UNIVERSIDAD DE MÁLAGA

E.T.S. INGENIERÍA INFORMÁTICA
UNIVERSIDAD DE MÁLAGA

GRADO EN INGENIERÍA INFORMÁTICA

Desarrollo de un dispositivo funcional de bajo coste para el seguimiento de personas en instalaciones. Aplicación al rastreo de contactos con positivos COVID-19 en empresas y organismos públicos.

Development of a low-cost functional device for tracking people in facilities. Application to the tracking of COVID-19 positive contacts in companies and public organizations.

Realizado por
Fco. Javier Azpeitia Muñoz

Tutorizado por
Prof. Dr. D. Vicente Manuel Arévalo Espejo

Departamento
INGENIERÍA DE SISTEMAS Y AUTOMÁTICA

MÁLAGA, DICIEMBRE, 2022

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA
GRADO EN INGENIERÍA INFORMÁTICA

Desarrollo de un dispositivo funcional de bajo coste para el seguimiento de personas en instalaciones. Aplicación al rastreo de contactos con positivos COVID-19 en empresas y organismos públicos.

Development of a low-cost functional device for tracking people in facilities. Application to the tracking of COVID-19 positive contacts in companies and public organizations.

Realizado por
Fco. Javier Azpeitia Muñoz

Tutorizado por
Prof. Dr. D. Vicente Manuel Arévalo Espejo

Departamento
Ingeniería de sistemas y automática

UNIVERSIDAD DE MÁLAGA
MÁLAGA, DICIEMBRE 2022

Fecha defensa: 23-1-2023

Resumen

El contenido de este proyecto de fin de grado se centra en rediseñar la solución de seguimiento de dispositivos bluetooth presentado en el trabajo fin de grado titulado "DESARROLLO DE UNA SOLUCIÓN DE BAJO COSTE PARA EL RASTREO DE CASOS POSITIVOS COVID-19 EN EMPRESAS Y ORGANISMOS PÚBLICOS" realizado por Isabel Teruel Parra, alumna del Grado en Ingeniería en Tecnologías Industriales, durante el curso 2020-2021.

En el anterior trabajo, se desarrolla una prueba de concepto que pretende resolver la problemática asociada al seguimiento de personas en entornos con control de acceso apoyándose en tecnología bluetooth. Sin embargo, la propuesta que se hace en dicho trabajo presenta ciertas limitaciones a nivel software y hardware que impiden su utilización en entornos reales.

Los cambios realizados, abarcan un rediseño global de la solución tanto a nivel hardware como software. Entre los aspectos que se abordan destacarían los siguientes:

- Añadir múltiples capas de seguridad para Tecnologías de la Información (IT).
- Añadir escalabilidad mediante el uso de servicios virtualizados.
- Añadir fiabilidad mediante el uso de servicios virtualizados replicados.
- Simplificación del hardware a través del uso de una plataforma orientada a *Internet of Things* (IoT).

Para cumplir con todas estas mejoras, el proyecto original ha sido rediseñado, manteniendo las principales funcionalidades, como:

- Detección de las direcciones MAC bluetooth a través de una placa Arduino.
- Envío de las direcciones a un servidor web, donde se realiza un proceso de filtrado.
- Almacenamiento de las direcciones filtradas en un servidor con un Sistema de Gestión de Bases de Datos (SGBD).
- Aplicación de consulta de la base de datos a través de un servidor con una aplicación web.

Palabras clave: Arduino; Bluetooth; Seguridad; Virtualización; Cifrado.

Abstract

The content of this final degree project focuses on redesigning the bluetooth device tracking solution presented in the final degree project entitled " Development of a low-cost functional device for tracking people in facilities. Application to the tracking of COVID-19 positive contacts in companies and public organizations" carried out by Isabel Teruel Parra, student of the Degree in Industrial Technologies Engineering, during the 2020-2021 academic year.

In the previous work, a proof of concept is developed that aims to solve the problem associated with the monitoring of people in environments with access control using Bluetooth technology. However, the proposal made in this work has certain limitations at software and hardware level that prevent its use in real environments currently.

The changes made include a global redesign of the solution both at hardware and software level. Among the aspects that are addressed, the following stand out:

- Bring multiple layers of security for Information Technology (IT).
- Bring scalability using virtualized services.
- Bring reliability with replicated virtualized services.
- Simplifying hardware using an Internet of Things (IoT) oriented platform.

To meet all these improvements, the original project has been redesigned, maintaining the main functionalities, such as:

- Detection of Bluetooth MAC addresses through an Arduino board.
- Sending the addresses to a web server, where a filtering process is performed.
- Storage of the filtered addresses in a server with a Database Management System (DBMS).
- Database query application through a server with a web application.

Keywords: Arduino; Bluetooth; Security; Virtualization; Encryption.

Índice

Figuras

1

Introducción

La seguridad en el mundo IT está cada vez más presente y es base esencial para cualquier desarrollo que se pretenda poner en producción con un mínimo de garantías. Cualquier organización que ponga un servicio a disposición del público o de uso interno, ha de tener en cuenta una variedad de características de seguridad que actualmente están alineadas con la mayoría de las certificaciones de seguridad.

Muchas certificaciones de seguridad comparten ciertas características de seguridad y algunas de ellas son aplicables al proyecto desarrollado durante el curso 2021 centrado en el seguimiento de dispositivos a través de Bluetooth. Algunas características no son alcanzables por las limitaciones propias del proyecto, por ejemplo, no se pueden tener replicados lo sistemas de alimentación de las máquinas que dan soporte a los distintos servidores.

La idea de aplicar múltiples capas de seguridad también da pie a la posibilidad de tener flexibilidad en la escalabilidad y mejorar considerablemente la fiabilidad del sistema.

La virtualización de servicios abre un amplio abanico de opciones en términos de escalabilidad y fiabilidad. También está el hecho de que el uso de un dispositivo Arduino orientado al IoT facilita la gestión del acceso a redes y algún otro aspecto.

1.1 Motivación

Durante el año 2021, se desarrolló un proyecto de trabajo de fin de grado con el propósito de realizar un seguimiento de los dispositivos que se encontraban dentro de un edificio o una habitación, y así poder trazar una relación de proximidad entre casos positivos de COVID-19, permitiendo localizar posibles casos de contagio.

Existen una variedad de normas que aplican a cualquier aplicación informática que vaya a ser puesta en producción, que cubren ámbitos como, la protección de datos, la seguridad en la administración de equipos y servicios, o la fiabilidad en la gestión de la información. Estas normas están recogidas en diversas certificaciones de seguridad IT como el Esquema Nacional de Seguridad (ENS) [16], la norma ISO 27001 [24] y su anexo 27002, la certificación Webtrust [46] y la certificación TISAX [42], Estas características actualmente se exigen tanto a empresas públicas como empresas privadas que proporcionen servicios o plataformas software, y no estaban incluidas dentro del alcance del trabajo fin de grado original.

Si bien es cierto que la pandemia asociada al COVID-19 actualmente en España está bajo control, y tanto el índice de contagios como la mortalidad se han reducido drásticamente [17][18], el proyecto sigue teniendo utilidad como herramienta de rastreo de personas en edificios o para otras aplicaciones como podrían ser, por ejemplo, el control acceso a áreas restringidas, futuras pandemias, etc...

1.2 Objetivos

Los objetivos principales de este Trabajo de Fin de Grado es dotar al proyecto inicial con una capa de seguridad IT, hacerlo escalable tanto horizontal como verticalmente y mejorar la fiabilidad. A continuación, se describen pormenorizadamente referidos objetivos.

1.2.1 Seguridad

Se pretenden añadir múltiples capas de seguridad alineadas con algunas de las certificaciones más relevantes que actualmente se exigen a cualquier empresa que oferte un servicio o plataforma software.

Las principales características de seguridad que se pretenden cumplir son;

- Cifrado en tránsito de la información.
- Cifrado en reposo de la información en la base de datos (TDE de MySQL).
- Control de acceso remoto a la administración de los servicios mediante Autenticación Multi-Factor (MFA). [35]

1.2.2 Escalabilidad

Al usar servicios virtualizados, más concretamente, los servicios web y el servicio de gestión de bases de datos, podemos escalar las necesidades del proyecto tanto de forma vertical como horizontal.

El escalado horizontal se basa en la agregación directa de máquinas virtuales que permitan balancear la carga de trabajo y así cada máquina tiene menos peticiones que servir.

El escalado vertical se basa en la mejora de los recursos disponibles de cada máquina virtual, como podría ser un aumento de memoria, mejorar la capacidad de la CPU, aumentar la capacidad de almacenamiento masivo, etc. En la Figura 1 se puede observar tanto el escalado horizontal como el vertical.

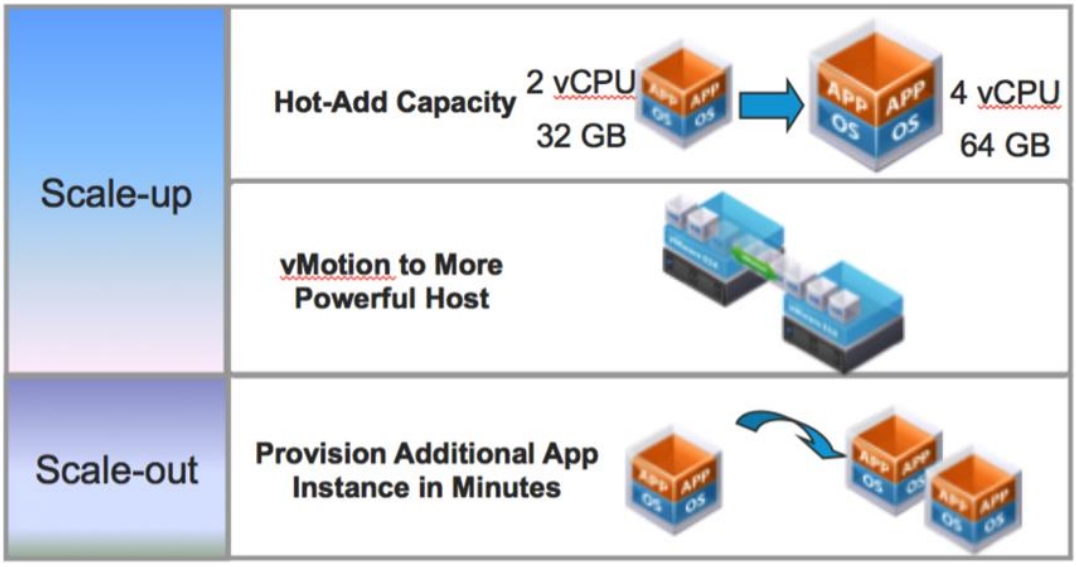


Figura 1. Escalado vertical y horizontal [15].

Si bien es cierto que en este proyecto el escalado, tanto vertical como horizontal, se ha de aplicar de forma manual, todo está preparado para poder usar herramientas orquestadoras de servicios, como *Docker Swarm* o *Kubernetes*,

las cuales permitirían el escalado automático y programado en función de la carga de trabajo o de otros factores relevantes.

1.2.3 Fiabilidad

La fiabilidad añadida al proyecto la proporciona principalmente el uso de servicios virtualizados separados y la gestión de copias de seguridad de la base de datos.

Gracias al uso de servicios separados tenemos cierta tolerancia a fallos ya que, si una máquina deja de funcionar, el resto no se ven afectadas, excepto en el caso de que deje de funcional el servidor donde se aloja la base de datos, en cuyo caso no se podrán almacenar nuevas direcciones captadas por el Arduino ni tampoco se podrán consultar las direcciones almacenadas previamente.

El servidor que contiene la base de datos dispone de una configuración Raid 1, aún que, la configuración de Raid 1 está sobre dos volúmenes lógicos alojados en un mismo dispositivo de almacenamiento físico, lo cual no aporta nada en caso de fallo en el disco, pero es una limitación por los medios disponibles.

También están configuradas las copias de seguridad, tanto de la estructura de la base de datos, como de los datos almacenados. Estas copias de seguridad tienen un impacto considerable en la capacidad de almacenamiento de la máquina virtual que aloja el servicio de gestión de la base de datos.

1.3 Plan de trabajo

La metodología seguida para el desarrollo y finalización de este proyecto ha sido *Rapid Application Development* (RAD) [34]. En la Figura 2 se muestra el ciclo completo de la metodología RAD.

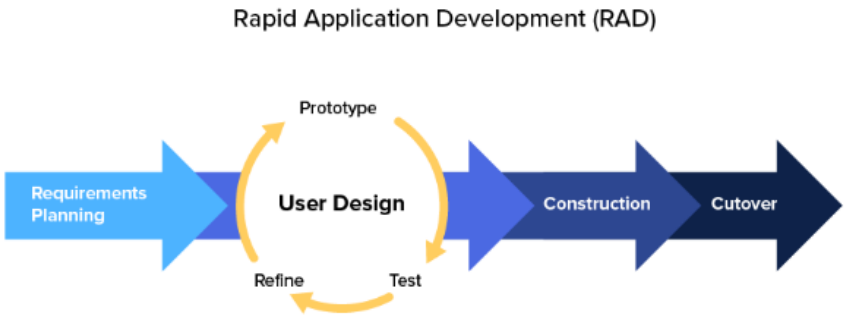


Figura 2. Ciclo de vida del método RAD [34].

En este proyecto, se ha seguido con este ciclo de vida, ligeramente adaptado para el tamaño y las necesidades del proyecto. En la Figura 3 se muestra el plan de trabajo con el que se ha desarrollado la aplicación:

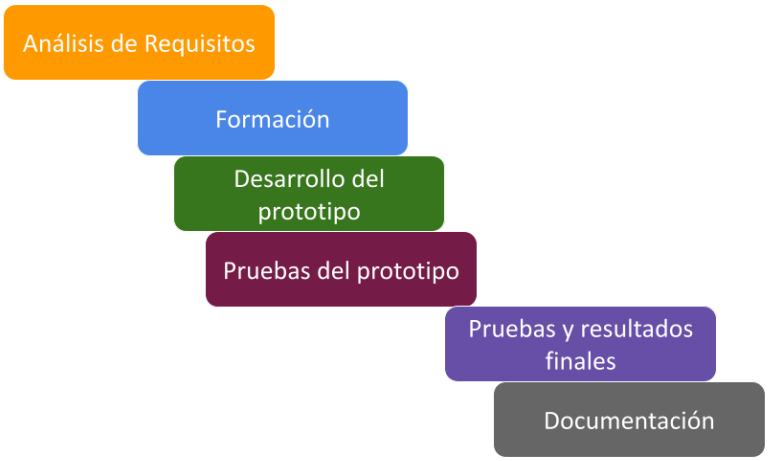


Figura 3. Diagrama del plan de trabajo.

