe su clave pública maestra, para poder cifrar el mensaje esta clave junto a la identidad. Cuando el mensaje llega al usuario "B" este contacta con el servidor generador de claves y solicita la clave privada para leerlo. El esquema utilizado para la firma desarrollado en este proyecto, realiza el proceso inverso. El nombre de este esquema criptográfico es "Full-Ident"[19].

3.2.3 Descripción de la implementación

En este apartado se especifican las pautas que se ha seguido para desarrollar la aplicación móvil:

1. Se instala Android Studio y se configura el proyecto.
2. Se diseñan e implementan las distintas ventanas de la aplicación, se agregan estilos propios de diseños, tanto para ventanas apaisadas, como para ventanas en modo retrato.
3. Se agrega el archivo que contiene las cadenas de caracteres para los distintos textos que utilizará la aplicación.
4. Se realiza una clase Serializable para representar la etiqueta y permitir la deconstrucción y reconstrucción de los datos entre las distintas pantallas (entre actividades).
5. Se implementan la funcionalidad de escritura y lectura en las etiquetas NFC. En esta fase, se estudian los distintos formatos y las especificaciones técnicas de las etiquetas NFC. Decidiéndose que el formato empleado será el NDEF. Implementando a su vez, una función que automáticamente de el formato comentado a las etiquetas que se utilicen para el triaje.
6. Se implementa el mecanismo de obtención de las coordenadas del dispositivo integrando el servicio de Google (gms).
7. Se empieza a implementar un mecanismo propio de comunicación con el servidor a través de la clase HttpUrlConnect. Al final se cambia integrando la interfaz Volley al proyecto.
8. Se agrega el algoritmo de encriptación HMAC. No toma mucho tiempo porque viene proporcionado por la propia librería de encriptación de java comentada en el apartado anterior.
9. Una vez se implementa el servidor, se realizan pruebas en las cuales, se envían las etiquetas y se consultan los datos guardados en el servidor, para verificar la integridad de los datos de las mismas.
10. Cuando se termina el sistema web, se procede a agregar una vista en la aplicación móvil para permitir cargar la web desde el dispositivo. En esta vista, se agregan los controles

necesarios para poder cargar la clave privada, desde el almacenamiento local del dispositivo, y realizar la firma, a través de el algoritmo IBE en la etiqueta.

3.2.4 Funcionamiento

Para iniciar el proceso de clasificación, el usuario tendrá que hacer clic en el botón “Iniciar Triage” y luego contestar las preguntas que irán apareciendo por medio de un sistema de paso de ventanas, que sigue el algoritmo de selección START, ilustrado en la Figura 3.1, para determinar el estado de gravedad de la víctima. Una vez llegue a la última ventana, el usuario debe hacer clic en el botón de “Escribir” y acercar el dispositivo móvil para realizar el proceso de escritura, con el color del estado de la víctima en formato texto, las coordenadas geométricas y el identificador único de la misma. Para garantizar la unicidad del identificador de la etiqueta, se genera un código de autentificación de mensajes en clave-hash (HMAC), combinando una función hash con el mensaje construido (color, posiciones geométricas e id de la etiqueta nativo) con una clave secreta. Posteriormente, si se desea ver el contenido de la etiqueta, solo basta con hacer clic en el botón de “lectura”, este botón lanzará un evento que mostrará un cuadro de texto con dicho contenido. Para terminar, si se desea enviar esta información, basta con hacer clic en el botón “Enviar al Servidor”. Si esta información no se puede enviar, por algún tipo de problema, se mostrará un mensaje especificando el tipo del error acontecido.

Como utilidad extra, se ha implementado la opción “Inspeccionar Etiqueta” ubicada en la pantalla principal, que permitirá al usuario, leer una etiqueta NFC y consultar el estado registrado en el sistema web, si lo tuviese.

Si la aplicación la desea usar un médico para el proceso de firma, el procedimiento será el siguiente. Primero, el médico tendrá que validarse contra el sistema web desde la aplicación móvil, haciendo clic en el botón de acceso e introduciendo los datos de acceso, una vez el sistema web emita la respuesta, si esta es positiva, el médico podrá continuar hacia la siguiente ventana, donde aparecerá un botón de carga y un botón para firmar la etiqueta. El médico deberá cargar la clave privada del triaje, haciendo clic en el botón de carga, desde el almacenamiento local del dispositivo, para posteriormente, habilitar el botón de firma y escribir en la etiqueta, lo que implica que la víctima asistida fue atendida en ese triaje.

