

Guía sobre ciberseguridad en aplicaciones IoT y entornos industriales 4.0

Memoria del trabajo

presentada para optar al grado de

Ingeniería Informática de Gestión y Sistemas de la Información

por

Julen Aristimuño Arana

Director: Isidro Calvo Gordillo



Universidad
del País Vasco Euskal Herriko
Unibertsitatea



**Escuela Universitaria
de Ingeniería
Vitoria-Gasteiz**

26 de febrero de 2018

Copyright ©2018 JULEN ARISTIMUÑO ARANA. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

AGRADECIMIENTOS

En primer lugar, gracias a mi madre, a mi padre, a mi hermano y a mi novia Erika por apoyarme en los momentos más difíciles de mi vida. Siempre habéis estado conmigo dándome fuerzas cuando las necesitaba y ayudándome a superar los obstáculos más importantes.

En segundo lugar, gracias a mi compañero y amigo Ander. Eres lo mejor que me llevo de la carrera, nunca se me olvidarán todas las aventuras que hemos pasado juntos (incluido el viaje a Madrid). Siempre has estado ahí para echarme una mano cuando lo he necesitado, aquí tienes un amigo para lo que quieras.

A continuación, quiero agradecer a mis amigos por ser otro apoyo fundamental en mi vida. Sin ellos yo hoy no sería lo que soy.

Por último, pero no por ello menos importante, gracias a mi director de trabajo de fin de grado Isidro. Me has ayudado a encontrar el camino cuando mis ideas no estaban claras, en este tiempo me has enseñado muchas cosas y me has hecho despertar curiosidad por temas que antes no conocía. Gracias por dedicarme parte de tu tiempo, este proyecto ha sido una experiencia muy enriquecedora.

ÍNDICE GENERAL

Agradecimientos	II
Resumen y organización de la memoria	XI
Resumen	XII
Abstract	XIII
Organización	XIII
I Alcance del trabajo	1
1. Descripción, Motivación y Objetivos del trabajo	3
1.1. Descripción	3
1.2. Objetivos	4
1.3. Motivación	4
II Estado del Arte de las Comunicaciones y Vulnerabilidades	5
2. Conceptos generales	7
2.1. Industria 4.0	7
2.2. Seguridad informática	12
2.2.1. Seguridad en el IoT	13
3. Análisis de vulnerabilidades	15
3.1. Vulnerabilidades en la comunicación	15

3.2. Vulnerabilidades en la aplicación	30
3.3. Vulnerabilidades en el sistema	36
4. Análisis de alternativas	41
4.1. Placa Base	41
4.2. Servidor Web	42
4.3. Sistema Operativo	47
4.4. Lenguaje de programación	49
4.5. Base de datos	51
4.6. Protocolo de comunicación entre la placa base y el VPS	53
4.7. Visualización de los datos	55
4.8. Plataforma seleccionada	56
III Recomendaciones y guía al programador	61
5. Propuesta	63
5.1. Guía al programador	66
5.1.1. Instalación y configuración básica de los elementos necesarios en la Raspberry Pi	67
5.1.2. Instalación y configuración básica de los elementos necesarios en el VPS	69
5.2. Recomendaciones	72
5.2.1. Recomendaciones de seguridad para Raspberry Pi	72
5.2.2. Recomendaciones de seguridad para el VPS	76
5.2.3. Recomendaciones de seguridad para Node-RED	78
5.2.4. Recomendaciones de seguridad para MQTT	80
5.2.5. Quick Check Guide	83
5.3. Riesgos	84
IV Caso de estudio	85
6. Manual de usuario del prototipo	87

V Análisis del trabajo	91
7. Conclusiones	93
8. Trabajo futuro	95
9. Viabilidad	97
9.1. Requisitos funcionales del trabajo	97
9.2. Planificación del tiempo	97
9.2.1. Estructura de descomposición del trabajo	98
9.2.2. Tareas	99
9.2.3. Entregables	107
9.2.4. Agenda del proyecto	107
9.2.5. Cronograma	107
9.3. Gestión de costos	110
9.3.1. Presupuesto	110
9.4. Gestión de riesgos	112
9.4.1. Explicación de los riesgos y plan de contingencia	113
VI Apéndices y bibliografía	117
A. Glosario	119
B. Conexión GPIO entre RPi y el sensor DHT 11	123
C. Creación y manejo de la base de datos MongoDB	127
D. Script cliente publicador MQTT	129
GNU Free Documentation License	133
1. APPLICABILITY AND DEFINITIONS	133
2. VERBATIM COPYING	135
3. COPYING IN QUANTITY	135
4. MODIFICATIONS	136
5. COMBINING DOCUMENTS	138
6. COLLECTIONS OF DOCUMENTS	138

7. AGGREGATION WITH INDEPENDENT WORKS	139
8. TRANSLATION	139
9. TERMINATION	139
10. FUTURE REVISIONS OF THIS LICENSE	140
11. RELICENSING	140
ADDENDUM: How to use this License for your documents	141
Bibliografía	142

ÍNDICE DE FIGURAS

2.1.	Componentes que forman la industria 4.0	8
3.1.	Funcionamiento visual de ARP Spoofing	18
3.2.	Funcionamiento visual de un Session Hijacking	21
3.3.	Funcionamiento de un servidor DNS	28
3.4.	Ejemplo de una inyección SQL	32
3.5.	Funcionamiento de un CSRF	35
4.1.	Uso de servidores en 2017	43
4.2.	Ejemplo de la interfaz de Node-RED	46
4.3.	Comunicación HTTP	53
4.4.	Comunicación MQTT	55
4.5.	Ejemplo de visualización de datos con Freeboard.io	56
4.6.	Logotipo Raspberry Pi	57
4.7.	Logotipo Node-Red	57
4.8.	Logotipo Raspbian	58
4.9.	Logotipo JavaScript	58
4.10.	Logotipo Python	59
4.11.	Logotipo MongoDB	59
4.12.	Logotipo MQTT	59
4.13.	Logotipo Clouding.io	60
5.1.	Diagrama de despliegue	63

5.2.	Diagrama de herramientas	64
5.3.	Diagrama de funcionamiento	65
5.4.	Diagrama de protocolos	66
5.5.	Ventana de selección del sistema operativo en NOOBS	68
5.6.	Login en Node-RED	79
5.7.	Node-RED sobre el puerto 1300	80
5.8.	Seguridad en el nodo Broker MQTT	82
6.1.	Diagrama de funcionamiento del prototipo	87
6.2.	Comando para ejecutar Node-RED	88
6.3.	Ejecución del cliente MQTT	89
6.4.	Mensajes recibidos por el cliente de Node-RED	89
6.5.	Últimas medidas guardadas en MongoDB	90
6.6.	Visualización de las últimas medidas en Freeboard.io	90
9.1.	Estructura de Descomposición del Trabajo 1	98
9.2.	Estructura de Descomposición del Trabajo 2	98
9.3.	Diagrama de Gantt 1	108
9.4.	Diagrama de Gantt 2	109
9.5.	Tabla de recursos	110
B.1.	Pines GPIO en Raspberry Pi 3	124
B.2.	Sensor DHT11	124
B.3.	Conexión completa del prototipo	125

ÍNDICE DE CUADROS

4.1.	Tabla comparativa de placas base	41
4.2.	Tabla total comparativa de placas base	42
4.3.	Tabla comparativa de servidores Web	45
4.4.	Tabla comparativa Apache vs Nginx	47
4.5.	Tabla total comparativa de servidores Web	47
4.6.	Tabla comparativa de sistemas operativos	49
4.7.	Tabla total comparativa de sistemas operativos	49
4.8.	Tabla comparativa de lenguajes de programación	50
4.9.	Tabla total comparativa de lenguajes de programación	50
4.10.	Tabla comparativa de bases de datos	52
4.11.	Tabla total comparativa de bases de datos	52
4.12.	Tabla resumen de las herramientas utilizadas	60
5.1.	Quick Check Guide	83
9.1.	Calendario de días festivos	107
9.2.	Coste de recursos trabajo	110
9.3.	Coste recursos materiales (Hardware)	110
9.4.	Coste recursos materiales (Software)	111
9.5.	Costo recursos trabajo	111
9.6.	Coste recursos materiales	111
9.7.	Amortizaciones de Hardware y Sofware	112
9.8.	Total presupuesto	112

9.9. Riesgos	113
B.1. Conexión de pines	125

RESUMEN Y ORGANIZACIÓN DE LA MEMORIA

Resumen

En este proyecto se lleva a cabo el diseño de una aplicación perteneciente al sector del Internet de las Cosas para posteriormente, añadir medidas de seguridad con el fin de que la aplicación eleve su grado protección.

Para conseguir este objetivo, previamente se realiza una pequeña introducción al estado de la industria 4.0 y al de la seguridad informática en la actualidad. Seguidamente, se realiza un estudio de vulnerabilidades y de alternativas para elegir las mejores tecnologías con las que realizar la aplicación

El resultado de este estudio es un sistema de recepción y envío de mensajes entre unas Raspberry Pi 3 y un Servidor Virtual Privado. Los mensajes enviados se tratan de mediciones de temperatura y humedad que realiza un sensor DHT11 conectado a la Raspberry Pi. Para la comunicación entre la RPi y el VPS se ha utilizado Mosquitto, una tecnología que está en auge. Seguidamente, con plataformas como Node-RED y Freeboard.io los mensajes son recibidos y mostrados en gráficas y finalmente guardados en la base de datos de MongoDB.

El proyecto ha sido desarrollado con dos lenguajes de programación Python y JavaScript utilizando librerías como Adafruit.io o Paho-MQTT utilizadas para la conexión RPi-sensor DHT11 y para la programación del cliente MQTT respectivamente.

El software desarrollado consta de un cliente programado desde 0 para realizar la recepción de las medidas realizadas por el sensor y la publicación de mensajes MQTT bajo un tópico. Posteriormente, desde Node-RED se ha desarrollado una paleta donde se ha implementado el bróker de MQTT encargado de gestionar los mensajes y se han implementado los nodos de MongoDB y Freeboard.io.

La primera versión que se desarrolla de la aplicación se encuentra sin ninguna medida de seguridad, por lo que será potencialmente vulnerable. En la segunda versión se implementan medidas elevando en gran medida el grado de seguridad de la aplicación.

Abstract

Organización

En la segunda parte de este proyecto se realiza una introducción al área de la seguridad informática, tanto en su impacto en la Industria 4.0 como en el auge del Internet de las Cosas. Dentro de esta parte también, se realizará un análisis de las vulnerabilidades más populares o peligrosas que existen hoy en día para que cualquier persona que no pertenezca al sector de la ciberseguridad pueda leerlo y tener un conocimiento básico de qué factores de riesgo existen y tomar conciencia de ello.

Después de conocer las distintas vulnerabilidades, comenzará el análisis de las alternativas que existen para realizar la aplicación. En este punto se expondrán diferentes tecnologías y se compararán entre sí para llegar a la conclusión de cuál es la más indicada. No siempre se elegirá la mejor tecnología ya que en este área se deben prestar atención a muchas opciones, por ejemplo: la usabilidad, la dificultad de la tecnología, la fiabilidad o la potencia que ofrecen.

En la tercera parte se realizará una guía al programador donde se detallarán todas las instalaciones necesarias para llevar a cabo la realización de la aplicación. A continuación, dentro de la sección de recomendaciones se explicarán las medidas de seguridad que se han decidido introducir y cómo llevar a cabo su implantación. Para terminar con esta parte se indicarán los riesgos que siguen existiendo.

El desarrollo de toda la aplicación viene en la cuarta parte. En este apartado se detallará el funcionamiento de la aplicación final paso a paso.

En la quinta parte del proyecto se realizará un estudio de la viabilidad del proyecto, se expondrán las conclusiones a las que hemos llegado después de tantas horas de investigación y trabajo de desarrollo y por último, se expondrá el posible trabajo futuro.

En el último apartado de apéndices se incluirán todas las palabras técnicas utilizadas y los acrónimos para que el usuario que lea el proyecto pueda ver los significados en caso de no entender el significado.

Parte I

Alcance del trabajo

CAPÍTULO 1

DESCRIPCIÓN, MOTIVACIÓN Y OBJETIVOS DEL TRABAJO

1.1. Descripción

En este trabajo se desarrollará un análisis de todos los riesgos de seguridad que puede haber en un sistema de recepción y entrega de datos dentro de cualquier organización que requiera este tipo de comunicaciones. Como resultado de este análisis, se implantarán medidas de seguridad sobre un sistema de envío y recepción de datos (IoT) para asegurar unos principios básicos de seguridad en las comunicaciones.

Hoy en día la seguridad es esencial en cualquier empresa que quiere proteger sus datos, por lo tanto, integrar unas bases de seguridad es fundamental. Durante este estudio se tratará de identificar y describir los diferentes tipos de vulnerabilidades que podrían afectar al sistema. Seguidamente se realizará un análisis entre las diferentes alternativas existentes para realizar el caso de estudio. Más adelante, se hará una guía al programador y unas recomendaciones para que sea capaz de construir una aplicación de comunicaciones con una base de seguridad. La toma de decisiones no será fácil ya que continuamente se valorará la seguridad contra la usabilidad con el fin de realizar un escenario más real.

Para afianzar los conocimientos se realizará un caso de estudio donde se va a configurar un sistema de transferencia de datos entre una Raspberry Pi 3 y un VPS(Virtual Private Server) teniendo en cuenta todas las vulnerabilidades mencionadas para hacerlo seguro.

Cabe remarcar que la aplicación que va a ser desarrollada se trata de una aplicación perteneciente al sector del IoT y las máquinas donde va a ser ejecutada solo podrán utilizarse para esa función en concreto y nada más. También, mencionar que este trabajo está dirigido hacia personas que trabajan en este sector pero no disponen del tiempo necesario para investigar sobre ciberseguridad.

1.2. Objetivos

El objetivo principal de este proyecto es analizar las diferentes vulnerabilidades que existen hoy en día, conocer los principios básicos que se necesitan para establecer comunicaciones seguras y realizar una guía para que una persona que trabaja en el sector y no sabe de seguridad, siguiendo los pasos que se van a establecer, consiga crear una comunicación mínimamente segura.

Al existir una gran variedad de tecnologías hoy en día, dentro del proyecto se realizará un análisis de alternativas donde se compararán las tecnologías más adecuadas para nuestro caso. Haciendo que el usuario vea con facilidad porque se ha seleccionado un componente en concreto.

Este trabajo al ser tan extenso tiene también como objetivo que pueda ser usado como base para otros proyectos sobre ciberseguridad o que cualquier persona quiera continuar desarrollando aun más la aplicación e implantando medidas de seguridad que yo por no alargar el proyecto más no he podido implantar.

Por último, en este proyecto toman parte diferentes áreas de la informática. La seguridad es la base pero el dominio de la programación y de las redes informática para realizar todas las conexiones que se verán a continuación hacen que el proyecto eleve el grado de dificultad y el resultado sea aún mejor.

1.3. Motivación

La pasión que tengo por esta área de la informática, hizo que buscara un proyecto adecuado para mis intereses actuales y futuros. Quería realizar un proyecto enfocado en seguridad informática para que me resultara divertido realizarlo y a su vez, adquirir conocimientos. Cuando supe que mi director de proyecto, Isidro Calvo Gordillo, quería realizar un trabajo similar al que yo estaba buscando, no dude en ponerme en contacto con él para empezar cuanto antes. Este trabajo surge del auge que está sufriendo la ciberseguridad ahora que está tomando fuerza el IoT y la Industria 4.0, un mundo conectado.

Mi motivación es ayudar a que personas sin conocimientos como he citado anteriormente, configuren un sistema seguro sin tener conocimiento sobre esta área.

Para terminar, todos sabemos que ningún sistema es 100 % seguro y que todos somos vulnerables, pero tenemos que hacer todo lo que esté en nuestra mano para ponerles dificultades a cualquiera que quiera perjudicarnos.

Parte II

Estado del Arte de las Comunicaciones y Vulnerabilidades

CAPÍTULO 2

CONCEPTOS GENERALES

2.1. Industria 4.0

Definición

“Estamos al borde de una revolución tecnológica que modificará fundamentalmente la forma en que vivimos, trabajamos y nos relacionamos. En su escala, alcance y complejidad, la transformación será distinta a cualquier cosa que el género humano haya experimentado antes” [1].

Tal y como dijo Klaus Schwab fundador del Foro Económico Mundial, la cuarta revolución industrial está ocurriendo a gran escala y a toda velocidad. Pero, ¿De dónde viene todo esto? ¿Qué es la Industria 4.0?

Definida de una manera sencilla, la Industria 4.0 representa la llegada de la nueva revolución industrial, cuya característica principal es la introducción del Internet of Things (IoT), el tráfico de datos masivo (Big Data) y servicios aplicados a la industria.

Esta nueva industria representa un cambio en el paradigma de producción centralizada a una producción descentralizada. A su vez, conecta tecnologías de producción en sistemas embebidos con procesos de producción inteligentes para dar paso a una nueva era de tecnología que cambiará radicalmente la industria y los modelos de negocio.

Los cambios en las tecnologías de la comunicación (Information and Communication Technology (ICT)) han roto los límites entre la realidad virtual y el mundo real. La idea detrás de la Industria 4.0 es la de crear una red donde las máquinas puedan comunicarse entre sí, este concepto generalmente es conocido como Internet de las Cosas. Además, los fabricantes también tienen que tomar parte en esas comunicaciones dentro de esas redes. De esta forma, todo estará conectado, creando lo que hasta ahora conocemos como un sistema de producción cyber-físico (Cyber Physical System (CPS)). Todo esto ayuda a la industria a integrar el mundo real dentro de la

realidad virtual permitiendo a las máquinas recolectar datos, analizarlos y tomar decisiones basadas en ellos.

Componentes

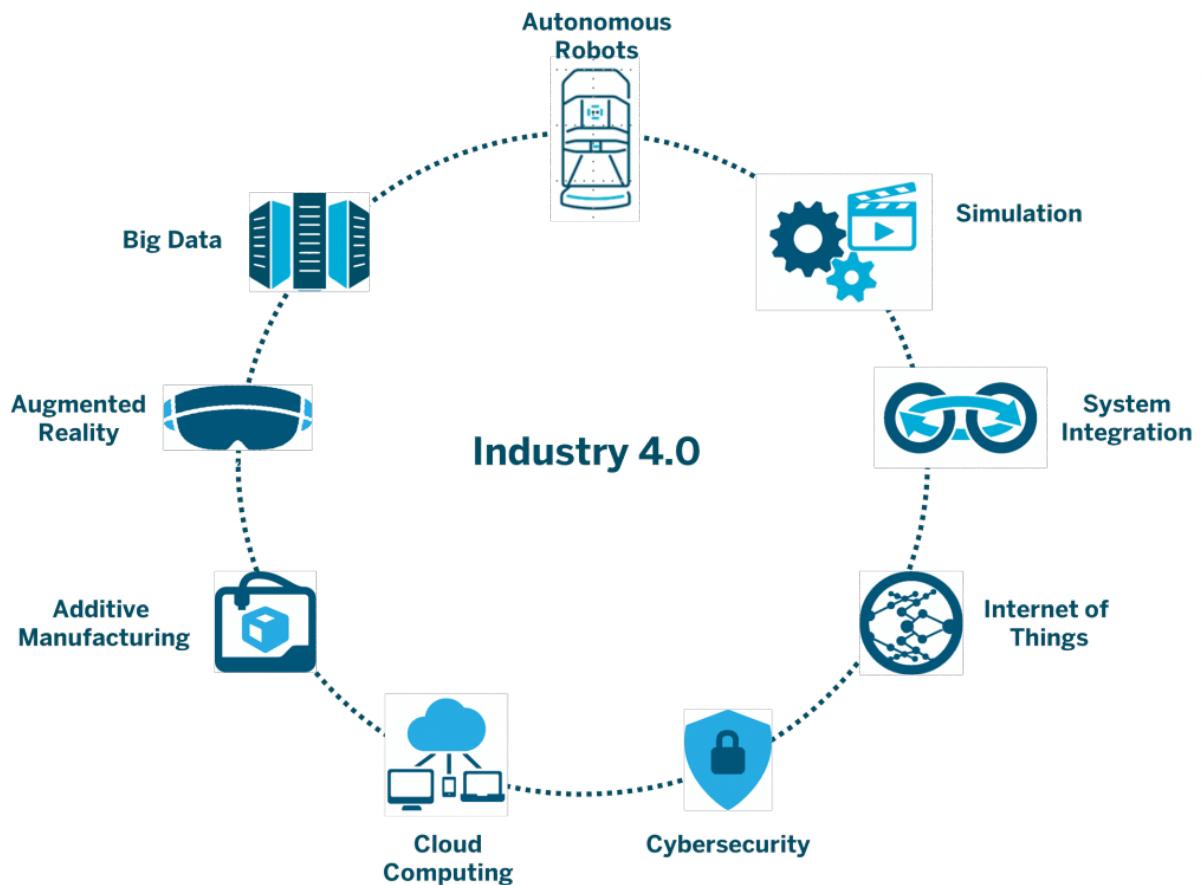


Figura 2.1: Componentes que forman la industria 4.0

Como se puede observar en la Figura 2.1 [2], la Industria 4.0 se apoya sobre 9 pilares. Células optimizadas y aisladas vendrán juntas completamente integradas en un flujo de producción optimizado y automatizado. Esto provocará un cambio en la producción tradicional y aumentando drásticamente la eficiencia.