Las partes etiquetadas como "*Formación*", "*Desarrollo del prototipo*" y "*Pruebas del prototipo*" se han aplicado para cada una de las cuatro partes en las que se ha dividido el proyecto.

- **Parte I** Dispositivo de rastreo Arduino.
- **Parte II** Servicio web y filtro de direcciones.
- **Parte III** Servicio de bases de datos.
- **Parte IV** Servicio web de consulta

1.4 Estructura de la memoria

La documentación asociada a la aplicación de este trabajo fin de grado contiene los siguientes capítulos:

▪ **Capítulo 1. Introducción.** Se presentan los antecedentes relativos al proyecto, la motivación, los objetivos y el plan de trabajo del proyecto.
▪ **Capítulo 2. Estado del arte.** Se describe la investigación y estudio de los distintos elementos tecnológicos actuales en el ámbito del proyecto.
▪ **Capítulo 3. Solución propuesta.** Se presentan las herramientas software y hardware necesarias para la implementación del proyecto.
▪ **Capítulo 4. Pruebas de rendimiento y resultados.** Se muestran una serie de resultados obtenidos para determinar si el cifrado de datos afecta al rendimiento de la placa Arduino.
▪ **Capítulo 5. Conclusiones y trabajo futuro.** Se presentan las conclusiones del proyecto realizado, y algunas ideas de mejora para el futuro.
▪ **Capítulo 6. Bibliografía.** Recopilación de las referencias bibliográficas

consultadas durante el desarrollo del proyecto.

Finalmente, se añaden una serie de apéndices con información adicional para complementar la información del proyecto.

2

Estado del Arte

En este capítulo se muestra la investigación realizada al inicio del proyecto con la intención de explorar otras soluciones que, o bien, sean similares a la aquí propuesta, o bien, persigan el mismo objetivo que este proyecto.

2.1. Descripción general de la solución inicial

Antes de describir los elementos hardware/softwares analizados en la realización de este proyecto, se describirá de forma breve el funcionamiento de la solución original [22].

El proyecto original (ver Figura 4) se centra en el desarrollo de una solución de bajo coste para poder realizar un rastreo de personas positivas en COVID-19 en empresas y organismos públicos.

Esta solución se basa en la detección de las MAC bluetooth de los dispositivos móviles y/o balizas (beacons) mediante una placa Arduino que se dispondrá en las entradas de edificios y/o estancias. Integrando en la placa los módulos bluetooth y Wifi se hacen posibles las funciones de detectar y enviar los datos directamente a una base de datos donde se almacena toda la información.

Simultáneamente, se implementa una pequeña aplicación para dispositivos Android capaz de dar de alta a usuarios y las distintas estancias de la empresa, quedando la posibilidad de poder eliminar o modificar cualquier dato.

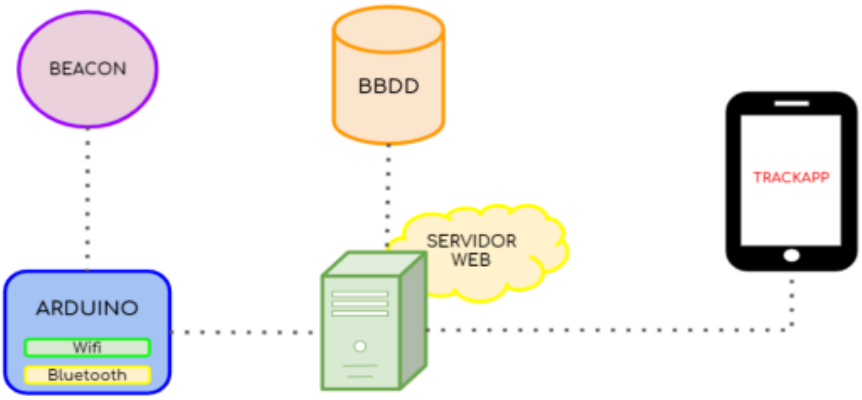


Figura 4. Esquema general del proyecto original [22].

2.2. Elementos hardware analizados

Al inicio del proyecto, una de las primeras decisiones a tomar fue decidir a qué dispositivos hardware se les iba a realizar el seguimiento de sus direcciones, porque en el caso de la tecnología, se va a trabajar con la misma propuesta en el proyecto original, esto es, direcciones MAC de Bluetooth Low Energy (BLE) [8][13].

Al igual que en el proyecto original, las beacons o balizas con tecnología BLE van a formar parte de los elementos rastreables. Las beacons son dispositivos de pequeño tamaño y peso, rondan los 70 gramos de peso, que emiten señales en broadcast para ser fácilmente detectadas por cualquier dispositivo cercano, en un rango de unos 65 metros de alcance máximo, aún que es cierto que la existencia de obstáculos, como muros, reducen considerablemente el rango de la señal.

También se consideró rastrear las direcciones de dispositivos móviles, pero esta idea no es actualmente viable debido a la capa de seguridad que proporciona BLE para proporcionar privacidad y evitar el seguimiento de dispositivos móviles.

Esta medida de seguridad se conoce como "*Resolvable Random Private Address*" o RPA (ver Figura 5), y es una característica que está presente y activa por defecto en cualquier dispositivo con BLE que no tenga como principal objetivo el ser sometido a seguimiento público. En el caso de las beacons, por defecto no tienen activo RPA, ya que su principal objetivo es el de proporcionar un seguimiento público fiable.

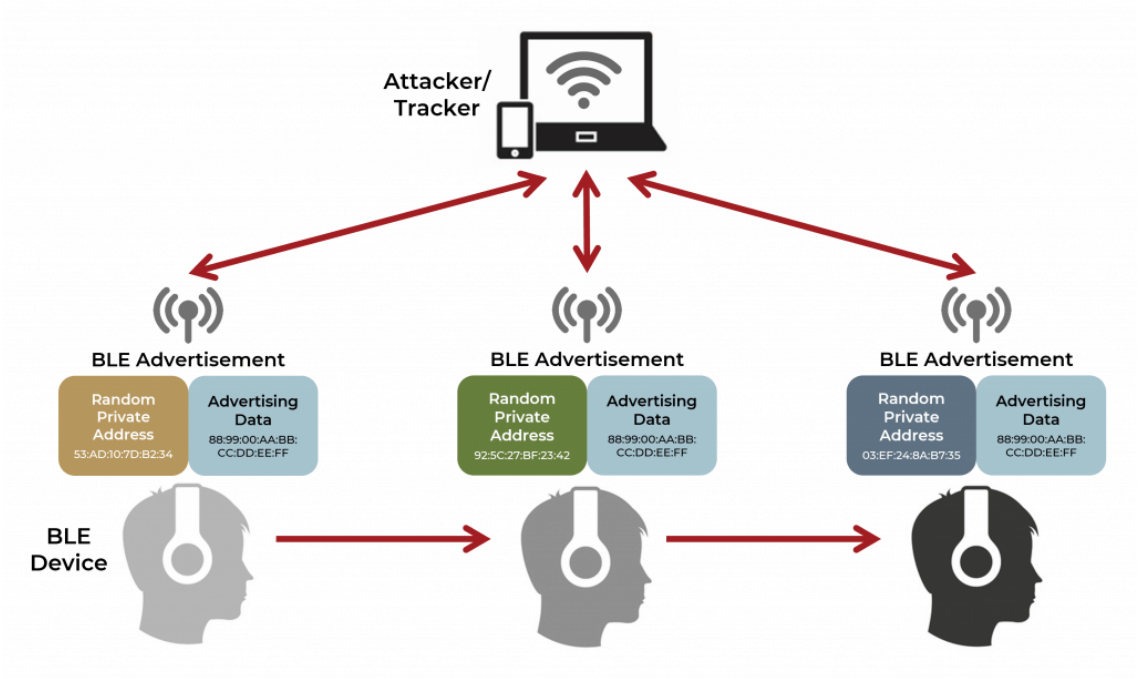


Figura 5. Resolvable Random Private Address (RPA) [9].

RPA es una característica de seguridad que proporciona privacidad e imposibilita el seguimiento de un dispositivo. Las direcciones MAC emitidas públicamente por los dispositivos BLE con RPA (ver Figura 6) son direcciones con el mismo formato que una dirección MAC estática de una interfaz Bluetooth fijada por el fabricante, con la diferencia de que dicha dirección no es estática y cambia frecuentemente para evitar el seguimiento del dispositivo.

Figura 6. Estructura de una dirección RPA.

En una dirección RPA el *prand* es un número de 24 bits donde los 2 bits menos significativos (LSB) son fijos, y los restantes 22 son aleatorios. Los 24 bits más significativos (MSB) son el resultado de aplicar un hash al "*Identity Resolving Keytienen*" (IRK) y el *prand* previamente generado de forma aleatoria. Esta estructura concuerda a la perfección en formato con una dirección MAC estática fijada por el fabricante, con la diferencia de que no es estática y varía con el tiempo. Este hecho imposibilita el seguimiento de dispositivos móviles que dispongan de BLE.

Si bien es cierto que BLE permite la desactivación manual de RPA, permitiendo así que el dispositivo sea rastreable públicamente en todo momento a través de su dirección MAC estática, todo dispositivo móvil del mercado que dispone de BLE tiene RPA activo por defecto y deshabilitarlo no es un proceso trivial. También se ha comprobado que RPA se encuentra activo por defecto en dispositivos de audio, tabletas y tarjetas Bluetooth USB, siempre que dispongan una versión de Bluetooth compatible con BLE.

Para la plataforma hardware encargada del rastreo de las direcciones públicas, se observó que las opciones se reducían a dos, placas Raspberry Pi (ver Figura 7), o placas Arduino (ver Figura 8).

Figura 7. Placa Raspberry Pi.

Las placas Raspberry Pi son computadores que disponen de buenas capacidades de cómputo para uso general, con una media de 7.5 GFLOPS, compatibilidad tanto con Bluetooth como con WIFI, conectividad para múltiples USB, entrada Ethernet, y bancos de memoria de entre 512 MB y 1 GB. Dispone de un sistema operativo propio, llamado Raspbian, basado en Debian y optimizado para las diferentes versiones de Raspberry Pi [47].

Las placas Arduino [5] son microcontroladores y en comparación con Raspberry Pi, su capacidad de cómputo es mucho menor, y su funcionamiento es bastante más sencillo y limitado. Estas placas cuentan con un pequeño software denominado monitor que se encarga se ejecutar cíclicamente el código, precompilado en una máquina externa, por lo que la interacción entre el hardware y el software se limita a la ejecución de código fuente que se le transfiere a través de un puerto serie utilizando el IDE desarrollo de la plataforma Arduino.

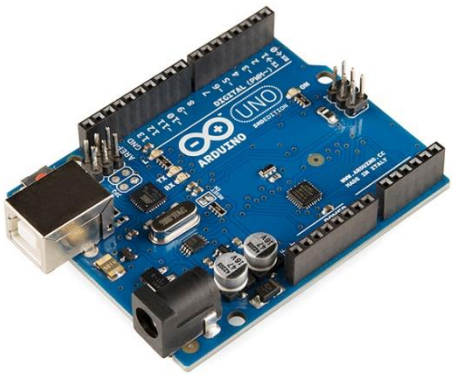


Figura 8. Placa Arduino [5].

El entorno de desarrollo integrado (IDE) usado para escribir y cargar programas compatibles con Arduino, llamado Arduino IDE [36], está elaborado con lenguaje Java. La última versión disponible en el momento de la elaboración de este proyecto es la 2.0.0. Es de código abierto, y además incorpora por defecto el acceso a un repositorio con diversas funcionalidades, especialmente para la gestión de entradas y salidas (I/O).

Existen distintos modelos de placas Arduino, la más popular es Arduino UNO, debido a su precio y al conjunto de opciones que ofrece. En el proyecto original, el modelo de Arduino empleado fue Arduino Mega 2560 (ver Figura 9), junto a dos placas adicionales. La placa Bluetooth HM-10, que proporciona conectividad Bluetooth, compatible con BLE y la placa WiFi ESP-8266-esp01, que proporciona conectividad Wifi. El modelo empleado en este proyecto es el Arduino MKR WIFI 1010 [4], que incorpora conectividad WIFI y Bluetooth compatible con BLE gracias a su interfaz u-blox NINA-W102 sin necesidad de placas externas. Se trata de una versión de Arduino pensada para el IoT.

Figura 9. Elementos Hardware del proyecto original. De izquierda a derecha, Arduino Mega 2560, Bluetooth HM-10 y Wifi Esp-8266-esp01.

El modelo elegido para el proyecto, Arduino MKR WIFI 1010, además dispone de un chip específico para criptografía, ATECC508 Crypto Chip, el cual permite entre otras cosas el almacenamiento de claves criptográficas, como claves SSL, TLS o claves asimétricas RSA.

Por diversos motivos, durante el desarrollo de este proyecto, el acceso o la facilidad para adquirir placas o circuitos integrados se encuentra muy limitado, lo que reduce las posibilidades de libre elección de elementos hardware considerablemente. Aun así, la elección de la placa Arduino MKR WIFI 1010 es en gran medida un acierto, especialmente por sus características enfocadas al IoT, pero con todas las limitaciones técnicas propias de Arduino, algunas de las cuales se discuten y elaboran en mayor profundidad más adelante.

2.3. Opciones de seguridad analizadas

Las principales características de seguridad que desde el principio formaban parte del proyecto eran que las transmisiones de información que pudieran dar caso a intercepción fueran cifradas, que el acceso remoto como administrador a los servicios virtualizados requiriese autenticación multi factor y que la información en reposo en la base de datos esté cifrada.

Estas tres características han sido seleccionadas ya que son características comunes para muchas de las certificaciones de seguridad IT más exigentes y solicitadas por las organizaciones que brindan cualquier servicio que este apoyado por una o varias aplicaciones o plataformas IT. Algunas de estas certificaciones son el ENS, Webtrust, ISO 27001 y TISAX.

2.3.1 Seguridad en las comunicaciones

Para la transferencia de información cifrada se consideraron opciones como usar comunicación HTTPS [20], "*HTTPS (protocolo seguro de Transferencia de Hiper-Texto) es un protocolo que permite establecer una conexión segura entre el servidor y el cliente, que no puede ser interceptada por personas no autorizadas.*".

El protocolo HTTP se ubica en la séptima capa del modelo OSI [29], la capa de aplicación, sin embargo, la parte del cifrado que permite usar HTTPS se

encuentra en la quinta capa, conocida como capa de sesión, a través de SSL o TLS [40].