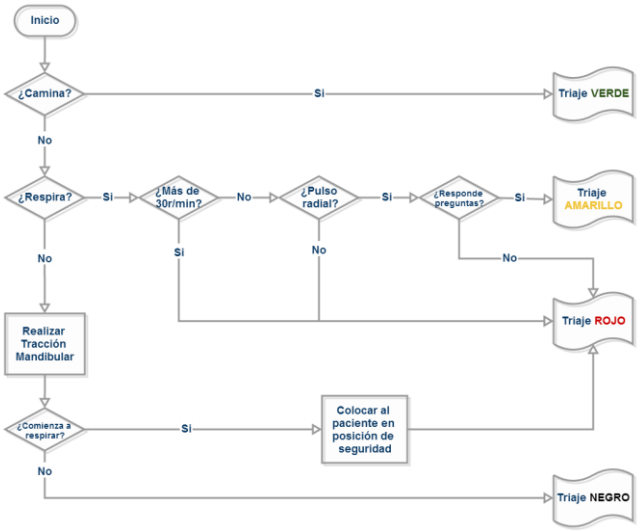


Figura 3.1: Protocolo de clasificación del triaje START.

A continuación, se enumerarán las ventanas del sistema de paso:

- La pantalla principal, donde el usuario tiene la capacidad de elegir entre, leer el contenido de la etiqueta o iniciar un nuevo proceso de clasificación.

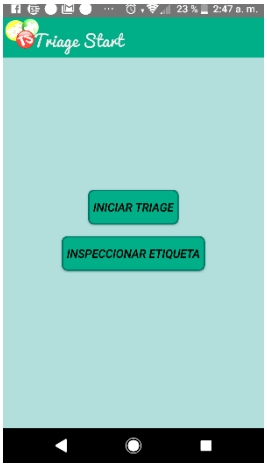


Figura 3.2: Pantalla principal, aplicación móvil

- La pantalla de información, donde se realiza la pregunta.

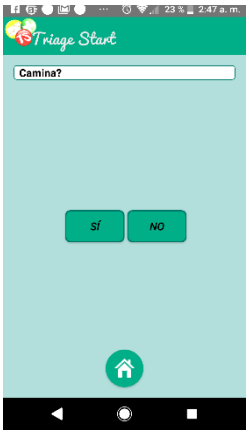


Figura 3.3: Pantalla de información, aplicación móvil

- La pantalla de resultados, se utiliza para escribir en la etiqueta y enviar los datos al sistema web.

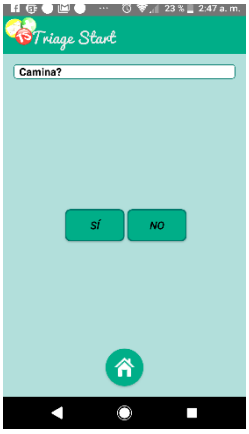


Figura 3.4: Pantalla resultados, aplicación móvil

- La pantalla de resultados donde se visualiza el dialogo de información, con el estado del proceso que se está llevando a cabo.

Figura 3.5: Pantalla resultados, aplicación móvil

- La pantalla de resultados con la información emitida al sistema web.

Figura 3.6: Pantalla resultados enviados, aplicación móvil

- La pantalla de lectura y de comprobación del estado grabado en la etiqueta:

Figura3.7: Pantalla de recuperación del estado desde el sistema web

3.3 Sistema Web

3.3.1 Descripción de la arquitectura software:

Es un sistema desarrollado con la infraestructura Express para Node.js, que se encarga de gestionar la creación y el control de acceso de los usuarios, almacenar las etiquetas recogidas en el triaje y generar nuevos eventos de triaje. Es una infraestructura, que permite un desarrollo minimalista y flexible para el entorno de ejecución Node.js, escrito en JavaScript y construido con el motor V8 de Chrome, el cual, garantiza un gran rendimiento en relación a el procesamiento y la gestión de los datos masivos. Concretamente, la versión de Express 4.17 con Node.js 10.15.