■ Robots autónomos:

Actualmente los robots forman parte de la industria realizando tareas complejas, pero están evolucionando para tener mayor utilidad. Se están haciendo cada vez más flexibles y cooperativos. Estos robots, podrán comunicarse con otros y a su vez con humanos consiguiendo trabajar de manera conjunta y segura. Costarán menos dinero mantenerlos y su rango de capacidades será mucho mayor comparado a los robots que hoy en día se utilizan para la fabricación.

■ Simulación:

Hoy en día ya se están usando sistemas de simulación, pero en un futuro, se usarán simulaciones más extensamente para todo tipo de operaciones incluidas las de planta.

Estas simulaciones volcarán datos en tiempo real para reflejar el mundo real en un modelo virtual que puede incluir máquinas, productos o humanos. Esto permitirá a los operarios testear y optimizar la configuración de las máquinas para maximizar la eficiencia y la calidad de los productos.

■ Integración de sistemas horizontales y verticales:

Muchos de los sistemas IT de hoy en día no están completamente integrados, dando lugar a pobres sistemas de conexión entre departamentos o entre las compañías y los proveedores. Pero con la llegada de la nueva industria, compañías, departamentos, funciones y capacidades van a ser más cohesivos.

■ Big Data:

Los análisis de datos masivos han aparecido recientemente en el mundo de la fabricación, estos análisis optimizan la calidad de la producción, ahorran energía y mejoran los servicios. En la industria 4.0, la recolección y evaluación de datos extraídos de diversas fuentes van a convertirse en un estándar para el apoyo en la toma de decisiones en tiempo real.

■ Internet of Things:

Hoy en día pocos de los sensores y máquinas destinadas a la fabricación están conectados y toman parte de la computación embebida. Están organizados típicamente en una pirámide vertical de automatización en donde los sensores y dispositivos con poca inteligencia alimentan a un sistema de control de proceso de fabricación general. Pero con el IoT industrial, muchos dispositivos van a enriquecerse utilizando tecnologías estandarizadas. Esto permitirá que los dispositivos interactúen entre sí y con controladores si es necesario.

■ Ciberseguridad:

Muchas compañías siguen confiando en gestionar sistemas que no están conectados o están cerrados. Con el crecimiento de las conexiones y uso de protocolos de comunicación que vienen impuestos en la Industria 4.0, la necesidad de proteger sistemas industriales críticos incrementa exponencialmente la obligación de integrar la ciberseguridad en cualquier entorno empresarial. Como resultado, son esenciales comunicaciones seguras y confiables, además de un potente control de acceso e identificación antes de utilizar cualquier máquina.

■ The Cloud:

Las compañías están usando actualmente software basado en la nube para algunas de sus aplicaciones. Con la llegada de la nueva era industrial, se van a requerir mayores flujos de datos fuera de los límites de las empresas. Al mismo tiempo, la ejecución de tecnologías basadas en la nube, van a lograr tiempos

de reacción de unos cuantos milisegundos. Esto provocará que los datos de las máquinas acabe desplegándose a la nube. Hasta sistemas de monitorización y control de procesos pueden convertirse en procesos basados en la nube.

- **Additive manufacturing:**

Las empresas han empezado a incluir la fabricación aditiva, como impresoras 3D, que utilizan habitualmente para crear prototipos y producir componentes. Con la nueva revolución industrial, estos métodos de fabricación aditiva van a ser ampliamente utilizados para producir pequeños lotes de productos personalizados que ofrecen ventajas de construcción. Los sistemas de fabricación aditiva descentralizada reducirán las distancias de transporte y el Stock en general.

- **Realidad aumentada:**

Los sistemas de realidad aumentada dan soporte a variedad de servicios, estos servicios se encuentran ahora en su infancia, pero en el futuro, las compañías van a realizar un uso más amplio de estos sistemas para proveer a los trabajadores de información en tiempo real para mejorar las toma de decisiones y los procedimientos de trabajo.

Beneficios y Desafíos

La nueva era va a revolucionar los procesos de producción. Sin embargo, cabe mencionar las ventajas y desafíos que van a tener las empresas.

Ventajas:

- **Personalización:**

Crear un mercado flexible va a ayudar a conocer las necesidades de la gente de una manera más rápida y fluida. La comunicación entre fabricantes y clientes será directa. Los fabricantes no van a necesitar comunicaciones internas ni externas lo que va a agilizar mucho la producción.

- **Optimización:**

La optimización de la producción es la ventaja clave de la nueva industria. La fábrica inteligente contará con cientos de dispositivos inteligentes que van a ser capaces de optimizar la producción para que no haya ningún retraso. Este es un factor muy importante para empresas que utilizan equipamiento caro.

- **Impulsar la investigación:**

La adopción de la nueva industria va a impulsar la investigación en algunos campos como la seguridad. Una nueva industria requerirá unas nuevas habilidades.

Desafíos:

■ Privacidad:

En una industria interconectada, los fabricantes van a necesitar recolectar y analizar información. Para un cliente esto puede suponer una violación de su privacidad.

■ Capital:

Una transformación de este calibre va a necesitar unas inversiones enormes de dinero para nueva tecnología.

■ Seguridad:

Este va a ser el desafío más importante que va a plantear la llegada de la revolución. Esta integración de sistemas online va a traer nuevas brechas de seguridad y fugas de datos. Todos los riesgos que supone la falta de seguridad hace casi obligatorio el impulsar la investigación en este campo.

2.2. Seguridad informática

Definición

La seguridad informática trata de minimizar los riesgos asociados al acceso y utilización de determinado sistema de forma no autorizada y en general malintencionada. La seguridad informática implica la necesidad de gestión, fundamentalmente gestión del riesgo. Para ello, se deben evaluar y cuantificar los bienes a proteger e implantar medidas preventivas y correctivas que eliminen los riesgos asociados o que los reduzcan.

Este campo cada día requiere más atención porque cada vez son más los dispositivos que se conectan a Internet. Las empresas, instituciones o gobiernos utilizan más la Web para intercambiar información o realizar transacciones, si un atacante logra entrar en alguno de esos servidores Web podría interceptar información o dinero [3].

A parte de esto, existen varios motivos más para preocuparnos sobre la ciberseguridad:

- **Comercio:**

En la actualidad, las compras online están a la orden del día. Existen multitud de servidores relacionados con el comercio y particularmente con el dinero. Existe seguridad para intentar que las transacciones sean seguras y que los usuarios puedan así realizar las transacciones sin preocuparse, pero nada puede garantizar la seguridad al 100 %. Estos servidores al albergar información financiera se han convertido en un blanco atractivo para los atacantes.

- **Información Confidencial:**

Las empresas han entrado en la Web, esto provoca que distribuyan una gran cantidad de información por las redes. Esta información fluye con gran sencillez entre empleados internos y socios externos a la empresa. Dentro de toda esta información hay información confidencial que si algún atacante intercepta podría poner en riesgo la propia empresa. Por lo tanto, esta información se convierte en un objetivo para atacantes o empresas de la competencia.

Objetivos

La seguridad informática es el conjunto de procedimientos y estrategias que permiten garantizar la integridad, la disponibilidad y la confidencialidad de la información de una entidad. Por lo tanto, se debe proteger la consistencia del triángulo C-I-D.

La **confidencialidad** consiste en asegurar que sólo los individuos autorizados tengan acceso a los recursos que se intercambian.

La **integridad** se basa en garantizar que los datos sean los que se supone que son, en otras palabras, asegurarse de que los datos no hayan sufrido alteraciones.

La **disponibilidad** trata de garantizar el acceso a los datos.

Aparte de estos tres conceptos, es necesario incluir otros dos fundamentales en este ámbito:

El **no repudio** consiste en que ninguna de las partes involucradas pueda negar en el futuro una operación realizada.

La **autenticación** consiste en la confirmación de la identidad de un usuario; es decir, la garantía para cada una de las partes de que su interlocutor es realmente quien dice ser.

2.2.1. Seguridad en el IoT

El IoT es el apartado más importante de la Industria 4.0 por ello es necesario conocer el concepto y sus riesgos de seguridad. Por otro lado, el Internet de las cosas está revolucionando Internet y sobre todo la ciberseguridad, por lo tanto, conocer las bases del IoT se hace necesario.

Introducción

El Internet of Things, Internet de las Cosas o IoT es un concepto que se refiere a la interconexión digital de los objetos cotidianos con Internet, convirtiéndose así en objetos inteligentes.

La clave está en las herramientas que convierten a los objetos en objetos inteligentes, permitiéndoles conectarse a Internet para ampliar las funciones que son capaces de realizar.

Un ejemplo claro sería un frigorífico. Hasta hace pocos años los frigoríficos no tenían otra función aparte de conservar los alimentos fríos. En cambio, un frigorífico inteligente conectado a Internet podría realizar estudios de los productos que compramos o incluso darnos avisos cuando algún producto esté agotándose.

El llamado IoT parece estar predestinado a cambiar por completo el escenario socioeconómico tal y como hoy lo conocemos. Tanto por dispositivos de domótica encargados de crear un hogar inteligente, como dispositivos ponibles como relojes que ampliaran sus funciones.

En 2020 el número de dispositivos conectados a Internet será superior a 26.000 millones, 30 veces más que los dispositivos que estaban conectados en el 2009.

La seguridad del IoT preocupa a los expertos

La seguridad siempre es un tema muy relevante a tener en cuenta, su importancia resulta aún mayor cuando nos encontramos con dispositivos conectados que sirven para conducir un coche o para guiar un marcapasos.

Una persona que no sabe nada de seguridad puede intuir el riesgo que supone que un atacante consiga acceder al sistema de un coche inteligente creado para personas

sin visión. El atacante podría cambiar algunas variables mientras está en funcionamiento y provocar un accidente.

El increíble desarrollo del IoT está cambiando por completo la percepción que se tenía de Internet, hacia una visión integrada de objetos que interactúan entre sí. Tanto es así, que el número de sensores y dispositivos conectados está creciendo exponencialmente.

En este nuevo paradigma donde puede manejarse información sensible, la pérdida de información o acceso incontrolado puede afectar seriamente a la privacidad de sus usuarios, por lo que la seguridad se constituye como un factor clave en el desarrollo y despliegue de estos nuevos escenarios. Este es un tema preocupante, ya que a día de hoy no se dispone de una guía clara de acción para la seguridad IoT.

La seguridad es la principal barrera para la implantación del IoT, debido a que las soluciones tradicionales de seguridad no pueden ser aplicadas a estos nuevos dispositivos. Muchas de las soluciones tradicionales requieren de un consumo de recursos elevado y estos dispositivos no carecen de ellos. Por lo tanto, el gran desafío que se plantea en la actualidad es el de dar soluciones de seguridad a estos dispositivos.

Dentro de este sector, estos son los vectores de ataque más comunes [4]:

- **Deficiencias de la seguridad en la transmisión de datos**
- **Deficiencias en la seguridad de la plataforma software**
- **Deficiencias de la seguridad de la funcionalidad y configuración**
- **Deficiencias de la seguridad del hardware**
- **Deficiencias en la cultura de seguridad de los usuarios**

A medida que avance el tiempo, más serán los dispositivos conectados y mayores serán los riesgos asociados a la seguridad. Sin embargo, las instituciones y empresas destinarán más recursos para la investigación en este ámbito lo que provocará un auge de la seguridad informática y se crearán soluciones para los riesgos que surgirán.

CAPÍTULO 3

ANÁLISIS DE VULNERABILIDADES

"Las empresas invierten millones en firewalls, cifrado y dispositivos para acceder de forma segura, y es dinero malgastado, porque ninguna de estas medidas corrige el nexo más débil de la cadena."

— Kevin Mitnick

Una vulnerabilidad es una debilidad o un fallo en un sistema de información que pone en riesgo la seguridad de la información pudiendo permitir que un atacante pueda comprometer la integridad, disponibilidad o confidencialidad de la misma, por lo que es necesario eliminarlas lo antes posible. Estos fallos pueden tener distintos orígenes por ejemplo: fallos de diseño o errores de configuración [5].

En este apartado del proyecto se realizará el análisis de las vulnerabilidades más conocidas hasta ahora y se hará especial hincapié en las que puedan aparecer en el caso de estudio posterior. Vamos a distinguir tres tipos de vulnerabilidades según su origen.

3.1. Vulnerabilidades en la comunicación

Ataque DoS y DDoS al servidor

Un Denial of Service (DoS)¹ o un ataque de denegación de servicio es aquel ataque que tiene como objetivo degradar la calidad de servicio de un sistema o una red llegando a dejarlo en un estado no operativo o inaccesible. Este estado puede lograrse saturando los recursos disponibles o causando un error grave que provoque que algún proceso crítico o todo el sistema deje de estar operativo [6].

¹En el apartado de anexos se ha realizado un glosario de términos y acrónimos

Como evolución de los ataques de denegación de servicio convencionales surgieron los ataques de denegación de servicio distribuidos o Distributed Denial of Service (DDoS), que consisten en la coordinación de múltiples ataques DoS desde distintas fuentes contra uno o varios objetivos.

Estos ataques son uno de los tipos de incidentes de seguridad más comunes incluso a día de hoy. Suelen tener efectos devastadores en los sistemas atacados y solo es posible una defensa eficaz mediante una amplia cooperación entre los diferentes actores que componen Internet, desde los proveedor de servicios (Internet Service Provider (ISP)) hasta los usuarios finales, que deben poner los medios para mantener los equipos libres de malware, como por ejemplo botnets, que puedan ser utilizadas en este tipo de ataques.

Existen varios tipos de ataques DoS pero dos son los más importantes:

- **Saturación:** Su objetivo es agotar o saturar alguno de los recursos clave del sistema, entre los que se incluyen el tiempo de Central Processing Unit (CPU), memoria, ancho de banda, accesos a sistemas externos, espacio en disco o alimentación de los sistemas. El grupo hacktivista Anonymous ha realizado múltiples ataques de este tipo.
- **Modificación de la configuración:** Su objetivo es alterar o eliminar la configuración de alguno de los elementos clave del sistema, típicamente servidores o routers. Estos cambios suelen provocar efectos críticos en los sistemas afectados, dejándolos fuera de servicio. Por último, la llegada del IoT ha provocado un incremento de este tipo de ataques. La falta de controles de seguridad que afecta a los dispositivos conectados que forman parte del Internet de las cosas se encuentra detrás del incremento de ataques de denegación de servicio y de denegación de servicio distribuido experimentado el año pasado [7]. Es decir, han aumentado drásticamente el número de dispositivos conectados a Internet, por lo que comprometer dispositivos para realizar ataques de denegación de servicio distribuidos es relativamente más sencillo.

Suplantación de identidad (Spoofing) y Man in the Middle (MiTM)

Internet alberga numerosas redes tanto públicas como privadas, gran parte del tráfico de datos mundial tiene lugar por medio de las redes públicas que no se encuentran cifradas. Los paquetes de datos fluyen por las redes entre el remitente y el destinatario. En su camino van a encontrarse con diferentes niveles de seguridad, como todos sabemos este tráfico de datos puede verse comprometido si la seguridad del canal no es la suficiente. De aquí surge la necesidad de crear medios de comunicación lo más seguros posibles. El ataque Man in The Middle (MiTM) surge cuando un atacante consigue hacerse pasar por alguno de los participantes en la comunicación (Spoofing) interviniendo en el tráfico de datos.

Estos ataques normalmente pretenden anular la codificación por medio de certificados Secure Sockets Layer (SSL)/Transport Layer Security (TLS) y poder acceder de este modo a toda la información sensible.

El esquema básico de este tipo de ataques es el siguiente:

El sistema A intenta crear una conexión codificada con el sistema B. En su lugar, un tercer partido malintencionado desvía el flujo de datos para establecer la conexión codificada del sistema A con el sistema C y que a partir de ahí se transmita al sistema B. Esto tiene como consecuencia que aquél que tenga el control sobre el sistema C (el atacante generalmente) pueda examinar, grabar o manipular el tráfico de datos, a menudo incluso sin que los participantes en la comunicación sean conscientes de ello. Una vez hecha la transmisión a la world wide web, el sistema C se presentará como servidor web ante el sistema A y como navegador web ante el sistema B.

Como se puede apreciar, para realizar un ataque de hombre en el medio es necesario que haya una suplantación de identidad. Dependiendo de qué elemento sea el suplantado, se distinguen algunos tipos de Spoofing: IP Spoofing, Address Resolution Protocol (ARP) Spoofing, DNS Spoofing, Web Spoofing, E-Mail Spoofing y GPS Spoofing.

IP Spoofing

Este tipo de ataque consiste en sustituir la dirección IP origen de un paquete TCP/IP por otra dirección IP a la cual se desea suplantar. Esta suplantación se logra gracias a programas creados para hacer este tipo de tareas. Debemos de tener en cuenta que las respuestas del host que reciba los paquetes irán dirigidas a la IP falsificada. Por ejemplo si se envía un ping spoofeado, la respuesta será recibida por el host al que pertenece la IP legalmente.

Para que se establezcan conexiones de capa de aplicación, se requiere que el host y el cliente se involucren en un proceso de verificación mutua, conocido como un apretón de manos TCP a tres bandas o TCP three-way handshake.

El proceso consiste en el intercambio de paquetes de sincronización (SYN) y de reconocimiento (SCK). Primero el cliente le envía un paquete SYN al host, seguidamente, el host responde con un paquete SYN-ACK y por último, el cliente acepta la recepción del SYN-ACK respondiendo con un paquete ACK [8]. Por lo tanto, para poder realizar IP SPOOFING en sesiones TCP, se debe tener en cuenta el comportamiento de dicho protocolo con el envío de paquetes SYN y ACK con su ISN específico y teniendo en cuenta que el propietario real de la IP podría cortar la conexión en cualquier momento al recibir paquetes sin haberlos solicitado. También hay que tener en cuenta que los routers actuales no admiten el envío de paquetes con IP origen no perteneciente a una de las redes que administra (los paquetes spoofeados no sobrepasarán el router) [9].

A día de hoy, es muy complicado realizar este tipo de ataque ya que la seguridad informática a sido desarrollada para lograr tapar estas brechas de seguridad.

Para terminar, este tipo de ataques son comúnmente utilizados dentro de dos ataques mencionados anteriormente:

- **MiTM:**

El IP Spoofing es utilizado para suplantar alguna identidad engañando a la víctima y haciendo que revele información sensible ya que el atacante confía en él.

- **DoS:**

Se utiliza el IP Spoofing para suplantar la IP de origen para hacer que el trazado y posterior detención del DoS sea tan difícil como sea posible.

ARP Spoofing

Es una especie de ataque en el que el atacante envía mensajes ARP falsificados a una LAN. La finalidad del atacante es vincular su dirección MAC con la dirección IP de un equipo legítimo en la red. Si el atacante logra su objetivo, va a recibir cualquier dato al que se pueda acceder desde esa dirección IP.

La falsificación de ARP permite a los atacantes maliciosos controlar y realizar cualquier acción con los datos que están en tránsito. Estos ataques ocurren en redes de área local que utilizan el protocolo de resolución de direcciones.

Si una empresa no es lo suficientemente segura y recibe un ataque de este tipo, puede verse gravemente afectada ya que los atacantes pueden robar toda la información sensible de ésta.

En la figura 3.1 [10] puede observarse el funcionamiento de un ARP Spoofing.

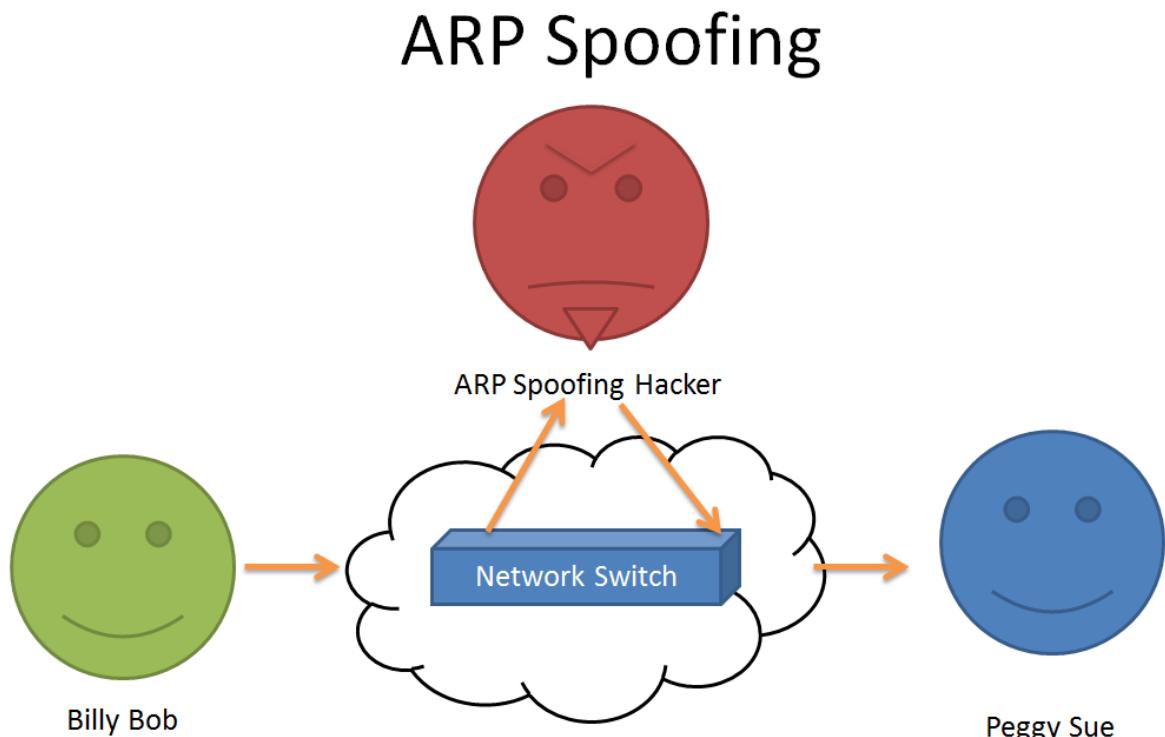


Figura 3.1: Funcionamiento visual de ARP Spoofing

Para realizar este tipo de ataques se suelen seguir los siguientes pasos [10]:

1. Primero el atacante utiliza una herramienta destinada a realizar este tipo de ataques, y establece la dirección IP de la herramienta para que coincida con la IP de un objetivo.
2. Después, escanea las direcciones MAC e IP de los hosts de la subred del objetivo.
3. Seguidamente, elige su destino y comienza a enviar paquetes ARP a través de la Local Area Network (LAN). Estos paquetes contienen la dirección MAC del atacante y la dirección IP de la víctima.
4. Como otras computadoras de la red los paquetes ARP de suplantación imitan el cache, los datos que los hosts envían a la víctima, una vez falseados se redirigen al atacante. El atacante puede realizar ahora las acciones que quiera con los paquetes que le son redirigidos.