El funcionamiento del protocolo HTTPS (ver Figura 10) consta de una serie pasos que se realizan para garantizar que la información va a ir por un canal cifrado donde previamente se ha verificado que el servidor es el auténtico servidor con el que el cliente quiere comunicarse. El procedimiento consiste en;

1. Cliente: Enviar un mensaje de "saludo" al servidor, indicando que se quiere iniciar una conexión segura.
2. Servidor: El servidor responde con otro "saludo" y un certificado conteniendo la clave pública de servidor.
3. Cliente: Verifica que el certificado se corresponde con unos de los certificados de confianza instalados en el navegador web o en el sistema operativo
4. Cliente: Envía al servidor una clave simétrica de sesión, dicha clave va cifrada con la clave pública del servidor.
5. Servidor: Descifra el mensaje enviado por el cliente y adquiere conocimiento de la clave simétrica enviada por el cliente.
6. Servidor: Cifra los mensajes que envía al cliente usando la clave simétrica de sesión.
7. Cliente: Cifra los mensajes que envía al Servidor usando la clave simétrica de sesión.

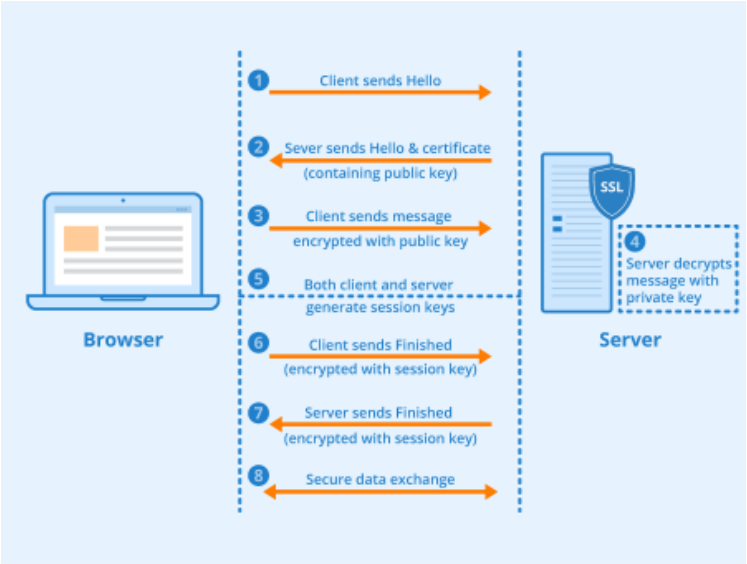


Figura 10. Protocolo HTTPS [21].

El cifrado en tránsito hay que aplicarlo en dos partes del proyecto, una es la comunicación que existe entre el dispositivo rastreador y el servidor web encargado de filtrar las direcciones y almacenarlas, y la otra parte es la comunicación entre el servidor web de consulta de información y los posibles terminales que soliciten hacer una consulta al servidor.

En el primer caso, usar HTTPS no ha sido posible, ya que las placa Arduino usada en el proyecto no permite almacenar una lista de certificados de confianza para la correcta verificación de que el servidor es el auténtico, es decir, no se puede cumplir con el paso número 3 "*3. Cliente: Verifica que el certificado se corresponde con unos de los certificados de confianza instalados en el navegador web o en el sistema operativo*".

Debido a esta limitación tecnológica del dispositivo rastreador, se han utilizado otros métodos, que, a pesar de cumplir con el objetivo de enviar y recibir la información cifrada, no tiene todas las características de seguridad que aporta HTTPS. Para aliviar la falta de algunas características de seguridad de HTTPS se han implantado algunas medidas compensatorias.

En el caso de las comunicaciones entre el servidor web de consulta y los terminales de usuarios si se ha podido emplear HTTPS como protocolo de comunicación entre ambos. Para la gestión de certificados del servidor web de consulta se ha usado un certificado TLS.

2.3.2 Seguridad en la administración remota de los servicios

Para la administración remota de las tres máquinas virtuales hay considerar diversos aspectos del proyecto, como la envergadura, los sistemas operativos de las máquinas virtuales, las opciones de seguridad disponibles y las necesidades del proyecto. Teniendo todo esto en cuenta, las principales opciones que satisfacen los requerimientos de administración remota son;

- Secure Shell (SSH) con OpenSSH.
- Virtual Network Computing (VNC) con TigerVNC.
- Remote Desktop Protocol (RDP) con Vinagre.

SSH [38] es un protocolo de red que permite el acceso remoto a equipos de manera segura a través de un canal cifrado entre el cliente y el servidor. OpenSSH

[30] es una de las aplicaciones más usadas en entornos Linux que implementa el protocolo SSH para la gestión remota de equipos de manera segura a través de un medio no seguro, como es internet. Las principales características de OpenSSH son su simplicidad, su fácil configuración y tiene una característica de gran valor para este proyecto, que es la posibilidad de añadir un segundo factor de autenticación (2FA) en el control de acceso a los servicios remotos.

VNC [45] es un sistema que permite el control remoto de máquinas a través de una interfaz gráfica. Está basado en *Remote Frame Buffer protocol* (RFB). Dependiendo de la plataforma existen diversas aplicaciones que hacen uso de VNC. En el caso de este proyecto, donde las máquinas virtuales son todas Linux, se ha estudiado la posibilidad de usar TigerVNC [41]. TigerVNC es una aplicación de código abierto con funcionalidad cliente/servidor multiplataforma, aún que su funcionalidad al completo solo está disponible para Linux. La principal ventaja de TigerVNC es que al estar basado en VNC permite la gestión remota a través de una interfaz gráfica, lo cual posibilita una gestión algo más intuitiva y, en algunos casos, de forma más rápida que en los sistemas remotos basados en consola como SSH.

RDP [43] es un protocolo cliente/servidor propiedad de Microsoft. RDP tiene interfaz gráfica que, al igual que VNC, hace algo más intuitiva la gestión remota. Aún que es un protocolo de Microsoft, tanto cliente como servidor son multiplataforma, y existen aplicaciones para Linux como Vinagre [44]. Esta opción es similar a TigerVNC en términos de funcionalidad, pero en la práctica, más concretamente en este proyecto, tras probarlo en varias ocasiones el resultado en términos de rendimiento ha sido algo precario, especialmente al compararlo con TigerVNC. La falta de rendimiento era especialmente evidente en la respuesta tardía del movimiento del ratón, así como las pulsaciones de teclas tanto del ratón como del teclado y con todo lo que ello conlleva.

Si bien es cierto que todas las aplicaciones probadas cubren las necesidades expuestas al principio de este subcapítulo, debido a su sencillez, rendimiento y familiaridad, se ha decidido emplear OpenSSH como aplicación para la gestión remota de las máquinas virtuales.

3

Solución propuesta

En este capítulo se expone y detalla cada una de las partes que forman parte del proyecto, más concretamente, el dispositivo de rastreo Arduino y todos los servicios que han sido virtualizados y dan soporte al proyecto.

3.1 Arduino

La placa Arduino con la que se ha trabajado es la MKR Wifi 1010. Esta placa está especialmente diseñada para entornos IoT, compatible con BLE y con algunas características criptográficas [6]. Todas estas características están perfectamente alineadas con los objetivos a cumplir en este proyecto.

El módulo *Nina W102* [28], nos proporciona conectividad Wifi y Bluetooth, no teniendo que hacer uso de módulos hardware externos para disponer de ambas opciones de conectividad.

3.1.1 Cifrado en Arduino

Una de las partes más importantes de este proyecto era la transmisión de información de forma segura, proporcionando privacidad de los datos en tránsito que se envían desde el dispositivo rastreados hasta el servidor encargado de procesar y almacenar la información.

Para cumplir con este objetivo, la información que es capturada por la placa Arduino debía ser enviada en un formato que no proporcionase ningún tipo de información útil si se capturaba en tránsito, es decir, la información debía ir cifrada.

Para el cifrado de la información se ha optado por emplear la librería *THiNX AESLib* [1]. Esta librería es un fork de otro repositorio que dejo de tener soporte por la comunidad. *THiNX AESLib* es un repositorio activo y su última reléase (v2.3.0) es del 21 de septiembre del año 2022. Es una de las librerías que se encuentran indexadas en los repositorios recomendados del propio Arduino IDE.

Con esta librería podemos cifrar las direcciones Bluetooth capturadas mediante AES128, es decir, algoritmo AES con una longitud de clave de 128 bits. Si bien es cierto que sería mejor aplicar AES192 o AES256, ya que ambos trabajan con claves de mayor tamaño, haciendo más difícil descifrar los mensajes por fuerza bruta, en el caso de Arduino, debido a sus limitaciones técnicas, vamos a trabajar con AES128, que hoy en día sigue siendo seguro.

Para poder trabajar con la placa Arduino MKR WIFI 1010, hemos de instalar el paquete de placas tipo *"Arduino SAMD boards (32-bits ARM Cortex M0+)"* y seleccionar la placa *"Arduino MKR Wifi 1010"*. Para tener acceso a internet instalamos la librería *"WiFiNINA"*, indicamos el SSID de la red Wifi a la nos vamos a conectar y la contraseña de esta. Para más detalles sobre la configuración inicial, consultar el código fuente asociado a este TFG.

El proceso de captura de direcciones Bluetooth no ha cambiado en lo que al proyecto original se refiere, la gran diferencia surge en el momento de enviar la información. En el proyecto original, la información era enviada en formato de texto plano, pero ahora no es el caso. La información capturada es cifrada mediante un conjunto de funciones y luego enviada como texto cifrado.

Para abordar satisfactoriamente el proceso de cifrado, dadas las características de los elementos hardware que forman parte de la solución original del proyecto, se han tomado varias decisiones. Puesto que la aplicación del dispositivo rastreado no está diseñada como una aplicación de usuario, se propone usar una clave de 128 bits sin padding, es decir, se crea un vector que contiene 16 valores de 1 byte cada uno, un total de 16 bytes, que son 128 bits. También hay que tener en cuenta que, si el tamaño del mensaje excede los 128 bits, hay que dividirlo en fragmentos de 128 bits, cifrar cada fragmento y luego concatenarlos antes de ser enviados.

En el caso concreto de Bluetooth, las direcciones MAC están formadas por 6 grupos de 1 byte cada uno, lo cual hace un total de 48 bits. Ya que los mensajes que se cifran son de 128 bits, y que tenemos que enviar dos direcciones MAC, una del dispositivo bluetooth que ha sido rastreado y otra del dispositivo que ha realizado dicho rastreo, tenemos que cifrar un mensaje de 96 bits, por lo que hay que añadir un padding de 32 bits, es decir, añadir 32 ceros al final de cada dirección antes de cifrar el mensaje resultante. La función de cifrado en la librería AESLib se denomina *"encrypt"*, y tiene como parámetros:

- Mensaje para cifrar
- Longitud del mensaje a cifrar
- Mensaje cifrado resultante
- Clave de entrada
- Tamaño de clave de entrada
- Vector de inicialización aleatoria para AES

```
int cipherlength = aesLib.encrypt((byte*)mensaje, longitud,
(byte*)textocifrado, clave, clave_len, iv_vector);
```

La única salida de esta función es el tamaño del mensaje cifrado resultante. El mensaje cifrado se encuentra en uno de los parámetros de la propia función de cifrado. El proceso de descifrado requiere de los mismos parámetros, pero en el caso concreto de este proyecto, no es necesario descifrar en el dispositivo rastreador. El proceso de descifrado será realizado en el servidor web al que se envía la información cifrada.

El envío del mensaje cifrado queda algo más simple que en el proyecto original gracias al módulo WifiNINA, el cual proporciona el objeto "client", que nos permite enviar datos de forma muy sencilla a cualquier servidor web que disponga de un script de recogida de datos en PHP. En la Figura 11 se muestran dos versiones de cómo se envía la información al servidor. En la parte superior de la figura se muestra el envío de la información recogida en el proyecto original, y en la parte inferior se muestra cómo se envía el bloque de datos cifrados para este proyecto.

```
ESP8266.println("AT+CIPSTART=\"TCP\",\"192.168.43.252\",80");
    if( ESP8266.find("OK"))
    {
        for(pos=0; pos<num_dispositivo ; pos++) //Construimos el encabezado de la peticion HTTP
        {
            mac_http = mac_buffer[pos];
            rssi_http = rssi_buffer[pos];
            name_http = name_buffer[pos];

            String request_1 = "GET /trackapp/device.php?aid=";
            String request_2 = "&mac=";
            String request_3 = "&rssi=";
            String request_4 = "&name=";
            String request_5 = " HTTP/1.1\r\nHost: 192.168.43.252\r\n\r\n";

            String request= request_1 + aid + request_2 + mac_http + request_3 + rssi_http + request_4 + name_http
        +request_5;

            ESP8266.print("AT+CIPSEND=");  //Enviamos el tamaño en caracteres de la petición HTTP:
            ESP8266.println(request.length());

            if(ESP8266.find(">")) // ">" indica que podemos enviar la petición HTTP
            {
                Serial.println("Enviando HTTP . . .");
                ESP8266.println(request);
```

```
if (client.connect(server, 80)>0) {  // Conexión con el servidor

    client.print("GET /enviardata.php?data=");
    client.print(data_cypher);  // Bloque cifrado de datos
    client.println(" HTTP/1.0");
    client.println("User-Agent: Arduino 1.0");
    client.println();

    }
    } else {
        if (mySerialPort == true){
            Serial.println("Error en la conexión");
        }
    }
```

Figura 11. Envío de datos a través de HTTP. Proyecto original (arriba) y proyecto actual (abajo).

3.2 Servicios Virtualizados

Uno de los pilares de este proyecto es el uso de servicios virtualizados que nos brindan características como escalabilidad, fiabilidad y la posibilidad de trabajar con sistemas distribuidos.

Para la gestión de todos los servicios virtuales se ha optado por el software de virtualización VMware Workstation Player, que nos permite crear máquinas virtuales para poner en marcha los servicios que vayamos a necesitar. En el caso concreto de este proyecto, vamos a crear 3 máquinas virtuales, cada una con varios servicios como, servidores web o servidores de base de datos.

En la Figura 12 se muestra el esquema general del proyecto, con las máquinas virtuales que dan soporte a los diferentes servicios.