Para poder realizar la implementación de este sistema se usaron las siguientes bibliotecas:7

- **MongoDb 4.0**: Es la base de datos orientada a documentos, que se usará para almacenar las etiquetas en formato JSON.
- **Mongoose 4.13**: Sirve para definir objetos con un esquema fuertemente tipado, que se asigna a un documento MongoDB, además de añadir ciertas funcionalidades para trabajar con estos documentos.
- **Body-parser 1.19**: Transforma el cuerpo de las solicitudes y respuestas en formato JSON.
- **Express-validator 6.2:** Añade funciones para validar los formularios utilizados en el sistema.
- **Errorhandler 1.5:** Añade funciones para el manejo de los errores del sistema.
- **Pug 2.0:** Es un motor generador de plantillas para node.js que convierte el código JavaScript como código HTML.
- **CSS 4.0:** Utilizada para crear estilos para las páginas de la web.
- **Google Map Api:** Permite trabajar con el mapa de Google, usado para visualizar las etiquetas en las coordenadas recibidas, con un color específico, dependiendo del estado de la víctima.
- **Bootstrap 4.0:** Es un framework, que permite diseñar vistas en un modo más flexible para que el sistema pueda ser responsivo.

3.3.2 Seguridad

Para garantizar la seguridad de las contraseñas, de los médicos registrados por el usuario administrador, se utiliza la biblioteca Bcrypt, específicamente la versión 3.0. Es una biblioteca que contiene una función de hashing de contraseñas, diseñado por Niels Provos y David Maxieres, que se basa en el cifrado BlowFish. Esta, incorpora un valor aleatorio que servirá para realizar un hash con la contraseña, y así evitar generar posibles colisiones entre las distintas contraseñas almacenadas en la base de datos. Evitando posibles ataques de fuerza bruta. Cuando se realice el hash de la contraseña, ésta se almacenará en la base de datos.[20].

3.3.3 Descripción de la Implementación

Para desarrollar la aplicación se han seguido las siguientes pautas:

- Debido al desconocimiento en el desarrollo web, se investiga acerca de la arquitectura de la aplicación y de las tecnologías de desarrollo web que más se adapten a los requerimientos de este trabajo.
- Se Instalan y configuran las herramientas de trabajo: Visual Studio Code, Node.js, Git Bash para el control de versiones y el framework Express para generar el esqueleto del proyecto.
- Se diseña y crea la base de datos en MongoDb con los modelos necesarios del sistema. Posteriormente, se integra la librería de mongoose.
- Se configura el sistema para convertirlo en una web RESTful (sin mantener estados entre las pantallas), donde los controladores se encargan de la lógica de las peticiones HTTP y los encaminadores de dirigir las rutas hacia el directorio necesario. El modelo proporciona entidades para representar interacciones con los recursos y las vistas se encargan de mostrar los resultados de las peticiones a los clientes.
- Con todo configurado, se diseñan las siguientes vistas: acceso, registro, gestión de usuarios y gestión del triaje.
- Se añade la lógica necesaria para conectar con la aplicación móvil y almacenar y visualizar en el mapa las etiquetas transmitidas.
- Se investigan los algoritmos de cálculos matemáticos para las rutas entre los pacientes del mapa. Posteriormente, se abandona la idea de usar el algoritmo, ya que, las etiquetas visualizadas en el mapa, podrían perder la referencia de sus coordenadas y no tendría sentido hacer este cálculo.
- Para hacer el sistema responsivo, se añade Bootstrap y se rediseñan las vistas.
- Se añade la lógica necesaria del sistema.
- Se añade el esquema de seguridad para las contraseñas almacenadas en la base de datos.

3.3.4 Funcionamiento

El sistema web estará compuesto por las siguientes interfaces visuales: la de registro del nuevo médico, la de acceso, la de gestión de los usuarios y la gestión del triaje.

A continuación, se detalla la función de cada una:

- Ventana de acceso: Es la interfaz que gestiona el control a usuarios permitidos al sistema. Se encuentra bajo la ruta “/login”, en esta interfaz, el usuario deberá introducir el número del colegiado y la contraseña de acceso, para continuar con el proceso de validación.
- Ventana de registro: Es la interfaz que permite la creación de nuevos usuarios. Se encuentra bajo la ruta “/signup”, para hacer el nuevo registro, basta con rellenar los campos que se visualizan, teniendo en cuenta que el número del colegiado debe ser un número de 9 dígitos con un formato válido. Ya que internamente se valida si el formato del texto es válido para un número de colegiado existente.