Al igual que con la suplantación de IP, este tipo de ataques también son utilizados para facilitar otro tipo de ataques:

■ **DoS:**

Utilizan ARP Spoofing para enlazar varias direcciones IP en una LAN con la dirección MAC de un solo objetivo. Debido a esto, el tráfico que está destinada a diferentes direcciones IP será redirigido a la dirección MAC del destino, sobrecargando así el objetivo con el tráfico.

■ **MiTM:**

Estos ataques pueden utilizar ARP Spoofing para interceptar y/o modificar el tráfico entre dos víctimas.

■ **Secuestro de sesiones:**

Usan de ARP Spoofing para robar los identificadores de sesión, garantizando así el acceso a los atacantes y los sistemas privados de datos.

Secuestro (Hijacking)

El hijack es un tipo de ataque donde el intruso toma control de algún elemento de una comunicación establecida entre un servidor y un usuario. Como su nombre indica, básicamente el atacante secuestra el elemento que quiera explotar con el fin de sacar algún beneficio.

Al igual que con las suplantaciones, existen varios tipos de secuestros como por ejemplo: IP hijacking, Domain Hijacking, Session hijacking y Browser Hijacking entre otros. Se tratará de explicar brevemente en qué consiste cada uno de ellos.

IP Hijacking o BGP Hijacking

Primero es necesario conocer en qué consiste el protocolo Border Gateway Protocol (BGP). Es el encargado de establecer las rutas que deben seguir los paquetes dentro de las comunicaciones existentes en Internet. Este protocolo establece que cada Autonomous System (AS) que va a transportar los paquetes de datos debe indicar a sus vecinos los bloques de direcciones IP que contiene. Estos anuncios van comunicándose a través de la red.

El secuestro de BGP se produce cuando, por ejemplo, en un sistema AS1 que anuncia que posee un bloque de IP que es propiedad de la red AS2, entonces el tráfico de una parte de Internet destinada a AS2 se enrutará a AS1.

Por otra parte, los atacantes han comenzado a usar otro tipo de secuestro de datos para hacer ataques más sigilosos. En vez de usar el protocolo BGP para conseguir el desvío de paquetes, el atacante cambia directamente la tabla de reenvío de los routers que estén en el camino. El atacante accederá a los routers de la ISP por medio de algún tipo de ataque como por ejemplo, usar un backdoor en el router. A este tipo de ataques se les llama IP Hijacking [11].

Session Hijacking

Este tipo de ataque consiste en engañar a un cliente o un servidor haciendo que crean que el host controlado por el atacante es el host legítimo. De este modo, el host del atacante va a poder manipular la red para convertirse a sí mismo en el destinatario de los paquetes.

El secuestro de sesiones es un método donde el cibercriminal toma la sesión de un usuario Web obteniendo su ID y disfrazándose como un usuario autorizado. Una vez que consigue suplantar al usuario legítimo, está autorizado a realizar cualquier cosa en la red.

En particular, este método se refiere al robo de una cookie² usada para la autenticación de un usuario a un servidor remoto. Tiene especial relevancia para los desarrolladores web, ya que las cookies HTTP utilizadas para mantener una sesión en muchos sitios web pueden ser fácilmente robadas por un atacante usando un ordenador intermediario o con acceso a las cookies guardadas en el ordenador de la víctima [12].

En la figura 3.2 [13] se puede observar el funcionamiento de un Session Hijacking.

²En el apartado de anexos se ha realizado un glosario de términos y acrónimos

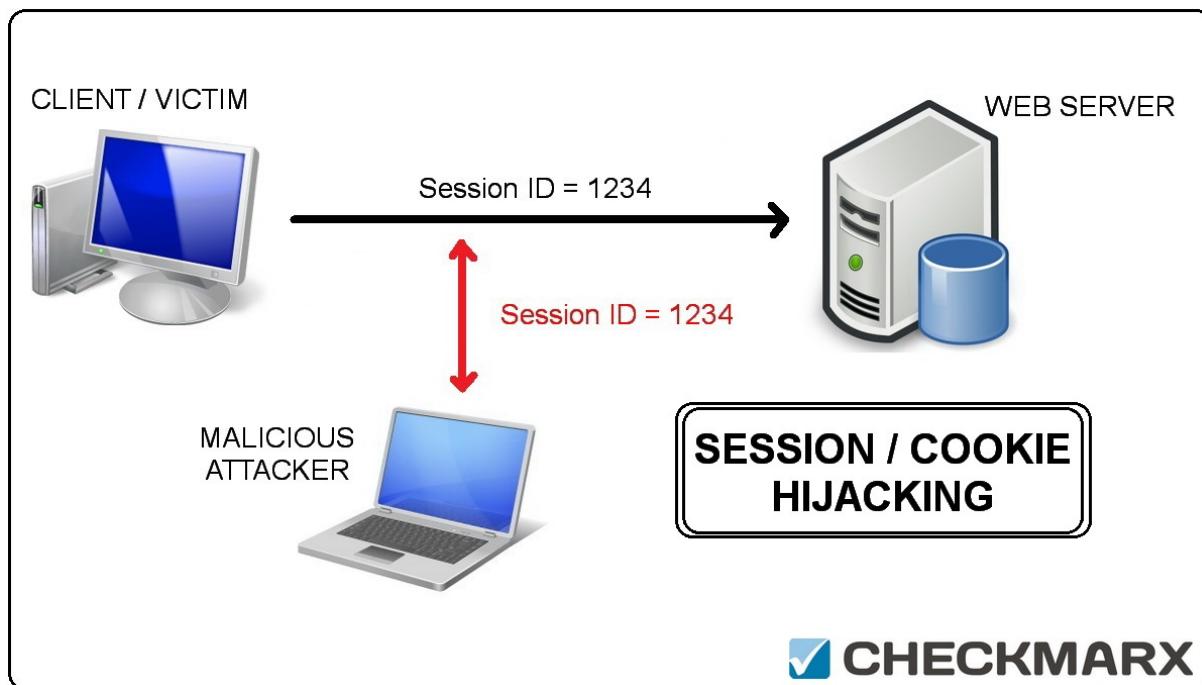


Figura 3.2: Funcionamiento visual de un Session Hijacking

Existen 2 métodos comunes para realizar este tipo de secuestros:

1. **Cross-Site Scripting**, el atacante engaña a la máquina del usuario para que ejecute un código malicioso, la máquina de la víctima confía en este código porque parece perteneciente al servidor. Este código permite al atacante robar las cookies o realizar cualquier otra operación.
2. **Session Sidejacking**, en este método el malo consigue robar la cookie de sesión monitorizando el tráfico de paquetes en una red. En la actualidad, la mayoría de las páginas Web utilizan un cifrado SSL/TLS para las páginas de inicio de sesión para evitar que los malos puedan ver datos sensibles como contraseñas o cuentas bancarias. Pero algunas de estas páginas dejan de utilizar el cifrado una vez el usuario se ha autenticado. Esto permite que cualquier atacante que pueda ver el tráfico de la red, tenga la posibilidad de interceptar los datos se mandan al servidor. Estos datos que son mandados al servidor, incluyen la cookie de sesión, por lo que un atacante podría suplantar la identidad del usuario incluso sin haber conseguido su contraseña.

Browser Hijacking

El secuestro de navegador es un tipo de malware que modifica la configuración del navegador de un ordenador, por lo que la víctima de este secuestro es redireccionada a Webs que no tenía intención de visitar.

La mayoría de los secuestradores de navegadores cambian las páginas de inicio

predeterminadas y las páginas de búsqueda a las de sus clientes, que pagan por ese servicio debido al tráfico que genera.

Domain Hijacking

El secuestro de dominio consiste en cambiar el registro de un dominio sin el permiso de su propietario original. Generalmente este secuestro se realiza de dos maneras: mediante la explotación de una vulnerabilidad en un sistema encargado de los dominios o mediante ingeniería social.

La táctica más usada por los atacantes es recabar información personal de la víctima para después utilizarla haciéndose pasar por él e intentar persuadir al registrador del dominio para modificar la información del registro o conseguir transferir el dominio a otro registrador. Básicamente, se trata de un robo de identidad. Una vez conseguido el objetivo, el atacante consigue el control total del dominio.

Existen más métodos como por ejemplo: Keyloggers, phishing sites y vulnerabilidades en el nivel de registro de dominio.

Malware

El “Malicious Software” da el nombre a la palabra malware, en castellano “Código Malicioso”, este tipo de Software la gran mayoría de las veces es creado por los cibercriminales para ser ejecutado por los usuarios y poder así llevar a cabo cualquier tipo de ataque [14].

La mayoría de las veces, el malware accede a los dispositivos a través de Internet o del E-mail, aunque también puede conseguir acceder a través de sitios web comprometidos o de descargas de alguna página Web no confiable.

Los términos nombrados a continuación son tipos de malware que existen en la actualidad, es necesario que sean mencionados y tener conciencia de su funcionamiento para poder tomar medidas preventivas:

- **Virus informáticos:**

Son programas que una vez los ejecutamos dentro de nuestra máquina, se propagan infectando otros programas ejecutables. Estos virus muy a menudo realizan también acciones de carácter malicioso como corromper o destruir archivos. Para que estos virus se propaguen, el usuario tiene que haber realizado una acción antes, ya sea instalar algún software malicioso, abriendo correos o de alguna otra forma.

- **Troyanos:**

Son programas maliciosos ocultos en algo atractivo para el usuario para invitarle a ejecutarlo. El troyano una vez ejecutado toma el control de la máquina ejecutando acciones con fines indeseables.

■ Gusanos o Worms:

Son programas que se transmiten a sí mismos, explotando vulnerabilidades de una red para infectar otros equipos. Su principal objetivo es infectar a la mayor cantidad de usuarios posible, y pueden contener instrucciones dañinas.

■ Botnets:

Son redes de ordenadores infectados o zombis que pueden ser controladas a la vez por un individuo para realizar distintas tareas. Las botnets mayormente son utilizadas para el envío masivo de SPAM o para lanzar ataques de denegación de servicio a organizaciones como forma de extorsión.

■ Backdoors:

Son métodos para evitarlos procedimientos habituales de autenticación al conectarse a un ordenador, siendo más fácil para el malware permanecer oculto ante posibles inspecciones.

■ Rootkits:

Son técnicas que modifican el sistema operativo de un ordenador para permitir que el malware permanezca oculto al usuario.

■ Adware:

El adware es un software que despliega publicidad de distintos productos o servicios. Estas aplicaciones incluyen código adicional que muestra la publicidad en ventanas emergentes, o a través de una barra que aparece en la pantalla simulando ofrecer distintos servicios útiles para el usuario.

■ Hoax:

La traducción al castellano es “bulo”, como su nombre indica, se tratan de avisos falsos, de alertas o de alguna denuncia falsa distribuida por correo electrónico con el objetivo de ser reenviada lo máximo posible por la red y molestar a los usuarios.

■ Keylogger:

Este tipo de malware se encarga de capturar todo lo que la víctima ingresa por el teclado y almacenarlo en archivos con el fin de recoger contraseñas, cuentas bancarias o algún tipo de información sensible. Gran cantidad de los troyanos cuentan con keyloggers dentro de su código.

■ Phishing:

Consiste en la falsificación de un sitio web o una aplicación de confianza con el objetivo de que el usuario introduzca sus credenciales sin saber que en realidad está siendo víctima de un robo. Una vez el usuario introduce sus datos o contraseña, esta información es recibida por el atacante.

- **Rogue:**

Un rogue es un Software falso, este software no va a realizar ningún tipo de acción en nuestro PC de las que promete. Los más conocidos en la actualidad son los “falsos antivirus”, estos enviarán a las victimas falsos avisos de infección que únicamente pueden ser arreglados si se compra su versión de pago.

- **Spam:**

El spam es el envío masivo de correos electrónicos no solicitados por parte de terceros. También es identificado como correo no deseado o correo basura.

- **Spyware:**

Este software se encarga de recopilar información sobre un particular o una organización sin su consentimiento. Esta información es explotada por motivos de mercadotecnia.

A parte de estos, existen más tipos de malware que no han sido explicados ya que serán tratados más adelante, como por ejemplo, el Ransomware.

Ataques a sistemas criptográficos

En este tipo de ataques, el atacante busca romper el cifrado de un sistema para obtener el texto original que se esconde detrás del texto cifrado. Para cumplir su objetivo, el atacante debe obtener la clave encargada de descifrar el texto. Una vez la clave secreta es obtenida, el sistema está comprometido [15].

Existen varios métodos de ataque a este tipo de sistemas:

- **Ciphertext-only attack (COA) o Ataque solo a texto cifrado:**

El atacante tiene acceso a una parte del texto cifrado, este ataque es exitoso cuando el texto plano puede hallarse con la porción de texto cifrado. En algunas ocasiones, el atacante puede sacar la clave de cifrado desde el texto cifrado que posee.

- **Known-plaintext attack (KPA) o Ataque a texto plano conocido:**

En este caso, el atacante tiene acceso a parte del texto plano, por lo tanto, debe utilizar este texto para hallar el resto de texto que está cifrado.

- **Chosen-plaintext attack (CPA) o Ataque a texto plano elegido:**

En método, el atacante es capaz de elegir una parte del texto plano para ser cifrado y tener acceso al texto cifrado resultante. El objetivo del atacante es revelar todo o parte de la clave de cifrado secreta.

- **Ataque de diccionario:**

Este ataque se basa en la compilación de un diccionario creado por el atacante con las posibles claves para intentar explotar el sistema si la clave secreta coincide con alguna de las introducidas en el diccionario.

- **Ataque de fuerza bruta:**

La fuerza bruta trata de adivinar la clave secreta utilizando para ello todas las claves posibles hasta dar con ella. Un ejemplo, si la clave fuese un número de 3 cifras, aplicar la fuerza bruta consistiría en probar todas las posibilidades entre 000 y 999.

- **Side-channel attack (SCA) o Ataque del lado del canal:**

Estos ataques se caracterizan por basarse en la información obtenida de la implementación de un sistema criptográfico. En otras palabras, el atacante busca información de todo tipo acerca del sistema, como por ejemplo el consumo de energía, con el objetivo de usar esa información para explotar el sistema. Muchos de estos ataques se apoyan en métodos estadísticos.

Ataques a contraseñas

Para saber cómo protegernos de un ataque de estas características, debemos de familiarizarnos con los tipos más conocidos de ataques que podemos sufrir. En este mundo de la seguridad cuanta más información tengamos de como los malos pueden atacarnos, mejores decisiones podremos tomar de cómo defendernos. Por este motivo voy a explicar las técnicas más conocidas que los cibercriminales utilizan en la actualidad para robar contraseñas.

Estas técnicas se dividen en 4 grandes grupos: Ataques Online Pasivos, Ataques Online Activos, Ataques Offline y Ataques sin Técnica [16].

Ataques Online Pasivos

Son los más comunes, el atacante intenta robar la contraseña sin comunicarse con la cuenta de la víctima.

- **Man in the Middle:**

Esta técnica está explicada anteriormente. Básicamente, el atacante intercepta el tráfico de datos de una comunicación y en esta comunicación puede estar la contraseña.

- **Ataque de reproducción:**

Son ataques que interceptan paquetes de datos y los reproducen, es decir, los reenvían tal como están al servidor receptor. Por lo tanto, una vez enviados esos paquetes, el atacante tendría los mismos derechos que el usuario.

- **Wire Sniffing:**

Es uno de los ataques más típicos en redes cableadas o inalámbricas. La contraseña es capturada en la fase de autenticación y después se compara con un archivo de diccionario o a una lista de palabras.

Ataques Online Activos

La característica principal de estos ataques es que el atacante intenta adivinar la contraseña de la víctima. Para ello, trata de insertar numerosas contraseñas una a una hasta dar con la correcta.

- **Password Guessing o Adivinar la contraseña:**

Este ataque solo funciona con contraseñas débiles, ya que adivinar una contraseña fuerte conllevaría meses o años de cómputos. El atacante crea un diccionario de palabras con todo tipo de combinaciones que pueden ser utilizadas como contraseñas y se ayudan de programas que generen miles de palabras por segundo.

Ataques Offline

En los ataques sin conexión, el atacante necesita los archivos de contraseñas o el archivo de diccionario dentro de su computadora, para poder así romper las contraseñas sin necesidad de conectarse a la red.

- **Fuerza Bruta:**

Como se ha mencionado anteriormente, este ataque consiste en explotar la contraseña probando todo tipo de combinaciones. Como es lógico, este ataque es el que más tiempo consume y no será efectivo si se intenta crackear una contraseña fuerte.

- **Ataque de diccionario:**

El atacante confecciona un diccionario de posibles contraseñas, después este diccionario se parte formando diferentes combinaciones y va probando una a una hasta dar con la válida. Obviamente, si la contraseña no se encuentra dentro del diccionario el ataque fallará.

- **Ataque Híbrido:**

En estos ataques se empieza por un diccionario de palabras y posteriormente se van añadiendo o sustituyendo números o símbolos, para tener más posibilidades de adivinar una contraseña fuerte.

- **Pre-Computed Hash:**

Se utiliza cuando la contraseña está cifrada, el atacante tendrá como objetivo encontrar la función hash con la que se cifró la contraseña para así cifrar el diccionario con la misma función hash y poder así realizar un ataque de diccionario. Este ataque solo es posible si un archivo contiene la contraseña cifrada en un formato legible.

- **Rainbow Attack o Ataque arcoíris:**

Es un pequeño avance de la técnica previamente mencionada. Se utiliza para romper contraseñas que se han cifrado con un hash. Una vez que un atacante obtiene acceso a la base de datos de contraseñas del sistema, el cracker de contraseñas compara la lista precompilada de la tabla arco iris de hashes potenciales con las contraseñas hash de la base de datos.

Ataques sin técnica

Son ataques que pueden realizarse sin tener ningún conocimiento sobre métodos o técnicas de crackeo de contraseñas.

- **Ingeniería social:**

Se basa en el arte de interactuar con las personas para intentar que confíen en ti con el fin de que te den información sensible o incluso sus contraseñas directamente. Gran cantidad de ciberdelincuentes han conseguido llevar a cabo ataques por medio de esta técnica.

- **Shoulder Surfing:**

El atacante en este caso está muy cerca de la víctima y puede ver como escribe la contraseña.

- **Dumpster Diving:**

La víctima deja su contraseña escrita en algún papel o en alguna nota en el escritorio y el atacante la obtiene con solamente echarle un vistazo.

Ataques SSL

SSL (Secure Socket Layer) es la tecnología que se implementó para conseguir comunicaciones confidenciales y protegidas en la red con el fin de evitar que los ciberdelincuentes robaran información sensible de estas comunicaciones.

Como empresa o como usuario de Internet, se debe de entender cómo funciona este protocolo y sus requerimientos para poder configurarlo de la mejor manera posible.

Estas comunicaciones son muy seguras pero nada puede garantizar una seguridad del 100 % por que como todas las tecnologías puede ser vulnerable. Por consiguiente, si no se configuran ni renuevan estos certificados correctamente, cualquier usuario o entidad puede ser víctima de los 5 vectores de ataque que se nombran a continuación [17]:

- **Heartbleed:**

Este error se encuentra en las bibliotecas criptográficas OpenSSL. Los atacantes explotando este fallo tienen la posibilidad de obtener información personal de los clientes, contraseñas o incluso acceso a cuentas.

- **POODLE:**

El fallo ocurrió cuando en una versión de SSL se remitía el uso a otras versiones de TLS (Transport Layer Security) o a versiones de SSL antiguas. De este modo, se podían obtener datos guardados en cookies y descifrar con ello todo lo necesario.

- **FREAK:**

El error fue provocado por defectos en el Software de SSL permitiendo que se realizaran ataques Man in the Middle.

- **Shellshock:**

Es de los más comunes entre usuarios Linux con un 70

- **Bar Mitzvah:**

Esta vulnerabilidad afectaba al algoritmo de cifrado RC4, dejando que los malos pudiesen interceptar credenciales en las comunicaciones. Hoy en día es muy difícil que se dé este caso ya que surgió el cifrado Advanced Encryption Standard (AES) para dejar RC4 obsoleto.