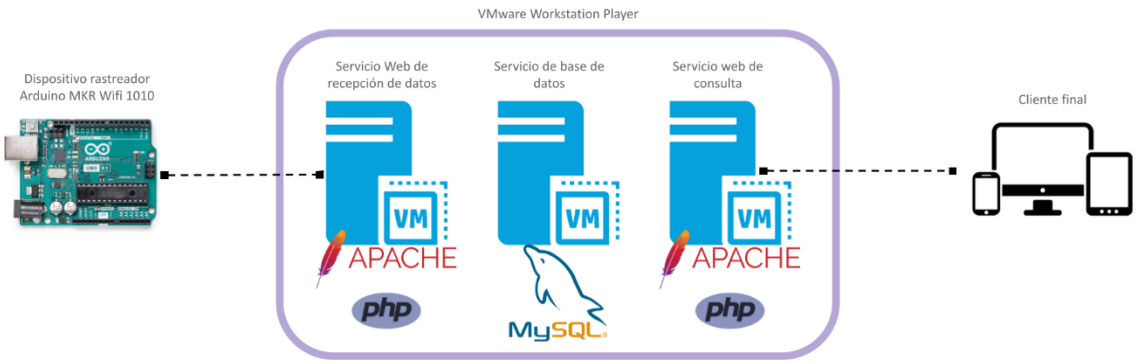


Figura 12. Esquema del proyecto.

La gestión de cada uno de los servicios ha sido a través de SSH, ya que se ha querido evitar la carga que supondría para las máquinas virtuales el disponer de editores de texto que faciliten el trabajo de edición de código o de configuración. En las máquinas virtuales se ha instalado OpenSSH y en la máquina anfitriona se ha usado la aplicación "PuTTY" como cliente SSH.

Para cumplir con la normativa propuesta en múltiples certificaciones de seguridad en entornos IT, se han seguido las siguientes dos reglas:

- El control de acceso a través de SSH requiere de un segundo factor de autenticación.
- El entorno de acceso a las máquinas virtuales está limitado a la máquina física que da soporto al software de virtualización, es decir, las máquinas virtuales solo son accesibles por la máquina física anfitriona o host.

Para cumplir con la primera regla, se ha configurado el servidor OpenSSH siguiendo las directrices que nos proporciona Ubuntu en su sección de tutoriales *"Configure SSH to use two-factor authentication"* [11]. Hay que tener en cuenta que el cliente SSH usado es PuTTY, y este no permite añadir una configuración para el control de acceso de forma automática desde el momento que añadimos el segundo factor de autenticación, por lo que cada vez que se quiere acceder hay que aportar el usuario, la contraseña y el *one time password* (OTP). La aplicación generadora de tokens usada ha sido Google Authenticator en un dispositivo Android.

Para limitar el acceso únicamente a la máquina host hay que hacerlo en dos pasos. El primer paso consiste en denegar el acceso por defecto de forma global, es decir, se prohíbe el acceso a través de SSH a cualquier dirección entrante. Para hacer esto hay que modificar el fichero de configuración *"/etc/hosts.deny"* y añadir una línea *"sshd: ALL"*, con esto restringimos todos los

accesos. El segundo paso consiste en agregar una o varias direcciones que si podrán acceder. En el fichero de configuración *"/etc/hosts.allow"* agregamos una línea *"sshd: 192.168.1.31"* siendo esta dirección IP la de la máquina host.

3.2.1 Servicio Web de recepción de datos

La máquina virtual encargada del servicio web de recepción de datos enviados por las placas Arduino tiene las siguientes características (ver Figura 13):

- CPU: 2 procesadores de 2 cores cada uno.
- Memoria: 8 GB.
- Almacenamiento: 20 GB.
- Adaptador de red (NIC1): Bridged.
- Adaptador de red (NIC2): NAT.
- SO: Ubuntu Server 22.04 LTS (Abril 2022).

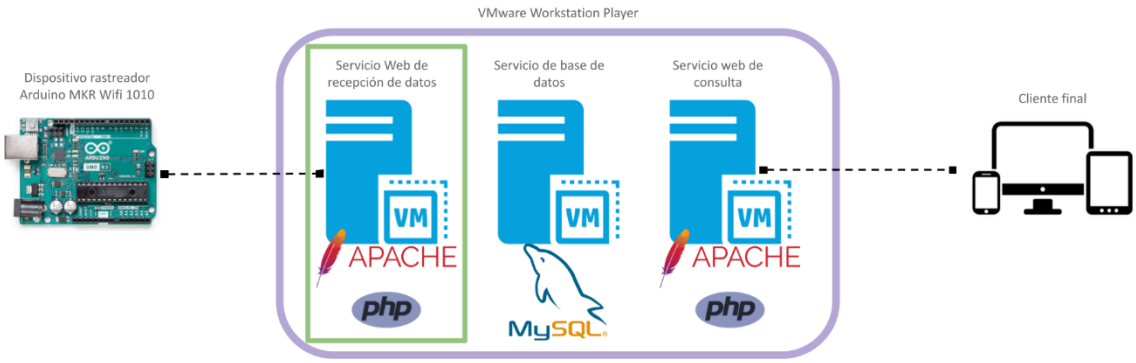


Figura 13. Servicio Web de recepción de datos.

Esta máquina virtual dispone de dos adaptadores de red, uno configurado como Bridged, y otro como NAT [10]. El primer adaptador de red, NIC1, está configurado como Bridged y su principal objetivo es tener acceso directo a Internet, y así facilitar el envío de información por parte de la placa Arduino al servidor de recepción de datos. Al tratarse de una configuración Bridge, comparte el estado de red de la tarjeta de red física del host, con una dirección IP diferente.

El segundo adaptador de red está configurado como NAT, y permite el acceso a otras máquinas virtuales que se encuentren en el mismo entorno físico, aún que también permite el acceso a internet a través del host. A través de esta interfaz de red se puede acceder a la máquina virtual que da soporte al servicio de base de datos. Para más detalles sobre la instalación y configuración de las

máquinas virtuales consultar el *Apéndice II. Instalación y configuración de las máquinas virtuales.*

Esta máquina virtual tiene instalado Ubuntu Server como sistema operativo. Durante la instalación del SO se pueden instalar algunos módulos adicionales como OpenSSH, y una vez finalizada la instalación del SO se han de instalar los siguientes paquetes:

- Apache HTTP Server [39].
- PHP [31].

Si bien es cierto que podríamos haber instalado *"Linux, Apache, MySQL and PHP"* (LAMP), en el caso de esta máquina virtual, no es necesario instalar MySQL, por lo que es preferible solo instalar los paquetes que realmente son necesarios. La información de instalación de Apache y PHP se encuentra más detallada en el *Apéndice II. Instalación y configuración de las máquinas virtuales.*

Apache es un servidor web capaz de recibir y enviar peticiones http a través de internet en el puerto 80. El principal objetivo de Apache en esta máquina virtual es posibilitar la recepción de información por parte de las placas de rastreo y lanzar algunos scripts PHP para la gestión y almacenamiento de la información proporcionada por la propia placa Arduino. Aún que la configuración por defecto de Apache ha ido mejorando con los años en términos de seguridad, se ha optado por realizar una configuración segura de Apache.

Los principales aspectos de configuración segura de Apache a tener en consideración y que están alineados con los propuesto por algunas de las organizaciones de seguridad más relevantes como, ISACA [23], CPACanada [14] o Apache HTTP Server [3] son;

- Ocultar información del servidor, como la versión o los tokens que van en el *header* de respuesta HTTP.
- Limitar los privilegios de Apache, siguiendo el principio de privilegios mínimos. Es importante asegurarse de que Apache no tiene permisos de superusuario.
- Deshabilitar los ficheros .htaccess que agregan parámetros de configuración adicionales para Apache.
- Restringir el acceso de administración del servidor Apache a ciertas IP.
- Prevención de ataques *Denial of Service* (DoS), mediante;
    - Habilitar el módulo de Apache *mod_reqtimeout*, el cual limita las conexiones vacías sin peticiones.
    - Habilitar y reducir el valor de tiempo de los parámetros *TimeOut* y *KeepAliveTimeout*.

- Habilitar y aumentar el valor del parámetro *MaxRequestWorkers* al máximo posible.
- Habilitar el módulo Multi-Processing Modules (MPM) [2].

Para más información sobre la configuración segura de Apache consultar el *Apéndice I. Configuración segura de Apache.*

PHP nos permite descifrar los paquetes de datos enviados por la placa Arduino, realizar ciertas verificaciones de seguridad de los datos y finalmente almacenarlos en la base de datos que se encuentra en otra máquina virtual. Se ha optado por usar PHP debido a que el módulo de comunicaciones WifiNINA de la placa Arduino facilita en gran medida el envío de paquetes de datos a servidores web que dispongan de PHP.

3.2.2 Servicio de base de datos

La máquina virtual encargada del servicio de base de datos tiene las siguientes características (ver Figura 14):

- CPU: 2 procesadores de 2 cores cada uno.
- Memoria: 8 GB.
- Almacenamiento: 40 GB.
- Adaptador de red: NAT.
- SO: Ubuntu Server 22.04 LTS (Abril 2022).

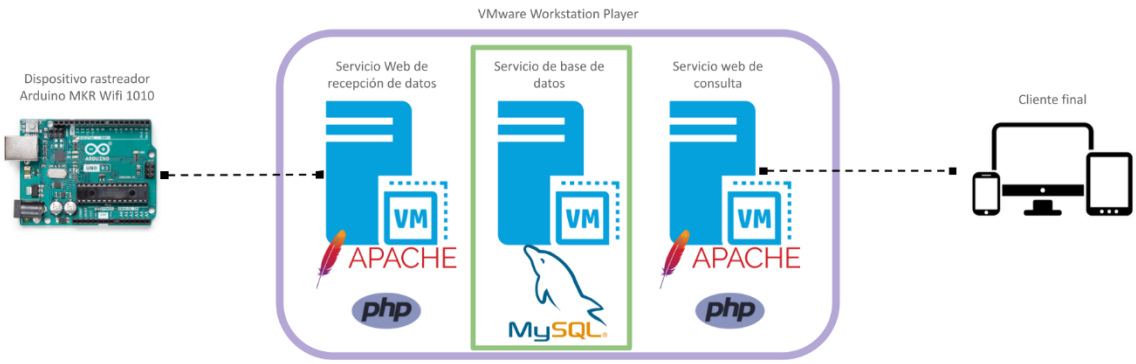


Figura 14. Servicio de base de datos.

Esta máquina virtual dispone de un único adaptador de red configurado en NAT. Su principal objetivo es tener conexión con la red interna del host y con todas las máquinas virtuales hospedadas en el mismo host, y así posibilitar el acceso a la base de datos desde cualquiera de las máquinas virtuales.

Al tratarse de una configuración NAT, también existe la posibilidad de tener acceso a internet, pero por razones de seguridad, el acceso al servidor de la base de datos se va a limitar exclusivamente para las dos máquinas virtuales que requieren acceso a la base de datos, haciendo dicho servicio inalcanzable para accesos externos. Para más detalles sobre la instalación y configuración de las máquinas virtuales consultar el *Apéndice II. Instalación y configuración de las máquinas virtuales.*

Esta máquina virtual tiene instalado Ubuntu Server como sistema operativo. Durante la instalación del SO se pueden instalar algunos módulos adicionales como OpenSSH, y una vez finalizada la instalación del SO se han de instalar los siguientes paquetes:

- MySQL [26].
- Apache HTTP Server.
- PHP.
- phpMyAdmin [32].

En esta máquina virtual, en la que se quiere disponer de un servidor de base de datos, también se va a instalar phpMyAdmin, una herramienta web que permite gestionar bases de datos de forma fácil y rápida, pero que nos obliga a tener un servidor web como Apache, y además tener PHP. Debido a esto, se ha visto conveniente instalar *LAMP stack* [7] que incluye exactamente lo necesario. La información de instalación de LAMP se encuentra más detallada en el *Apéndice II. Instalación y configuración de las máquinas virtuales.*

MySQL es un servidor de bases de datos que permite la gestión de grandes cantidades de datos de una forma eficiente y sólida. En este proyecto los registros que se van guardando son de pequeño tamaño, pero trabajamos con cantidades bastantes altas de registros, ya que se van rastreando direcciones bluetooth de forma constante en intervalos de unos 3 segundos, por ejemplo:

$n \rightarrow Número\ de\ direcciones\ rastreables.$
$t \rightarrow Tiempo\ que\ se\ rastrea\ activamente\ (s).$
$f \rightarrow Función\ que\ indica\ el\ número\ de\ registros\ almacenados.$
$$\boldsymbol{f(t) = n * \frac{t}{3}}$$

$Si\ tenemos\ 10\ direcciones\ rastreables, en\ una\ hora:$
$$f(3600) = 10 * \frac{3600}{3} = \ 12000\ registros\ almacenados.$$

Con esto en mente, es importante que la longitud de cada uno de los registros sea lo más pequeña posible, por este motivo los datos que se almacenan para cada dirección rastreada son:

- address
    - MAC: Dirección MAC Bluetooth que ha sido rastreada (varchar(20)).
    - Name: Nombre del dispositivo rastreado (varchar(20)).
    - board_id: Dirección MAC de la placa Arduino que envía la información (varchar(20)).
    - Timestamp: Fecha y hora en la que se rastreó la dirección (TIMESTAMP).

Para automatizar las copias de seguridad, y que estas se realicen de forma periódica, se emplea un script en Python [33] que lanza una aplicación propia de MySQL para crear copias de seguridad llamada, *"mysquldump"*. El script Python es lanzado a través de la aplicación Cron de Ubuntu, que permite ejecutar tareas bajo ciertas condiciones de periodicidad. A continuación, se muestra la entrada de la Crontab que se corresponde con la ejecución del script para crear las copias de seguridad de la base de datos:

```
0 2 * * 1,3,6 /usr/local/bin/python3 /home/scripts_p/ dbbackup.py
```

Esta entrada en la Crontab de Cron lanza el script "dbbackup.py" a las 2:00am cada lunes, miércoles y sábados, creando una copia de seguridad a través de mysqldump.

Para el cifrado de la información almacenada en la base de datos, la intención inicial era usar *Transparent Data Encryption* (TDE) [27], pero esta tecnología solo está disponible en la versión Enterprise de MySQL, que no es gratuita, por lo que para este proyecto quedó descartada. Pero, aun así, existe una tecnología proporcionada por MySQL [12] que permite cifrar a nivel de *tablespace* (ver Figura 15).

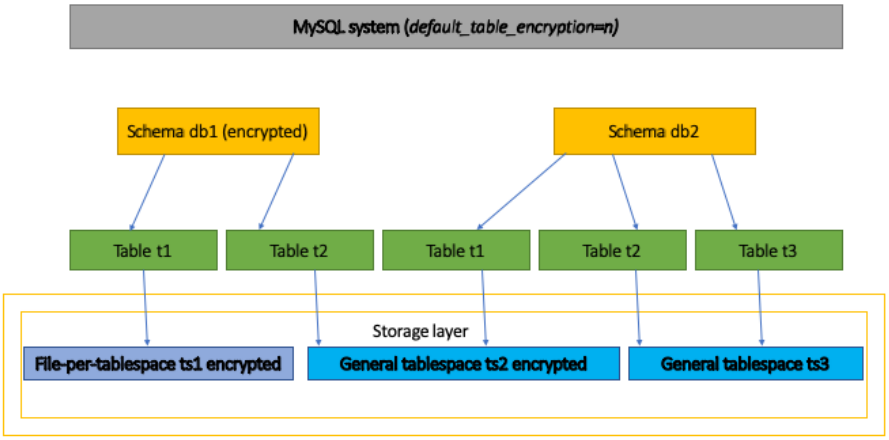


Figura 15. Cifrado de datos en MySQL [12].

