Escuela Técnica Superior de Ingeniería de Telecomunicación

Universidad Politécnica de Cartagena

UNIVERSIDAD POLITÉCNICA DE CARTAGENA

Escuela Técnica Superior de Ingeniería de Telecomunicaciones

Identificación electrónica mediante Tarjeta Universitaria Inteligente (TUI) utilizando una Raspberry Pi.

TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA TELEMÁTICA

Autor: Jesús Navarro Zaplana

Director: Mathieu Kessler

Codirector: Daniel Pérez Berenguer

Cartagena, a 10 de diciembre del 2020

Autor	Jesús Navarro Zaplana
E-mail del autor	Jesus.nav.zap@hotmail.com
Director	Mathieu Kessler
Codirector	Daniel Pérez Berenguer
Email director y codirector	mathieu.kessler@upct.es daniel.perez@upct.es
Tirulo del TFG	Identificación electrónica mediante Tarjeta Universitaria Inteligente (TUI) utilizando una Raspberry Pi

Resumen

En este trabajo se ha desarrollado un sistema de autenticación capaz de reconocer tecnologías NFC y de leer y decodificar códigos QR generados a través de la aplicación AppCRUE desarrollada por la CRUE. El objetivo de este proyecto es incorporar el sistema desarrollado a la plataforma UPCTstream.

Para ello, se comienza realizando un estudio previo de las tecnologías existentes, así como de las características de las salas de UPCTstream.

El sistema propuesto permite mejorar la seguridad, fiabilidad y tolerancia a fallos de la autenticación en las salas UPCTstream. Además, se evita el contacto con los dispositivos táctiles de las salas UPCTstream, mejorando las condiciones higiénico-sanitarias impuestas por el estado actual de pandemia COVID-19.

Titulación	Grado en Ingeniería Telemática
Intensificación	
Departamento	Centro de Producción de Contenidos Digitales
Fecha de presentación	Diciembre - 2020

Agradecimientos

A mi familia, por todo el apoyo y la fuerza que me ha dado durante este periodo de mi vida.

A mis amigos, que siempre me han tendido una mano

A todos mis compañeros del CPCD, que me han ayudado y de los que he aprendido tanto y, en particular, a Daniel Pérez y Mathieu Kessler que me han dado la oportunidad de realizar este gran trabajo.

A Todos, simplemente. **GRACIAS.**

Índice General

1. Introducción

1.1 Motivación

El uso de sistemas de streaming y videoconferencia para docencia online o semipresencial han cobrado especial importancia en el contexto actual de pandemia generado por el COVID-19.

Las universidades han realizado un importante esfuerzo en modernizar sus infraestructuras adoptando nuevas tecnologías que les permitan adaptarse a la situación actual. Esas medidas van, desde optar por clases cien por cien online a modalidades semipresenciales, en la que la asistencia a clase es optativa, bajando así los aforos para poder cumplir las normativas de seguridad sanitaria.

En este sentido el **Centro de Producción de Contenidos Digitales** (CPCD) de la **Universidad Politécnica de Cartagena** (UPCT) ha trabajado en los últimos años en el desarrollo de herramientas que faciliten al docente su tarea.

UPCTstream es uno de los proyectos desarrollados en este contexto, a partir del cual todos los profesores de la UPCT pueden retransmitir en directo su clase al mismo tiempo que queda grabada y publicada en su perfil de *UPCTmedia*, otro de los sistemas desarrollado en el CPCD para el almacenamiento de contenido multimedia por parte del profesorado, y desde donde podrá compartir ese material con estudiantes o con otros docentes a través del Aul@ Virtual.

De este modo, podrían cumplirse de una forma efectiva las modalidades completamente online y semipresenciales, al proporcionar a aquellos que opten por no acudir al centro, de clases en streaming, con opción incluso a poder realizar preguntas a través del chat de UPCTstream. O bien poder reproducir estas clases grabadas en cualquier momento a través de Aul@ Virtual, dando una completa libertad de horarios a los alumnos.

Llegando incluso a la posibilidad de realizar los exámenes de forma online, como sucedió en la convocatoria de junio 20/21, donde debido a la situación de la pandemia, la gran mayoría de exámenes realizados a alumnos de la UPCT fueron de forma online, gracias a la aplicación *UPCTevalúa* desarrollada por el CPCD.

El sistema de UPCTstream está compuesto de un agente de captura que permite capturar hasta cuatro entradas SDI, una entrada HDMI y una entrada VGA. El agente de captura permite capturar para grabación y/o emisión en streaming dos de estas señales. Por ejemplo, el ordenador del docente y la cámara del aula.

Figura 1.1 Ejemplo de grabación/streaming dual UPCTstream

En las salas UPCTstream designadas, podemos encontrar una Raspberry Pi conectada a una pequeña pantalla táctil. Además, disponemos de un monitor para ver las imágenes de la Cámara y la imagen que nos llega desde el ordenador instalado en el aula o el portátil que se haya conectado al sistema. Estas serán las imágenes que se graben y se suban al perfil de UPCTmedia o se emitan en directo.

Figura 1.2 Ejemplo Aula Stream, situada en la Facultad de Ciencias de la Empresa. Destacando la Raspberry sobre el resto de la imagen.

En nuestra Raspberry, podremos seleccionar la posición en la que estará la cámara al inicio de la grabación, el título que esta tendrá en nuestro perfil de UPCTmedia y el paso más importante para este trabajo dado que es aquí donde intervendremos para modificar el sistema, autenticarnos con nuestro DNI.

Actualmente para introducir el DNI del usuario que va a usar la sala, se usa un teclado virtual que aparece en la misma pantalla táctil. Esto presenta grabes fallos de seguridad en el sistema, dado que solo con el conocimiento de este DNI podríamos acceder al sistema y comenzar la grabación/streaming, lo que permite la suplantación de un usuario. Además, teniendo en cuenta la situación COVID-19, debemos limitar el uso de la pantalla táctil lo máximo posible para mejorar la seguridad sanitaria de la sala.

Por ello, la motivación principal en este TFG ha estado dirigida al diseño e implementación de un sistema de autenticación para las aulas de UPCTstream, mediante el uso de tecnologías que ayuden no solo a simplificar el trabajo de estos docentes, si no a hacerlo más seguro. Corrigiendo o mitigando el problema de suplantación y creando una interfaz que requiera de un contacto mínimo con los dispositivos del sistema.

Para este propósito, haremos uso de las Tecnologías de NFC y códigos QR, que no solo garantizaran que el usuario es quien dices ser, sino que lo hará sin tener contacto alguno con los dispositivos colocados en la sala. Estas tecnologías, tienen como característica principal que su transferencia de datos se realiza sin necesidad de poner en contacto físico directo los dispositivos implicados en la transmisión de la información.

Actualmente la escuela cuenta con diez aulas/salas dotadas con equipos de grabación y retransmisión.

1.2 Objetivos

En la actualidad todas las salas UPCTstream están dotadas, además de los diferentes equipos de grabación, de una Raspberry Pi con una pequeña pantalla táctil desde la cual se identifica al usuario tecleando su DNI en esta pantalla y se le dota de las herramientas para iniciar/parar los diferentes procesos.

El presente trabajo se centra en el desarrollo de un sistema que permita la identificación electrónica en las salas UPCTstream de la UPCT. Para ello se fijan los siguientes objetivos.

1. Diseño e implementación de lectura de tarjetas Near-field communication (NFC) integrada en las Tarjetas Universitaria Inteligente (TUI).
2. Diseño e implementación de una arquitectura basada en servicios REST que permita obtener los datos del docente identificado a partir de los datos obtenidos en la lectura de la TUI.
3. Activación del sistema de grabación y retransmisión en streaming de las salas UPCTstream en función de la autenticación realizadas y del sistema de reservas.

Una vez iniciado el proyecto, el Santander anunció que dejaría de soportar la emisión de las tarjetas TUI tal y como venía realizando en los últimos años. Esto podría provocar que el uso de la TUI se fuera reduciendo gradualmente quedando para casos de uso específicos. Además, la Conferencia de Rectores de las Universidades Españolas (CRUE), anunció el desarrollo de una aplicación, AppCRUE, para la implantación de la TUI virtual. Estos dos hechos nos llevaron a añadir los siguientes objetivos al proyecto sin renunciar a los anteriores:

4. Desarrollo de un sistema para lectura de QR dinámicos mediante cámara conectada a Raspberry Pi.
5. Modificación de la arquitectura diseñada en el objetivo 2, para permitir la verificación de códigos QR a través de llamadas a un servicio REST proporcionado por la aplicación AppCRUE.

2. Estado del arte

2.1 Análisis de aplicaciones similares

En la actualidad, hay infinidad de sistemas que utilizan **NFC** para identificar al usuario de uno o varios servicios contratados: Tarjetas de crédito, abonos de transporte o incluso sistemas de seguridad de tiendas físicas. Se hace o bien a través de una tarjeta física o con un dispositivo electrónico.

En este proyecto se ha utilizado una tarjeta física TUI para identificar al usuario, pero estas tarjetas físicas se utilizan para todo tipo de servicios que requieres el comprobar la identidad del usuario.

Uno de los ejemplos más utilizados en el día a día, es el pago por Contacless. Con este tipo de pago puede llegar a realizar compras de hasta 20€ con solo acercar tu tarjeta al datafono, pero si realizas una compra de más de 20€ has de introducir también tu PIN. Dado que este proceso evita el contacto con el datafono o con el personal del servicio a adquirir, durante la pandemia de COVID-19 este límite de 20€ aumento hasta 50€, con el objetivo de mantener el menor contacto posible entre cliente y vendedor.

Figura 2.1 Tarjeta bancaria

Este paso es una muestra de la cierta inseguridad de este proceso, dado que para el sistema el simple hecho de poseer dicha tarjeta te hace ser el propietario de esta, dejando un fallo de seguridad en el caso de la posibilidad de la sustracción por parte de un tercero de nuestra tarjeta, y dándole pleno acceso al sistema sin ser el auténtico usuario. Por eso se generó un nuevo paso en los gastos mayores de cierta cantidad, añadiendo la necesidad de introducir un PIN que solo el auténtico usuario debería conocer.

Dado que actualmente la gran mayoría de smartphones, incorporan la tecnología NFC entre sus características. Los bancos trasladaron este servicio a sus aplicaciones de banca móvil dando como resultado un paso más en la seguridad, importando la seguridad (PIN, patrones, huellas, etc.) propia de un smartphone al proceso de pago por Contacless.