Ataques DNS

Antes de entrar de lleno en los tipos de ataques que pueden existir, se explicará en qué consiste un servidor DNS (Domain Name Service) y lo importante que es en Internet.

El sistema de nombres de dominio se encarga de traducir las direcciones IP en nombres de dominio. Gracias a su trabajo, los usuarios no tenemos que acordarnos de direcciones IP y podemos acceder a cualquier Web escribiendo su nombre de dominio [18].

El funcionamiento de estos servidores se muestra en la figura 3.3 [19]:

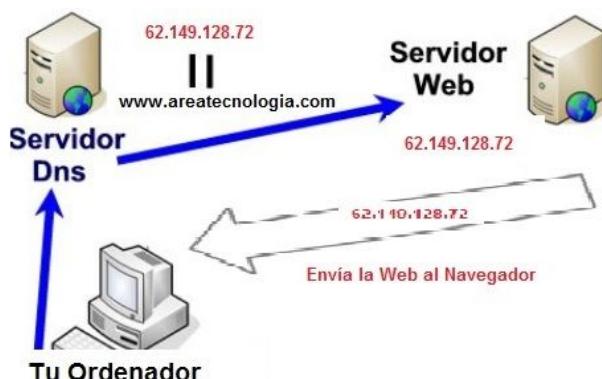


Figura 3.3: Funcionamiento de un servidor DNS

Existen varios tipos de ataques a los nombres de dominio o DNS, pero voy a resumirlos en dos bloques: Ataques no dirigidos específicamente a DNS y Ataques dirigidos a DNS [20].

Ataques no dirigidos específicamente a DNS

Estos ataques están más enfocados a explotar la parte más administrativa de los nombres de dominio en lugar de a las infraestructuras o servidores.

- **Cibersquatting o Ciberocupación:**

Esta técnica consiste en registrar un nombre de dominio de manera malintencionada con el objetivo de sacar algún beneficio de la víctima o para causarle algún daño directamente.

- **"Name-jacking":**

El atacante roba el dominio apropiándose de él directamente o por medio técnicas como modificar los servidores que hostean el sitio.

- **Ataques sociales:**

Como he mencionado anteriormente, estos ataques se basan en engañar a personas haciendo que confíen en el atacante para darle información personal.

Ataques dirigidos a DNS

Estos ataques sí que están enfocados hacia las infraestructuras y se necesita de conocimiento de técnicas para poder explotarlos.

- **DNS Cache Poisoning [21]:**

El servidor DNS del usuario, envía una petición a un DNS autoritativo preguntando por la IP del nombre solicitado. El atacante en este momento suplanta al DNS autoritativo y devuelve una IP falsa. Entonces, la petición del usuario finalmente será dirigida a un servidor que el usuario no quiere acceder. Además, la cache de este servidor DNS será envenenada albergando en ella la entrada falsa obtenida para darle la misma respuesta a todos los usuarios que quieran acceder al mismo nombre.

- **DOS o DDOS:**

Este tipo de ataques ya están explicados anteriormente. Tratan de dejar un servidor sin servicio.

- **Reflected Attacks:**

Este ataque consiste en enviar miles de solicitudes desde la dirección de la víctima, para que al recibir de vuelta todas las respuestas la infraestructura se vea afectada por la sobrecarga.

- **Fast Flux:**

En estos ataques los atacantes tratan de ocultar su identidad utilizando técnicas que cambian rápidamente la información sobre su localización para ocultar de donde viene el ataque.

Ransomware

Probablemente, el tipo de ataque del que más se habla en la actualidad después de que Wanna Cry (un tipo de ransomware) afectara a medio mundo y mostrando que la Seguridad Informática es una necesidad.

El ransomware es un tipo de Software malicioso que bloquea datos o equipos informáticos hasta que recibe el pago que solicita como rescate. En otras palabras, un secuestro del ordenador [22].

Se pueden dividir los tipos de este Software malicioso en dos bloques: El ransomware de cifrado y el ransomware de bloqueo.

- **Ransomware de cifrado:**

En este tipo, los datos del usuario son cifrados con algún mecanismo.

Una vez queda el ordenador infectado, este malware busca por todas las unidades accesibles ficheros con extensiones conocidas con el objetivo de cifrarlas. Una vez ha terminado, muestra al usuario un mensaje indicándole que su ordenador está infectado y el rescate que debe abonar.

Realizado el pago, el secuestrador puede o no, darle las claves necesarias para descifrar de nuevo sus archivos.

- **Ransomware de bloqueo:**

Este malware, impide el uso del dispositivo, hasta que se efectúa el pago del rescate.

Una de las técnicas de manipulación utilizadas para que se proceda con el pago, es utilizar una pantalla en la que se indica al usuario que ha cometido algún delito y que por tanto debe pagar una multa.

Según se ha ido perfeccionando esta técnica, han conseguido que el mensaje pueda adaptarse a diferentes países, utilizando el idioma local y presentando imágenes identificativas de las fuerzas del estado competentes, así como, la imagen de la víctima capturada con la webcam en caso de disponer de ella.

3.2. Vulnerabilidades en la aplicación

Cross Site Scripting (XSS)

El XSS es un tipo de ataque en el cual el ataque tiene la capacidad de injectar scripts en el output de una Web de manera que se ejecuta en el navegador del usuario.

Normalmente se debe a la falta de los controles necesarios en la página, pensados para evitar la ejecución de comandos desde la misma página web [23].

Esta vulnerabilidad puede conllevar unos daños muy elevados ya que si un ciberdelincuente es capaz de realizar inyecciones de scripts en una Web, va a ser capaz de ejecutar código malicioso en los navegadores de los usuarios. Los objetivos más comunes de estos ataques son los de robar datos sensibles, redireccionar al usuario a un sitio comprometido o instalar el malware.

Es importante tener en cuenta que el atacante utiliza la confianza que tienen los usuarios en un sitio Web particular [24].

Los ataques XSS se dividen en dos grupos dependiendo de cómo envíen el código: XSS no persistente o XSS reflejado y XSS persistente.

- **XSS reflejado:**

En este tipo el atacante envía directamente el código a la víctima en vez de almacenarlo en el servidor. El ejemplo más común es hacer que un navegador ejecute algún mensaje de alerta para intentar que la víctima ejecute un comando determinado dentro de la página. Una vez logrado el objetivo, el atacante puede robar las cookies para después robar la identidad de la víctima.

- **XSS persistente:**

En este caso, el malware ya está validado y almacenado. Este código puede ser cualquier tipo de sección del sitio web que solicita una entrada de datos al usuario. El código es ejecutado automáticamente cuando esta la información donde está almacenado el malware se presenta en la página web.

Inyección SQL

En la actualidad, la mayoría de las páginas web utilizan bases de datos. La manera de interactuar con éstas para que realicen las acciones necesarias es por medio del lenguaje SQL creado para interactuar con las bases de datos.

A raíz de este lenguaje apareció una de las vulnerabilidades más conocidas en el mundo de la seguridad informática, la inyección de código SQL. Esta vulnerabilidad consiste en la inyección de código en los datos de entrada del lado del cliente en el sitio web. Explicado de otra manera, el atacante puede modificar las consultas que realiza la operación y ejecutar otras diferentes por medio de la inserción de código. La finalidad es acceder a la herramienta y poder robar información o alterar la integridad de la base de datos.

Si el administrador de la base de datos no ha asignado adecuadamente los privilegios de los usuarios, un atacante podría entrar en las bases de datos con privilegios que le podrían permitir no solo acceder a la base de datos de la aplicación si no que podría acceder a cualquier tabla que estuviese dentro del servidor, lo que lo hace altamente peligroso.

Lo comentado anteriormente es posible gracias a que el uso de ciertos caracteres en los campos de entrada de información por parte del usuario, ya sea mediante el uso de los campos de los formularios que son enviados al servidor mediante POST o bien por medio de los datos enviados mediante GET en las urls de las páginas web, posibilitan coordinar varias consultas SQL o ignorar el resto de la consulta, permitiendo al atacante ejecutar la consulta que elija [25].

Dentro de esta vulnerabilidad existe un tipo de ataque muy conocido llamado Blind SQL injection. El ataque a ciegas por inyección SQL ocurre cuando una página web no muestra mensajes de error cuando una consulta a la base de datos no es válida o es incorrecta, lo que es un fallo de seguridad.

Esta técnica se utiliza combinada con técnicas de fuerza bruta o diccionarios para conseguir obtener la contraseña o usuario que esté dentro de la base de datos atacada. Para que esto suceda se utiliza código SQL encargado de comprobar carácter por carácter consiguiendo acumular uno a uno los resultados positivos. Es decir, va adivinando la contraseña carácter a carácter hasta tenerla completa y poder entrar así a la base de datos.

En la figura 3.4 [26] se explica cómo funciona la consulta más famosa de SQL injection:

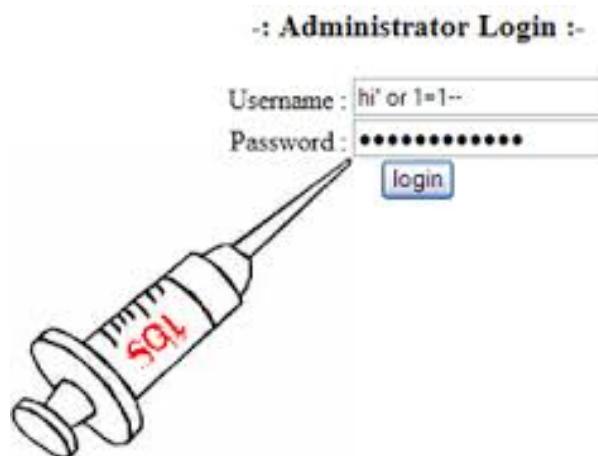


Figura 3.4: Ejemplo de una inyección SQL

La inyección SQL con la famosa sentencia 'or '1'='1 se produce cuando el atacante recibe un formulario de ingreso de datos en alguna página web.

La consulta de la base de datos tiene que ser algo similar a esto:

```
Select id  
from usuarios  
where usuario='$usuario' and pass='$pass';
```

Si la consulta está mal construida, es decir, concatenada, si el atacante introduce la sentencia famosa en el usuario o contraseña o en las dos, accederá al sistema.

```
Select id
```

```
from usuarios  
where usuario='admin' and pass="or '1'='1";
```

Aunque parezca una vulnerabilidad fácil de corregir, hoy en día se pueden encontrar cantidad de sitios web vulnerables a esta sentencia.

Inyección LDAP

Estos ataques se parecen de alguna forma a las inyecciones de SQL, ya que el acceso de LDAP suele ser muy parecido al acceso a alguna base de datos. En este caso, el atacante no ataca un servidor SQL, sino que ataca a un sistema de validación de usuarios [27].

Antes de continuar es necesario conocer el protocolo LDAP (Lightweight Directory Access Protocol). El Protocolo Ligero de Acceso al Directorio se encarga de gestionar las listas de control de acceso de un dominio determinado. El protocolo LDAP se ejecuta sobre protocolos de transporte de Internet, como TCP. Las aplicaciones web pueden usar datos introducidos por el usuario para crear sentencias LDAP para peticiones de páginas web dinámicas.

El ataque tiene éxito cuando la aplicación Web no tiene seguridad establecida sobre los datos que establece el usuario, por lo tanto, el cibercriminal puede alterar las sentencias LDAP provocando un proceso que se ejecutará con los mismos permisos del componente que ejecutó el comando. Puedes imaginarte el riesgo de estos ataques si un atacante consigue apoderarse de privilegios para realizar cualquier tipo de acción incluso la de borrado de elementos del árbol de LDAP.

Almacenamiento Criptográfico Inseguro

Este tipo de vulnerabilidad se encuentra entre una de las 10 más peligrosas dentro de Open Web Application Security Project (OWASP). Por lo que debemos de tomarla muy en cuenta a la hora de mirar por la seguridad de nuestra empresa o nuestras aplicaciones [28].

Esta vulnerabilidad ocurre cuando una aplicación no cifra adecuadamente sus datos sensibles, no tiene nada que ver con la comunicación de los datos, únicamente su almacenamiento.

El fallo de seguridad se da en los siguientes casos:

1. Los desarrolladores no cifran los datos que son almacenados en la base de datos.
2. Los desarrolladores cifran los datos con un cifrado local. Esto es, los desarrolladores deciden crear sus propios métodos de cifrado en vez de utilizar algunos métodos ya desarrollados que son de toda confianza.

Como la mayoría de las vulnerabilidades, el almacenamiento criptográfico inseguro puede darse debido a varias causas que voy a explicar a continuación:

- **Falta de cifrado:** Si un atacante consigue acceder a los datos almacenados en una aplicación y estos datos no están cifrados, puede obtener todos los datos en cuestión de segundos.
- **Uso de Hashes débiles:** Este es otro de los fallos más comunes. Se trata de cifrar los datos pero utilizar para ello un algoritmo débil que puede ser explotado en poco tiempo.
- **Gestionar las claves de manera insegura:** En este caso el cifrado se ha realizado con un algoritmo fuerte pero el manejo de las claves secretas es débil. Es como si blindamos una puerta pero dejamos una copia de la llave debajo del felpudo.
- **Almacenamiento de datos innecesarios:** Se refiere al almacenamiento de datos sensibles de los clientes como pueden ser cuentas bancarias, en las bases de datos de una aplicación web. En caso de que un atacante consiga acceder a los datos, puede quedarse con toda la información.

Desbordamiento del Buffer

Esta vulnerabilidad ocurre cuando un programa informático supera el uso de memoria asignada en un área de memoria y los bits que sobran se escriben sobre zonas de memoria adyacentes, sobrescribiendo lo que estas memorias contenían [29].

Cuando un programa no controla la cantidad de datos que se van a copiar en la memoria, si el atacante envía un caudal de datos mayor de los que puede recibir, el búfer que es el encargado de almacenar datos temporalmente, se desborda. Además, si el programa atacado tiene alguna función que no permite el desbordamiento, el programa puede bloquearse permitiendo al atacante ejecutar código para acceder al sistema.

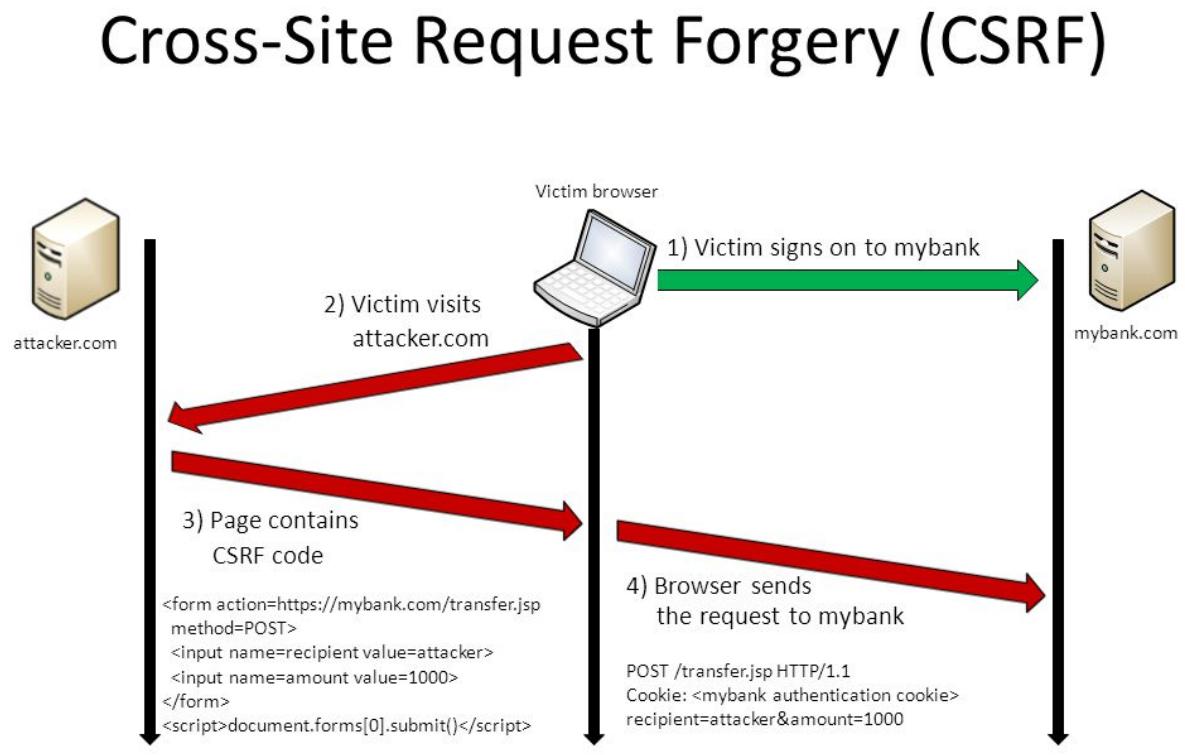
Cross Site Request Forgery (CSRF) o Falsificación de Petición en Sitios Cruzados

Este ataque se basa en la confianza que una aplicación web tiene en el usuario que va a ser víctima del ataque, al contrario que en las vulnerabilidad XSS. El atacante va a forzar al navegador de la víctima a enviar una petición a una web vulnerable. Seguidamente, el atacante, sin que la víctima sea consciente de ello, aprovechándose de la confianza de la web, va a poder ejecutar scripts para que la víctima los ejecute [30].

Un ejemplo para que quede más claro: La víctima esta navegando y tiene abierta una aplicación web donde está autenticado, a su vez, en otra pestaña, sin saberlo,

podría estar ejecutando sin saberlo código destinado a un ataque CSRF contra esa aplicación. Como la aplicación donde está autenticado no puede distinguir si las peticiones que le llegan son de la víctima o del navegador, el atacante puede ejecutar acciones dentro de la aplicación sin que la víctima se entere.

En la figura 3.5 [31] se contempla el funcionamiento de una falsificación de petición en sitios cruzados:



39

Figura 3.5: Funcionamiento de un CSRF

Fallo de Restricción de Acceso a la URL

Es una de las vulnerabilidades más fáciles de explotar porque requieren poco conocimiento. Básicamente, esta vulnerabilidad consiste en acceder a URL que deberían ser seguras.

Esto se debe a que ha habido un fallo en la administración de la autorización y autenticación. Lo que ocasiona que los niveles de acceso que cada usuario dependiendo del rol que tengan deberían tener no existan, por lo tanto, un trabajador normal del portal podría tener el mismo acceso que un administrador. Este mal control podría ocasionar que un simple usuario escribiendo directamente la URL podrá acceder sin problemas a cualquier información [32].

También se ha de comentar que en algunas ocasiones estos fallos se producen por una mala programación, por lo que, podría haberse arreglado si el trabajador hubiese estado más concentrado o hubiese tenido los conocimientos necesarios.

Referencia Directa a Objetos Insegura

Estas vulnerabilidades permiten a los cibercriminales tener acceso a referencias de la aplicación para tener acceso a objetos privados. En otras palabras, si una página web tiene esta vulnerabilidad, un atacante por medio de modificar la URL puede tener acceso a cualquier información alojada en esta aplicación [33].

Imaginar que un atacante accede a una página web desde la cual quiere descargar un PDF, si este ataque modifica esa URL podría hacer que la aplicación en vez de devolverle ese PDF descargable le devolviese cualquier otro archivo de configuración de la aplicación o del sistema operativo.

Pérdida de Autenticación y Gestión de Sesiones

La pérdida de autenticación y gestión de sesiones permite a un atacante suplantar a un usuario, lo que conlleva que este atacante pueda administrar completamente esa cuenta pudiendo saltarse los controles de autorización. Esto permite que el atacante pueda llegar a obtener toda la información que esté en el servidor comprometido [34].

Existen varias situaciones en las que una aplicación tiene esta vulnerabilidad, pero en general, la gestión de las contraseñas y los procesos de recuperación de las mismas, la expiración de las sesiones o el proceso de cierre de las mismas son las más comunes.

Redirecciones y Reenvíos no validados

Esta vulnerabilidad es realmente simple. En el mundo de las comunicaciones las páginas web interactúan entre sí obligando al usuario a ser redireccionado entre varios sitios web. Si no existe una validación confiable, el atacante puede redirigir a sus víctimas a una página web que esté comprometida [35].

3.3. Vulnerabilidades en el sistema

Software mal configurado

Cualquier sistema por muy robusto que parezca puede estar mal configurado. Esto puede deberse a intentar construir el sistema lo más rápido posible sin conocer muy bien el software utilizado o por tener demasiada confianza en este software. La mala

configuración suele ser responsabilidad del usuario aunque por defecto, algunos programas vienen más seguros que otros. Cuando se expone a Internet un sistema con una configuración por defecto, si un atacante consigue obtener una vulnerabilidad de nuestro tipo de sistema, cualquier exploit por defecto funcionará.

Software Desactualizado

Actualizar el software es muy importante, debemos mentalizarnos de que cuando aparecen en el equipo mensajes de alguna aplicación que necesita ser actualizada, se debe instalar esa actualización lo antes posible para evitar riesgos de seguridad.

Las actualizaciones realizan muchas tareas, son necesarias por proporcionar revisiones para nuestro equipo, introducir nuevas funciones, actualizar drivers... Pero lo más importante es que eliminan funciones obsoletas, corren errores y reparan las vulnerabilidades que han sido detectadas [36].