Lo único que debemos hacer es cambiar el valor de un parámetro de configuración de la propia *"mysql> SET default_table_encryption='y';"*. En el caso de este proyecto, se está cifrando toda la base de dates, que dispone de dos tablas, por lo que no hay que añadir ningún otro tipo de parámetro de configuración para definir schemas u otros detalles.

Como se muestra en el estudio comparativo "*Performance impact analysis of enabling Transparent Data Encryption (TDE) on SQL Server*" [37] llevado a cabo por SQLShack, el rendimiento con TDE apenas se ve afectado para operaciones de consulta, inserciones o copias de seguridad, pero en las operaciones de actualización el rendimiento se ve seriamente afectado.

3.2.3 Servicio Web de consulta

La máquina virtual encargada del servicio web de consulta de información tiene las siguientes características (ver Figura 16):

- CPU: 2 procesadores de 2 cores cada uno.
- Memoria: 8 GB.
- Almacenamiento: 20 GB.
- Adaptador de red (NIC1): Bridged.
- Adaptador de red (NIC2): NAT
- SO: Ubuntu Server 22.04 LTS (Abril 2022)

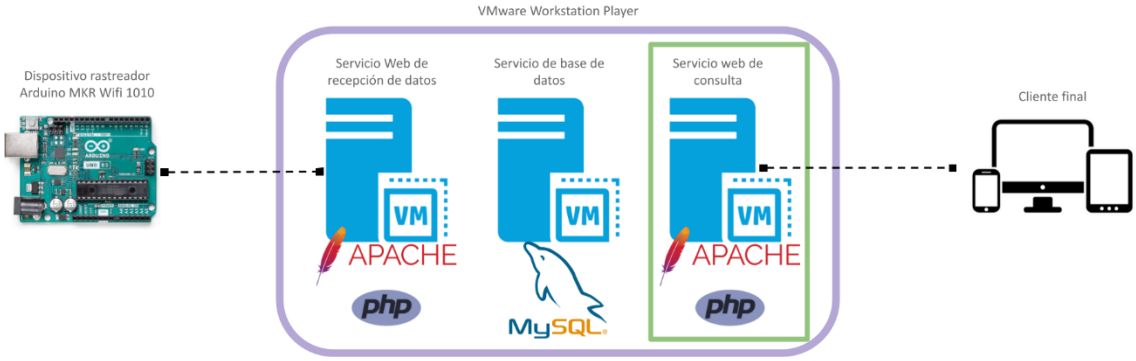


Figura 16. Servicio Web de consulta.

Esta máquina virtual es un clon de la máquina virtual encargada de la recepción de datos enviados por la placa Arduino, y como tal, la configuración de las diferentes interfaces de red es la misma que en el caso de la máquina virtual descrita en el apartado *3.2.1 Servicio Web de recepción de datos.*

Esta máquina virtual dispone del mismo SO que la máquina virtual de recepción de datos, así como de las aplicaciones básica, es decir;

- Apache HTTP Server.
- PHP.

El principal objetivo de Apache en esta máquina virtual es la de servir páginas web dinámicas creadas con PHP con información almacenada en la base de datos. Para esto se ha creado un sitio web formado por múltiples páginas web que muestran información con diferentes filtros. La configuración de Apache sigue las directrices descritas en el *Apéndice I. Configuración segura de Apache.*

En esta máquina virtual, las principales tareas desarrolladas a través de PHP son:

- Realizar consultas a la base de datos y obtener información.
- Procesar la información obtenida.
- Renderizar la información procesada a formato HTML.

Las principales páginas web dinámicas que proporciona el sitio web son dos. En la primera (ver Figura 17) podemos ver todo el contenido almacenado en la base de datos en un formato legible. Se muestra la entrada más reciente de cada una de las direcciones MAC rastreadas, así como la dirección de la placa Arduino responsable de la detección y la fecha y hora. El principal objetivo de esta página es tener acceso a todas las direcciones MAC bluetooth rastreadas sin tener la sobrecarga de ver repetidas dichas direcciones en cada instante de tiempo en el que fueron rastreadas.

MAC	NAME	TRACKER	LATEST ENTRY
D6:F3:88:76:80:35	nut3	80:7D:3A:87:EB:F4	2-12-2022 14:17:40
F5:CB:8B:40:F4:37	nut1	80:7D:3A:87:EB:F4	2-12-2022 14:17:40
00:13:EF:D2:56:FF	tile_s	80:7D:3A:87:EB:F4	28-11-2022 18:29:33
EF:2F:1C:61:25:A2	nut2	80:7D:3A:87:EB:F4	2-12-2022 14:17:40
84:47:6F:5C:A7:03	nut4	80:7D:3A:87:EB:F4	2-12-2022 14:17:40
18:17:14:54:D4:A2	emt_lc	80:7D:3A:87:EB:F4	30-11-2022 20:05:19

Figura 17. Primer filtro web de consulta.

La segunda página del sitio web (ver Figura 18) filtra las direcciones MAC bluetooth rastreadas en los últimos 5 segundos, mostrando las direcciones MAC y la fecha hora donde se detectó inicialmente. Con este filtro podemos ver las direcciones de los dispositivos que actualmente están activas y siendo capturadas por la placa Arduino.

MAC	NAME	TRACKER	CURRENT ENTRY
EF:2F:1C:61:25:A2	nut2	80:7D:3A:87:EB:F4	2-12-2022 14:57:03
84:47:6F:5C:A7:03	nut4	80:7D:3A:87:EB:F4	2-12-2022 14:57:03
D6:F3:88:76:80:35	nut3	80:7D:3A:87:EB:F4	2-12-2022 14:57:03
F5:CB:8B:40:F4:37	nut1	80:7D:3A:87:EB:F4	2-12-2022 14:57:03

Figura 18. Segundo filtro web de consulta.

El sitio web se encuentra bajo el dominio “tfgtrackapp.es”, el cual ha sido cedido de forma gratuita. El sitio web dispone de un certificado TSL, permitiendo la navegación web a través de HTTPS. Al disponer de un certificado TLS, toda la información que proporciona el sitio web va a ir cifrada de punto a punto, es decir, la información va a viajar de forma segura y privada entre el servidor web y cualquier cliente web, típicamente un navegador web.

4

Pruebas de Rendimiento y Resultados

En este capítulo se van a realizar pruebas al proceso de cifrado que se lleva a cabo en la placa Arduino con el objetivo de analizar cómo afecta el cifrado de información al rendimiento de la placa Arduino. En primer lugar, se van a describir los procedimientos y condiciones de las pruebas y en segundo lugar se expondrán los resultados comparativos de las pruebas realizadas.

4.1 Pruebas de rendimiento en Arduino

El objetivo de estas pruebas es determinar una métrica que proporcione información objetiva y permita evaluar el rendimiento del cifrado de información que se realiza en la placa Arduino.

La métrica que se ha utilizado es el tiempo transcurrido (tt) o *time elapsed* por un proceso, desde que este comienza, hasta que termina. Ya que el objetivo es determinar en qué medida afecta al rendimiento el proceso de cifrado, vamos a tomar el tiempo transcurrido de dos escenarios.

En el primer escenario se tomará el tiempo transcurrido desde que se detecta una dirección MAC bluetooth válida hasta que se envía con éxito, sin

cifrar ninguna información, es decir, enviando la información en texto plano a través de internet. En este primer escenario, los valores de tiempo transcurrido no se verán afectados por el proceso de cifrado.

En el segundo escenario se tomará el tiempo transcurrido desde que se detecta una dirección MAC bluetooth válida hasta que se envía con éxito, pero habiendo cifrado la información antes de ser enviada. En este segundo escenario, los valores de tiempo transcurrido se verán directamente afectados por el proceso de cifrado de información.

Para llevar a cabo la tarea de toma de tiempos, Arduino dispone de algunas funciones para obtener el tiempo en el momento de ser llamadas, más concretamente, la función "*micros()*" [19]. Está función devuelve el número de microsegundos transcurridos desde que la placa Arduino comenzó a ejecutar el programa actual. En la Figura 19 se muestra la estructura seguida para la toma de tiempos en cada uno de los escenarios estudiados.

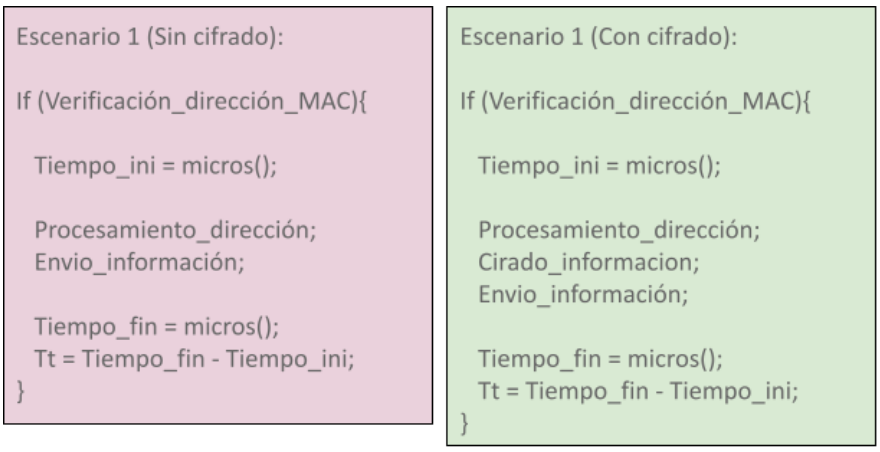


Figura 19. Estructura de las pruebas.

4.1.1 Resultados

Para disponer de unos resultados fiables y objetivos, las pruebas descritas en la sección anterior se realizaron varias veces, y en cada ocasión se tomó el valor del tiempo transcurrido como muestra.

Para determinar si la cantidad de muestra obtenidas son suficientes, se evaluó el error existente con la toma de cada muestra, hasta obtener un error muestral menor al 1% con una confianza del 95%. Cada uno de los dos escenarios analizados ha requerido de una cantidad de muestras diferentes, tal que;

Error menor al 1% *con una confianza del* 95%

$E1 \rightarrow Escenario\ 1\ (Sin\ cifrado)$
$E2 \rightarrow Escenario\ 2\ (Con\ cifrado)$

$N(E1) = 124\ muestras$
$N(E2) = 312\ muestras$

La cantidad de muestras necesarias para el escenario dos son más del doble que las necesarias para el escenario uno. El principal motivo de esto es que la desviación estándar de los valores muestrales del escenario dos es mucho mayor que la del escenario uno.

En la Figura 20 podemos ver en una gráfica cada uno de los 124 valores tomados para el escenario uno, donde el eje de abscisas representa cada una de las muestras, y el de ordenadas el tiempo transcurrido para cada muestra en microsegundos:

Figura 20. Valores muestrales del Escenario 1.

El valor medio del tiempo transcurrido en el escenario uno es de 0.0997 microsegundos.

En la Figura 21 se muestra la gráfica para el escenario dos:

Figura 21. Valores muestrales del Escenario 2.

En este segundo escenario, el valor medio del tiempo transcurrido es de 0.00659 milisegundos, es decir, 6.59 microsegundos.

Observando estos resultados, especialmente los valores medios para cada uno de los escenarios, donde en el escenario uno tenemos una media de tiempo transcurrido de 0.0997 microsegundos, y en el escenario dos 6.59 microsegundos, siendo el segundo caso 66 veces superior al primero, resulta más que evidente que el cifrado tiene una repercusión muy considerable en el rendimiento de la placa Arduino.

4.2 Pruebas de rendimiento en servidores web

El objetivo de estas pruebas es determinar el número de conexiones simultáneas que es capaz de gestionar tanto el servidor web de recepción de datos como el servidor web encargado de las consultas.

Para determinar la cantidad de conexiones que puede llegar a servir de forma correcta cada uno de los dos servidores web se ha tomado la decisión de usar la herramienta Apache JMeter [25].

La aplicación Apache JMeter es un software de código abierto, diseñado para probar el comportamiento funcional y medir el rendimiento ante diferentes situaciones de carga. Se diseñó originalmente para probar aplicaciones web, pero desde entonces se ha ampliado a otras funciones de prueba.

Las pruebas se han realizado tanto al servidor de consultas como al servidor de recepción de datos. Para determinar cuántas conexiones simultaneas es capaz de gestionar cada servidor, se ha creado un plan llamado “LoadTest”. El objetivo de esta prueba será lanzar unas 3000 peticiones. En el caso del servidor de consulta, como disponemos de 3 páginas, es decir, la página de inicio, o índex, la página con el primer filtro y la página con el segundo filtro se ha decidido que cada petición se envíe de forma aleatoria a alguna de las tres. En el caso del servidor de recepción de datos, únicamente se dispone de un recurso, así que todas las peticiones llegan al mismo punto.

En la Figura 22 se muestra la pantalla de configuración de pruebas de Apache JMeter para el servidor web de consultas, donde se van a lanzar 3000 hebras a alguna de las tres páginas disponibles de forma aleatoria.

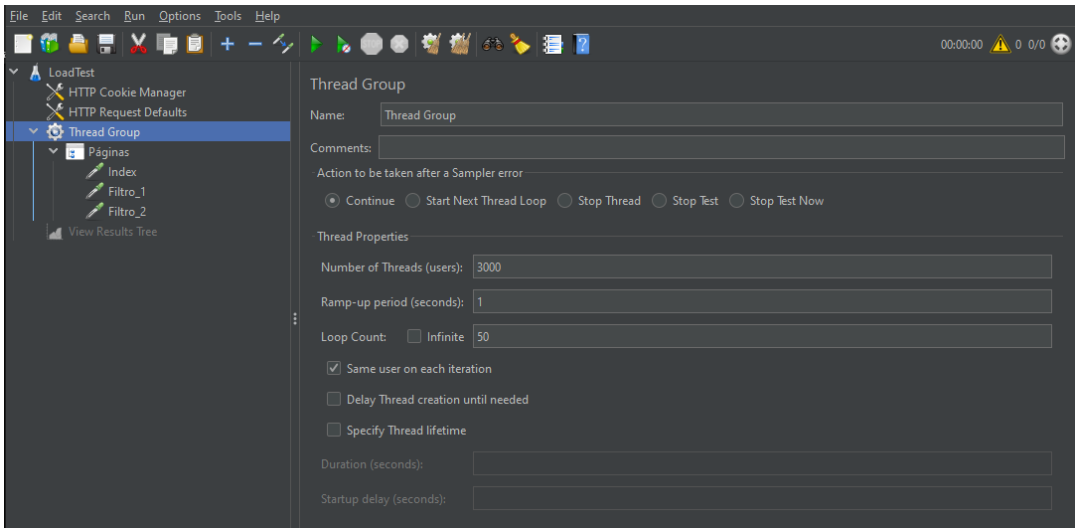


Figura 22. Prueba de carga al servidor de consultas con Apache JMeter.