Este tipo de autenticación por NFC se usa en muchas empresas como método para acceder a ciertas zonas restringidas a cierto personal. De hecho, en el edificio ELDI Edificio de Laboratorios de Investigación) de la UPCT, este sistema se utiliza para abrir las puertas de despachos y aulas. Las cerraduras electrónicas se abren con el uso de las TUI del personal o ante las tarjetas RFID designadas. Este sistema sigue siendo vulnerable a los duplicados o substracciones por parte de terceros.

Con respecto al uso de códigos **QR dinámicos** de un solo uso, tras una exhaustiva búsqueda de aplicaciones similares, podemos decir que esta tecnología apenas tiene usos significativos en el mercado o en funciones públicas.

La mayoría de los usos de códigos QR están ligado a la lectura de enlaces para redirigir a webs donde se explica o da un servicio en concreto. Y estos usan códigos QR estáticos para esa función por lo que se aleja de nuestro propósito.

Cabe destacar el uso que realizan en diferentes universidades, por ejemplo, la Universidad de Murcia (UMU), dispone de códigos QR estáticos en las puertas de sus aulas, estos han de ser leídos por los alumnos al comienzo de la clase y así se registra su presencia en esa aula durante esa clase. Pero al ser un QR estático, de forma maliciosa, con una foto de este podríamos realizar el proceso de registro sin estar en el aula, y provocar datos falsos sobre la asistencia o aforo de esa clase.

En [1], podemos consultar toda la información que da la UMU sobre la implementación de este proceso.

También en la universidad de Cantabria, han usado un sistema muy similar, pero extendiendo esto a todos sus servicios desde las aulas hasta los comedores, para así conseguir una mayor trazabilidad de sus usuarios e identificar posibles contactos de riesgo. Pero al seguir siendo QR estáticos, estos tienen el error descrito anteriormente.

En [2], podemos consultar toda la información proporcionada por la universidad sobre su sistema.

Por lo tanto, y una vez analizado los sistemas similares podemos comprobar que ambos casos se usan códigos QR estáticos que los usuarios leen para registrar su asistencia ante el sistema.

Estas medidas se llevan a cabo con el objetivo de poder aplicar los protocolos sanitarios establecidos para el control del COVID-19. Con lo que un uso malintencionado de este sistema podría llevar a actuaciones erróneas en esos protocolos.

3. Análisis

El proyecto se centra en buscar un método para que el proceso de autenticación del personal docente, a el sistema UPCTstream en las aulas designadas, sea lo más simple para ellos y así les resulte menos complejo usar el sistema. En concreto, buscamos automatizar a la vez que securizar este proceso.

Durante el desarrollo de este trabajo, los objetivos se vieron alterados por el anuncio de la retirada de las tarjetas TUI físicas. Por lo que, se decidió dividir este proyecto en dos fases no excluyentes entre sí, que fueran capaces de cumplir con los objetivos actuales y futuros de una manera eficaz.

3.1 Fase 1

Como idea original, este trabajo comenzó proponiendo el uso de la tecnología NFC integrada en las TUI físicas de la UPCT. Todo el personal de la universidad tiene acceso a una de estas tarjetas, su emisión es gratuita y te autentifica para usar cualquier servicio restringido al personal de la UPCT.

Figura 3.1 Imagen TUI física

Con esta tarjeta, eliminaríamos la acción de teclear el DNI en la pantalla de la Raspberry, además, nos aseguramos de que la persona es la que dice ser.

En [4] podemos encontrar el TFE, de una estudiante de la UPCT, en el cual podemos informarnos al completo de todas las funciones de la TUI física de las cuales se hace mención.

Aunque el uso de estas está ya normalizado en la comunidad universitaria, para acceder a la biblioteca o incluso como identificador durante un examen. La tecnología NFC integrada en la TUI, apenas tiene uso dentro de la universidad. Por lo que, para llevar a cabo este uso fue necesaria la ayuda de la unidad de informática de la UPCT, que desarrollo una api para tratar los datos leído de la TUI (este proceso será detallado en el apartado “Diseño”).

Para el uso de esta tecnología, deberíamos incluir un lector de NFC en las Raspberry instaladas en las aulas de UPCTstream.

3.2 Fase 2

Durante el desarrollo de la Fase 1, se anunció, desde la **Conferencia de Rectores de las Universidades Españolas** (CRUE), el encargó de una nueva aplicación para smartphones llamada "AppCRUE" (Esta aplicación será explicada detalladamente en la sección de Tecnologías "AppCRUE"), la cual entre sus funciones incluye tener una tarjeta TUI virtual.

Figura 3.2 Menú inicio AppCRUE UPCT

Desde esta App, si tu smartphone dispone de la tecnología NFC integrada en el dispositivo, se está planteado el poder usarlo como si fuera la TUI física, aunque aún no ha sido implementado por parte de los desarrolladores de la AppCRUE, pero el cual la haría compatible con todo lo ya desarrollado en la Fase 1.

Lo interesante de esta App es que incluye una nueva función a esta TUI virtual. Además del QR estático ya incluido en el reverso de la TUI física, en la virtual tenemos un sistema de generación de códigos QR dinámicos de un solo uso y unipersonales (Tecnología descrita detalladamente en el apartado de Tecnologías "QR"), es en esta función en la que se basará la Fase 2, dado que nos proporcionará un nivel de seguridad muy alto a la hora de autenticar al usuario del sistema UPCTstream.

Para realizar esta fase es necesario añadir una pequeña cámara a las Raspberry de las aulas/salas UPCTstream, con el único objetivo de leer los QR que se les proporciona.

Aunque tanto en sistema de la Fase 1 como el sistema de la Fase 2 no son excluyentes entre sí y ambos han sido desarrollados. Es el sistema de la Fase 2, es el que se ha priorizado y el que se implantará en las salas de UPCTstream, puesto que, es este el que nos garantiza una autenticación más fiable ante el sistema y, dado que las tarjetas TUI físicas van a ser retiradas, el sistema de la Fase 2 va a ser el único funcional por el momento.

4. Diseño

En este apartado, vamos a estudiar las herramientas usadas en nuestro sistema para llegar al diseño de una solución óptima de los objetivos de este trabajo.

4.1 Tecnologías

Aquí explicaremos con detalle todas la tecnologías y herramientas usadas en nuestro trabajo, con el fin de dar una visión amplia de los conocimientos necesarios para llevar a cabo el diseño de nuestro sistema.

4.1.1 Raspberry Pi

Este es el dispositivo sobre el que hemos basado todo este proyecto. Se trata de una serie de ordenadores de placa simple, en la que se intenta reducir al máximo su coste y tamaño para poder llegar a una potencia y capacidad de cómputo muy parecida a la de los ordenadores domésticos, pero a un precio muchísimo más reducido.

Desde su lanzamiento en 2012, dado su bajo coste, buscaba el poder poner en manos de las personas con menos recursos el poder de la informática para que pudieran llevar a cabo sus proyectos digitales, más en concreto su objetivo era la enseñanza de informática en las escuelas. Aunque su rápida popularización hizo que acabara extendiéndose a otros sectores, como el de la robótica.

Su sistema operativo “Raspberry Pi OS” es de código abierto, siendo este una versión adaptada de Debian, un sistema basado en Linux. Aunque desde su lanzamiento se han llegado a portar otro tipo de sistemas, incluso hay una versión de Windows 10 para Raspberry.

Mas en concreto, el modelo usado en este desarrollo es una **Raspberry Pi 3 modelo B+.**

Figura 4.1 Raspberry Pi 3 modelo B+

En la Bibliografía se pueden consultar el datasheet en [3]. Con las especificaciones técnicas, entre las que podemos destacar que las dimensiones son: 82 mm x 56 mm x 19,5 mm y apenas un peso de 50 gramos.

De estas especificaciones las que más nos interesan, son que disponemos de un puerto "**MIPI DSI display port**", al cual va instalada nuestra pantalla táctil. También, tenemos acceso a un "**GPIO de 40 pines**" por el cual conectaremos nuestra placa PN532, que nos dotara de la capacidad de leer tarjetas NFC, necesario para la Fase 1. Un puerto de "**MIPI CSI camera port**" para poder hacer uso de una pequeña cámara con la que leeremos los códigos QR. Y, por último, un puerto ethernet con el que conectar esta Raspberry a nuestro sistema.

4.1.2 AppCRUE

Tras el anuncio de la retirada del soporte para la emisión de tarjetas TUI físicas, nos vimos obligados a buscar alternativas para autenticación de los usuarios. Desde el CRUE se anunció el desarrollo de una aplicación que pretendía sustituir estas tarjetas por una aplicación para smartphones, es decir, pasar del formato físico actual al formato digital.

Figura 4.2 Logo AppCrue

Este Proyecto **AppCRUE**, comenzó en septiembre de 2015 cuando el comité permanente de CRUE aprobó la creación de una plataforma móvil para todas las universidades españolas, orientada a todos los colectivos de la universidad: Alumnos, Personal Docente e Investigador (PDI) y Personal de Administración y Servicios (PAS).

La aplicación fue desarrollada para todas las universidades pertenecientes al CRUE, pero el sistema permite que cada institución la personalice en función de sus necesidades. Se puede personalizar toda imagen corporativa de su propia aplicación, del nombre y publicación que quiera darle.

La AppCRUE, permite a las Universidades crear y potenciar un innovador modelo de relación con sus alumnos, profesores, investigadores y personal de administración a través de los teléfonos móviles, ofreciendo un catálogo de servicios que se irá ampliando progresivamente con el trabajo conjunto de todos.

Algunos de los servicios integrados actualmente son: Servicios académicos para sus alumnos, como consulta de notas, calendarios académicos, etc. También podemos consultar datos administrativos o extracurriculares, servicios generales de la CRUE, comunicaciones institucionales, notificaciones personales, **Tarjeta Universitaria Inteligente (TUI)** y sus servicios asociados entre ellos los generadores de **QR dinámicos**, ofertas exclusivas para estudiantes y servicios financieros del Banco Santander.

Cabe destacar que todo el desarrollo tecnológico del proyecto es colaborativo, compartido y libre de licencia para las universidades pertenecientes al CRUE con el fin de que una mejora pueda ser aplicable a todas ellas.

Por lo que, podemos esperar la implementación de nuevos servicios por parte de los desarrolladores de cada universidad.

4.1.3 NFC

Near-field communication o en sus siglas NFC, significa **comunicación de campo cercano**, se basan en una tecnología de comunicación inalámbrica, de corto alcance y alta frecuencia en la banda de los 13,56MHz, que permite el intercambio de datos entre dispositivos a una distancia de entre cuatro a veinte centímetros.