Si no se actualiza el software se deja a propósito un agujero de seguridad en la máquina que cualquier atacante puede utilizar para introducir cualquier tipo de malware y provocar pérdidas.

Ausencia de Copias de Seguridad

Cuando una empresa no dispone de un buen sistema de copias de seguridad, esta exponiéndose a un riesgo enorme de perder un porcentaje elevado de sus datos.

Si por un fenómeno natural o un ataque masivo, dicha empresa sufre una desconexión o la incapacitación total de algunos equipos, los datos albergados en ellos desaparecerían.

Por lo tanto, para que una empresa no pierda todos sus datos si ocurre un incidente, deberá tener unas políticas de copias de seguridad perfectamente establecidas, realizando guardados de datos diarios, semanales y mensuales, almacenando estos datos en sitios seguros realizando varias copias y con alguna copia fuera del lugar físico donde se encuentra la empresa.

Control de Cuentas de Usuario

Toda empresa que quiera tener un buen funcionamiento en Internet, debe establecer una política estricta en el control de las cuentas de usuario. Es decir, todas las personas que trabajan en un sistema web van a tener diferentes roles dentro de él, por lo tanto, deberán tener acceso solo a las partes que necesiten.

Todos los trabajadores deben comprometerse en seguir estas políticas y debe haber supervisores que se aseguren del cumplimiento de las mismas.

Estas políticas deben aclarar perfectamente todo lo que cualquier empleado debe hacer tanto en la construcción de una contraseña segura, la confidencialidad de la

misma y la necesidad de ser modificada periódicamente, como se debe actuar en caso de detectar comportamientos extraños dentro del sistema...

Si la empresa logra instalar unas políticas de este tipo conseguirá ser más robusta y tener menos vulnerabilidades.

Ausencia de Seguridad en Archivos Digitales y Físicos

Este riesgo es muy sencillo de solucionar y es muy común en empresas con poca seguridad. Básicamente consiste en establecer un grado más de seguridad en archivos con información sensible sobre la empresa o clientes. Es decir, una empresa no puede permitirse guardar información muy importante en texto plano o no puede permitirse tener algún tipo de archivo físico como puede ser unos folios donde se resume la nueva aplicación que quiere fabricar la empresa a la vista de cualquiera.

Los datos digitales deben ser cifrados para evitar un robo de información si sufrimos cualquier ataque y los archivos físicos deben ser guardados en lugares seguros por la misma razón.

Cualquier nivel de seguridad que una empresa pueda establecer siempre será beneficioso para la misma.

Desconocimiento de Políticas por parte de los responsables de Informática

Esto se produce cuando aparece algún error grave en la gestión de políticas instaladas en una organización, pueden ser de seguridad, de gestión de algún sistema, etc.

Cuando esto ocurre se produce un desconcierto en la empresa porque nadie sabe cuál es el camino correcto a seguir y se necesita urgentemente tomar medidas porque estamos expuestos a cualquier tipo de ataques.

Para evitar estos riesgos, los responsables de informática deben estar perfectamente formados y para ello la empresa va a tener que ofrecerles esa formación, el dinero estará perfectamente invertido ya que esta formación hará que estos responsables sepan instalar un buen sistema de políticas y guiar a los empleados que no tengan conocimientos sobre esta rama.

Por otra parte, cualquier tipo de empleado debe estar familiarizado con las políticas que una empresa desea seguir para que no haya fallos de ningún tipo.

Centro de cómputo sin UPS

En primer lugar, un UPS (Uninterruptible Power Supply) o SAI (Sistema de alimentación ininterrumpida) es una fuente de suministro eléctrico que posee una batería con el fin de seguir suministrando electricidad aunque se produzca una interrupción

eléctrica [37].

Como es lógico esto es muy importante en una empresa ya que si ocurre cualquier fallo eléctrico los ordenadores seguirán funcionando y no se apagarán de golpe perdiendo todo aquello que se estaba procesando o desarrollando en ese momento. El SAI tiene poca autonomía pero la suficiente para poder terminar lo que se estaba haciendo y poder guardar todo correctamente para continuar cuando la electricidad se establezca de nuevo.

Pantallas sin bloqueo por el usuario

Este tipo de riesgo también es bastante sencillo de arreglar y de entender. Todos los empleados de una empresa que trabajan con un ordenador tienen que estar obligados a configurar el bloqueo de pantalla para que se bloquee cuando este el ordenador mucho tiempo inactivo o para cuando el usuario lo requiera sin necesidad de apagar el ordenador del todo.

Estos bloqueos son necesarios para que nadie más aparte del usuario pueda acceder al ordenador y poder ver los datos que están guardados dentro del mismo.

Contraseñas débiles o por defecto

El problema de las contraseñas débiles o por defecto es que cualquier atacante que intenta adivinarla, va a conseguirlo en poco tiempo.

Una contraseña debe ser personal y confidencial, no sirve de nada tener una contraseña fuerte si la está apuntada en algún papel a la vista de cualquiera o guardadas en sitios donde cualquiera puede obtenerla [38].

Lo primero que se debe hacer es evitar tener contraseñas por defecto, contraseñas muy frecuentes como 1234 o palabras que puedan aparecer en algún diccionario, con esto se puede evitar que algún atacante que utilice fuerza bruta o ataques de diccionarios pueda romper la contraseña en cuestión de minutos.

También se debe evitar que la contraseña sea algún dato personal, como por ejemplo el apellido, fecha de nacimiento, etc. Un atacante por medio de ingeniería social podría obtener esta información y crear un diccionario para realizar un ataque.

Para crear una contraseña fuerte podemos utilizar algún programa que genere este tipo de contraseñas o crear una nueva sabiendo que debe tener más de 8 caracteres y debe combinar mayúsculas, minúsculas y números.

Ingeniería Social

Este ataque es el más peligroso dentro de la seguridad informática. El 97 % de los ataques utilizan técnicas de este tipo para conseguir credenciales y poder vulnerar los sistemas [39].

Por este motivo, aun teniendo construido el sistema más seguro del mundo si cualquier empleado que no sabe nada de seguridad está haciendo un uso inadecuado de sus claves, no va a valer de nada.

La ingeniería social es el método de manipular psicológicamente a las personas para que comparten información personal que le pide el atacante o ejecuten algún tipo de acción que les pida el atacante. Los ataques suelen hacerse por medio telefónico de E-mail. Se trata de un tema absolutamente humano, el ciberdelincuente se hace pasar por un ente de confianza para la víctima y aprovechándose de sus pocos conocimientos y de su buena voluntad convencen a la víctima para extraer información o realizar alguna acción.

Muchos de los hackers más conocidos consiguieron romper cantidad de sistemas incluyendo algunos de corporaciones muy conocidas utilizando esta técnica. Consiguieron acceder al sistema por medio de la técnica que requería menos conocimiento de técnicas de hacking ético, únicamente hacer que un trabajador te de acceso a su usuario.

Conclusiones aplicadas al prototipo

En este apartado señalar las vulnerabilidades más importantes que afectan a nuestra aplicación.

CAPÍTULO 4

ANÁLISIS DE ALTERNATIVAS

En este apartado del proyecto, se realizará un estudio de las diferentes tecnologías o componentes que pueden utilizarse hoy en día para realizar aplicaciones relacionadas con el caso de estudio que se realizará posteriormente. Como se ha mencionado en la descripción del proyecto, el caso de estudio estará basado en aplicaciones del sector del IoT o la nueva industria 4.0.

4.1. Placa Base

Para empezar, se hará un estudio de los diferentes tipos de placas base de bajos recursos que realizarán la mayor parte de las tareas. Este dispositivo será el encargado de recibir la información de los sensores, y posteriormente, enviarlos a un ordenador que los recogerá.

La tabla 4.1 muestra la comparación entre los diferentes tipos de placas base que a día de hoy lideran el mercado, se valorará cada opción entre 0 y 5 puntos, 0 si consideramos que la opción es muy insatisfactoria y 5 muy satisfactoria [40] [41] [42] [43].

	Procesador	Memoria	Periféricos disponibles	Precio	Información disponible	Curva de aprendizaje
Raspberry Pi 3 Model B	3	3	5	4	5	5
JaguarBoard	3	3	3	4	3	3
BeagleBone Black	2	2	4	3	3	4
Nano Pi Neo Plus 2	3	3	3	5	2	3

Cuadro 4.1: Tabla comparativa de placas base

	Total
Raspberry Pi 3 Model B	25
JaguarBoard	19
BeagleBone Black	18
Nano Pi Neo Plus 2	19

Cuadro 4.2: Tabla total comparativa de placas base

Como se puede observar en la tabla 4.2, todas las placas están igualadas respecto a procesamiento.

En cuanto a la versatilidad de las placas y los periféricos que tienen disponibles para trabajar con ellas las Raspberry Pi tiene una clara diferencia sobre el resto. En Internet pueden encontrarse cantidad de trabajos hechos con este tipo de placas base, desde una consola de videojuegos hasta impresoras 3D.

Un punto muy importante es el precio, una empresa buscará la mejor calidad-precio para obtener mayor margen de beneficios.

Por último y uno de los puntos en los que apoyarse para tomar la decisión de qué placa utilizar es la información al respecto que puede encontrarse en Internet de cada uno de los dispositivos. Con mucha diferencia la ganadora es Raspberry Pi ya que en Internet pueden encontrarse cantidad de manuales o guías para sacarle el máximo partido. Se ha formado una comunidad enorme alrededor de esta marca de dispositivos lo que aumentó drásticamente la cantidad de información acerca de ellos.

4.2. Servidor Web

Seleccionar un servidor Web puede ser una decisión muy difícil, ya que una vez se elija el servidor que se va a utilizar, lo más probable es que se mantenga ese servidor durante mucho tiempo.

Entre los servidores de código abierto, dos son los más utilizados en los entornos Web con clara diferencia respecto al resto. Estos servidores son Apache y Nginx. Aparte de estos dos existen otros servidores como LiteSpeed, Microsoft-IIS o Node.js que son usados en la actualidad.

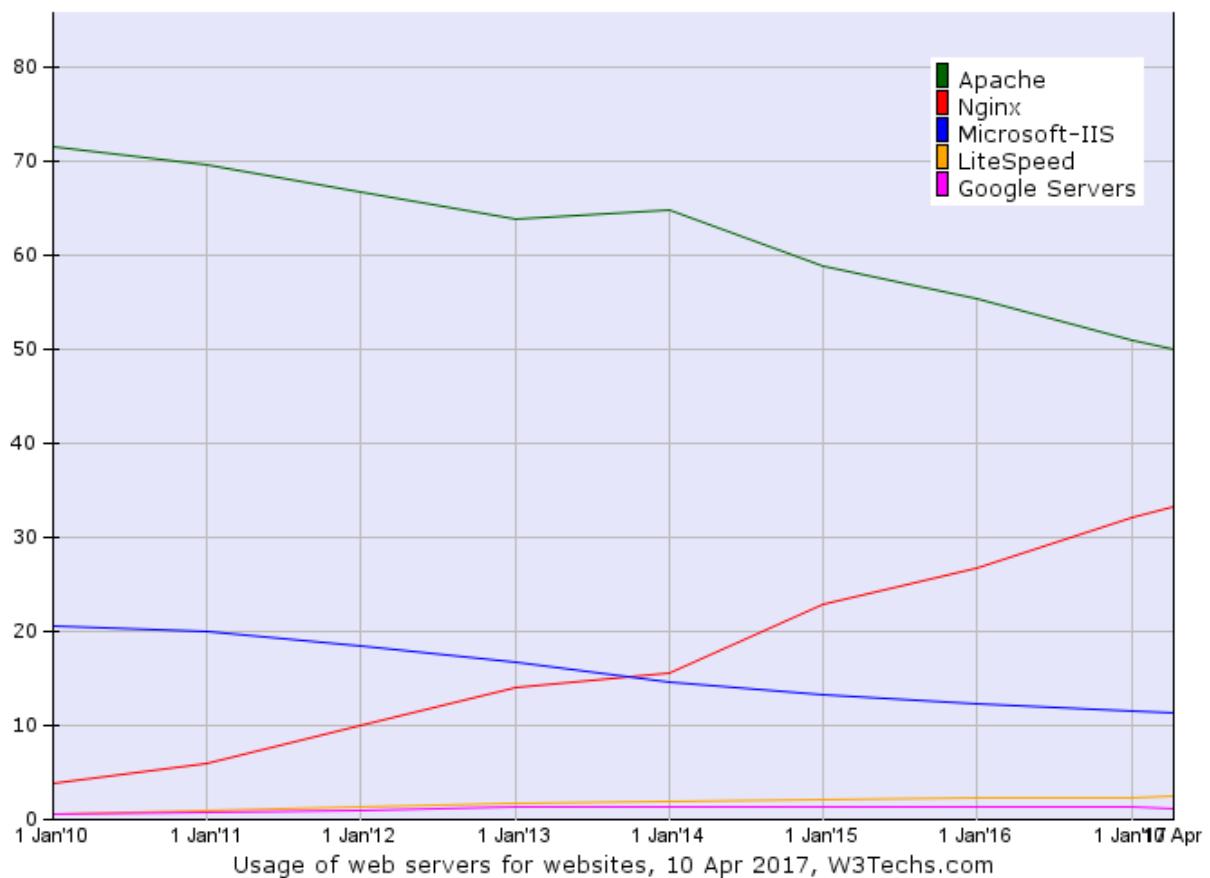


Figura 4.1: Uso de servidores en 2017

En la figura 4.1 [44] se puede observar el uso de servidores en 2017.

Para poder tomar la decisión se reducirán las opciones a las tres siguientes: APACHE, Nginx y Node-RED basado en Node.js.

APACHE

El servidor HTTP de Apache¹ es una aplicación de servidor Web de código abierto, por lo que los usuarios que trabajen con él pueden editar el código para adaptarlo a sus necesidades y ayudar al desarrollo del mismo. Tiene una comunidad detrás de usuarios que están dispuestos a dar soporte y arreglos de errores.

Un dato positivo es que este tipo de servidor funciona en la mayor parte de sistemas operativos.

Apache es más lento que su rival Nginx, aunque en realidad, Apache tiene una carga y tiempo de actualización ligeramente mejor para las páginas dinámicas. En cuanto a características que puede soportar el servidor, Apache es la claro ganador. Muchas de estas características se implementan como módulos que amplían la funcionalidad principal.

¹<https://www.apache.org/>

Algunos de los módulos más importantes son los siguientes [44]:

- Soporte del lenguaje de programación del lado del servidor
- Módulos de autenticación
- Módulo de soporte SSL/TLS
- Excelente seguridad
- Módulos Proxy
- Hosting Virtual

Nginx

Al igual que Apache se trata de un servidor Web de código abierto y al igual que Apache tiene una gran comunidad detrás que le ayudan a desarrollarse, aunque esta comunidad no es tan grande como la de su competencia. Al ser un servidor más nuevo, tiene menos soporte y menos información que Apache.

El punto principal de venta de Nginx² es que es un servidor Web asíncrono, es decir, Nginx no crea hilos separados por cada proceso con los problemas que esto puede causar cuando se tienen demasiados procesos activos al mismo tiempo. En su lugar utiliza un subproceso maestro y procesos dentro de un hilo principal. De esta forma, cada trabajador se asigna a un evento en vez de a un cliente en particular, estos eventos pueden ser una solicitud de recursos o un procesamiento de un cliente. En consecuencia, un trabajador puede atender a múltiples clientes al mismo tiempo, haciendo que este servidor sea muy eficiente para sitios Web con mucho tráfico.

En cuanto al rendimiento, Nginx supera a Apache con diferencia. Pruebas independientes informan que Nginx puede manejar aproximadamente un 40 % más de tráfico en 300 % menos de tiempo. Esto hace que Nginx sea aproximadamente 4.2 veces más rápido que Apache.

En la tabla 4.3 se describen las principales diferencias entre ambos servidores [45]:

²<https://nginx.org/en/>

	APACHE	NGINX
Sistemas Operativos	Linux, Unix, Windows, Mac OS	Linux, Unix, Windows-parcial, Mac OS
Soporte y soluciones de usuarios	Soporte corporativo y comunidad de usuarios	Comunidad de usuarios
Coste y Desarrollo	Gratis, código abierto	Gratis, código abierto
Seguridad	Excelente	Muy buena
Arquitectura	Flexible a través de módulos, puede estar basado en hilos, procesos o eventos	Arquitectura basada en eventos. Procesamiento asíncrono
Rendimiento	El uso de memoria y CPU es bueno, pero depende de la configuración y selección de módulos. Puede fallar cuando existe alto tráfico	Mejora el rendimiento considerablemente. Debido a la eficiencia de su algoritmo y arquitectura consume menos recursos
Contenido estático/-dinámico	Apache es flexible y puede manejar contenido estático y dinámico sin problemas mediante el uso de MPMs(Módulos de multiprocesamiento)	Nginx maneja el contenido estático muy rápidamente debido a su arquitectura. Para manejar contenido dinámico debe usar un componente externo
Características y prestaciones	Soporte para lenguajes de programación del lado del servidor; soporte para Perl, Python y PHP; módulos útiles de autenticación; soporte para SSL/TLS; módulos para proxy; archivos log personalizables; soporte de filtros; reescritura de URL; hosting virtual; métodos de compresión como gzip; entre otras	Sirve archivos estáticos; soporte SSL/TLS; proxy inverso; balanceo de carga; compresión; control de acceso; reescritura de URL; registro de logs personalizado; hosting virtual; entre otros

Cuadro 4.3: Tabla comparativa de servidores Web

Node-RED basado en Node.js

El auge del lenguaje de programación JavaScript ha propiciado que tecnologías como Angular.js, la base da datos Mongo DB o en este caso Node.js³ hallan vivido un crecimiento enorme en los últimos años.

Node Es un entorno de programación en JavaScript para el Backend basado en el

³<https://nodejs.org/es/>

motor V8 de JavaScript del navegador Google Chrome. Posee una gran comunidad de desarrolladores que están logrando hacer este entorno más rápido y más robusto.

Apache es el servidor más común para crear servidores hoy en día, Apache Tomcat se utiliza para aplicaciones JAVA en la web. El problema de Apache es la creación de hilos simultáneos para cada conexión hasta que la página no soporta más y se desborda, Node.js crea un sólo hilo que permanece a la espera de los posibles eventos.

Node puede ser una gran opción a tener en cuenta a pesar de ser menos popular y menos utilizada que APACHE y Nginx, debido a su sencillez de uso, a los frameworks que posee y al alto grado de escalabilidad y de integración que posee si se combina con tecnologías como Angular.js, MongoDB y Express.js formando así el denominado MEAN full stack.

Sabiendo que la aplicación que se va a desarrollar se trata de una aplicación del área del IoT, la inclusión en el proyecto de la tecnología Node-RED es más que interesante. Se trata de una aplicación basada en Node.js, es decir, una plataforma back-end Javascript que utiliza un modelo de entradas y salidas para crear aplicaciones de envío y recepción de datos en tiempo real a través de diferentes dispositivos [46].

Node-RED provee una interfaz basada en un navegador web que permite crear flujos de eventos y la posibilidad de interconectarlos, esto lo hace muy flexible y sencillo de usar dando la posibilidad de ahorrar horas de escribir código. Además, el gestor de paquetes de node (npm) puede usarse para instalar nodos adicionales y permitiendo crear nuevas conexiones con diferentes dispositivos o servicios.

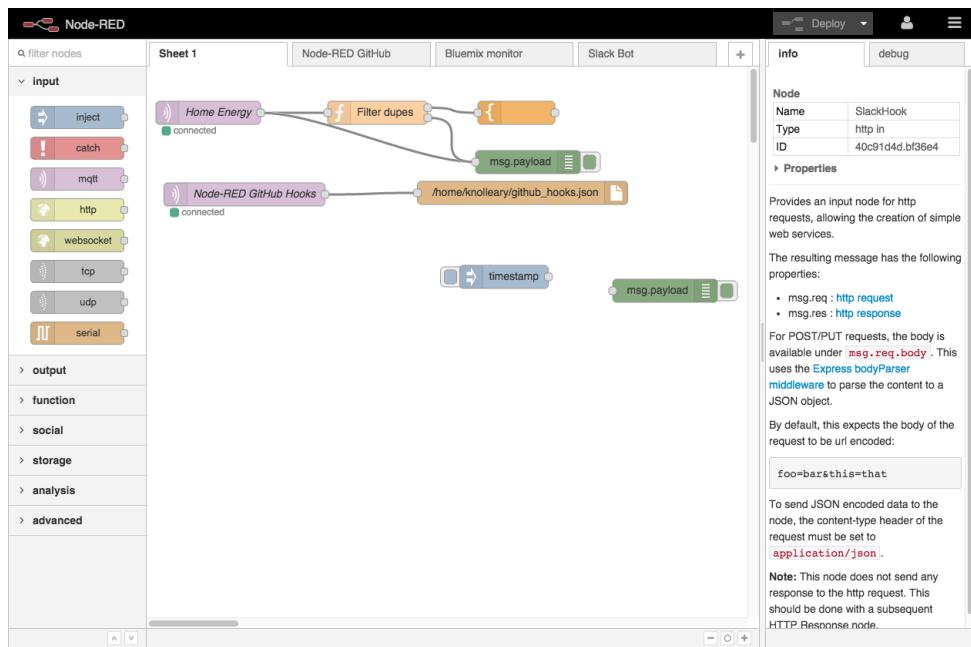


Figura 4.2: Ejemplo de la interfaz de Node-RED

Para tomar la decisión final con más precisión se realizará una tabla comparativa entre los dos servidores puntuando las características principales de ambos:

	Rendimiento	Seguridad	Curva de aprendizaje	Popularidad	Tiempo en funcionamiento
Apache	3	5	4	5	5
Nginx	4	3	3	3	3
Node-RED	3	3	4	2	4

Cuadro 4.4: Tabla comparativa Apache vs Nginx

	Total
Apache	22
Nginx	16
Node-RED	16

Cuadro 4.5: Tabla total comparativa de servidores Web

En este caso aunque Apache y Nginx sean superiores a Node-RED, el sistema será desarrollado con Node-RED por la gran integración y funcionalidades que nos brinda respecto a sus competidores.