- Ventana de gestión de usuarios: Es la interfaz usada por un usuario administrador, que permite añadir, eliminar o modificar a los usuarios médicos validados en el sistema. Se encuentra bajo la ruta “/users”. En esta interfaz aparecerán los médicos listados en una tabla con los controles para añadir, eliminar o modificar algún médico de la tabla.
- Ventana de gestión del triaje: Es la interfaz usada por un usuario administrador, que permite iniciar un nuevo triaje, para asociar las etiquetas transmitidas al sistema en un lapso máximo de tiempo de veinticuatro horas, al identificador de dicho triaje. Además, se visualizarán en un mapa las etiquetas encontradas en ese lapso de tiempo. Se encuentra bajo la ruta “/triageManagement”.

Capítulo 4
Presupuesto

En este capítulo se recogen los costes estimados del trabajo. Se separan en: costes hardware, costes del personal y coste total.

1. Costes hardware y software

Hardware	Unidades	Valor Unitario (eur)
Móvil XPERIA	1	133
Etiqueta NFC	1	0.33
Ordenador Portátil	1	600
Servidor Google Cloud	12	28,27/mes
Total		1.072,54

Tabla 1: Tablas costes hardware

Software	Unidades	Valor Unitario (eur)
Android Studio	1	0
Licencia Android Dev	1	22,0
Total		22,0

Tabla 2: Tabla costes Software

2. Costes del personal

Software	Jornadas	Sueldo/Horas (eur)
Sistema Móvil	30	14
Sistema Web	35	14
Total		3.120,0

Tabla 3: Tabla costes del personal

3. Costes Totales

	euros
Coste Software	1.072,54
Coste Hardware	22,0
Coste Personal	3.120,0
Total	4.214,54

Tabla 4: Tabla coste total

Capítulo 5
Conclusiones y líneas futuras

En conclusión, el sistema implementado agiliza el proceso de selección, dado que las personas que no tienen conocimiento en la materia pueden ayudar con sus dispositivos, lo que implica, que la ventana de asistencia de las personas heridas o graves disminuya. En contra parte, veo difícil la realización de la asistencia en zonas con poca cobertura, ya que sería difícil realizar la geolocalización de los pacientes y, por lo tanto, garantizar la asistencia de todos correctamente. Otro factor en contra, es la precisión que puede alcanzar el zoom del mapa de Google cuando dos personas están a corta distancia, llegados a ese caso, si un médico se acercara a alguna de ellas sería difícil distinguir cuál de las dos ha sido auxiliada, lo cual no permite la realización de un sistema de posicionamiento efectivo.

Por otro lado, este trabajo refleja que la tecnología bien empleada, puede ayudar en gran medida a desarrollar herramientas informáticas, que permitan mejorar la eficiencia a la hora de actuar ante una adversidad.

Como trabajo para líneas futuras, se debería pensar en implementar un algoritmo de cifrado IBE que permita relacionar al paciente con el médico y la temporalidad del triaje, permitiendo realizar un control más específico del triaje, además de garantizar quién identidad del médico que atendió a la víctima. Por ejemplo, se podría usar el número del colegiado con el identificador del triaje, como identidad y asociar las claves generadas para el proceso de firma.

Bajo desde mi punto de vista, he aprendido mucho a lo largo del desarrollo del sistema ya que nunca me había enfrentado al desarrollo de una web y una aplicación que utilizara estás tecnologías, aumentando mi grado de interés en el desarrollo de aplicaciones web y la seguridad en la informática, además de darme cuenta, de que el conocimiento adquirido a lo largo de la carrera, puede servir de mucho para resolver problemáticas que afectan a la salud de las personas.

Summary and Conclusions

In conclusion, the implemented system speeds up the selection process, since people who do not have knowledge in the matter can help with their devices, which implies, the assistance window for people injured or diminished graves. On the other hand, I find it difficult to carry out assistance in areas with little coverage, since it would be difficult to perform the geolocation of patients and, therefore, the assistance of all correctly. Another factor against it is the accuracy that the Google map zoom can reach when two people are at close range, if that is the case, if a doctor approaches one of their difficulties difficult to distinguish from the two that have been helped , which does not allow the realization of an effective positioning system.

On the other hand, this work reflects that well-used technology can greatly help to develop computer tools, improve efficiency when acting in the face of adversity.

As work for future lines, we should consider implementing an IBE encryption algorithm that allows the patient to be related to the doctor and the timing of the triage, to perform a more specific control of the triage, in addition to who the identity of the doctor who attended to the victim. For example, you could use the collegiate number with the triage identifier, as an identity and associate the keys generated for the signing process.