La tasa de transferencia de datos entre dispositivos que usan esta tecnología es muy baja y solo puede alcanzar diversas velocidades comprendidas entre: 106, 212, 424 u 848 Kbit/s. Por lo que, su enfoque es más para la transmisión de pequeñas cantidades de datos, por ejemplo: identificación y validación de equipos/personas.

Esta tecnología se comunica mediante inducción en un campo magnético, donde dos antenas de espiral son colocadas dentro de sus respectivos campos cercanos para la transferencia de datos. Por lo que, en la tecnología NFC podemos encontrar dos tipos de dispositivos según su implicación en la transferencia de datos:

- **Activo**, en este modo ambos equipos generan un campo electromagnético e intercambian datos.
- **Pasivo**, de esta forma solo hay un dispositivo activo y el otro aprovecha ese campo para intercambiar la información.

Este último modo Pasivo, es el que utilizaremos para la transferencia de datos, dado que la antena insertada en la TUI es una antena pasiva, también llamada “tag” RFID (esta tecnología es de la se derivó el actual sistema de comunicaciones NFC), no tiene alimentación eléctrica propia y se alimenta de la señal magnética que les llega de la antena activa, que funciona como lector de la información contenida en la TUI.

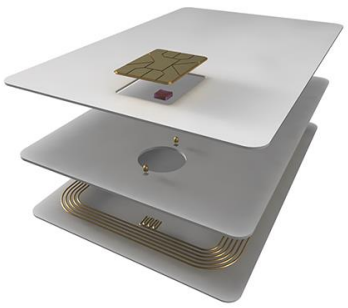

Figura 4.3 Imagen interior de una TUI

En este trabajo utilizaremos la información de nuestra TUI para identificar al usuario y darle acceso al servicio.

4.1.4 QR

Las siglas de **QR** provienen del inglés "**Quick Response**", que significan **respuesta rápida**, por lo que podemos definir los códigos QR como códigos de respuesta rápida.

Los Códigos QR son una evolución de los códigos de barras que constan principalmente, de una matriz bidimensional de módulos de dos colores contrastados, en principio blancos y negros. Hay varias versiones de códigos QR con diferentes módulos que admiten diferentes tamaños de información.

Una de las utilidades de estos códigos es que no es necesario que se compongan únicamente de módulos blancos y negros, esto hace que sean personalizables mientras mantengan los colores suficientemente contrastados, para que sigan siendo legibles por los sistemas de lectura.

Figura 4.4 Códigos QR de con diferentes diseños

Existen desde 1994, su uso se popularizo en Japón alrededor del 2000. Pero gracias a este nivel de personalización y dada la facilidad a la hora de generar estos códigos. La utilización de este sistema se abrió al uso comercial y no solo a el uso industrial. Siendo en el 2010 cuando estos se comenzaron a usar en Estados Unido y Europa. Hoy en día, su uso es muy común en campañas de publicidad, educación, turismo, etc.

En este trabajo usaremos la AppCrue para solicita un código QR dinámico cifrado de un solo uso y que caduca en unos treinta segundos, con lo que conseguiremos autenticar al usuario de forma segura para usar nuestro servicio.

4.1.5 Lenguajes de programación

Dados los objetivos a cumplir, el entorno de trabajo necesario para crear este sistema debe de trabajar con varios lenguajes de programación, que se comunicaran entre sí para forma una plataforma que satisfaga los objetivos de una forma eficiente a la vez que eficaz.

En consecuencia, en esta sección daremos unas nociones básicas que nos permita ver el entorno tecnológico, en el que se ha desarrollado esta plataforma.

4.1.5.1 Desarrollo web

SQL

SQL que viene de las siglas en inglés de Structured Query Language. Es un Lenguaje de Consulta Estructurado. Un tipo de lenguaje de programación que te permite manipular y descargar información de una base de datos. Tiene capacidad de hacer cálculos avanzados y álgebra, para realizar estas consultas.

Es utilizado en la mayoría de las empresas que almacenan su informacion en una base de datos. Ha sido y sigue siendo el lenguaje de programación más usado para bases de datos relacionales.

PHP

PHP proviene de su acrónimo en inglés: Hypertext Preprocessor es un lenguaje de código abierto muy popular, especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.

El código de PHP es ejecutado en el servidor, generando HTML y enviándolo al cliente.

HTML5

HTML es un lenguaje de marcado que se utiliza para el desarrollo de páginas de web. Se trata de la siglas que corresponden a HyperText Markup Language, es decir, Lenguaje de Marcas de Hipertexto.

Sirve para definir el contenido de una página web, como el texto, la imágenes, los videos, etc.

CSS3

CSS podría definirse como un tipo de lenguaje que permite definir y crear la presentación de un documento ya estructurado y escrito en un lenguaje de marcado como puede ser HTML. Es decir, permite generar el diseño visual de páginas web e interfaces de usuario.

JavaScript

JavaScript es un lenguaje poderoso, capaz de aportar soluciones eficaces en la mayoría de los ámbitos de la tecnología.

Es especialmente importante porque es el único lenguaje de programación que entienden los navegadores, con el que se desarrolla la parte de la funcionalidad "frontend" en sitios web y aplicaciones web modernas.

Ajax

La tecnología AJAX (Asynchronous JavaScript + XML) consiste esencialmente de actualizar el contenido de una página accediendo al servidor, pero sin recargar la página.

4.1.5.1 Desarrollo software

Python

Python es uno de los lenguajes de programación dinámicos más populares que existen entre los que se encuentran Perl, Tcl, PHP y Ruby.

Aunque es considerado a menudo como un lenguaje "scripting", es realmente un lenguaje de propósito general. En la actualidad, Python es usado para todo, desde simples "scripts", hasta grandes servidores web que proveen servicio ininterrumpido 24×7.

C

C es un lenguaje compilado, es decir, convierte el código fuente en un fichero objeto y éste en un fichero ejecutable. Es un lenguaje de propósito general y estructurado, ya que permite crear procedimientos en bloques dentro de otros procedimientos.

4.2 Diseño de la solución

Aquí trataremos de forma más exhaustiva los procesos y técnicas usadas para desarrollar la infraestructura, así como todos los pasos que fueron llevados a cabo para el funcionamiento del producto final.

En ambas fases del proyecto el objetivo común a conseguir es asociar lo datos de entrada, tanto los recibidos por el lector NFC (Fase 1) como por la lectura del QR (Fase 2), con el DNI de la persona que quiere autenticarse ante el sistema.

4.2.1 Fase 1

En esta Primera Fase como se ha explicado en apartados anteriores, llevaremos a cabo la autenticación de nuestros usuarios a través de la tecnología NFC y la etiqueta RFID de las tarjetas TUI físicas o digitales instaladas en nuestros smartphones usando la AppCrue. La arquitectura seguida para el desarrollo de esta fase es la siguiente:

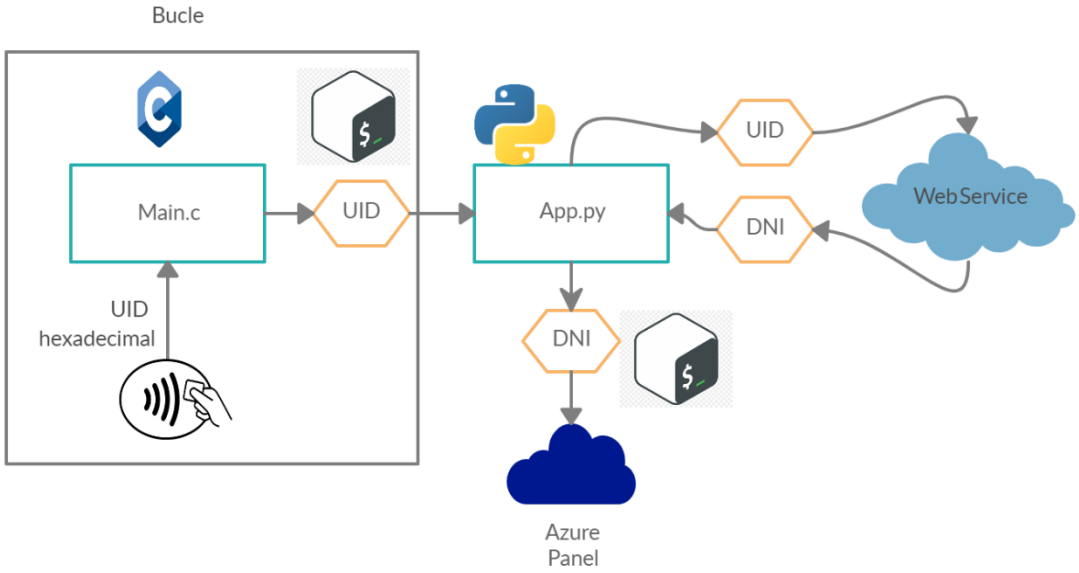

Figura 4.5 Esquema arquitectura seguida

De esta arquitectura, cabe destacar el bucle creado para leer del UID, este bucle se comienza al iniciar el sistema y estará comunicándose con el lector NFC hasta que este lea datos de una tarjeta.

Una vez recibido el UID, pasaremos esos datos a otro script donde iniciaremos una transmisión WSDL para recuperar el DNI asociado al UID obtenido en la lectura.

Ya con el DNI del usuario, le habilitaremos los servicios del aula/sala UPCTstream a través del panel alojado en Azure.

4.2.1.1 Desarrollo

Para comenzar con el desarrollo de nuestra arquitectura, hay que destacar que para la lectura de estar tarjetas tanto físicas como digitales usaremos el dispositivo PN532 Breakout Board. Se seleccionó este dispositivo porque es compatible con la tecnología MIFARE, en concreto con MIFARE Classic 4K y DESFire EV1 4K, que son los usados por nuestra TUI.

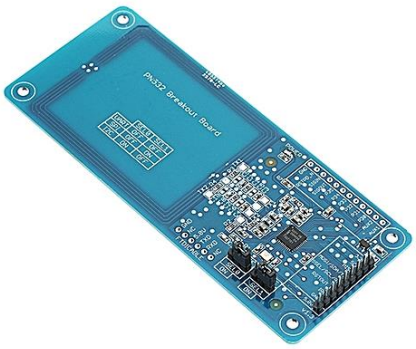

Figura 4.6 dispositivo PN532 Breakout Board

Este dispositivo se conecta a nuestra Raspberry a través del **GPIO,** más específicamente, a los pines que controlan la comunicación SPI, en concreto hemos usado los pines 1 y 9 para dar energía al lector y los pines 19,21,23 y 24 para las comunicaciones entre la Raspberry y el PN532.