Hay que tener en cuenta que aún que estas pruebas se han desarrollado en local, las peticiones son igualmente tratadas por el servidor web encargado de servir cada una de las peticiones "Get", por lo que los resultados obtenidos son tan válidos como si la prueba se hiciera en remoto al dominio público de la aplicación en lugar de ir directos a la máquina en concreto donde se encuentran los recursos, y al ser en local, las peticiones se lanzan mucho más rápido.

4.2.1 Resultados

JMeter proporciona una forma fácil y rápida de ver cada una de las peticiones y su resultado. En la Figura 23 podemos observar varias peticiones aleatorias que han tenido éxito tanto en la petición como en la respuesta.

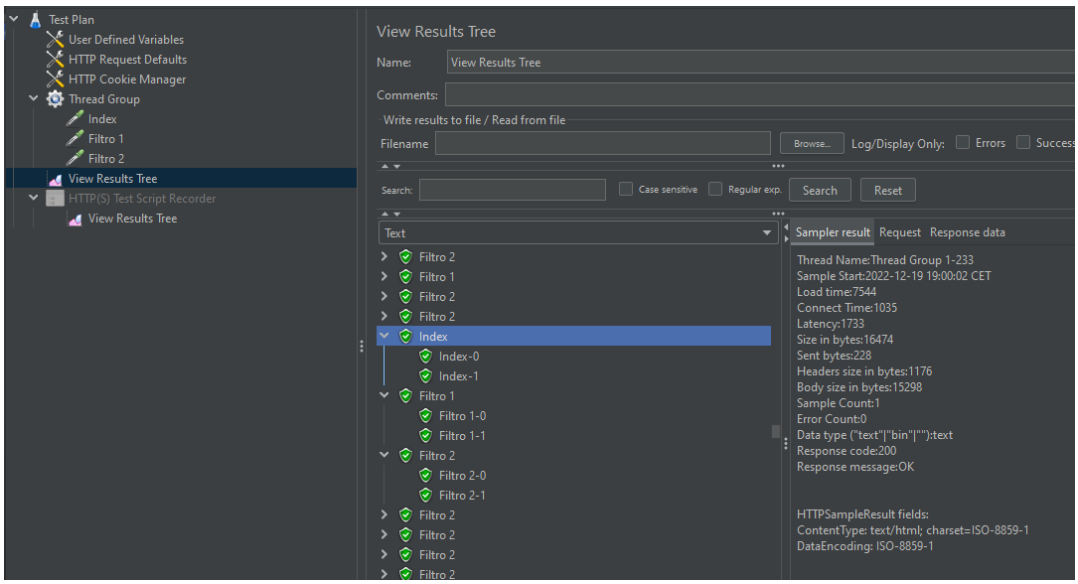


Figura 23. Resultados exitosos de Apache JMeter.

En el caso de suceder algún error, JMeter también nos los indica y nos proporciona toda la información referente al propio error como se muestra en la Figura 24.

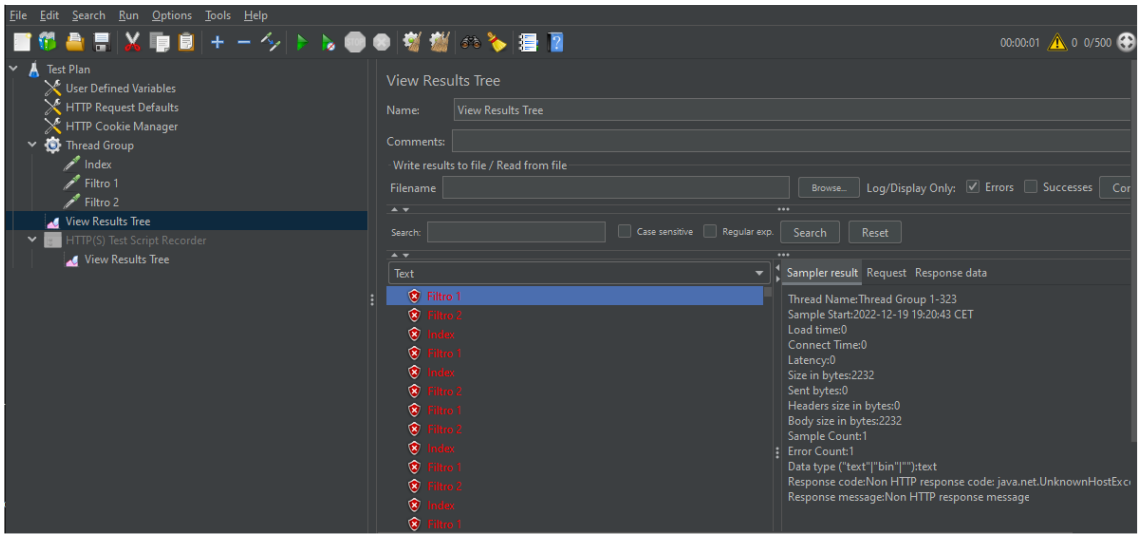


Figura 24. Resultados fallidos de Apache JMeter.

Los resultados obtenidos tanto para el servidor de consulta como para el servidor de recepción de datos han sido los mismos, es decir, el porcentaje de peticiones exitosas ha resultado ser el mismo. El 36.7% de las 3000 peticiones, es decir, unas 1100, han sido resuelta satisfactoriamente por el servidor, mientras que las restantes 1900 han sido rechazadas por el servidor.

Estos valores se han repetido con una diferencia de menos del 2% en cada simulación realizada para el servidor de recepción de datos, es decir, una diferencia de menos de 60 peticiones de desviación en cada simulación de 3000 peticiones. Sin embargo, en el caso del servidor de consultas la variación ha sido algo mayor, entorno al 5%, es decir, una variación de menos de 150 peticiones en cada simulación realizada.

Se ha probado a lanzar más peticione, por ejemplo 4000 o 5000, pero en estos casos, la simulación llegó a producir un error en el propio JMeter, y no era capaz de recoger correctamente los datos resultantes ni podía lanzar todas las hebras encargadas de las peticiones.

Las conexiones rechazadas se deben principalmente a las medidas establecidas en la configuración de Apache HTTP Server, ya que muchas conexiones son directamente rechazadas para evitar la que el servicio se detenga por sobrecarga si no hay suficientes recursos para atender más peticiones, es decir, se aplican medidas para evitar ataques DoS (*Denial of Service*).

5

Conclusiones

La seguridad en el mundo IT está cada vez más presente y es base esencial para cualquier desarrollo que se pretenda poner en producción con un mínimo de garantías. Cualquier organización que ponga un servicio a disposición del público o de uso interno, ha de tener en cuenta una variedad de características de seguridad que actualmente están alineadas con la mayoría de las certificaciones de seguridad.

5.1 Conclusiones generales

El principal objetivo de este proyecto era añadir seguridad, escalabilidad y fiabilidad al proyecto original, acercándolo a una versión más cercana a las exigidas por las aplicaciones profesionales en producción actualmente.

Los objetivos de seguridad alcanzados son;

- Añadir cifrado para cualquier información que fuera transmitida por medios no seguros como internet, lo cual se ha conseguido cifrando la información de la placa Arduino hasta el servidor de recepción, y a través de HTTPS en el servidor de consultas.
- Añadir cifrado de los datos en reposo, lo cual se ha conseguido gracias al cifrado de MySQL.
- Añadir un segundo factor de autenticación para el control de acceso de la gestión remota, y esto lo resuelve el segundo factor de autenticación añadido a los servidores OpenSSH de cada máquina virtual y exigido para cualquier aplicación cliente.

Los objetivos de escalabilidad alcanzados son;

- Emplear máquinas virtuales para la gestión de toda la información suministrada por la estación rastreadora, brindando así la posibilidad de poder escalar la estructura virtual tanto a nivel vertical como a nivel horizontal. Gracias al uso de VMware Workstation Player, todos los servicios están virtualizados con éxito.
- Si bien es cierto que este proyecto no contemplaba la agregación de orquestadores para servicios distribuidos, la estructura virtual en la que se han dispuesto todos los servicios facilitaría mucho el escalado a nivel organizativo.

Los objetivos de fiabilidad alcanzados son;

- Distribuir los servicios en diferentes máquinas virtuales, de forma que, si una máquina falla, provoque el menor impacto en el funcionamiento del todo el sistema. Esto se ha cumplido al haber creado tres máquinas virtuales con diferentes servicios, aún que es necesario crear y configurar réplicas de dichas máquinas virtuales para que realmente exista una verdadera tolerancia a fallos.
- La base de datos debe de automatizar la creación de copias de seguridad, lo cual se ha cumplido gracias a un script en Python y al administrador de tareas de Linux, Cron.

La metodología de trabajo que se propuso al inicio del proyecto se ha ido cumpliendo de acuerdo con lo establecido, con la salvedad de que el proceso de investigación ha sido constante y ha estado presente durante todo el proyecto, es decir, desde el comienzo del proceso de implementación del primer sistema, hasta el final de la implementación del último sistema, cada día se ha tenido que investigar y aprender algo nuevo o pulir detalles. La única excepción ha sido durante el desarrollo de la documentación, donde las exigencias de investigación son menores.

5.2 Conclusiones personales

Este proyecto ha requerido de esfuerzo, aprendizaje, paciencia y disciplina. He aprendido a trabajar con tecnologías que eran totalmente desconocidas para mí, nunca las había visto ni en la carrera ni en mi vida profesional. Ya que este proyecto abarca una amplia variedad de tecnologías y opciones considerables, el hecho de encontrarme con algo nuevo ha sido parte del día a día.

Mi pasión por la seguridad informática y mis ganas de aprender han sido vitales para el desarrollo del proyecto. Gran parte del trabajo que ha requerido el

proyecto ha estado dentro de las áreas de conocimiento que más me gustan, lo cual me ha permitido lidiar con las dificultades y los incesantes errores que han ido surgiendo por el camino.

5.3 Trabajo futuro

Al finalizar el desarrollo del proyecto es cuando se pueden apreciar los diferentes aspectos de mejora, especialmente surge la posibilidad de ampliar diversas funcionalidades como:

- Empaquetar los servicios en aplicaciones como Docker para poder realizar despliegues distribuidos y que sean orquestables.
- Añadir filtros o funcionalidades extra a la aplicación web de consultas.
- Agregar un servicio de usuarios para la gestión de las consultas.
- Desarrollar una aplicación Android o iOS que conecte con la aplicación web de consultas o bien incruste la aplicación web en la propia APP.
- Cambiar la placa Arduino por una Raspberry Pi, la cual permite el tráfico HTTPS, almacenando los certificados de confianza en su Sistema Operativo Raspbian.

Desde el comienzo del proyecto, he intentado que todas las tecnologías usadas permitan la posibilidad de añadir mejoras al proyecto, evitando usar tecnologías obsoletas o en decadencia, con el objetivo de facilitar futuras mejoras.

6

Referencias

[1] "AESLib project repository", github.com. https://github.com/suculent/thinx-aes-lib (Accedido: 20-dic-2022).

[2] "Apache HTTP Server Project Multi-Processing Modules (MPMs)", httpd.apache.org. https://httpd.apache.org/docs/2.4/mpm.html (Accedido: 20-dic-2022).

[3] "Apache HTTP Server Project Security Tips", httpd.apache.org. https://httpd.apache.org/docs/2.4/misc/security_tips.html (Accedido: 20-dic-2022).

[4] "Arduino MKR WiFi 1010", docs.arduino.cc. https://docs.arduino.cc/hardware/mkr-wifi-1010 (Accedido: 20-dic-2022).

[5] "Arduino Official Website", arduino.cc. https://www.arduino.cc/ (Accedido: 20-dic-2022).

[6] "ATECC508A CryptoAuthentication Device Complete Data Sheet", docs.arduino.cc. https://docs.arduino.cc/resources/datasheets/ATECC508A-datasheet.pdf (Accedido: 20-dic-2022).

[7] "AWS What Is A Lamp Stack?", aws.amazon.com. https://aws.amazon.com/what-is/lamp-stack/ (Accedido: 20-dic-2022).

[8] "Bluetooth Core Specification Version 5.2 Feature Overview",

bluetooth.com. https://www.bluetooth.com/wp-content/uploads/2020/01/Bluetooth_5.2_Feature_Overview.pdf (Accedido: 20-dic-2022).

[9] "Bluetooth LE security and privacy in wireless audio devices", cardinalpeak.com. https://www.cardinalpeak.com/blog/bluetooth-le-security-and-privacy-in-wireless-audio-devices (Accedido: 20-dic-2022).

[10] "Bridged vs NAT connection", communities.vmware.com. https://communities.vmware.com/t5/VMware-Workstation-Pro/Bridged-vs-NAT-connection/m-p/2290902#M137141 (Accedido: 20-dic-2022).

[11] "Configure SSH to use two-factor authentication", ubuntu.com. https://ubuntu.com/tutorials/configure-ssh-2fa#1-overview (Accedido: 20-dic-2022).

[12] "Controlling table encryption in MySQL 8.0", dev.mysql.com. https://dev.mysql.com/blog-archive/controlling-table-encryption-in-mysql-8-0/ (Accedido: 20-dic-2022).

[13] "Core Specification 4.0", bluetooth.com. https://www.bluetooth.com/specifications/specs/core-specification-4-0/ (Accedido: 20-dic-2022).

[14] "CPACanada Official Website", cpacananda.ca. https://www.cpacanada.ca/ (Accedido: 20-dic-2022).

[15] "Critical Factors to consider when virtualizing Business Critical Applications: (Part 1 of 2)", blogs.vmware.com. https://blogs.vmware.com/apps/2014/02/virtualize-business-critical-applications-vsphere-updated.html (Accedido: 20-dic-2022).

[16] "Esquema Nacional de Seguridad, ENS", ccn.cni.es. https://ens.ccn.cni.es/es/esquema-nacional-de-seguridad-ens (Accedido: 20-dic-2022).