4.3. Sistema Operativo

Otra de las decisiones clave a la hora de montar un sistema se trata de elegir el mejor sistema operativo posible. En esta área tenemos cantidad de opciones que supondrán un gasto de tiempo elevado en documentación y toma de decisiones.

Uno de los requisitos del proyecto es que el sistema operativo sea Linux, por lo tanto, existen algunos sistemas que podrían ser validos para cumplir la función. Para Raspberry el sistema operativo más conocido se trata de Raspbian, se trata de una distribución de GNU/Linux basado en Debian. Por otro lado, Arch Linux, Fedora y Ubuntu Mate también distribuciones de Linux, podrían ser serias candidatas.

Fuera de Linux, hace poco salió a la luz el sistema operativo llamado Windows IoT core, se trata de una edición de Windows 10 específica para los pequeños dispositivos del Internet de las Cosas.

Para terminar, se tendrá en cuenta un sistema operativo basado en la seguridad y desarrollado por la compañía Kaspersky, Kaspersky OS. Este sistema no se basa en ningún otro sistema conocido y fue diseñado desde cero pensando en la seguridad.

La decisión final será tomada entre los 3 sistemas que más se ajustan a las necesidades del proyecto, los sistemas operativos estudiados serán los siguientes: Raspbian, Arch Linux y Fedora.

Raspbian

Raspbian⁴ es el sistema operativo más popular para Raspberry Pi. Cuenta con el apoyo oficial por ser la más completa y optimizada de todas las existentes [47]. Este sistema viene con alrededor de 35.000 paquetes precompilados para que su instalación sea sencilla [48].

Como se ha comentado anteriormente, se trata de un sistema libre y gratuito basado en Linux en la distribución Debian.

Este sistema está en continuo desarrollo y ha conseguido formar una comunidad de usuarios que la soportan de manera gratuita.

Fedora

Fedora⁵ se trata de una distribución Linux desarrollada por la comunidad Fedora y promovida por la compañía estadounidense Red Hat. Se trata de una distribución muy potente y al igual que Raspbian mantenida por los usuarios.

Fedora tiene diferentes versiones, estación de trabajo, servidor o cloud. Cada una de ellas con un propósito diferente. Cabe comentar que esta distribución está orientada a usuarios familiarizados con el uso de sistemas operativos Linux y no es apta para usuarios novatos que están empezando [49].

Uno de los puntos fuertes de esta distribución es la seguridad. Fedora cuenta con SELinux ("Security-Enhanced Linux") incorporando una gran cantidad de políticas de seguridad.

Arch Linux

Arch Linux⁶ es una distribución GNU/Linux independiente desarrollada para x86-64. Siguiendo un modelo de lanzamiento continuo intenta estar siempre actualizada.

Esta distribución destaca por su simplicidad, al ser instalada el sistema base es mínimo para que el usuario después configure todo lo que necesite. Al igual que Fedora, esta distribución se centra en el usuario, destinada a cubrir las necesidades de los usuarios que contribuyen a ella.

Arch Linux es una distribución de propósito general. Tras la instalación, solo se proporciona un entorno de línea de órdenes, se ofrece al usuario la posibilidad de crear un sistema personalizado, eligiendo entre miles de paquetes de alta calidad presentes en los repositorios oficiales [50].

⁴<https://www.raspbian.org/>

⁵<https://getfedora.org/es/>

⁶<https://www.archlinux.org/>

	Facilidad de uso	Comunidad de usuarios	Continuidad del SO	Seguridad
Raspbian	4	5	4	4
Fedora	3	4	3	4
Arch Linux	2	4	4	5

Cuadro 4.6: Tabla comparativa de sistemas operativos

	Total
Raspbian	17
Fedora	14
Arch Linux	15

Cuadro 4.7: Tabla total comparativa de sistemas operativos

4.4. Lenguaje de programación

A la hora de programar existe una gran cantidad de lenguajes con los que se pueden realizar los algoritmos o programas. En el caso de estudio que se realizará posteriormente la aplicación consistirá en envío y recepción de datos, es decir, una aplicación basada en el Internet de las Cosas. Por lo tanto, se necesitará buscar un lenguaje que sea capaz de hacer frente a estas necesidades y que posea paquetes o librerías que ayuden en el desarrollo de la aplicación.

Debido a la cantidad de lenguajes existentes, la comparativa se centrará en tres de ellos: Java, JavaScript y Python. Aunque muchos de los expertos en este sector incluyen otros lenguajes como C, C++, Ruby, PHP, SWIFT o Lua [51].

Java

Java⁷ es el lenguaje de programación más utilizado a día de hoy. Se trata de un lenguaje compilado y multiplataforma. Otra de sus características más importantes es que es un lenguaje orientado a objetos lo que lo hace muy simple de manejar.

JavaScript

Es el lenguaje de programación que mayor crecimiento ha experimentado en los últimos años. Es uno de los lenguajes más usados y sigue en aumento. Tal es su popularidad que en un gran número de aplicaciones desarrolladas recientemente el back-end o front-end están desarrollados en este lenguaje.

⁷<https://www.java.com/es/download/>

JavaScript⁸ cuenta con una gran cantidad de librerías que ayudarán a que la programación sea aun más sencilla y tenga un mayor alcance. Por otro lado, cuenta con Frameworks como el popular Node.js para desarrollar las aplicaciones.

Se trata de un lenguaje de scripting, multiplataforma y orientado a objetos. En general, es un lenguaje muy ligero y sencillo lo que lo hace perfecto para aquellos usuarios que están empezando a programar.

Python

Python⁹ se trata de uno de los mejores lenguajes de programación en la actualidad. Es menos utilizado que Java o C pero gran cantidad de empresas de prestigio, tales como Google, Youtube o Facebook, lo utilizan, lo que lo hacen uno de los lenguajes más demandados.

Es un lenguaje interpretado y multiplataforma. Al igual que Java está orientado a objetos lo que lo hace ser un lenguaje simple. Por otro lado, al igual que JavaScript cuenta con Frameworks que facilitan el trabajo de la programación.

Uno de los puntos fuertes de Python es la cantidad de librerías que posee. Estas librerías pueden desarrollar cualquier tipo de aplicaciones, como juegos, apps para smartphones o utilidades para comunicaciones de redes. Puede utilizarse con Arduino lo que lo hace idóneo para ser un lenguaje muy utilizado en el Internet de las Cosas.

	Facilidad de uso	Curva de aprendizaje	Documentación	Librerías
Java	2	2	5	4
JavaScript	3	4	4	5
Python	4	4	4	5

Cuadro 4.8: Tabla comparativa de lenguajes de programación

	Total
Java	13
JavaScript	16
Python	17

Cuadro 4.9: Tabla total comparativa de lenguajes de programación

⁸<https://www.javascript.com/>

⁹<https://www.python.org/>

4.5. Base de datos

Entrando en el tema de almacenamiento y administración de datos, toca tomar la decisión de cuál será el mejor sistema para gestionar la base de datos. Cabe destacar la importancia que tienen las bases de datos en las aplicaciones ya que sin ellas sería casi imposible el manejo de la cantidad de datos que se alojan en Internet.

Al igual que en los diferentes puntos tratados, existen varias alternativas a la hora de analizar cuál es la base de datos más adecuada para el caso de estudio.

Las bases de datos más populares en la actualidad son las siguientes: MySQL, Oracle, PostgreSQL, Microsoft SQL Server, MariaDB, SQLite y MongoDB. De todas estas opciones se realizará la comparativa entre tres de ellas que se adaptan más al entorno del Internet de las Cosas, MySQL, SQLite y MongoDB.

MySQL

Es la base de datos de código abierto más popular. Desarrollada por MySQL AB se trata de un sistema de gestión de bases de datos multihilo, relacional y multiusuario [52].

Destaca su adaptación a múltiples lenguajes de programación y su integración en diferentes sistemas operativos.

MySQL¹⁰ es rápido y fácil de utilizar, en comparación con sistemas más grandes resulta más fácil de configurar y manejar lo que ayuda enormemente al usuario.

Para terminar, aparte de destacar de que se trata de un sistema gratuito, se trata de un sistema con alta conectividad y muy seguro. Dispone de controles de acceso y puede ser accedido desde cualquier lugar de Internet.

SQLite

SQLite¹¹ es la otra opción viable que se contempla. Es una herramienta de Software libre sencilla de utilizar, potente rápida. Puede utilizarse en dispositivos móviles o sistemas de escritorio sin tener que realizar procesos de exportación e importación complejos ya que tiene una compatibilidad total entre todas las plataformas posibles [53].

La base de datos completa se encuentra en un solo archivo lo que la hace muy fácil de acceder y localizar. Por otro lado, puede trabajar únicamente en memoria haciendo extremadamente veloz. Cuenta con librerías para soportar muchos lenguajes de programación y soporta funciones SQL definidas por el usuario.

¹⁰<https://www.mysql.com/>

¹¹<https://www.sqlite.org/>

MongoDB

MongoDB¹² es la base de datos NoSQL más importante que existe en la actualidad. Gracias a su gran agilidad, su escalabilidad y rendimiento muchas han sido las organizaciones que han terminado utilizándola. MongoDB brinda un elevado rendimiento, tanto para lectura como para escritura, potenciando la computación en memoria. Otro punto importante de este tipo de base de datos es que ofrece una tolerancia a fallos automática y contiene una replicación nativa de MongoDB ofreciendo mucha fiabilidad y flexibilidad.

En las bases de datos no relacionales, no tiene porque existir ninguna relación entre los datos ni entre colecciones. Otra característica es que en las bases de datos NoSQL no existen las tablas, lo que se utiliza son colecciones de documentos y los documentos son objetos JavaScript Object Notation (JSON) que dentro de MongoDB se conocen como Binary JSON (BSON) [54].

Al final este tipo de bases de datos no tienen un esquema definido, dentro de ella hay colecciones y dentro de ellas hay documentos guardados en los cuales está toda la información embebida.

Los datos se almacenan de manera binaria para aumentar el rendimiento. Además este tipo de base de datos es muy sencilla y se puede empezar a trabajar con MongoDB fácilmente.

	Facilidad de uso	Curva de aprendizaje	Documentación	Integridad
MySQL	3	4	5	2
SQLite	3	4	4	3
MongoDB	4	2	4	5

Cuadro 4.10: Tabla comparativa de bases de datos

	Total
MySQL	14
SQLite	14
MongoDB	15

Cuadro 4.11: Tabla total comparativa de bases de datos

¹²<https://www.mongodb.com/es>

4.6. Protocolo de comunicación entre la placa base y el VPS

Con todos los criterios de Hardware y Software ya elegidos, falta la comunicación entre ellos, por lo tanto, se debe hacer un pequeño estudio sobre los diferentes protocolos y tratar de elegir el que mejor encaje con el sistema.

Contando con que la aplicación que se va a desarrollar sobre IoT el estudio se realizará sobre 3 de los protocolos más importantes en la actualidad: HTTP, MQTT y CoAP.

HTTP

HTTP o HyperText Transfer Protocol es un protocolo cliente-servidor que facilita los intercambios de mensajes entre clientes web y servidores HTTP.

Este protocolo funciona sobre TCP/IP y funciona como el resto de los servidores de los entornos UNIX, el servidor escucha en algún puerto TCP y espera a las conexiones de los clientes. Una vez que se establece la conexión, el protocolo TCP se encarga de mantener la comunicación y garantizar un intercambio de datos libre de errores.

En la figura 4.3 [55] se aprecia el funcionamiento de una petición HTTP:

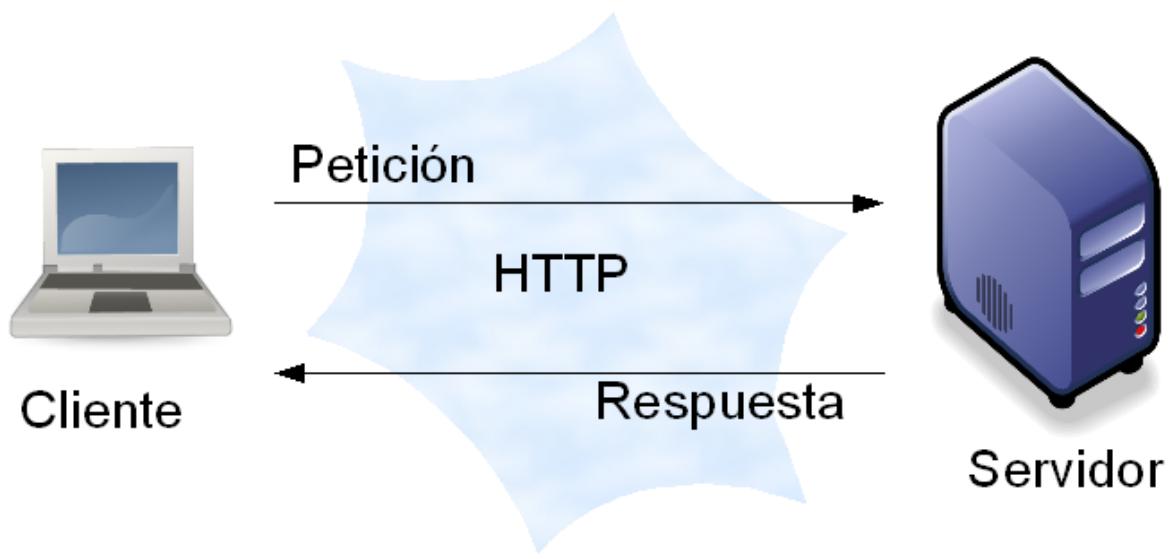


Figura 4.3: Comunicación HTTP

Para terminar, HTTP se trata de un sistema muy robusto de intercambio de mensajes y admite mensajes de gran tamaño lo que puede hacerlo algo pesado para

aplicaciones con multitud de envío y recepción. Sobre la seguridad, HTTP no es nada seguro, para incluir capas de seguridad sobre HTTP necesitamos usar HTTPs que básicamente es lo mismo que HTTP pero funciona sobre SSL o TLS para que los mensajes estén cifrados.

MQTT

MQTT¹³ o Message Queue Telemetry Transport se trata de un protocolo creado por IBM enfocado a la conectividad Machine to Machine (M2M). Es utilizado principalmente en aplicaciones donde los mensajes que se comunican son de un tamaño pequeño que no requiera consumir mucho ancho de banda. Es un protocolo muy ligero y de bajo consumo. Estas características, sin duda, lo hacen un protocolo excelente para la comunicación de sensores y en definitiva del IoT.

La arquitectura de MQTT sigue una topología de estrella. Posee un nodo central llamado Broker con capacidad para manejar miles de clientes y que se encarga de gestionar toda la red y transmitir los mensajes.

La comunicación de este protocolo se basa en tópicos o "topics". Es decir, un cliente crea un tópico y sobre ese tópico comienza a enviar mensajes, esos mensajes llegan al Broker y a continuación el bróker redirecciona esos mensajes a los clientes que están suscritos a ese tópico. Un "topic" se representa mediante una cadena y tiene una estructura jerárquica. Cada jerarquía se separa con '/'.

A continuación, en la figura 4.4 [56] se puede observar gráficamente el funcionamiento de una comunicación MQTT

¹³<http://mqtt.org/>

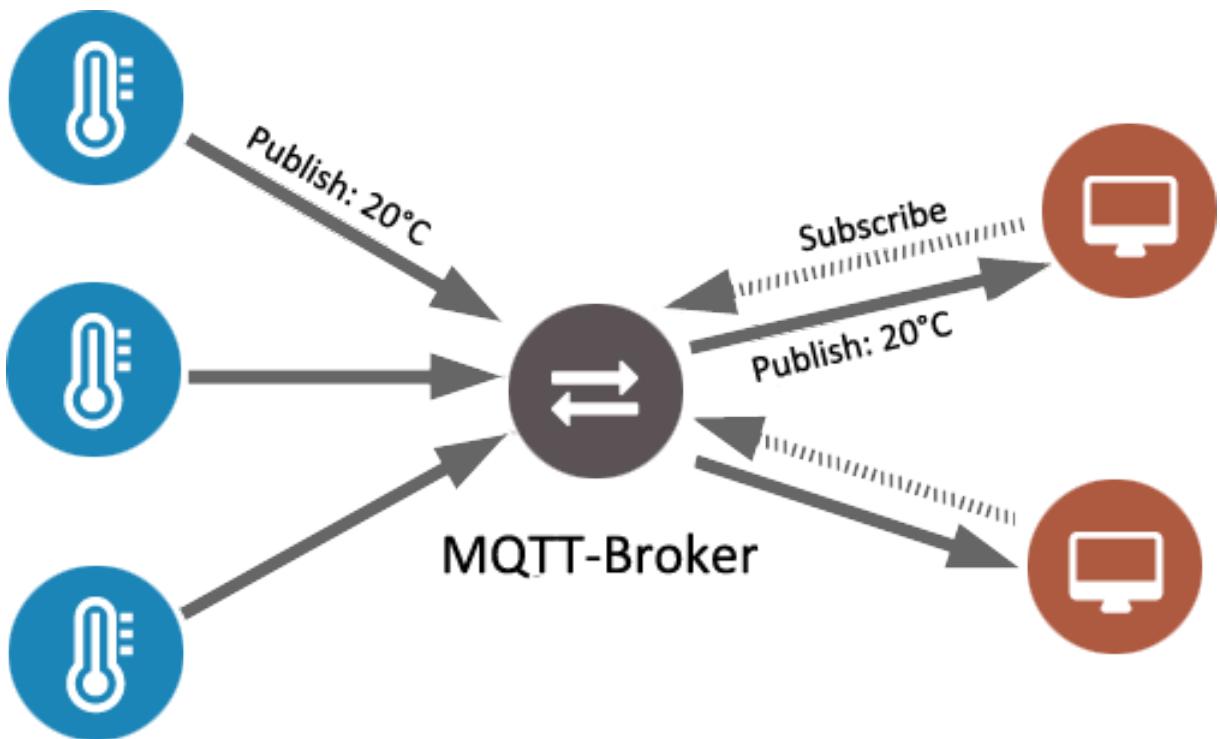


Figura 4.4: Comunicación MQTT

CoAP

CoAP¹⁴ o Constrained Application Protocol es un protocolo diseñado para adaptar HTTP a dispositivos.

Este protocolo funciona sobre UDP y los mensajes que transmite son mucho más pequeños pero manteniendo la misma arquitectura que HTTP. Además, incluye nuevas características como multicast, bajo overhead y simplicidad y el modelo Machine to Machine. Todo esto lo hace un protocolo potente a tener en cuenta dentro del área del IoT.

Para finalizar, cabe destacar que CoAP implementa el modelo Representational State Transfer (REST) de HTTP, usa cabeceras reducidas, y limita el intercambio de mensajes.

4.7. Visualización de los datos

Para terminar con el estudio de las alternativas se debe seleccionar la plataforma donde se visualizarán los datos para que el cliente pueda tomar medidas según los datos que arrojen las diferentes mediciones de los sensores.

¹⁴<http://coap.technology/>

Hoy por hoy existen varias alternativas como Adafruit.io¹⁵, Cayenne, ThingSpeak, Emoncms pero a diferencia de los demás casos, en cuanto a la visualización de datos se refiere, se ha marcado el objetivo de utilizar la plataforma Freeboard.io

Freeboard.io

Se trata de una plataforma desarrollada por Bug Labs¹⁶. Este dashboard programado en JavaScript ofrece un panel web sencillo que muestra la información de los diferentes dispositivos conectados en tiempo real.

Tiene la capacidad de funcionar completamente en el navegador como una aplicación web estática lo que la hace muy recomendable como front-end para aplicaciones con dispositivos de capacidad limitada.

Una de las características más atractivas es la integración que posee con Node-RED. Solo basta con instalar el nodo de Freeboard.io mediante el gestor de paquetes de Node. Una vez instalado, se podrá utilizar en la paleta de Node-RED.



Figura 4.5: Ejemplo de visualización de datos con Freeboard.io

4.8. Plataforma seleccionada

Placa base

La placa seleccionada para realizar este proyecto se trata de la **Raspberry Pi 3 Model B**. La Raspberry Pi 3 es la tercera versión de la plataforma embebida de mismo nombre. Se trata de un mini ordenador completo de bajo consumo en una única placa.

¹⁵ Adafruit a parte de tener librerías para gestionar la conexión entre sensores y los pines GPIO de la Raspberry, posee una interfaz gráfica para mostrar datos en aplicaciones

¹⁶<https://buglabs.net/>

Esta nueva versión mejora en todos los aspectos a las anteriores, pudiendo llegar a tener cientos de utilidades diferentes.