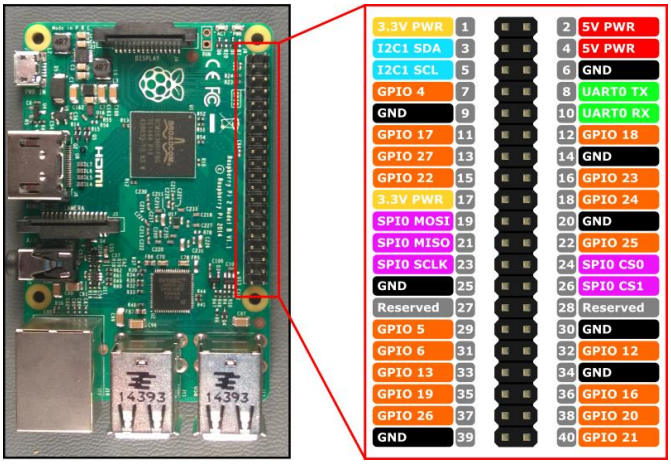

Figura 4.7 Disposición pines Raspberry 3 b+

Una vez que las conexiones son correctas, utilizamos la librería "**Libnfc**" para controlar el lector y realizar las acciones que necesitamos llevar a cabo. El lenguaje necesario para usar esta librería es **C**. Podemos encontrar toda la documentación sobre esta librería en [6].

Cuando realizamos la lectura, si todo ha sido correctamente conectado y configurado, recibiremos el UID, identificado de usuario, de nuestra tarjeta TUI. Este UID se lee en formato hexadecimal, por lo que, lo primero a realizar en el mismo script una vez tenemos la lectura, es convertir a decimal ese valor.

Después de que tengamos el valor decimal, debemos enviar ese valor al servicio web, siguiendo una estructura de mensajes WSLD, creado por la unidad de Informática y este nos devolverá el DNI asociado a ese UID. Para la comunicación con este servicio se optó por usar la librería "**Zeep**" de **Python**. Podemos consultar toda la información sobre esta librería en [7].

Para pasar el valor obtenido en nuestro script de lectura programado en **C**, a un nuevo script programado en **Python**, debemos hacer uso en ambos lenguajes de sus respectivas llamadas a consola.

Desde el script en **C** y para evitar que se detenga el programa de lectura, una vez obtenemos la lectura de la tarjeta, creamos el siguiente hilo:

```
void * ejecutoScrip(void *arg)
{
 pthread_t ntidth= ntid;
 system("pkill chromium");
 char comando[255]="python3 app.py ";
 strcat(comando,uid);
 system(comando);
 return((void *)0);
 pthread_detach(ntid);
}
```

Figura 4.8 Función que ejecuta script de Python en C

Lo más importante de este, es la creación de la variable comando, esta incluye la llamada por consola al script "app.py" al que se le concatena el valor de la UID recibido y pasado a decimal. Una vez creado el hilo ponemos a dormir el programa principal unos segundos, para no enviar de forma repetida el valor obtenido.

En este momento, ya hemos ejecutado el script "app.py". En este enviamos una petición WSDL pasando como parámetro el UID y una serie de credenciales.

Por último, en esta fase como respuesta a la anterior petición, obtenemos el DNI asociado a ese UID y lo enviaremos por Get al panel alojado en el servidor de Azure. Haciendo uso nuevamente de las llamadas por consola, esta vez de **Python**.

Una vez realizado todo este proceso pasaríamos a la sección "Azure Panel". Esta, es común a ambas fases, y en ella se explica detenidamente el proceso que seguimos en la parte del servidor.

4.2.2 Azure Panel

En esta sección daremos toda la información sobre el tratamiento de los datos recibidos, más concretamente, el DNI asociado al UID obtenido de la lectura desde las Raspberry de nuestro sistema.

Cabe destacar que se ha trabajado para homogeneizar, desde las Raspberry, él envió de estos datos. Con el objetivo de conseguir que, para los procesos llevados a cabo en el servidor, la obtención de esos datos sea transparente.

Es decir, que no sea necesario para nuestro servidor saber el método que usamos en las Raspberry para obtener el DNI y así poder usar esos datos recibidos, sin necesidad de saber si se obtuvieron con los métodos desarrollados en la Fase 1 (NFC), o en la Fase 2 (QR).

Todas la infraestructura de servicios usados para el funcionamiento de los sistemas que ha sido usado es este trabajo, particularmente el sistema de UPCTstream, tanto la web como la base de datos, están alojados en **Microsoft Azure Cloud Services**.

En Microsoft Azure tenemos una *reducción de costes* significativa, dado que no debemos invertir en adquirir y mantener equipos hardware, pagando únicamente por la capacidad que se requiera en cada momento. Por lo que tenemos gran una *flexibilidad* a la hora de obtener más o menos recursos según nuestras necesidades. Además, en cuestión de minutos se puede *reescalar* los recursos de nuestros servidores. También, se nos garantiza una disponibilidad continua de nuestros datos y servicios con un alto nivel de seguridad. En referencias [12] tenemos toda la documentación sobre Microsoft Azure.

Por todo ello, el alojamiento de todos los servicios, por parte del CPCD, se realizó en la nube de Azure, por sus grandes ventajas frente a los alojamientos de servicios web clásicos.

En consecuencia, para hacer las modificaciones pertinentes en el sistema ya creado de UPCTstream, debemos de ser capaces de poder conectar con nuestros servicios en la nube de Azure. Para ello, se ha usado diferentes softwares.

En el caso de la conexión con los servicios de **FTP** se ha usado el programa de FileZilla y para el acceso a la base de datos **SQL**, empleamos el programa MySQL Workbench.

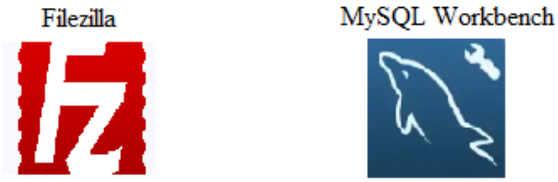

Figura 4.12 logos programas FileZilla y MySQL

4.2.2.1 Base de Datos

Para poder crear un registro propio sobre las reservas de las salas y los demás servicios de UPCTstream. Se creó una base de datos en la que los usuario se registran desde la web, con los datos de acceso del CAS.

Esta base de datos fue creada cuando se desarrolló el sistema UPCTmedia, anterior a este trabajo. Dado que partimos de una base de datos ya creada y en servicio, debemos extremar las precauciones con su uso o modificación.

Tras estudiar esta base de datos de forma exhaustiva, se decidió no modificarla y usar solo algunas de sus tablas. En concreto, para poder autentificar al profesor y comprobar que está en el aula que reservó, utilizaremos tres tablas de la base de datos "upctmedia".

Descripción de las tablas:

- Tabla "profesor": Esta tabla contiene los datos de los profesores que han creado su perfil en el sistema a través de la web de UPCTmedia. Estos datos fueron importados de los datos de acceso del CAS. De todos los datos importados, el único que usaremos es el DNI.
- Tabla "reservas_upctstream": Aquí es donde quedan registradas las reservas de las diferente salas, con la fecha, hora de inicio y fin. En esta tabla se generan entradas desde la web de UPCTstream, cuando los usuarios crean una reserva desde el formulario.
- Tabla "salas_upctstream": Esta tabla guarda la información de las salas disponibles actualmente en nuestro directorio.

Estas siguen el siguiente modelo relacional:

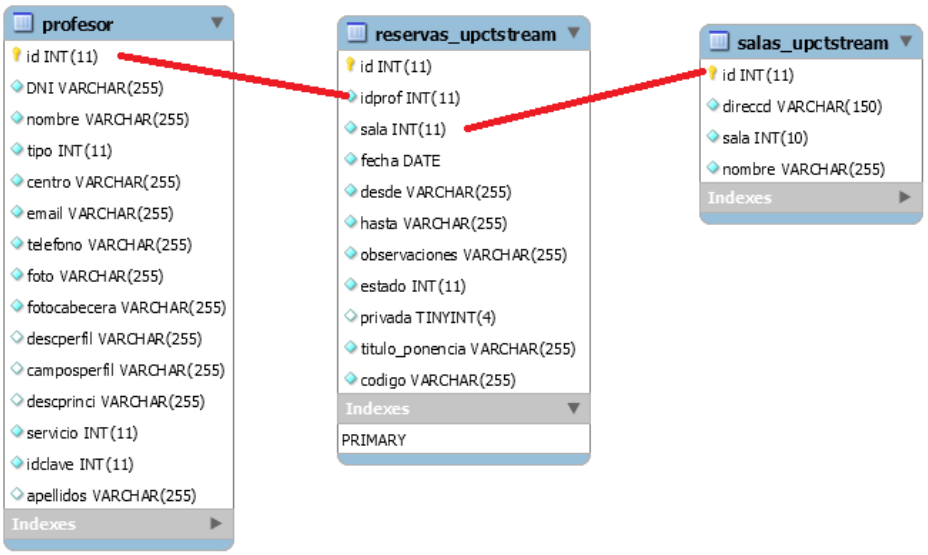

Figura 4.13 Modelo relacional de la base de datos.

Para poder autenticar y dar acceso a el aula reservada al usuario. Debemos comprobar que su DNI este en la tabla "profesor". Además, la id de ese profesor debe estar en la tabla "reservas_upctstream" en la comuna "idprof", y debe tener una reserva hecha para ese día y hora. Por último, la sala desde la que intenta acceder tiene que ser la que reservo.

4.2.2.2 Desarrollo Web

Como describimos en apartados anteriores, este sistema ya estaba desarrollado y en uso antes de iniciar este proyecto. Por lo que, previamente a introducir las modificaciones, se debe hacer un estudio minucioso del código del sistema actual, para introducir estos cambios sin afectar a la funcionalidad actual del sistema.

Partimos de una interfaz web ya creada y debemos modificar esta para dar soporte a el nuevo sistema de identificación.

Figura 4.14 interfaz web inicial y modal sobre el que aplicaremos la información

El objetivo principal es recibir el DNI obtenido desde las Raspberry de nuestro sistema, y con esta información, realizar consultas a nuestra base de datos para poder dar o restringir acceso al inicio de la grabación.

Con este objetivo, se estudia detalladamente el código y se determina que hay que modificar los siguientes scripts:

- El script “index.php” con el cual se crea la interfaz web
- El script “main.js” responsable de la interactividad con la interfaz web.

Y además hay que crear un nuevo script:

- El script “consultas.php” donde realizaremos las consultas SQL a nuestra base de datos.

Una vez identificado los ficheros, pasaremos a describir las modificaciones realizadas a cada uno de ellos y explicar la creación del nuevo script “consultas.php”.