[17] "Estimación del número de defunciones semanales. Últimos datos", ine.es. https://www.ine.es/dyngs/INEbase/es/operacion.htm?c=Estadistica_C&cid=1254736177074&menu=ultiDatos&idp=1254735573175 (Accedido: 20-dic-2022).

[18] "Evolución pandemia". cnecovid.iscii.es.

https://cnecovid.isciii.es/covid19/#evoluci%C3%B3n-pandemia (Accedido: 20-dic-2022).

[19] "Functions: micros()", arduino.cc. https://www.arduino.cc/reference/en/language/functions/time/micros/ (Accedido: 20-dic-2022).

[20] "HTTPS", es.ryte.com. https://es.ryte.com/wiki/HTTPS (Accedido: 20-dic-2022).

[21] "HTTPS", seobility.net. https://www.seobility.net/en/wiki/HTTPS (Accedido: 20-dic-2022).

[22] I. Teruel Parra, "Desarrollo de una solución de bajo coste para el rastreo de casos positivos COVID-19 en empresas y organismos públicos", Trabajo fin de grado, Departamento de Ingeniería de Sistemas y Automática, UMA, Málaga, España, 2021.

[23] "ISACA Official Website", isaca.org. https://www.isaca.org/ (Accedido: 20-dic-2022).

[24] "ISO/IEC 27001 and related standards Information security management", iso.org. https://www.iso.org/isoiec-27001-information-security.html (Accedido: 20-dic-2022).

[25] "Apache JMeter", jmeter.apache.org. https://jmeter.apache.org/ (Accedido: 20-dic-2022).

[26] "MySQL Community Server 8.0.31", dev.mysql.com. https://dev.mysql.com/downloads/mysql/ (Accedido: 20-dic-2022).

[27] "MySQL Enterprise Transparent Data Encryption (TDE)", mysql.com. https://www.mysql.com/products/enterprise/tde.html (Accedido: 20-dic-2022).

[28] "NINA-W10 series", u-blox.com https://www.u-blox.com/en/product/nina-w10-series-open-cpu (Accedido: 20-dic-2022).

[29] "Open Systems Interconnection (OSI) Reference Model", docs.oracle.com. https://docs.oracle.com/cd/E19504-01/802-5886/intro-45828/index.html (Accedido: 20-dic-2022).

[30] “OpenSSH Offcial Website”, openssh.com. https://www.openssh.com/ (Accedido: 20-dic-2022).

[31] “PHP Official Website”, php.net. https://www.php.net/ (Accedido: 20-dic-2022).

[32] “phpMyAdmin Official Website”, phpmyadmin.net. https://www.phpmyadmin.net/ (Accedido: 20-dic-2022).

[33] “Python script for taking mysqldump”, gist.github.com. https://gist.github.com/trafficinc/a0b7cfa58a00e794bf8c7dd0626c2765 (Accedido: 20-dic-2022).

[34] “Rapid Application Development (RAD) Model: An Ultimate Guide For App Developers in 2022”, kissflow.com. https://kissflow.com/application-development/rad/rapid-application-development/ (Accedido: 20-dic-2022).

[35] “Recomendaciones de Seguridad para Autenticación Multi-Factor”, ccn.cni.es. https://www.ccn.cni.es/index.php/es/docman/documentos-publicos/boletines-pytec/353-pildorapytec-oct2020-autenticacion-multifactor/file (Accedido: 20-dic-2022).

[36] “Software Arduino IDE 2.0.3”, arduino.cc. https://www.arduino.cc/en/software (Accedido: 20-dic-2022).

[37] “SQLShack Performance impact analysis of enabling Transparent Data Encryption (TDE) on SQL Server”, sqlshack.com. https://www.sqlshack.com/performance-impact-analysis-of-enabling-transparent-data-encryption-tde-on-sql-server/ (Accedido: 20-dic-2022).

[38] “SSH Academy”, ssh.com. https://www.ssh.com/academy/ssh/protocol (Accedido: 20-dic-2022).

[39] “The Apache Software Foundation”, projects.apache.org. https://projects.apache.org/project.html?httpd-http_server (Accedido: 20-dic-2022).

[40] “The Transport Layer Security (TLS) Protocol Version 1.3”, rfc-editor.org. https://www.rfc-editor.org/rfc/rfc8446 (Accedido: 20-dic-2022).

[41] “TigerVNC Offcial Website”, tigervnc.org. https://tigervnc.org/ (Accedido: 20-dic-2022).

[42] "TISAX Trusted Information Security Assessment Exchange", enx.com, https://enx.com/en-US/TISAX/ (Accedido: 20-dic-2022).

[43] "Understanding the Remote Desktop Protocol (RDP)", learn.microsoft.com. https://learn.microsoft.com/en-us/troubleshoot/windows-server/remote/understanding-remote-desktop-protocol (Accedido: 20-dic-2022).

[44] "Vinagre project repository", gitlab.gnome.org. https://gitlab.gnome.org/Archive/vinagre (Accedido: 20-dic-2022).

[45] "Virtual Network Computing from ORL", web.mit.edu. http://web.mit.edu/cdsdev/src/docs.html (Accedido: 20-dic-2022).

[46] "Webtrust Principles and Criteria", cpacanada.cn. https://www.cpacanada.ca/en/business-and-accounting-resources/audit-and-assurance/overview-of-webtrust-services/principles-and-criteria (Accedido: 20-dic-2022).

[47] "Welcome to Raspbian", raspbian.org. https://www.raspbian.org/ (Accedido: 20-dic-2022).

Apéndice I. Configuración segura de Apache

Si bien es cierto que Apache HTTP Server ha ido mejorado las condiciones de seguridad en su configuración por defecto, es decir, la configuración al ser instalado sin modificaciones, siguen existiendo algunos detalles que se pueden modificar para aportar un mayor nivel de seguridad contra amenazas tanto externas como internas. El objetivo de esta sección es describir los cambios realizados para este proyecto.

I. Control de permisos.

Todo el entorno de ejecución de Apache, es decir, el *ServerRoot*, debe de estar protegido contra escrituras no autorizadas, y para esto, es conveniente dar permisos de escritura únicamente al root para todos los directorios y ficheros de este *ServerRoot*, para ello, debemos crear los directorios de la siguiente menera:

1. Creamos los directorios que van a ir dentro del directorio raiz de Apache:

```
mkdir bin conf logs
```

2. Asignamos el usuario root como propietario

```
chown root:root . bin conf logs
```

3. Asignar el grupo root como grupo propietario

```
chgrp root:root . bin conf logs
```

Finalmente asignamos permisos de lectura, escritura y ejecución a cada tipo de usuario, es decir, propietario, grupo y otros. Todos los permisos para el propietario, y permisos de lectura y ejecución para grupos y otros.

```
chmod 755 . bin conf logs
```

Si se instala httpd dentro del directorio *bin*, también debe de tener las mismas condiciones que todo el *ServerRoot*, pero los permisos no son los mismos, ya que, al propietario le asignamos permisos de lectura y ejecución, y tanto al grupo como a otros únicamente les asignamos permisos de ejecución;

```
chmod 511 /usr/local/apache/bin/httpd
```

II. Deshabilitar los scripts CGI.

Common Gateway Interface (CGI) es un método por el cual un servidor web puede interactuar con programas externos de generación de contenido. Estos scripts CGI son una reconocida fuente de ataques a servidores Apache, y ya que este proyecto no necesita del modulo CGI de Apache HTTP Server, es conveniente deshabilitarlo. A través del siguiente comando, el módulo CGI queda deshabilitado después de reiniciar el servicio de Apache:

```
a2dismod cgi
sudo service apache2 restart
```

Hay que tener en cuenta que el servidor Apache de este proyecto lanza multiples scripts PHP, y por defecto lo hace a través del modulo CGI, por lo que al dehabilitar dicho módulo, hay que habilitar la ejecución de scripts PHP. La última versión de PHP es la 8.2, publicada el 8 de diciembre de 2022. Para que Apache lance ficheros PHP.

1. Durante la instalación de PHP:

```
sudo apt install php8.2 libapache2-mod-php8.2
```

2. Después de la instalación de PHP:

```
sudo a2enmod php
sudo service apache2 restart
```

III. Limitar los directorios accesibles.

El servidor Apache puede acceder a multiples directorios y ficheros en busqueda de algún recurso que se le solicite a través de las reglas de mapeo de direcciones URL, y esto permitiría a los usuarios navegar a través de directorios que no deberían ser accesibles. Para cambiar esto, hay que añadir el siguiente parametro en el fichero de configuración de apache *apache2.conf*.

```
<Directory "/">
      # Se deniegan TODAS las opciones
      Require all denied
</Directory>
```

Con esto se prohibe el acceso a todo el sistema de ficheros, por lo que ahora hay que indicar a que directorios si que puede acceder Apache.

```
<Directory "/usr/users/*/safedirs">
    # Se habilitan las necesarias
    Require all granted
</Directory>
<Directory "/usr/local/httpd">
    # Se habilitan las necesarias
    Require all granted
</Directory>
```

IV. Protección contra denegación de servicio (DoS).

Hay ciertas modificaciones que se pueden tener en consideración para ayudar a mitigar el problema de los ataques de denegación de servicio o ataques *DoS*.

- El método *RequestReadTimeout* limita el tiempo máximo para ciertas operaciones durante el establecimiento de las conexiones:

```
RequestReadTimeout handshake=8 header=14
```

- El método *TimeOut* debe ser reducido, su valor por defecto es 60 segundos, y es conveniente que sea de unos 10 o 15 segundos en los casos donde el módulo CGI está deshabilitado.

```
TimeOut 15
```

- El método *KeepAliveTimeout* tiene el valor 5 por defecto, y es mejor que sea un valor algo mas bajo, como 3.

```
KeepAliveTimeout 3
```

- Los siguientes métodos deben ser evaluados y modificados de acuerdo a las necesidades del servicio:

```
LimitRequestBody 100000              # Límite de 100KB
LimitRequestFields 80                # Límite de 80 campos
LimitRequestFieldSize    6000        # Límite de 6KB
LimitRequestLine 6000                # Límite de 6KB
LimitXMLRequestBody 1000000          # Valor por defecto
```

- El método *AcceptFilter* en Ubuntu cubre las necesidades y no es necesario modificarlo:

```
AcceptFilter http data
AcceptFilter https data
```

- El método *MaxRequestWorkers* permite indicar el máximo número de conexiones simultaneas, y el método *ListenBacklog* permite limitar el número de conexiones que quedarán en cola a la espera:

```
MaxRequestWorkers 1000   # Número máximo de conexiones
ListenBacklog 512        # Número máximo en cola
```

Apéndice II. Instalación y configuración de las máquinas virtuales

En este apéndice se describe la instalación y configuración de los servicios necesarios en cada una de las máquinas virtuales. Todas las máquinas virtuales han sido creadas y gestionadas con el software de virtualización *VMware Workstation Player v17*.

I. Añadir una segunda interfaz de red:

Dos de las máquinas virtuales de este proyecto requieren de dos interfaces de red, una configurada como NAT y otra como Bridged. A continuación se describe el proceso de agregación de una nueva interfaz de red:

1. Apagar la máquina virtual.
2. Ir a **Settings** de la máquina virtual.
3. Pulsar **Add** y elegir **Network Adapter** (ver Figura 25).

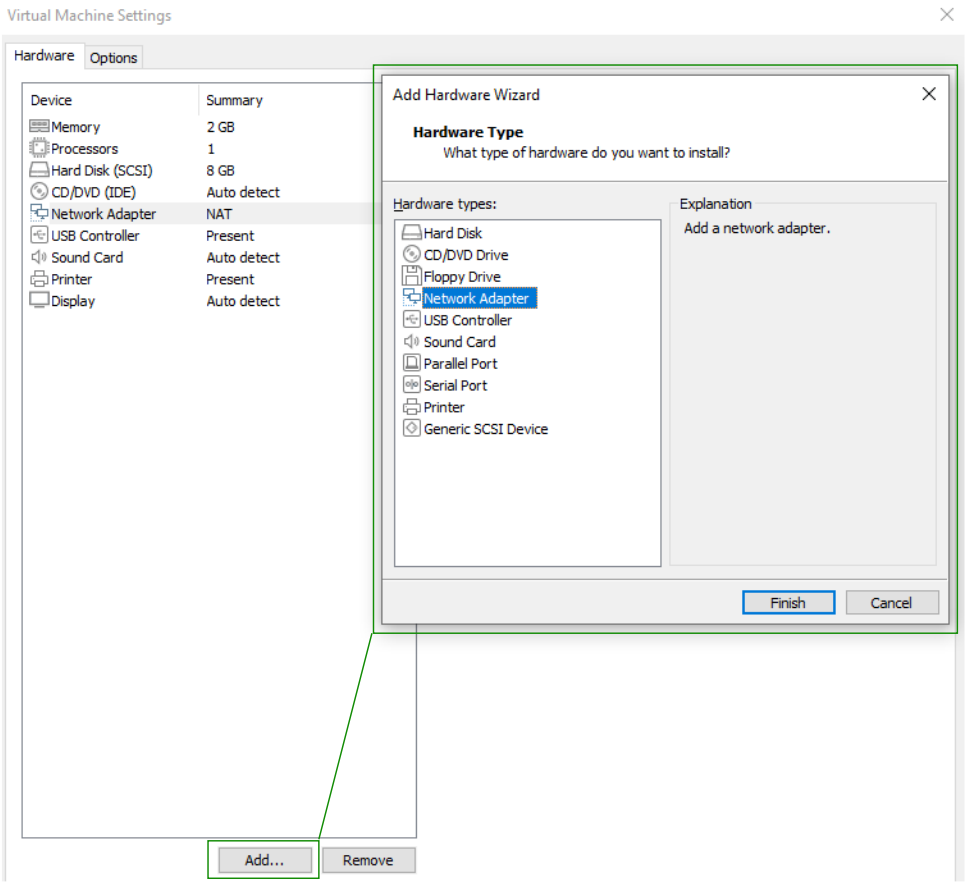


Figura 25. Agregación de una nueva interfaz de red.

4. Configuramos la nueva interfaz de red como **Bridged** y marcamos la opción de replicar el estado de la conexión física (ver figura 26).