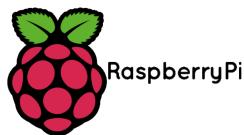


Figura 4.6: Logotipo Raspberry Pi

Entre sus principales características destacaremos:

- Un procesador ARM-Cortex-A53 de cuatro núcleos a 1,2 GHz.
- 1GB LPDDR2 de RAM
- Conectividad Wifi, Bluetooth 4.1 y Ethernet.

Al ser un ordenador de bajo coste y de bajo consumo de recursos lo hacen perfecto para funcionar en aplicaciones relacionadas con el IoT, envío de datos masivos desde sensores u otro tipo de dispositivos.

Para terminar, este tipo de placa base cuenta con un gran número de periféricos disponibles que facilitarán la realización del proyecto y dan opción a realizar múltiples acciones con este mini ordenador. Además, Raspberry cuenta con una comunidad enorme de personas desarrollando aplicaciones o proyectos realmente divertidos, lo que provoca que haya mucha información sobre esta placa y sus utilidades.

Servidor Web

A la hora de elegir el servidor Web aparecen varias opciones que más que validas. Finalmente, la opción elegida será **Node-RED basado en Node.js** como servidor Web. Aun sabiendo que APACHE o Nginx son opciones más potentes puesto que ofrecen mejores características, Node.js ha sido seleccionado ya que encaja mejor con la finalidad del proyecto y los componentes que aparecen en él. Poder incluir la tecnología Node-RED va a ser una gran oportunidad para adquirir mayor conocimiento sobre el área y va a añadir un elemento novedoso al proyecto.

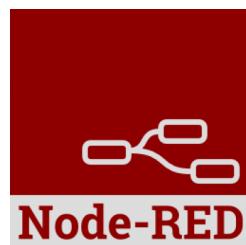


Figura 4.7: Logotipo Node-Red

El motivo mas importante por el que hemos elegido esta tecnología es por su fácil integración con el protocolo MQTT, facilitando así las conexiones. A parte, Node-RED nos permite almacenar los datos en la base de MongoDB de una forma muy sencilla y visualizarlos en el dashboard de Freeboard.

Sistema Operativo

En cuanto al sistema operativo de la Raspberry Pi la opción seleccionada ha sido **Raspbian**. Lo que se ha tomado en cuenta a la hora de tomar la decisión ha sido su facilidad de uso y su gran comunidad de usuarios. Aunque existen cantidad de sistemas operativos muy potentes para esta placa, Raspbian es sin duda el sistema operativo más popular.

Por último, cabe destacar que se prevé que todavía tenga muchos años por delante para seguir desarrollándose haciendo que la aplicación del proyecto pueda funcionar a lo largo del tiempo sin ningún tipo de problema.

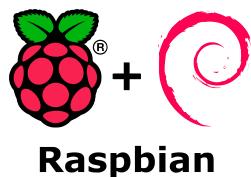


Figura 4.8: Logotipo Raspbian

Lenguaje de programación

En cuanto al lenguaje de programación se ha decidido utilizar **Python y JavaScript**, estos dos lenguajes son los predominantes en cuanto a aplicaciones de IoT se trata. JavaScript se usará dentro de Node-RED y Freeboard.io ya que están desarrolladas bajo este lenguaje de programación.



Figura 4.9: Logotipo JavaScript

En cambio, el cliente MQTT encargado de enviar los mensajes al bróker será programado en Python.



Figura 4.10: Logotipo Python

Base de datos

Para la persistencia de los datos la base elegida ha sido **MongoDB**. A pesar de que su competencia ofrece unas bases de datos potentes como MySQL, se ha optado por esta base. El motivo principal por el que ha sido seleccionada se trata de la integridad que posee con Node-RED y el lenguaje JavaScript. Esta base de datos NoSQL es la más popular por el gran rendimiento y agilidad que ofrece, por lo tanto, será una fantástica opción para el proyecto a realizar. Además el formato de los datos enviados en la aplicación es JSON lo que hace MongoDB perfectamente compatible ya que se trata de la base de datos que mejor trabaja con JSON.



Figura 4.11: Logotipo MongoDB

Protocolo de comunicación entre la placa base y el VPS

Con el fin de comunicar la Raspberry con nuestro servidor Virtual Private Server (VPS) el protocolo seleccionado ha sido **MQTT**. Este protocolo ligero ha sido seleccionado principalmente porque es el que más se ajusta a nuestras necesidades. Un protocolo sencillo que sea capaz de manejar mensajes de tamaño reducido y sea capaz de conectar diferentes dispositivos. Además su integración con Node-RED lo hace todavía mas atractivo para ser utilizado en aplicaciones de Internet of Things.



Figura 4.12: Logotipo MQTT

VPS

Para añadir un grado de dificultad al proyecto se ha decidido incluir un VPS para poseer y gestionar un servidor propio.

El servidor que se ha contratado pertenece a clouding.io¹⁷



Figura 4.13: Logotipo Clouding.io

El objetivo de utilizar un servidor propio es el hecho de aprender a gestionar un servidor desde cero, aprender a gestionar las comunicaciones, el Firewall del mismo y establecer un nombre de dominio asociado a nuestro servidor, en nuestro caso www.tfgjulen.es. Por último, comentar que el sistema operativo seleccionado para el VPS ha sido Ubuntu Server 17.04.

Visualización de los datos

Por último como se ha comentado anteriormente, la plataforma seleccionada es **Freeboard.io**. Desde el primer momento se contempló como la mejor opción debido a la alta integración con Node-RED y al hecho de esta todo programado bajo el lenguaje de programación JavaScript. En cuanto a la seguridad, una vez instalado el nodo de Freeboard.io en Node-RED conseguimos una comunicación segura.

Tabla resumen de las herramientas utilizadas

Alternativas	Herramienta seleccionada
Placa base	Raspberry Pi 3
Servidor Web	Node-RED basado en Node.js
Sistema operativo	Raspbian
Lenguaje de programación	JavaScript y Python
Base de datos	MongoDB
Protocolo de comunicación entre la placa base y el VPS	MQTT
Visualización de los datos	Freeboard.io

Cuadro 4.12: Tabla resumen de las herramientas utilizadas

¹⁷<https://clouding.io/>

Parte III

Recomendaciones y guía al programador

CAPÍTULO 5

PROUESTA

Funcionamiento básico del sistema

Con todos los componentes de Hardware como de Software seleccionados, podemos empezar a plantear cual va a ser el mejor esquema a seguir para la realización de la aplicación. Hay que destacar que la primera versión de la aplicación no tendrá medidas de seguridad, éstas serán aplicadas en la segunda versión de la aplicación después de asegurar su correcto funcionamiento.

Diagrama de despliegue

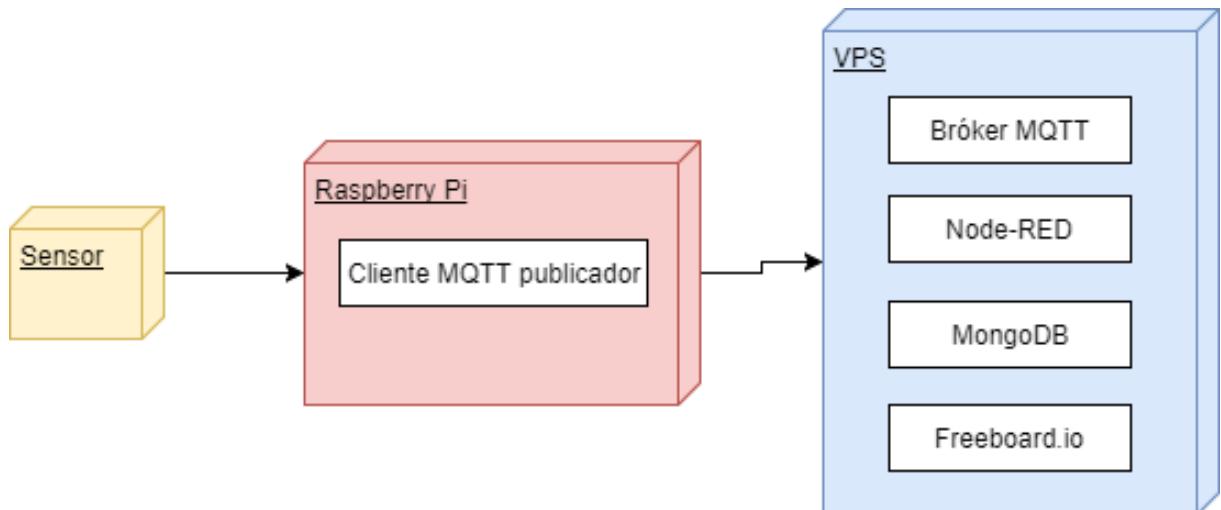


Figura 5.1: Diagrama de despliegue

Como se observa en la figura¹ 5.1, la aplicación que se va a desarrollar consta de tres elementos fundamentales: Sensor, placa base y VPS. Los mensajes serán enviados desde la placa base al VPS y los datos de mediciones serán recibidos por la placa base.

Diagrama de herramientas

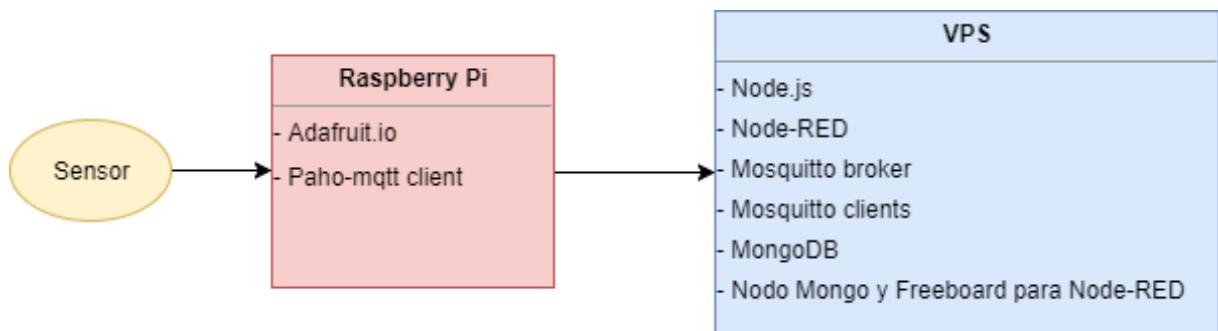


Figura 5.2: Diagrama de herramientas

En el diagrama 5.2 se puede observar las herramientas que se van a utilizar en el prototipo. Dentro de la Raspberry Pi estarán instaladas las librerías Adafruit.io y Paho-mqtt. Ambas van a ser utilizadas a la hora de programar el cliente encargado de enviar los mensajes. El VPS en cambio, contará con Node.js para poder instalar Node-RED, el bróker y clientes de MQTT, la base de datos MongoDB y los nodos de Mongo y Freeboard para poder utilizarlos con Node-RED.

¹Los diagramas se han realizado en la plataforma <https://www.draw.io/>

Diagrama de funcionamiento

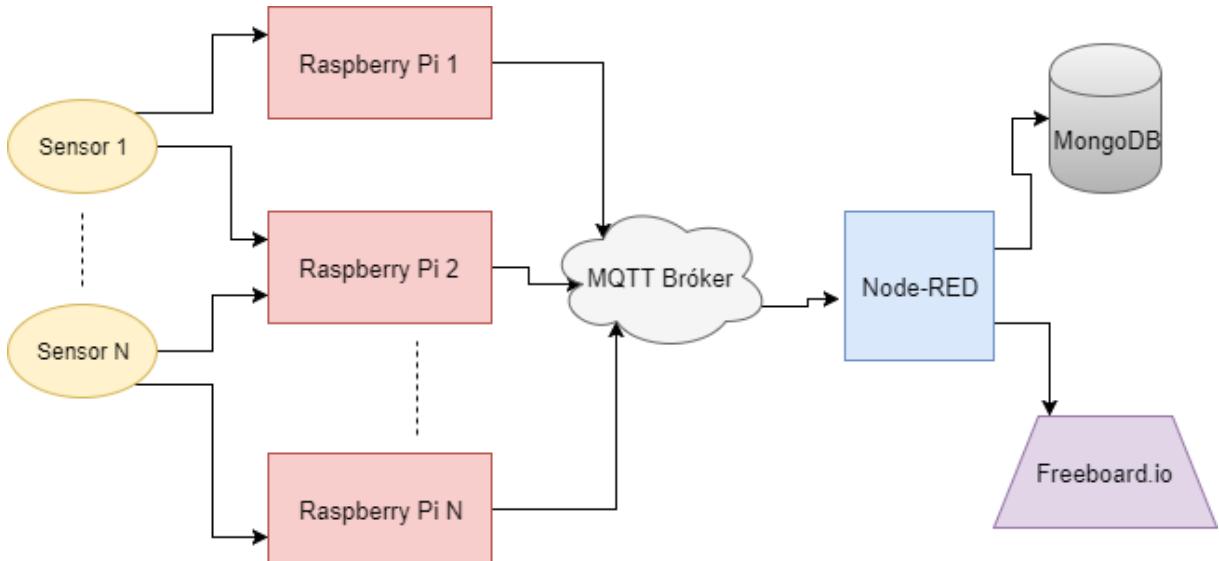


Figura 5.3: Diagrama de funcionamiento

En el diagrama 5.3 se puede observar que el sensor es el encargado de iniciar el flujo de datos. En el caso de estudio utilizaremos únicamente un sensor, pero el número de sensores que pueden utilizarse en una aplicación de alto nivel puede escalar hasta N.

Las placas base son las encargadas de recibir los datos de las mediciones que los sensores realizan y los clientes MQTT programados dentro de ellas los envían mediante tópicos al bróker que estará alojado en el servidor VPS.

El VPS a parte del bróker MQTT contiene el Node-RED que es el encargado de recibir los mensajes transmitidos por los clientes con los datos de las mediciones y posteriormente almacena estos datos en la base de MongoDB creada y los envía a la plataforma de Freeboard.io para ser visualizados por el cliente.

Como se ha comentado anteriormente, este sistema posee una gran escalabilidad ya que se podrían tener miles de sensores midiendo todo tipo de variables y gran cantidad de Raspberries recogiendo la información y enviándola al bróker de MQTT con diferentes tópicos para ser tratada.

Diagrama de protocolos

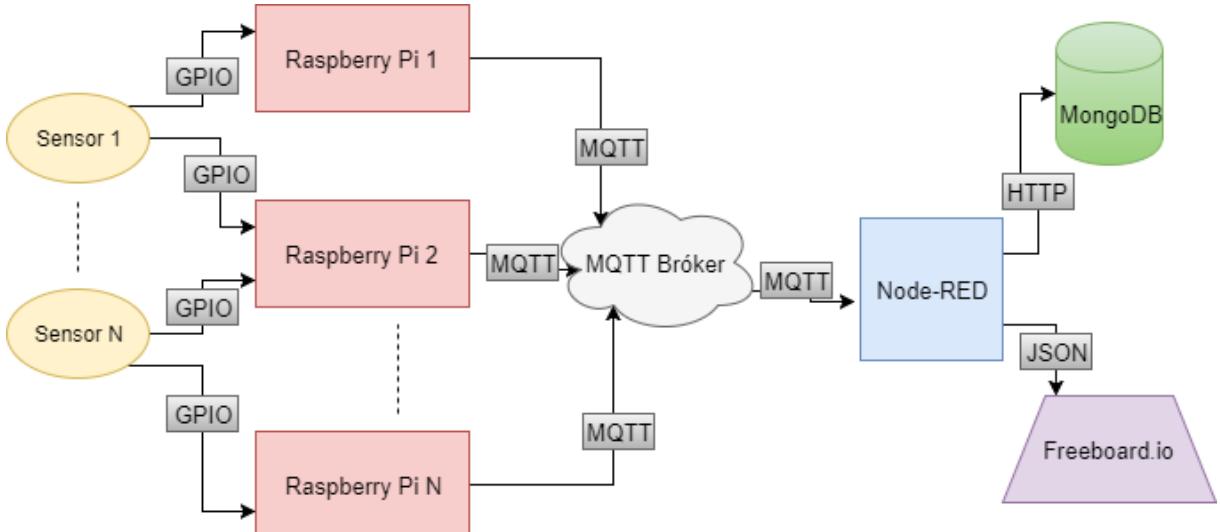


Figura 5.4: Diagrama de protocolos

En el diagrama 5.4 se pueden observar los diferentes protocolos que van a ser utilizados para transmitir los mensajes correctamente.

La comunicación sencilla entre el sensor y la Raspberry se realiza mediante GPIO (General Purpose Input/Output), básicamente, se trata de pines de entradas y salidas digitales.

A continuación, la comunicación completa de las placas base con el servidor VPS se realiza vía MQTT que como se ha explicado anteriormente se trata de un protocolo de comunicación ligero orientado a este tipo de aplicaciones.

Por último, Node-RED transfiere los datos tanto a MongoDB como a Freeboard.io. La comunicación con la base de datos se realiza mediante HTTP y utiliza un protocolo basado en JSON para transferir los datos a la plataforma de visualización.

Destacar que el cliente MQTT envía los datos al bróker en formato JSON, durante la transmisión del mensaje entre el bróker y el Node-RED se parsea, pero dentro del flujo del Node-RED vuelve a parsearse de nuevo a JSON y este objeto es guardado en la base de datos y mostrado en la gráfica.

5.1. Guía al programador

En este apartado del proyecto se va a realizar una guía para conseguir que cualquier persona que no haya podido aprender conceptos sobre este área o no tenga el tiempo suficiente para investigar pueda realizar la aplicación desarrollada. Recordar que la primera versión de la aplicación no contará con seguridad.

5.1.1. Instalación y configuración básica de los elementos necesarios en la Raspberry Pi

Instalación de Raspbian

La puesta en marcha de la Raspberry es un punto importante para empezar a desarrollar la aplicación. Lo primero que se debe hacer es instalar el sistema operativo para ello se necesitan los siguientes elementos:

- **Una Raspberry Pi**
- **Un teclado**
- **Un ratón**
- **Un cable HDMI**
- **Un monitor**
- **Una tarjeta SD**

Estos elementos son necesarios tanto para transmitir las imágenes y sonido como para poder manejar la placa base como si fuese un PC normal.

El primer paso será la descarga de NOOBS² en el ordenador. Seguidamente se descomprimirá la carpeta descargada dentro de la tarjeta de memoria. Para asegurarse de que funcione correctamente, es recomendable que la tarjeta SD esté formateada y tenga más de 8GB de memoria. Una vez la tarjeta esté preparada comenzará la siguiente fase.

Insertar la tarjeta SD en la Raspberry y conectar los componentes antes mencionados. Al conectar la Raspberry a la corriente podrá encenderse y NOOBS se ejecutará de forma automática.

El proceso de instalación es muy sencillo, NOOBS da la opción de elegir el sistema operativo que se necesite (en nuestro caso Raspbian) y solamente con seleccionarlo y pulsar sobre instalar comenzará la instalación. El menú de instalación permite elegir el idioma tanto para el teclado como para el instalador. Hay que remarcar que necesitaremos conectar un cable Ethernet ya que en la instalación el Wifi no está activado. Cuando tengamos todo preparado pulsamos sobre instalar y el sistema operativo empezará a instalarse.

²Instalador de sistemas operativos para la Raspberry Pi, enlace de descarga: <https://www.raspberrypi.org/downloads/noobs/>

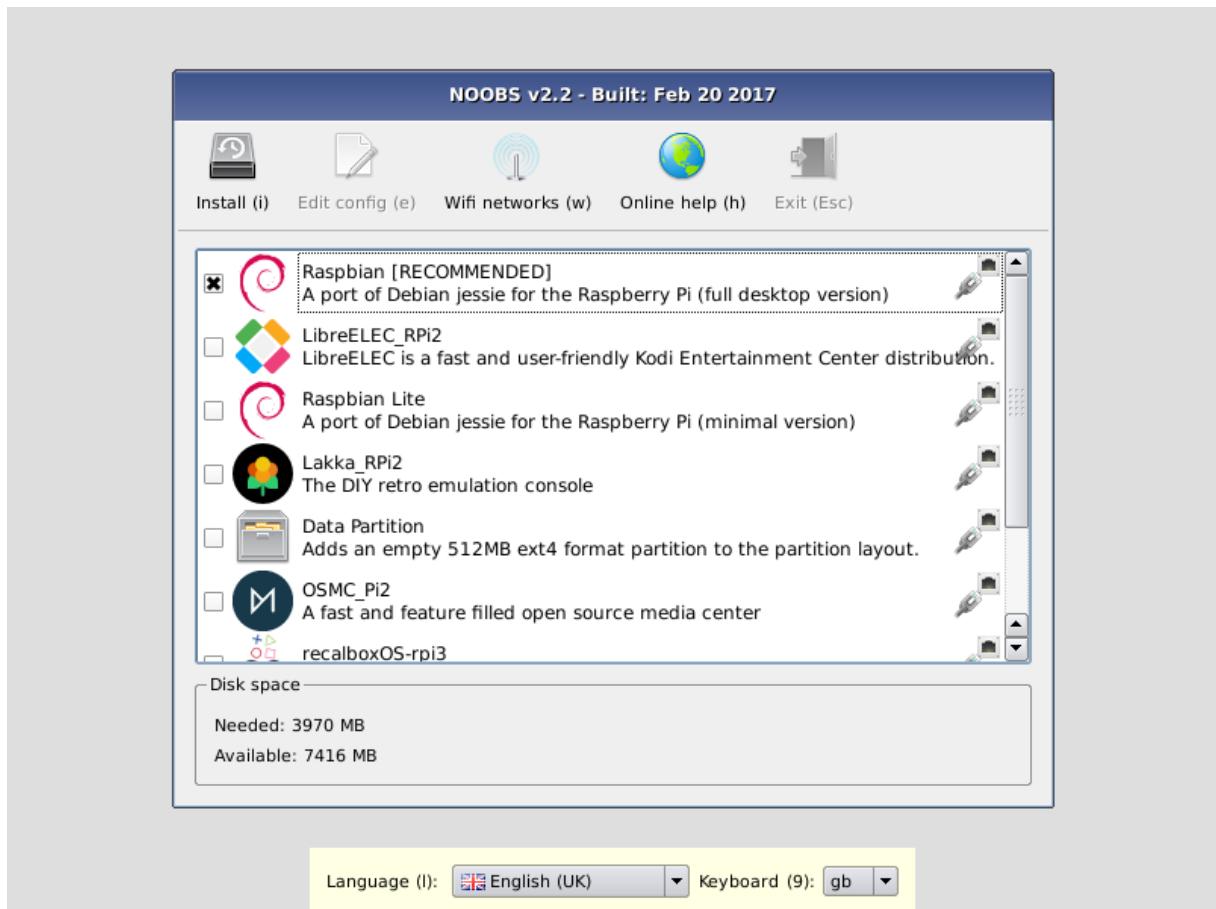


Figura 5.5: Ventana de selección del sistema operativo en NOOBS

Al terminar la instalación se pulsará sobre OK y terminarán de instalarse los últimos paquetes, por último, aparecerá el entorno gráfico.