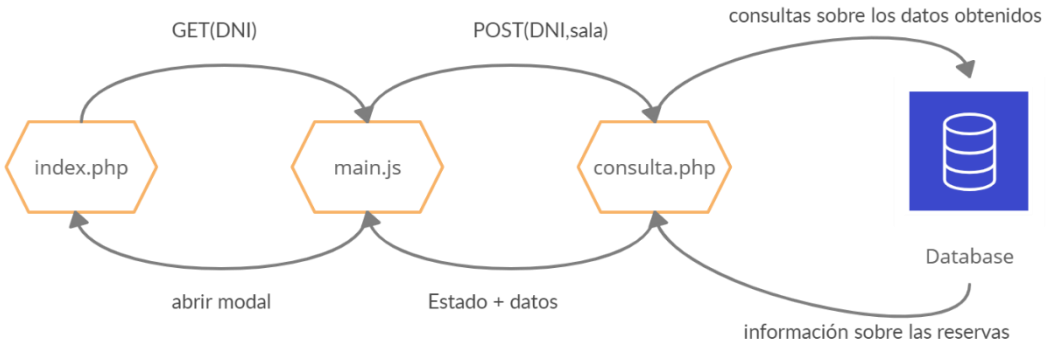

Figura 4.15 Flujo de información entre script y base de datos

Index.php

Este fichero es el responsable de la interfaz web mostrada en el figura 4.14.

En la funcionalidad previa a este trabajo, el usuario debía presionar el botón de grabar para abrir el modal donde debía introducir, a través de un teclado virtual, su DNI y el título con el cual se nombraría al video resultante de la grabación en su perfil de UPCTmedia.

Por lo que, debemos modificar este para que una vez recibamos el DNI desde la Raspberry por GET, simulemos las acciones del usuario correspondientes a presionar sobre el botón de grabar e introducir los datos.

Conjuntamente, al introducir este nuevo sistema, debemos crear una serie de alertas para comunicar al usuario información sobre su reserva o posibles errores.

Todas estas alertas, se han creado como modales, pequeñas ventanas emergentes, que son lanzados desde el script "main.js" y que informan de errores en los datos introducidos, como: no hay reservas para ese día o esa sala, aún no ha llegado la hora de la reserva o ha expirado, errores de lectura, etc.

Teniendo en cuenta que los datos sean correctos, desde "main.js" obtendremos los datos necesario para rellenar los campos de DNI y título, completando así la fase de autenticación.

Una vez pasado completado este modal, podremos iniciar la grabación y/o el streaming, terminando así el proceso implementado en este trabajo.

Main.js

Este JavaScript se encarga de controlar las acciones, sobre los equipos de grabación, de las que disponen los usuarios, como: iniciar la grabación, pararla, emitir en directo, cambiar la posición de la cámara, etc.

Para cumplir el objetivo de este trabajo, será modificado añadiendo una función llamada "dni" para que, en caso de recibir el DNI del usuario a través de un GET, ejecute una petición Ajax al fichero "consultas.php".

En esta petición, pasaremos los datos de la sala desde la que se realiza la petición de autenticación y el DNI del usuario.

Como respuesta, obtendremos diferentes estados que identificaran el tipo de información obtenida por las consultas SQL del fichero "consultas.php", además de información relevante sobre la reserva.

Una vez se recibe la respuesta, desde esta función, abrimos diferentes modales de "index.php". Si se encuentra una coincidencia en los datos se abrirá el modal de la figura 4.14, con los campos de DNI docente y título rellenados. Y para el caso contrario, se abrirán modales alertando de algún fallo o error por parte del usuario.

Consultas.php

El objetivo de este script consiste en crear una conexión, con la base de datos, destinada a realizar consultas SQL para comprobar y obtener los datos necesarios para la autenticación del usuario, y verificar la reserva del aula/sala UPCTstream.

Para comenzar con este proceso, debemos tener en cuenta, los datos que nos llegan por POST desde la petición Ajax realizada en "main.js". Estos datos son el DNI del usuario y la sala desde donde se está intentando realizar el proceso. Una vez obtenidos esos datos, procederemos a realizar la primera consulta SQL a nuestra base de datos.

Dado el modelo relacional de nuestra base de datos mostrado en la figura 4.13, necesitaremos realizar varias consultas para poder corroborar los datos de las diferentes tablas.

Un ejemplo de consulta SQL es:

SELECT * FROM profesor WHERE DNI= ?

En ella obtendremos toda la información de la tabla profesor donde la columna "dni" coincida con el DNI enviado. Si en esta consulta no encontramos ningún resultado, querrá decir que no tenemos ese DNI en nuestra base de datos, por lo que no podremos dar acceso a ese usuario.

Sin embargo, si encontramos una coincidencia, pasaremos a la siguiente consulta:

"SELECT * FROM reservas_upctstream WHERE idprof= ? AND fecha=? "

Previa a esta consulta, hemos obtenido la fecha completa de nuestro servidor y con ella hemos realizado la consulta, para obtener así todas las reservas realizadas por ese profesor en el día actual. Si existe una, quiere decir que ese profesor tiene una reserva para ese día, de lo contrario no tiene reserva y el sistema lo no le permitirá el acceso.

Para el caso de que tenga reserva, tendremos que comprobar si aún sigue activa, es decir, que está dentro del rango de horas que se le asignó en la base de datos. Si no, se le lanzará un aviso para recordarle el horario de su reserva.

Mientras que, si está dentro del horario, pasaremos a la última comprobación. Esta se basa en comprobar por último si está en la sala que reservo. De ser así, se le dará acceso al sistema y podrá comenzar la grabación.

En cambio, si la sala desde donde se está accediendo no coincide con la de la reserva, se realizará una última consulta para determinar cuál es la sala en la que se reservó, y de esa forma poder informar a usuario de cuál es el aula correcta.

Con esta batería de comprobaciones, podremos dar al usuario la información necesaria para solventar el fallo y poder iniciar la grabación correctamente.

4.2.3 Fase 2

Para la segunda fase del proyecto, y tras la noticia de que las tarjetas TUI físicas iban a dejar de recibir soporte, con la intención de ser retiradas o restringir su uso en la comunidad educativa.

Se preciso de otro método de autenticación de los usuarios del sistema, para ello y tras el lanzamiento de la AppCrue, se comenzó a desarrollar un sistema por el cual se leyeran códigos QR con las Raspberry usadas en el sistema. En esta fase del proyecto se requiere leer los códigos QR dinámicos generados por la AppCrue.

Lo más interesante de este tipo de códigos es que son códigos de lectura única y tiene una caducidad de treinta segundos, es decir, que tras la primera lectura por nuestro sistema o cuando pase ese tiempo de vida, el código será descartado por el sistema y dado de baja.

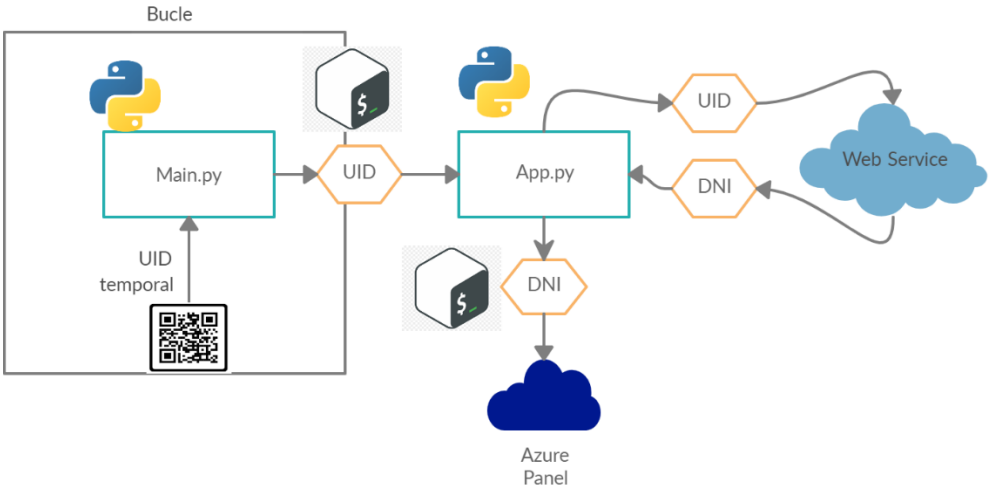

Figura 4.9 Arquitectura Fase 2

Partimos de la arquitectura que tenemos de la Fase 1, pero la modificamos para obtener la lectura de la cámara instalada para poder leer los códigos QR.

Para ello, debemos generar un nuevo script que procese las imágenes recibidas de la cámara y pueda detectar y decodificar los códigos QR que se le muestren. Para este propósito dado las librerías necesarias, se ha usado Python para generar este script.

Una vez leído y decodificado el QR, obtenemos el valor del UID temporal. A través de un script secundario, se lo comunicamos a el servicio web que comprobará ese UID y nos devolverá el DNI asociado.

Ya con el DNI del usuario, como en la fase 1, le habilitaremos los servicios del Aula/sala UPCTstream a través del panel alojado en Azure.

4.2.3.1 Desarrollo

Para la lectura de los códigos se ha escogido la cámara "Joy-It Cámara" del fabricante "Joy-it". Lo más importante de esta elección es la relación prestaciones-precio, para esta cámara se buscaba que tuviese un amplio Angulo de grabación y una buena resolución para que la lectura del código sea más fácil de reconocer por nuestro programa. Además, tiene un precio muy asequible para su instalación en todas las salas UPCTstream y ampliable para su uso a gran escala en toda la universidad.

Figura 4.10 Joy-It Cámara

Esta cámara hace uso de puerto "**MIPI CSI camera port**" de la Raspberry, por lo que no hace falta usar ningún puerto USB, haciendo más fácil su instalación y complicando su manipulación por terceros. Para más información sobre las especificaciones de esta cámara podemos leer su datasheet en [8].

Para el reconocimiento de imágenes se ha usado la librería "**OpenCV**" de **Python**. Con esta librería se pueden realizar seguimientos y reconocimientos en tiempo real de las imágenes captadas por una cámara. Encontraremos toda las características de esta librería en [9].

Una vez podemos captar un QR por la entrada de la cámara para decodificarlo usamos la librería "**Pyzbar**" de **Python.** Con ella podemos obtener el código numérico que contiene el QR y así podremos tratar la información obtenida. En [10], podemos ver todas las especificaciones de esta librería.

Estas son las librerías clave para la correcta lectura del código QR proporcionado por el usuario para su autenticación.