From my point of view, I learned a lot throughout the development of the system, since I had never faced the development of a web and an application that uses, you are, my degree of interest in the development of web applications and security in the computing, in addition to realizing, that the knowledge acquired throughout the career, can be very useful to solve problems that affect the health of people.

Bibliografía

[1] Triage, S. (2019). SATS Mobile Triage for Android - APK Download. Recuperado 7 septiembre, 2019, de https://apkpure.com/es/sats-mobile-triage/sats.triage

[2] App, Q. (2019). Quick Triage App 1.0 Descargar APK para Android - Aptoide. Recuperado 7 septiembre 2019, de https://quicktriage.es.aptoide.com/

[3] EMERGENCIAS. (2019). Recuperado 7 septiembre 2019, de https://www.triajeset.com/productos/emergencias/crima-imv/

[4] Gómez J. Urgencia, gravedad y complejidad: un constructo teórico de la urgencia basado en el triaje estructurado. Emergencias 2006; 18: 156-164ACM

[5] Jiménez, J. Gómez. Clasificación de pacientes en los servicios de urgencias y emergencias: Hacia un modelo de triaje estructurado de urgencias y emergencias. Emergencias, 2003, vol. 15, p. 165-174

[6] Garner, Alan, Lee, Anna, Harrison*, Ken, & Schultz, Carlh. (2001, 29 junio). Comparative Analysis of Multiple-Casualty Incident Triage Algorithms. Annals of emergency medicine, 38(5), 541-548

[7] Keith P. Cross, MD, MS, Mark X. Cicero, MD (junio del 2013). "Head to head comparison of disaster triage methods in pediatric, adult and geriatric patients". Disaster Medicine/Original Research, 668-676

[8] Moscaritolo, A. Angela. (2017, 17 febrero). El 99.6% del mercado móvil le pertenece a Android y iOS. Recuperado 7 septiembre, 2019, de http://latam.pcmag.com/sistemas-operativos-moviles/18490/news/el-996-del-mercado-movil-le-pertenece-a-android-y-ios

[9] Arquitectura de la plataforma Android Developers. (2019). Recuperado 7 septiembre 2019, de https://developer.android.com/guide/platform?hl=es-419

[10] Near Field Communication Technology Standards. (s.f.). Recuperado 7 septiembre, 2019, de http://nearfieldcommunication.org/technology.html

[11] Near field communication. (s.f.). Recuperado 7 septiembre, 2019, de https://es.wikipedia.org/wiki/Near_field_communication

[12] JavaScript vs PHP vs Ruby | What are the differences?. (2019). Recuperado 7 septiembre 2019, de https://stackshare.io/stackups/javascript-vs-php-vs-ruby

[13] Introducción a NodeJS (JavaScript del lado del Servidor) - Oscar Blancarte - Software Architecture. (2019). Recuperado 7 septiembre 2019, de https://www.oscarblancarteblog.com/2017/05/29/introduccion-a-nodejs-2/

[14] Express - Infraestructura de aplicaciones web Node.js. (2019). Recuperado 7 septiembre 2019, de https://expressjs.com/es/

[15] XZ, X., & XZ, X. (2019). Especificaciones de Xperia XZ - Sony Mobile (España). Recuperado 7 septiembre 2019, de https://www.sonymobile.com/es/products/phones/xperia-xz/specifications/

[16] (2019). Recuperado 7 septiembre 2019, de http://circuitosdigitalesdemexico.com.mx/wp-content/uploads/2018/11/CDMEX-ME-

RNXX-0006.pdf

[17] NFC Forum Technical Specifications. (s.f.). Recuperado 7 septiembre, 2019, de https://nfc-forum.org/our-work/specifications-and-application-documents/specifications/nfc-forum-technical-specifications/.

[18] Villanueva, J. (2019). What Is HMAC And How Does It Secure File Transfers?. Recuperado 10 septiembre 2019, from https://www.jscape.com/blog/what-is-hmac-and-how-does-it-secure-file-transfers

[19] Chatterjee, S., & Sarkar, P. (2011). Identity-based encryption. New York: Springer.

[20] Vicente, D. (2019). Encriptación de password en NodeJS y MongoDB: bcrypt. Recuperado 7 septiembre 2019, de https://solidgeargroup.com/password-nodejs-mongodb-bcrypt/?lang=es