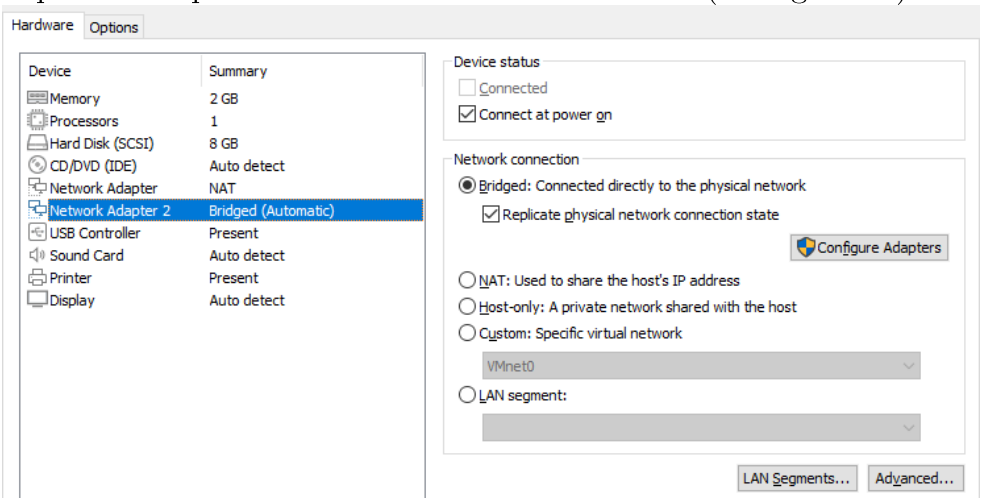


Figura 26. Configuración de la segunda interfaz de red.

II. Direcciones estáticas

Las tres máquinas virtuales necesitan tener una dirección IP estática en sus correspondientes interfaces de red NAT para permitir el acceso entre las propias máquinas virtuales. A continuación se describen los pasos a seguir;

1. Identificarse como root:

```
sudo -i
```

2. Editar el fichero */etc/network/interfaces*, para cada una de las máquinas. (Las direcciones IP deben ser diferentes, en mi caso he usado 10.0.0.15/16/17 respectivamente).

```
# Interfaz NAT, no la Bridged
allow-hotplug eth0
iface eth0 inet static
address 10.0.0.15
netmask 255.255.255.0
```

3. Editar el fichero /etc/hosts y añadir una linea para la resolución del nombre *server_data*, *server_db* y *server_query*, dependiendo de la máquina en la que estemos:

```
127.0..0.1          localhost
10.0.0.15           server_data
```

```
127.0..0.1          localhost
10.0.0.16           server_db
```

```
127.0..0.1          localhost
10.0.0.17           server_query
```

4. Ejecutar

```
update-rc.d exim4 disable
```

5. Reiniciar la máquina virtual.

III. Apache HTTP Server y PHP

Hay dos servicios que las tres máquinas virtuales necesitan tener instalados, Apache HTTP Server y PHP. Para instalar ambos, y adicionalmente instalar el módulo de gestión de PHP para Apache (Ya que no vamos a usar CGI como parser de contenido dinámico) debemos de seguir estos pasos:

1. Actualizar Apt caché:

```
sudo apt update
```

2. Instalar Apache:

```
sudo apt install apache2
```

3. Dar permisos al servicio Apache en el firewall de Ubuntu, *Uncomplicated Firewall* (UFW):

```
sudo ufw allow in "Apache"
```

4. Instalar PHP y todos los paquetes extra necesarios:

```
sudo apt install php libapache2-mod-php
```

5. Por defecto, Apache servirá el documento *index.html* antes del *index.php*, para evitar esto, y que *index.php* tenga mayor preferencia hay que editar el fichero */etc/apache2/mods-enabled/dir.conf*, y colocar *index.php* antes de *index.html*.

```
<IfModule mod_dir.c>
        # Se listan en el orden de preferencia
        DirectoryIndex index.php index.html
</IfModule>
```

6. Reiniciar el servicio Apache:

```
sudo apache2ctl restart
```

7. Podemos probar el correcto funcionamiento de PHP. Vamos al directorio */var/www/html/info.php* (Ya que cada máquina va a servir como mucho un sitio web, no tiene mucho sentido crear y habilitar otros *virtual hosts* aparte del que viene por defecto). Ahora creamos un fichero y lo editamos con el siguiente contenido:

```
nano /var/www/html/info.php
```

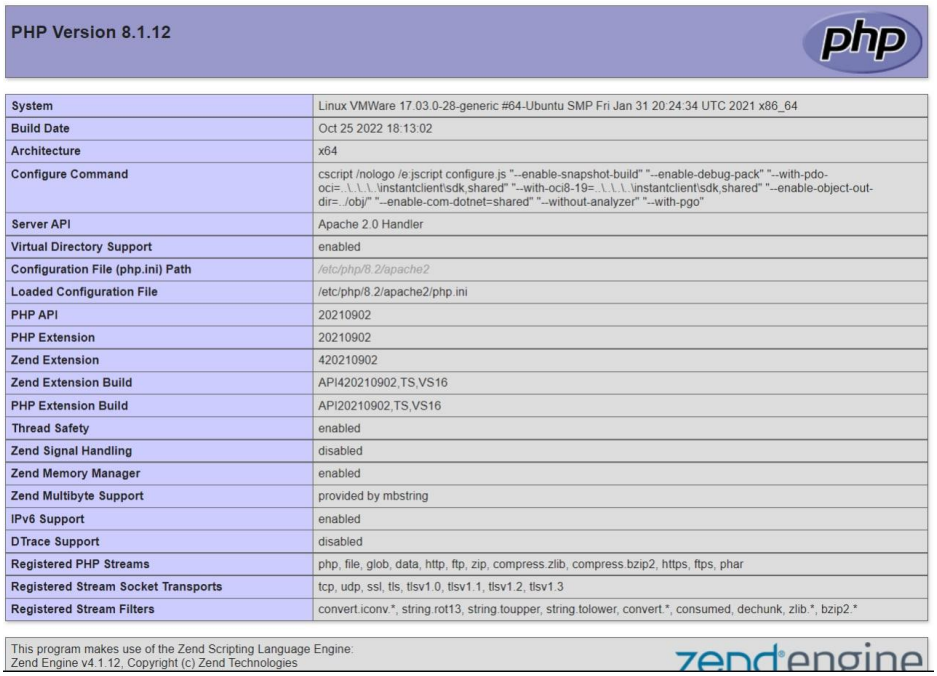
```
<?php

// Show all information
phpinfo ();

?>
```

8. Finalmente, a través de un navegador web, poniendo la siguiente dirección, obtendremos algo parecido a la Figura 27:

```
http://10.0.0.17/info.php
```



PHP Version 8.1.12

System	Linux VMWare 17.03.0-28-generic #64-Ubuntu SMP Fri Jan 31 20:24:34 UTC 2021 x86_64
Build Date	Oct 25 2022 18:13:02
Architecture	x64
Configure Command	cscript /nologo /e:jscript configure.js "--enable-snapshot-build" "--enable-debug-pack" "--with-pdo-oci=..\..\..\instantclient\sdk,shared" "--with-oci8-19=..\..\..\instantclient\sdk,shared" "--enable-object-out-dir=../obj/" "--enable-com-dotnet=shared" "--without-analyzer" "--with-pgo"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	/etc/php/8.2/apache2
Loaded Configuration File	/etc/php/8.2/apache2/php.ini
PHP API	20210902
PHP Extension	20210902
Zend Extension	420210902
Zend Extension Build	API420210902,TS,VS16
PHP Extension Build	API20210902,TS,VS16
Thread Safety	enabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	disabled
Registered PHP Streams	php, file, glob, data, http, ftp, zip, compress.zlib, compress.bzip2, https, ftps, phar
Registered Stream Socket Transports	tcp, udp, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2, tlsv1.3
Registered Stream Filters	convert.iconv.*, string.rot13, string.toupper, string.tolower, convert.*, consumed, dechunk, zlib.*, bzip2.*

This program makes use of the Zend Scripting Language Engine:
Zend Engine v4.1.12, Copyright (c) Zend Technologies

Figura 27. Resultados de phpinfo().

IV. MySQL, phpMyAdmin y Python

La máquina virtual encargada de dar servicio de base de datos, *server_db*, aparte de Apache y PHP, requiere la instalación de MySQL, phpMyAdmin y Python.

MySQL es un gestor de bases de datos, que además dispone de un servidor (MySQL Server), y es gracias a este servidor que podemos envíar y recibir peticiones de otras máquinas.

PhpMyAdmin es una herramienta web que permite una gestión rápida y cómoda de la base de datos, así como de sus usuarios.

Python nos facilita la creación de un script que realiza copias de seguridad de la base de datos a través de una herramienta propia de MySQL llamada *mysqldump* y podemos lanzar el script de forma automática y regular a través de la aplicación Cron de Linux.

1. En primer lugar hay que instalar MySQL, para lo que usaremos de nuevo *Apt*:

```
sudo apt update
sudo apt install mysql-server
```

2. Ahora vamos a instalar phpMyAdmin, a través de *Apt*:

```
sudo apt install phpmyadmin
```

3. Reiniciamos el servicio de Apache para cargar todos los componentes de phpMyAdmin:

```
sudo service apache2 restart
```

4. Hay que instalar Python, ya que el script encargado de las copias de seguridad está escrito en Python:

```
sudo apt install python3
```

5. Ya que tenemos instalado phpMyAdmin, podemos acceder y crear la base de datos que contendrá toda la información del proyecto. Para acceder a phpMyAdmin, necesitamos un navegador web, y acceder a la dirección:

```
http://10.0.0.16/phpmyadmin
```

6. Ahora podemos crear la base de datos tfg_trackapp, y sus dos tablas principales, *address* y *trackers*. El resultado se muestra en las Figuras 28, 29 y 30

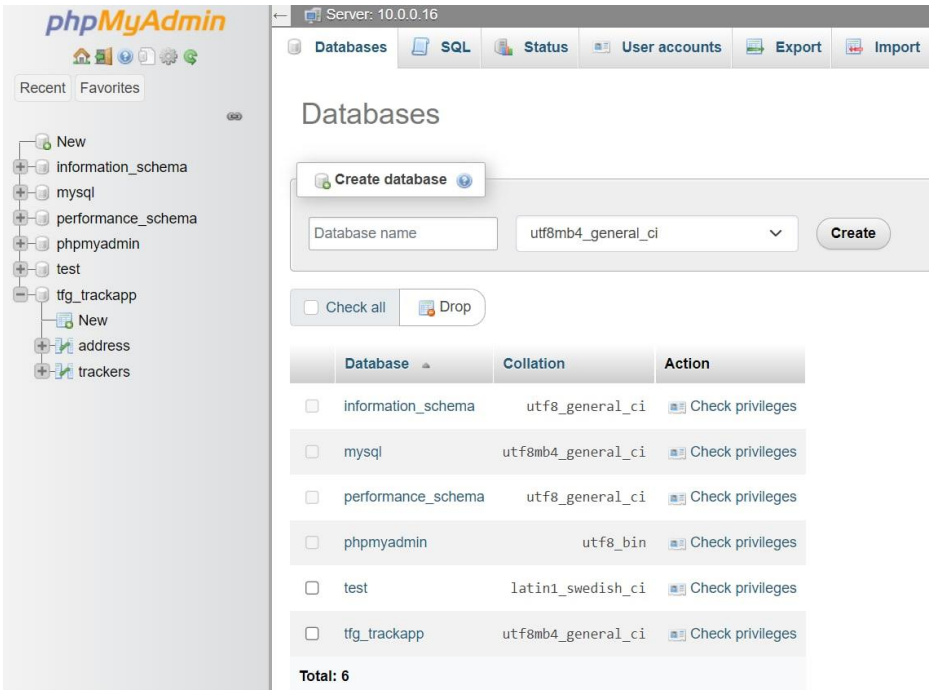


Figura 28. Base de datos *tfg_trackapp* a través de phpMyAdmin.

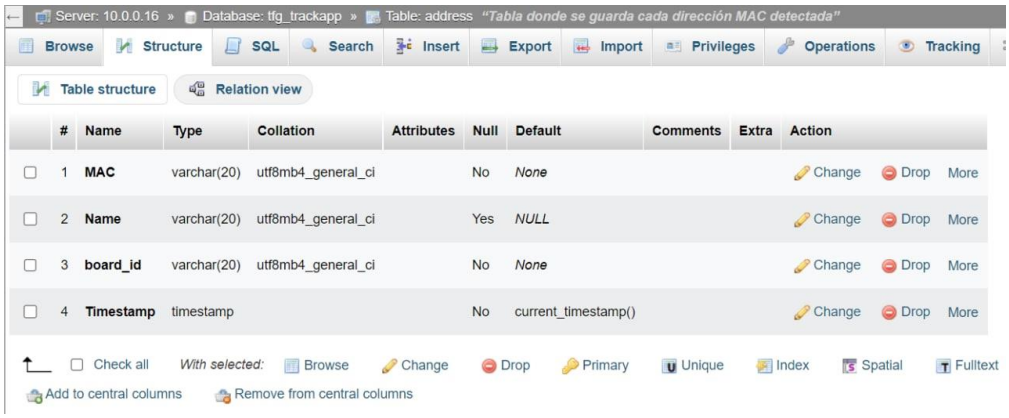


Figura 29. Estructura de la tabla *address*.

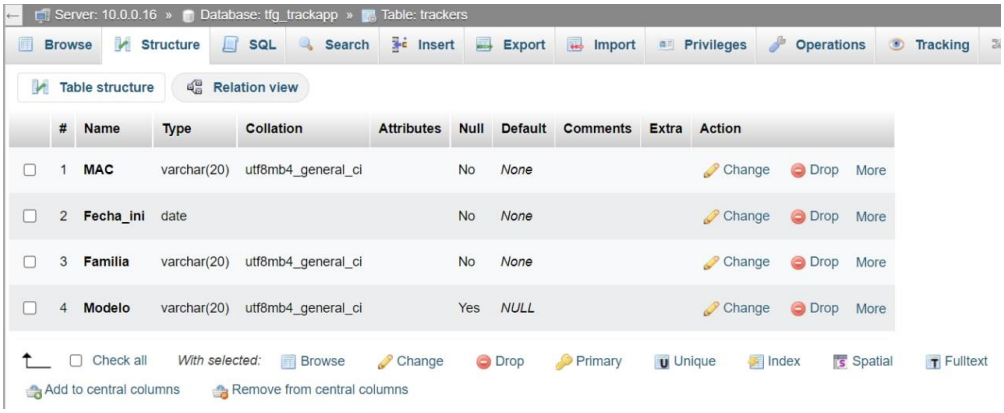


Figura 30. Estructura de la tabla *trackers*.

UNIVERSIDAD DE MÁLAGA | uma.es

E.T.S de Ingeniería Informática
Bulevar Louis Pasteur, 35
Campus de Teatinos
29071 Málaga

E.T.S. DE INGENIERÍA INFORMÁTICA