Cabe destacar que, para el script principal de lectura del QR, se partió del código en **Python** del usuario de GitHub "Cuicaihao", podemos encontrar ese código en su perfil de GitHub: "https://github.com/cuicaihao", en el proyecto "Webcam_QR_Detector".

Con el este código se puede realizar una lectura correcta y decodificación de un código QR, pero este usuario lo creo para mostrar en la misma imagen el resultado de la decodificación del código, de esta forma:

Figura 4.11 Ejemplo resultado de una lectura QR con el código base de "Cuicaihao"

Por lo que el script ha sido modificado, para llevar a cabo la funciones que nos interesan para nuestro sistema. En concreto se ha eliminado la función del ejemplo, por lo que no se muestra por pantalla el resultado de la lectura, y una vez obtenido el UID temporal, este es pasado a nuestro script secundario "App.py".

En este, enviaremos el UID temporal, obtenido de la lectura del QR generado por el usuario, al servicio web proporcionado por los desarrolladores de la AppCrue y el servicio nos devolverá el DNI asociado.

Solo devolverá ese DNI si este UID cumple las condiciones de no haber sido recibido anteriormente, es decir si no ha sido usado, y si aún está dentro del periodo de treinta segundos de validez.

Como en la fase 1, ese DNI es enviado al panel de Azure donde trataremos los datos como describimos en la sección "Azure Panel".

4.3 Seguridad

Esta sección tiene el propósito de detallar las medidas de seguridad y los posibles fallos de estas, previos a la realización de este trabajo y como se ha ido solucionando con los procesos incorporados a el sistema.

4.3.1 Llegada y acceso a la sala

Todas las salas/aulas en las que se ha instalado el sistema de UPCTstream, están situadas dentro de las diferentes escuelas de las UPCT.

Actualmente, para poder acceder a estas, además de haber hecho la reserva en la web de UPCTstream, debes comunicarlo a la consejería de la escuela donde este situada el aula en cuestión, para que estos te den acceso al interior. Por lo que, solo el personal autorizado, podrá acceder a ellas.

4.3.2 Acceso remoto

Para la gestión y mantenimiento del sistema, podemos acceder a ellas de forma remota a través de un VNC servidor instalado en las Raspberry de nuestro sistema. Este servidor nos permite controlarlas usando un VNC cliente.

Las comunicaciones utilizan el protocolo RFB para el acceso remoto a la interfaz gráfica, además usa una encriptación AES-GCM de 128-bit o 256bit dependiendo del tipo de suscripción. En la Bibliografía podemos encontrar el whitepaper del servicio VNC en [11], en el tenemos todos los detalles sobre seguridad de este programa.

Como método de seguridad añadido para evitar el acceso de terceros de este sistema. Todos los equipos (Raspberry, cámaras y agentes de grabación) del sistema UPCTstream han sido aislados del resto de la red de la UPCT en su propia VPN, para así crear un entorno de trabajo más controlado y seguro.

Conjuntamente a esta VPN, todos los puntos de acceso donde están conectados estos dispositivos han sido restringidos a la MAC e IP estática de ese equipo. Con esto se pretende evitar en todo lo posible la entrada de equipos externos a nuestra red, con el consiguiente riegos que esto produciría en nuestro sistema.

Asimismo, dado que la ni conexión wifi ni bluetooth de nuestra Raspberry son usadas en ningún momento, estas han sido deshabilitadas para evitar posibles riegos en la seguridad del sistema provenientes de ambas tecnologías.

También han sido bloqueadas físicamente, las entradas USB de las Raspberry con el objetivo de impedir el uso de estos para la entrada de software malicioso o cualquier otro dispositivo malintencionado.

4.3.3 NFC

La forma de autenticar al usuario frente al sistema, previo a este trabajo, se basaba en teclear el DNI de este en la pantalla táctil de nuestras Raspberry. Es decir, el hecho de conocer el DNI te permitía autenticarte como usuario del sistema.

Para este método de autenticación, la seguridad se basaba en el control de acceso previo a la sala y en que habría tenido que ser reservada previamente en la web de UPCTstream donde ha tenido que identificarse a través del CAS con sus credenciales como docente de la UPCT.

Con la implementación de esta tecnología al sistema conseguimos aumentar la veracidad de que el usuario es quien dice ser. Dado que estas TUI son personales e intransferibles, podemos decir con certeza que la persona poseedora de esa tarjeta es el usuario que reservo la sala.

Cabe destacar que, con este método el sistema no puede reconocer si usuario portador de la TUI, que quiere autenticarse ante el sistema, es el auténtico titular de esta. Por lo que, podría fallar si la TUI es sustraída por un tercero de forma maliciosa.

Para solucionar este fallo en la autenticación, una vez el auténtico titular de la tarjeta diese el aviso de su sustracción, desde la unidad de informática se procedería a dar de baja el UID asociado. Con lo cual, la TUI sustraída se vería invalidada a todos los efectos y se podría crear una nueva con un UID diferente.

Al ser una tecnología que no requiere contacto entre los dispositivos y al sustituir el anterior de autenticación por teclado, hemos conseguido evitar al máximo el contacto de los usuario con los equipos de nuestro sistema. Ayudando así a mejorar las condiciones higiénico-sanitarias ante el COVID-19 de las salas UPCTstream.

4.3.4 QR

Gracias a la aplicación de AppCRUE, tenemos acceso a un nuevo método de autenticación de usuarios que, incorporado a nuestro sistema, dará varias capas de más de seguridad al sistema.

La característica más importante a nivel de seguridad, dado que usamos códigos QR dinámicos, es que son códigos de lectura única y tiene una caducidad de treinta segundos.

Con lo que, tras la primera lectura por nuestro sistema del código QR generado por el usuario desde la AppCrue, este código es deshabilitado y si intentamos volver a usarlo dará error y no nos permitirá acceder al sistema. Conjuntamente, cuando pase su tiempo de vida, el código será descartado por el sistema.

En consecuencia, y dada la suma de estas restricciones en su uso. Aunque un tercero consiguiera hacerse con uno de estos código QR y este no estuviese ya usado por el usuario. Solo podríamos usarlo durante treinta segundo, haciendo muy difícil el intento de suplantación por un tercero.

A estas restricciones sobre el uso del QR, tenemos que añadir que su generación se hace desde el smartphone del usuario. Por esta razón, para poder generar uno de estos códigos QR tendremos que, no solo tener acceso al dispositivo del usuario, si no que poder deshabilitar todas las medidas de seguridad ya incluidas en los smartphones, como pueden ser: PIN, patrones, huellas, reconocimiento facial y demás sistemas de seguridad disponibles en los diferentes smartphones del mercado.

De modo que, con este sistema de generación de códigos QR dinámicos, podemos decir, que las posibilidades de acceder al sistema de forma maliciosa se reducen en gran medida, dadas todas las capas de seguridad que se necesitan atravesar para poder hacer uso de este.

Además, Tanto con este sistema como con el de NFC, al ser ambas tecnologías que no requiere contacto entre los dispositivos. Hemos conseguido limitar las superficies de contacto de los usuario con los equipos de nuestro sistema. Ayudando así a mejorar las condiciones higiénico-sanitarias ante el COVID-19 de las salas UPCTstream.

5. Prototipos, pruebas y costes

5.1 Prototipos

Para finalizar este trabajo y llevar a cabo todo el proceso de diseño, descrito en el anterior apartado. Se han realizado dos prototipos, cada uno de ellos implementando los métodos desarrollados en las diferentes fases del trabajo.

5.1.1 Prototipo Fase 1 (NFC)

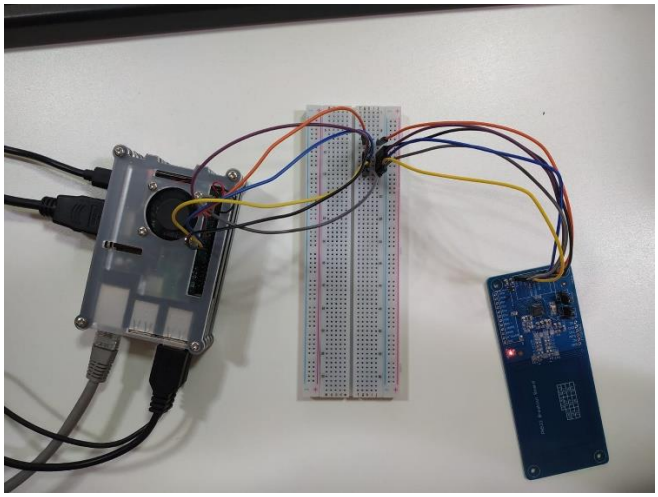

Figura 5.1 Imagen Raspberry con PN532 conectado por SPI

Como ya se ha explicado detalladamente en el diseño de la "Fase 1", para este prototipo se ha requerido incorporar a la Raspberry Pi un lector de tarjetas PN532. Para dotar a esta de la capacidad para leer tarjetas NFC más en concreto, para poder usar nuestras TUI.

Desde el CPCD, se está trabajando en un soporte generado por impresión 3D para poder colocarlo en las aulas/salas UPCTstream.

Para lograr la conexión entre los pines SPI de la Raspberry y el PN532, en este caso se ha precisado de una Breadboard intermedia, pero será eliminada en diseños posteriores.

5.1.2 Prototipo Fase 2 (QR)

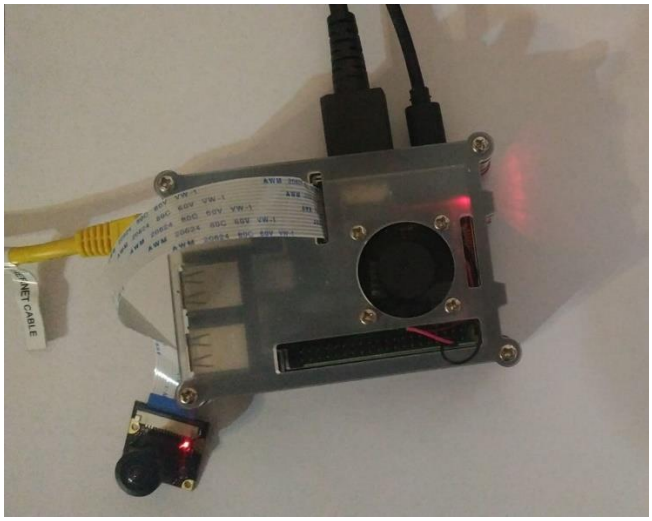

Figura 5.2 Imagen Raspberry con cámara Joy-it conectada

El prototipo para la "Fase 2" incluye una pequeña cámara conectada a la Raspberry a través de su puerto "**MIPI CSI camera port**". Gracias a esto le hace tener un diseño más simple y fácil de posicionar.

Como en el prototipo anterior, desde el CPCD se está trabajando en un soporte que pueda ser colocado en las aulas, para ofrecer un buen ángulo a la cámara y simplificar así la lectura de los códigos QR generados por los smartphones de los usuarios.

5.2 Pruebas funcionales

Antes de comenzar las pruebas de nuestros prototipos, estudiaremos el uso que las salas UPCTstream han tenido desde su creación en 2017. Para ello realizaremos un análisis de la información sobre reservas que se encuentra en la base de datos.

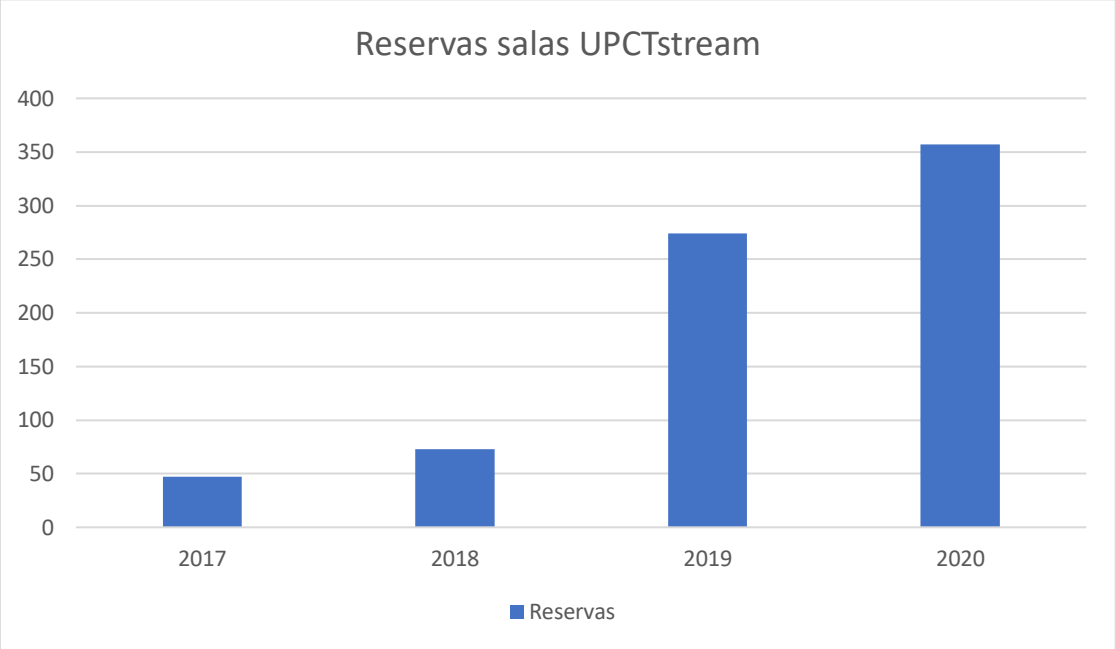

Figura 5.3 Grafico evolucion

Con este grafico, podemos ver como el uso de las salas a aumentado bastante en en los dos ultimos años, en especial este ultimo año, debido al gran aumento de clases online por motivo de la pandemia COVID-19.

Otro factor a tener en cuenta es el aumento de salas disponibles, que ocurrio en 2019. Estas pasaron de tres salas, a las diez que hay actualemente.

Con esta información, debemos tener en cuenta, que las Raspberry, deberían permanecer encendidas durante largos periodos de tiempo de forma ininterrumpida. A demás deben estar ejecutando, los programas descritos en las fases, a la espera de usuarios.

Por lo que, las pruebas se han centrado en comprobar el uso del sistema en periodos largos de tiempo. Para ellos se dispusieron una seria de batería de pruebas en las que, se mantenía encendido y monitorizado el equipo mientras se realizaban usos de los sistemas de autentificación de las distintas fases.

Durante las primeras pruebas se consiguió detectar y solventar ciertos errores en el código que provocaban un malfuncionamiento o que paraban los programas de autenticación.

Para una última prueba, se realizó una simulación de uso real, en la que durante cinco días (una semana laboral), se realizaban intentos de autenticación ante el sistema. Con el fin de lograr acercarnos lo máximo posible a el uso de una sala UPCTstream. Durante esos cinco días, todos los intentos de autenticación, tanto por NFC como por código QR, fueron ejecutados de forma correcta.

Además, durante esos cinco días se monitorizo el uso de los recursos de ambos métodos de autenticación.

La CPU de la Raspberry en standby, es decir, a la espera de recibir una lectura solo ejecutando el código, tiene un uso de entre 20-29% para ambas fases. Una vez reciben la lectura, se observa un incremento hasta llegar al 50-56% de su uso, debido a que cuando se recibe una lectura, creamos una llamada al navegador lo que incrementa momentáneamente el uso de recursos del sistema, pero pasado unos pocos segundos volvemos a los niveles de standby.

Con estos resultado, podemos decir que nuestro prototipo responde de forma correcta al uso que tendrá, una vez sea puesto en funcionamiento dentro de las salas.

Y por lo tanto que está listo para ser integrado dentro de la infraestructura de UPCTstream.

5.3 Costes

Para completar toda la información sobre nuestro sistema, en esta sección se proporcionará una visión más específica sobre los costes de implementación de este nuevo sistema de autenticación a nuestro actual sistema de UPCTstream y su posible escalamiento a la toda red de aulas/salas repartidas por la UPCT.

Para comenzar con este informe de costes debemos especificar que, en la actualidad, en las diez salas desplegadas por la universidad. Ya podemos encontrar una Raspberry con su correspondiente pantalla táctil y un soporte, creado por impresión 3D, que la sujeta a la pared.

Teniendo en cuenta esto, para actualizar este sistema por completo debemos proveernos del número suficiente de dispositivos hardware necesarios, para llevar a cabo las dos fases del proyecto.

A continuación, desglosaremos todos los costes dependiendo de los posibles despliegues de los métodos de las diferentes fases que se pretenden implementar, o de si se quieren implementar los métodos de ambas fases.

Debemos recordar que estos costes serían sumados a los costes ya existentes en los equipos necesarios para la creación de una sala/aula UPCTstream.

5.3.1 Costes Fase 1

Para poder implementar la Fase 1, en nuestro sistema, debemos de incorporar a las Raspberry un módulo NFC PN532.

Entre los diferentes vendedores de este producto el precio oscila entre los 20-30€. Así mismo el soporte de la Raspberry tendría que ser actualizado con un módulo para sujetar el lector NFC. Por lo que la factura total de implementar este sistema en una sala sería:

Modulo NFC PN532: 30€

Impresión 3D: 12€

Mano de obra: 10€

Coste total: 52€

Con esta información, podríamos decir que el coste total, al alza, de la instalación de este nuevo sistema estaría rondando los 52€. Por lo cual, si extrapolamos con la intención de cubrir la instalación, con este método de autenticación, de las diez salas.

El coste total para cubrir toda nuestra red de UPCTstream con el método de autenticación por NFC descrito en la sección de “Fase 1”, rondaría los 520€.

5.3.2 Costes Fase 2

En el caso de la Fase 2, necesitamos incorporar a las Raspberry una cámara para que, podamos efectuar la lectura de códigos QR mostrados a través de la pantalla de un smartphone.

Con este objetivo, y tras un estudio de las diferentes cámaras disponibles para Raspberry, se decidió usar el modelo de "Joy-It gran angular de 5MP 160º" con un precio que ronda entre los 27-30€. Además, como en el caso anterior deberíamos actualizar el soporte al que ira sujeta esta cámara. Por lo que, usando estos datos para generar nuestra factura:

Cámara:	30€
Impresión 3D:	12€
Mano de obra:	10€
Coste total:	52€

Por lo que, los costes de una instalación en toda nuestra red de UPCTstream alcanzaría de unos 520€.

Como hemos podido demostrar, los coste de instalación máximos en ambas fases, podrían alcanzar el mismo valor, eso quiere decir que el coste de implementar cualquiera de las dos fases es muy similar o igual.

Con esta información y observando las distintas características ya descritas en la sección de "Análisis". Podemos concluir, que en el caso de vernos forzados a decir cuál de las dos fases llevar a producción. La opción que nos proporciona una mayor seguridad a la hora de autenticarnos y además tendrá una disponibilidad mayor en el tiempo. Es el método de la Fase 2 y la lectura de QR dinámicos generados por la aplicación "AppCrue".

5.3.3 Costes implementación Fase 1 y Fase 2

En el caso de que se decida, implementará ambas fases del trabajo, el resultado de la factura será:

Modulo NFC PN532: 30€

Cámara: 30€

Impresión 3D: 12€

Mano de obra: 10€

Coste total: 82€

Dado que el soporte solo se actualizaría una vez y el proceso de actualización se haría en una solo sesión, podemos fijar los valores de mano de obra e impresión 3D. Lo que concluiría en un coste de 82€ por sala y un coste total de 820€ por la instalación de todas las salas de la red UPCTstream.

6. Conclusiones

Para ultimar, señalaremos las ideas y sensaciones principales que se han ido obteniendo durante el desarrollo teórico-práctico de este trabajo.

Se consideran los objetivos cumplidos, puesto que como se ha ido exponiendo durante este informe, todos y cada uno de los requisitos necesarios para dar solución a los objetivos principales han sido subsanados por las soluciones propuestas.

6.1 Trabajo futuro

6.1.1 Líneas de ampliación

Existe varias líneas de ampliación de los objetivos fijados en este trabajo.

Una de ellas sería un sistema de ahorro energético, con el cual, teniendo en cuenta que disponemos de los datos uso futuro que se va a hacer del aula, gracias al sistema de reservas. Podemos crear un sistema que apague y encienda los equipos de la sala, e incluso iluminación y climatización.

Esta medida podría llevarse a cabo, implementado en las salas, un sistema de relés eléctricos controlados a través de un Arduino o incluso una Raspberry conectado a nuestro sistema, el cual reciba ordenes de activar o desactivar estos relés en función de si hay o no reserva para ese día y esa hora.

Con esto se conseguiría reducir de forma considerable el uso de energía de las salas, contribuyendo a un ahorro económico importante, además de contribuir a la lucha contra el cambio climático.

Otra de las posibles medidas a incorporar, seria añadir a el sistema de autenticación creado en este trabajo, tanto QR como NFC, un sistema de cerraduras electrónicas para las salas/aulas UPCTstream.

Así estas salas serían completamente autosuficientes, eliminándose esa carga de las respectivas consejerías de la diferentes escuelas. Además, al evitar el contacto personal entre el personal de las conserjerías y los usuarios de nuestro sistema, mejoraríamos las condiciones higiénico-sanitarias, medidas muy importantes durante la pandemia de. COVID-19.

6.1.2 Extrapolaciones posibles del sistema

Podemos llegar a extrapolar los conocimiento adquiridos, el hardware y software desarrollado, a otros ámbitos dentro de la propia universidad. Por ejemplo y dado el actual estado de emergencia sanitaria por la pandemia de COVID-19, modificando la infraestructura usada en este trabajo, podríamos crear un sistema de registro y trazabilidad COVID-19.

Durante el estado de emergencia y posterior "nueva normalidad", que está aconteciendo actualmente en nuestro país y prácticamente en cada rincón de nuestro mundo. Se ha demostrado que el seguimiento y cuarentena de todos los afectados por el COVID-19, es el único sistema verdaderamente efectivo para la lucha contra el virus.

Con este objetivo en mente, desde la OMS se recomienda un ratio de unos 18 rastreadores por cada 100.000 habitantes. Es decir, para tener bajo control y evitar el progreso de la enfermedad, se requiere un gran número de rastreadores, personas que contactan con los pacientes de COVID-19 y hacen un seguimiento exhaustivo de las personas con las que han tenido contacto estos pacientes, para así determinar posibles contactos de riesgo y confinarlos, con el fin de evitar su propagación.

Por ello y para hacer el trabajo de estos rastreadores menos tedioso y más efectivo, desde los respectivos servicios públicos se han creado diferentes sistemas, con el fin de anticipar la información que estos rastreadores necesitaran para hacer el seguimiento de un paciente.

Figura 6.1 Logo Radar COVID España

Un ejemplo conocido, es la aplicación Radar COVID, con ella, a través de bluetooth de nuestros smartphones, conseguimos en caso de ser positivo, alertar a todos los posibles contactos de riesgo de nuestra situación y viceversa.

Desde las universidades se han desarrollado diferentes medidas de control de aforos y registro de acceso, bien mediante control por parte del personal de la universidad o bien usando diferentes plataformas webs para que los usuario se identifiquen manualmente, y así puedan dejar constancia de su estancia en ciertas localizaciones de la universidad.

Teniendo en cuenta la importancia de conseguir un sistema verdaderamente eficaz y automático para los usuarios de las universidad, para su registro y trazabilidad por las estancias de la universidad.

Desde nuestro sistema actual de autenticación de usuarios de salas UPCTstream, se propone como un posible método eficaz, la modificación de este sistema, para que sirva como control de acceso, a los diferentes recintos de la universidad, trazabilidad de posibles contactos de riesgo COVID-19 y limitador de aforos.

Se podría crear una red de Raspberry, como la red actual de las salas UPCTstream, para que registre el acceso y salida de los usuarios a los diferentes recintos de la UPCT, como: escuelas, bibliotecas, aulas, salones, etc.

Por el cual, mediante su autenticación por QR de la AppCRUE, como en la Fase 2 (disponibles tanto para personal docente como para estudiantes). Y gestionado por una base datos en las que se grabe tanto la fecha, como la hora, el lugar donde estuvo y todos los datos de interés.

En el caso de que uno de los usuarios fuese positivo en COVID-19. La figura del coordinador COVID de nuestra escuela, dispondrá de una interfaz creada expresamente para él,y solo con identificar al positivo ante el sistema, tras una serie de consultas a los parámetros que define a un posible contacto de riesgo, le devuelva todos los posibles estos contactos para, bien pasar los datos de contacto a los rastreadores regionales o que sea el propio coordinador COVID el encargado de ponerse en contacto con ellos para alértales de su situación.

Incluso dando un paso más a la automatización, ser el propio sistema el que se ponga en contacto con esos usuarios mediante medios electrónicos (correo, SMS, alertas es su AppCRUE, etc.) para alértale de que ha tenido contacto con un paciente COVID.

Estas medidas de seguridad aumentarían en mucho la eficacia de los rastreadores actuales, contribuyendo a una mayor efectividad en las medidas de prevención, y poniendo nuestro granito de arena para mejorar la dramática situación por la que atraviesa actualmente nuestro país y, en conjunto, nuestro mundo.

6.2 Reflexión final

Como ultimo y en resumen de todas las experiencias vividas durante el desarrollo del trabajo. Debo destacar que, se han llevado a la práctica, no solo todos los conocimientos adquiridos durante el grado, si no, que se ha requerido adquirir destreza en otras muchas habilidades como pueden ser:

- ✓ La adquisición de nociones avanzadas en lenguajes de programación variados (Python, C, bash). Y el uso de librerías en diferentes lenguajes, realmente complejas, pero con un potencial extraordinario.
- ✓ El estudio de diferentes dispositivos hardware que me permitieran cumplir con los objetivos marcados.
- ✓ Incorporarme a un proyecto ya en marcha, para modificar las funciones de un sistema en uso. Lo que conlleva a dedicar un estudio previo exhaustivo de toda la infraestructura con el objetivo de completar las modificaciones requeridas sin alterar sus funcionabilidades ya existentes.
- ✓ La obtención de nociones y manejo de la Raspberry Pi, un dispositivo con un potencial enorme, tanto para la docencia como para la industria.
- ✓ La capacidad de adaptar todos estos conocimientos a nuevos objetivos durante el desarrollo de un proyecto.

Como conclusión, he de distinguir que la suma de todas esta experiencias, ha hecho de este trabajo una forma emocionante y realmente compleja de poner a prueba mis habilidades para llevar a cabo un proyecto apasiónate, que me ha hecho darlo todo, y del cual estoy satisfecho, al dejar una herramienta desarrollada y lista para su uso en la universidad que me dio la oportunidad de llevarlo a cabo.

7. Bibliografía

7.1 Referencias

[1] Universidad de Murcia uso QR [en línea] [consulta: 28 noviembre 2020]. Disponible en: https://digital.um.es/nuevo-sistema-para-registrar-la-presencia-en-las-aulas/

[2] Universidad de Cantabria uso QR [en línea] [consulta: 28 noviembre 2020]. Disponible en: https://web.unican.es/sistema-de-rastreo-mediante-codigos-qr

[3] Datasheet Raspberry Pi 3 modelo B+ [en línea] [consulta: 5 octubre 2020]. Disponible en: https://static.raspberrypi.org/files/product-briefs/200206+Raspberry+Pi+3+Model+B+plus+Product+Brief+PRINT&DIGITAL.pdf

[4] García, I. (2016). Casos prácticos de uso de la Tarjeta Universitaria Inteligente basados en QR y otros usos. TFE. ETSIT. UPCT. (Citado Fecha). Recuperado a partir de: https://repositorio.upct.es/handle/10317/7144

[5] Wikipedia Mifare [en línea] [consulta: 26 noviembre 2020]. Disponible en: https://es.wikipedia.org/wiki/Mifare

[6] Librería Libnfc para Python [en línea] [consulta: 26 noviembre 2020]. Disponible en: http://nfc-tools.org/index.php/Libnfc

[7] Librería Zeep para Python [en línea] [consulta: 26 noviembre 2020]. Disponible en: https://docs.python-zeep.org/en/master/

[8] Datasheet cámara Joy-it [en línea] [consulta: 5 octubre 2020]. Disponible en: https://joy-it.net/files/files/Produkte/rb-camera-WW/rb-camera-WW-Datasheet.pdf

[9] Librería OpenCV para Python [en línea] [consulta: 28 noviembre 2020]. Disponible en: https://pypi.org/project/opencv-python/

[10] Librería Pyzbar para Python [en línea] [consulta: 28 noviembre 2020]. Disponible en: https://pypi.org/project/pyzbar/

[11] Whitepaper VNC [en línea] [consulta: 14 octubre 2020]. Disponible en: https://static.realvnc.com/media/documents/vncconnect-security-whitepaper.pdf

[12] Documentación Microsoft Azure [en línea] [consulta: 25 octubre 2020]. Disponible en: https://docs.microsoft.com/es-es/azure/?product=featured

7.2 Imágenes

JS [en línea] [consulta: 29 noviembre 2020]. Disponible en: https://ayudawp.com/defer-parsing-javascript/

Ajax [en línea] [consulta: 29 noviembre 2020]. Disponible en: http://www.v-espino.com/~chema/daw2/ajax/ajax.htm

Python [en línea] [consulta: 29 noviembre 2020]. Disponible en: https://es.wikipedia.org/wiki/Historia_de_Python

C [en línea] [consulta: 29 noviembre 2020]. Disponible en: https://disenowebakus.net/lenguaje-c.php

Figura 4.6 dispositivo PN532 Breakout Board

Vendo [en línea] [consulta: 26 noviembre 2020]. Disponible en: https://vendo.ma/details/nfc-pn532-module-rfid-near-field-communication-reader-13-56mhz-compatible-with-arduino-prix-maroc-jumia-un329el14lhu4nafamz

Figura 4.7 Disposición pines Raspberry 3 b+

Prometec [en línea] [consulta: 26 noviembre 2020]. Disponible en: https://www.prometec.net/usando-los-gpio-con-python/

Figura 4.8 Función que ejecuta script de Python en C. [Elaboración propia]

Figura 4.9 Arquitectura Fase 2. [Elaboración propia]

Figura 4.10 Joy-It Cámara

PCcomponentes [en línea] [consulta: 5 octubre 2020]. Disponible en: https://www.pccomponentes.com/joy-it-camara-gran-angular-de-5mp-160-para-raspberry-pi

Figura 4.11 Ejemplo resultado de una lectura QR con el código base de “Cuicaihao”

Github [en línea] [consulta: 28 noviembre 2020]. Disponible en: https://github.com/cuicaihao/Webcam_QR_Detector/blob/master/README.md

Figura 4.12 logos programas Filezilla y MySQL. [Elaboración propia]

Figura 4.13 Modelo relacional de la base de datos. [Elaboración propia]

Figura 4.14 interfaz web inicial y modal sobre el que aplicaremos la información. [Elaboración propia]

Figura 4.15 Flujo de información entre script y base de datos. [Elaboración propia]

Figura 5.1 Imagen Raspberry con PN532 conectado por SPI. [Elaboración propia]

Figura 5.2 Imagen Raspberry con cámara Joy-it conectada. [Elaboración propia]

Figura 6.1 Logo Radar COVID España

Google Play [en línea] [consulta: 5 diciembre 2020]. Disponible en: https://play.google.com/store/apps/details?id=es.gob.radarcovid&hl=es&gl=US