Instalación de elementos necesarios para programar el cliente MQTT

Dentro de este punto, tenemos que tener en cuenta el lenguaje de programación que se va a utilizar para programar el cliente MQTT y la librería necesaria para gestionar la conexión con el sensor.

En este caso el script va a ser programado en Python por lo que después de ejecutar los siguientes comandos:

```
$ sudo apt-get update
$ sudo apt-get upgrade
```

Necesitamos instalar los paquetes necesarios para utilizar el lenguaje de programación Python. A su vez, se utilizará la librería de Adafruit.io³ para poder establecer

³<https://io.adafruit.com/>

comunicaciones con los pines GPIO de la Raspberry Pi. Para instalar lo mencionado anteriormente se deben insertar los siguientes comandos:

```
$ sudo apt-get install build-essential python-pip python-dev python-smbus  
      git  
$ sudo git clone https://github.com/adafruit/Adafruit_Python_GPIO.git  
$ cd Adafruit_Python_GPIO  
$ sudo python setup.py install
```

Una vez terminadas las instalaciones todas las librerías y paquetes necesarios para el manejo de Python y la librería Adafruit estarán disponibles, por lo tanto, se podría establecer ya la conexión entre la Raspberry y el sensor DHT11.

Instalación del cliente MQTT

Para terminar con las instalaciones en nuestra placa base, falta el elemento más importante encargado de transmitir los datos de las medidas al bróker, el cliente MQTT.

Para realizar la aplicación se ha tomado la decisión de utilizar Paho MQTT. Se trata de un cliente MQTT que utiliza Python como lenguaje de programación por lo que será perfecto para integrarlo con la librería de Adafruit para la recepción y envío de los datos al servidor.

La instalación de este cliente es muy sencilla, basta con introducir la siguiente orden:

```
$ sudo pip install paho-mqtt
```

Después de tener todo instalado ya se podrá empezar a programar el script del cliente cuyo objetivo es el de establecer una conexión con el bróker, recoger las medidas del sensor DHT11 y enviarlas al servidor.

5.1.2. Instalación y configuración básica de los elementos necesarios en el VPS

Instalación de Node-RED

Lo primero que se instalará en el servidor VPS será Node-RED. Para ello es necesario tener instalado Node.js y su gestor de paquetes npm.

Antes de nada, se deben introducir los comandos básicos de update y upgrade para recibir las últimas actualizaciones de todos los paquetes instalados en el servidor.

A continuación, si **curl** no está instalado en el servidor, se instala introduciendo el siguiente comando:

```
$ sudo apt-get install curl
```

Cuando esté todo listo, se podrá descargar y ejecutar el script que ofrece Node-Source:

```
$ curl -sL https://deb.nodesource.com/setup_9.x | sudo -E bash -
```

La instalación de todos los paquetes tardará unos cuantos minutos. Después de terminar el proceso se introducirá el siguiente comando para completar la instalación:

```
$ sudo apt-get install -y nodejs build-essential
```

Este comando instalará Node.js si antes no existía en la máquina o lo actualizará si había una versión anterior.

Finalmente, para comprobar que la instalación ha sido correcta, se deben introducir los dos siguientes comandos:

```
$ node -v  
$ npm -v
```

Los dos comandos anteriores sirven para ver la versión tanto del Node.js como del npm.

Al finalizar con la instalación de Node.js se procederá a instalar Node-RED. Lo primero que se debe hacer en este paso es instalar **git**. La instalación es muy sencilla, únicamente se debe insertar el comando que muestro a continuación:

```
$ sudo apt-get install git
```

Seguidamente, para instalar el paquete de Node-RED se debe introducir el siguiente comando:

```
$ sudo npm install -g --unsafe -perm node-red
```

Comenzará el proceso de instalación que durará unos minutos y cuando termine, estará listo para ser ejecutado.

Por último, se introducirá el comando que se encuentra a continuación para asegurar la correcta instalación:

```
$ sudo node-red
```

Instalación del bróker MQTT

En este apartado se va a proceder a instalar la parte más importante de la comunicación MQTT, el bróker. La instalación es muy sencilla ya que solo hace falta introducir el siguiente comando para conseguirlo:

```
$ sudo apt-get install mosquitto mosquitto-clients
```

Cuando la instalación termine el bróker ya estará listo para ser utilizado.

Instalación de MongoDB y los nodos de Mongo y Freeboard en Node-RED

Finalmente, cuando se instalen los elementos que vienen a continuación, el esquema de la aplicación estará creado.

Para empezar, lo primero que se instalará será la base de datos MongoDB. Al igual que las instalaciones vistas hasta ahora se trata de una instalación sencilla. Basta con introducir:

```
$ sudo apt-get install mongodb
```

Para terminar con las instalaciones, se instalarán dos nodos para el entorno de Node-RED. Para instalar los siguientes nodos se debe estar dentro de la carpeta de **node-red**, esta carpeta se encuentra normalmente en la siguiente ruta: */usr/lib/node_modules*.

El primer nodo que se instalará será el de MongoDB, para ello se debe ejecutar el comando que aparece a continuación:

```
$ sudo npm install node-red-node-mongodb
```

El segundo nodo será el de Freeboard.io, encargado de gestionar la conexión con la plataforma para la visualización de los datos.

```
$ sudo npm install node-red-contrib-freeboard@0.0.5
```

Remarcar que se instalará la versión 5 porque la 7 viene con errores y no es posible completar la instalación.

5.2. Recomendaciones

Hasta este punto se ha desarrollado una aplicación que cumple a la perfección el objetivo del envío y la recepción de los mensajes, pero sin ningún tipo de seguridad, por lo que es vulnerable a multitud de ataques.

En este punto del proyecto se recomendarán medidas de seguridad para que la aplicación sea segura. Cabe recalcar que el objetivo del proyecto es el de hacer posible que alguien sin conocimientos sobre seguridad pueda realizar la aplicación con unas bases de seguridad. Se implantarán las medidas necesarias y más importantes explicadas paso a paso y el problema que mitigan.

5.2.1. Recomendaciones de seguridad para Raspberry Pi

El primer punto donde se empezará a implantar seguridad es en nuestra placa base. Este aparato esta siendo utilizado para multitud de proyectos en Internet por lo que su seguridad debe ser revisada. El problema es que muchos usuarios se olvidan de cambiar el usuario que la Raspberry ofrece por defecto y permanecen vulnerables. Es algo muy básico y a la vez muy importante.

Cambiar usuario y contraseña por defecto

El primer paso será cambiar la contraseña que tiene el usuario Pi por defecto. Para lograrlo introducimos se introducirá en la terminal:

```
$ passwd
```

El sistema pedirá introducir la contraseña que viene por defecto e introducir la nueva. Una vez terminado la contraseña habrá sido cambiada.

En el siguiente paso se va a cambiar el usuario por defecto pi por el nuevo que se quiera introducir. Lo primero que hay que hacer es activar el usuario **root** que por razones de seguridad viene desactivado por defecto. El siguiente comando activa el usuario root pero antes se debe introducir una contraseña, por lo tanto, se introducirá una contraseña fuerte para evitar posibles vulnerabilidades de contraseña.

```
$ sudo passwd root
```

Una vez terminado este paso se debe cerrar el usuario pi e iniciar una nueva sesión en el usuario root. Dentro de root insertamos el siguiente comando:

```
$ usermod -l nuevo-usuario pi -md /home/nuevo-usuario
```

Este comando lo que hace es renombrar el usuario por defecto pi y mover todo su contenido a la carpeta /home del nuevo usuario.

Después, iniciamos sesión con el nuevo usuario y por seguridad volvemos a deshabilitar el usuario root:

```
$ sudo passwd -l root
```

Seguridad de SSH

Este punto es muy importante ya que se protegerá a la Raspberry de accesos y conexiones no autorizadas.

El primer paso a realizar es el de proteger nuestra red. Para ello se debe editar el archivo `/etc/sysctl.conf`, en la terminal introducimos el siguiente comando:

```
$ sudo nano /etc/sysctl.conf
```

Editar el archivo descomentando las líneas que se ven a continuación:

```
...  
# Uncomment the next two lines to enable Spoof protection (reverse-path  
# filter)  
# Turn on Source Address Verification in all interfaces to  
# prevent some spoofing attacks  
net.ipv4.conf.default.rp_filter=1  
net.ipv4.conf.all.rp_filter=1  
  
...  
# Do not accept ICMP redirects (prevent MITM attacks)  
net.ipv4.conf.all.accept_redirects = 0  
net.ipv6.conf.all.accept_redirects = 0  
# _or_  
# Accept ICMP redirects only for gateways listed in our default  
# gateway list (enabled by default)  
# net.ipv4.conf.all.secure_redirects = 1  
#  
# Do not send ICMP redirects (we are not a router)  
net.ipv4.conf.all.send_redirects = 0  
#  
# Do not accept IP source route packets (we are not a router)  
net.ipv4.conf.all.accept_source_route = 0  
net.ipv6.conf.all.accept_source_route = 0
```

Al modificar este archivo se ha establecido una seguridad extra contra ataques MITM, Spoofings y algunos métodos más.

Por otro lado, Raspbian instala una shell de acceso remoto a la que se puede acceder desde cualquier lugar, esto puede ser muy peligroso por lo que se debe cambiar esa configuración para que solamente máquinas con una clave Secure Shell (SSH) autorizada puedan conectarse.

Para editar el archivo de configuración SSH se debe introducir el siguiente comando:

```
$ sudo nano /etc/ssh/sshd_config
```

Al igual que en el caso anterior se deben descomentar algunas ordenes. En este caso el archivo quedaría de la siguiente manera:

```
# Authentication:  
LoginGraceTime 2m  
PermitRootLogin no  
StrictModes yes  
MaxAuthTries 6  
MaxSessions 10  
PubkeyAuthentication yes  
  
AuthorizedKeysFile      .ssh/authorized_keys .ssh/authorized_keys2  
  
...  
# To disable tunneled clear text passwords, change to no here!  
#PasswordAuthentication yes  
PermitEmptyPasswords no  
  
...  
# Set this to 'yes' to enable PAM authentication, account processing,  
# and session processing. If this is enabled, PAM authentication will  
# be allowed through the ChallengeResponseAuthentication and  
# PasswordAuthentication. Depending on your PAM configuration,  
# PAM authentication via ChallengeResponseAuthentication may bypass  
# the setting of "PermitRootLogin without-password".  
# If you just want the PAM account and session checks to run without  
# PAM authentication, then enable this but set PasswordAuthentication  
# and ChallengeResponseAuthentication to 'no'.  
UsePAM no
```

Al editar este archivo se están modificando varias acciones importantes. En primer lugar se aprecia como el login del usuario root no está permitido. Seguidamente, se han modificado los intentos de iniciar sesión a 6 y el máximo de sesiones a 10. Aparte de esto se han modificado factores relacionados con el manejo de las claves. Las contraseñas vacías no están permitidas y por último, se ha quitado el uso de PAM, es decir, se deshabilitarán los módulos de autenticación enchufables o la autenticación nativa de Linux y sólo permitirá a los usuarios iniciar sesión con una clave.

Una vez se hayan guardado los cambios, se debe comprobar que la configuración del fichero es correcta. Este punto es crucial, ya que si el fichero tiene algún error, la siguiente vez que se intente establecer una conexión el servidor SSH no arrancará.

```
$ sudo sshd -t
```

Implantación de un IDS

Para ir un paso más allá en la seguridad de la Raspberry se va a instalar un sistema de detección de intrusos. Este elemento evita el éxito de ataques de fuerza bruta o de diccionario que se realicen sobre la máquina protegida. En la aplicación el Intrusion Detection System (IDS) seleccionado será Fail2Ban⁴, este sistema escanea automáticamente todos los logs de la Raspberry y banea las IPs que considera maliciosas.

Para instalar el IDS basta con introducir el siguiente comando:

```
$ sudo apt-get install fail2ban
```

Después de instalarse se debe acceder al archivo que contiene la configuración:

```
$ sudo nano /etc/fail2ban/jail.conf
```

Una vez dentro se debe editar el archivo quedando de la siguiente manera:

```
...
#[DEFAULT]
bantime = 3600
#
#[sshd]
enabled = true

...
# "ignoreip" can be an IP address, a CIDR mask or a DNS host. Fail2ban will
# not
# ban a host which matches an address in this list. Several addresses can
# be
# defined using space (and/or comma) separator.
ignoreip = 127.0.0.1/8 192.168.1.0/24
```

Obviamente, si el usuario lo requiere puede modificar este archivo de la manera que más le convenga.

Cuando haya terminado toda la configuración, se puede reiniciar el servicio, comprobar el status o incluso ver las IP baneadas con los siguientes comandos:

```
$ sudo /etc/init.d/fail2ban restart
$ sudo /etc/init.d/fail2ban status
$ sudo fail2ban-client status sshd
```

Para terminar con este punto, el archivo donde se guardan los intentos de login a la Raspberry es el siguiente:

```
$ sudo cat /var/log/fail2ban.log
```

⁴<https://www.fail2ban.org/>

Instalación de un Firewall

Se necesitará instalar un cortafuegos para evitar que la Raspberry esté expuesta a recibir cualquier tipo de ataque desde Internet. La misión de un Firewall es la de gestionar los puertos de comunicaciones, controlando así todas las conexiones entrantes y salientes de la máquina en la que está instalado. Existen varias alternativas pero se ha elegido el cortafuegos Uncomplicated Firewall (UFW) para cumplir la tarea por ser el más fácil de configurar.

Lo primero que se debe hacer para instalarlo es ejecutar el comando siguiente:

```
$ sudo apt-get install ufw
```

Una vez instalado, se introducirán dos órdenes básicas por defecto. La primera permite todo el tráfico saliente y la segunda impide todo el tráfico entrante.

```
$ sudo ufw default deny incoming  
$ sudo ufw default allow outgoing
```

Como en el prototipo, la Raspberry solamente se va a utilizar para transmitir los datos hacia el bróker, únicamente se van a habilitar dos puertos de entrada que van a ser el puerto 22 correspondiente al puerto estándar de SSH y el puerto 3389 que corresponde al servidor de acceso remoto xrdp.

```
$ sudo ufw allow ssh  
$ sudo ufw allow 3389/tcp
```

De esta forma la Raspberry solo admitirá el tráfico que se transmite por esos dos puertos.

Por último, para poner el Firewall en marcha se debe ejecutar el siguiente comando, una vez ejecutado, el Firewall ya estaría en funcionamiento:

```
$ sudo ufw enable
```

5.2.2. Recomendaciones de seguridad para el VPS

Gestionar el control de acceso

El primer paso para poder crear un servidor con una base de seguridad es asegurar el acceso que se puede tener al mismo. Es decir, que solo tú puedas entrar.

Lo primero que se debe hacer es inhabilitar el acceso a root, todos los servidores Linux tienen root como usuario lo que facilita a los hackers hacer ataques de fuerza bruta contra ese usuario y conseguir acceder. Este paso básico añadirá una capa extra de seguridad.

Al deshabilitar el superusuario, se debe crear un usuario nuevo y utilizar el comando **sudo** para ejecutar órdenes de root.

Para deshabilitar el usuario root, al igual que en la RPi hay que editar el archivo que tenemos a continuación cambiando el “PermitRootLogin” que por defecto será Si a No:

```
$ sudo nano /etc/ssh/sshd_config
```

Siguiendo con el tema de control de acceso, un tema a tener en cuenta es la política de contraseñas. Como se ha explicado anteriormente en el proyecto, el tener una buena política de contraseñas es crucial para tener nuestros sistemas protegidos. Por lo tanto se deberán crear contraseñas fuertes y deberán ser modificadas periódicamente.

Para finalizar con este punto, sería muy buena opción el instalar un IDS como el que se ha explicado en el anterior punto. Existen varios IDS que podrían cumplir perfectamente la función en el servidor. Por ejemplo: Tripwire, Aide o Psad entre otros. Esto evitaría cualquier tipo de ataque por fuerza bruta lo que haría al servidor muy difícil de acceder.

Consejos generales sobre seguridad

Para empezar con los consejos básicos de seguridad en servidores, siempre hay que tener el Software actualizado. Gran cantidad de hackers aprovechan el Software desactualizado para atacar a la víctimas. Si se consigue tener un servidor actualizado diariamente se evitarán muchos tipos de ataques.

El VPS contratado viene con un Firewall por defecto el cual se puede modificar para cerrar o abrir los puertos necesarios. Por lo tanto, es muy buena opción el de utilizarlo correctamente para tapar brechas de seguridad.

Otro consejo importante es el de evitar descargar archivos o programas de sitios no oficiales. El hecho de descargar archivos o programas a nuestro servidor desde una fuente en la que no confiamos conlleva un gran riesgo de seguridad. Esto no significa que todo lo que se descargue de sitios web no oficiales sea un virus, pero si se puede evitar, el servidor no correrá ningún riesgo.

Nunca se debe usar File Transfer Protocol (FTP), en su lugar se deberá usar SSH File Transfer Protocol (SFTP), la autenticación en FTP se envía en texto plano por lo que si alguien intercepta la comunicación puede ver los detalles del login. En cambio, SFTP esta basado en la misma seguridad del protocolo SSH, es decir, la comunicación está cifrada.

Continuando con el tema de FTP, nunca se debe dejar que usuarios anónimos suban archivos a nuestro servidor. Aunque parezca mentira, algunas configuraciones de servidores tiene esta opción habilitada por defecto dejando una brecha de seguridad enorme.

Para terminar con este tema, añadir los escáneres de malware. Es sabido que

Linux es menos amenazado que Windows en cuanto a Malware pero esto no quiere decir que el riesgo no exista. El instalar un escáner de malware puede ayudar de gran forma a identificar algún tipo de actividad que indique que un malware ha sido instalando en el servidor. El escáner más popular en Linux es Linux Malware Detect, más conocido como **maldet** or **LMD**. Su instalación y manejo son sencillos por lo que sin duda es una buena opción para fortificar aun más.

5.2.3. Recomendaciones de seguridad para Node-RED

Node-RED es muy poco seguro en su estado por defecto haciéndolo vulnerable a ataques potenciales, por lo tanto, es necesario aplicar unas medidas de seguridad para reducir estas vulnerabilidades y conseguir fortificar este elemento de la aplicación.

Incluir usuario y contraseña

Por defecto cualquiera con acceso a Internet puede conectarse a nuestro dashboard de Node-RED y realizar modificaciones. Para evitar esta vulnerabilidad, se implementará un usuario y una contraseña para que solo el usuario legítimo pueda tener acceso y control de los paneles.

Lo primero que hay que hacer es acceder a la carpeta de Node-RED mediante el siguiente comando:

```
$ cd .node-red
```

Una vez dentro de la carpeta, se deberá modificar el archivo de configuración:

```
$ nano settings.js
```

Dentro del archivo se deberán descomentar las siguientes líneas de código:

```
adminAuth: {
    type: "credentials",
    users: [
        {
            username: "admin",
            password: "$2a$08$zZWtXTja0fB1pzD4sHCMyOCMYz2Z6dNbM6t18sJogENOMcxWV9DN.",
            permissions: "*"
        }
    ],
}
```

Ahora, se debe cambiar el hash del password que viene por defecto. Para ello debemos instalar el paquete node-red-admin con el siguiente comando:

```
$ sudo npm install -g node-red-admin
```

Cuando termine la instalación, se introducirá el siguiente comando para generar un hash de la contraseña que se desee:

```
$ node-red-admin hash-pw
```

Una vez se introduzca la nueva contraseña, se generará un hash el cual hay que copiar y pegar dentro de la variable password en el archivo de configuración de Node-RED. Quedará de la siguiente manera:

```
adminAuth: {  
  type: "credentials",  
  users: [{  
    username: "admin",  
    password: "$2a$08$k3v80qAC5r0/9v0yj37YCOCGYnyW9AvBTkyxJ56SFJtQJYGvvUX3a",  
    permissions: "*"  
  }]  
},
```

Cuando todo haya terminado, se reiniciará el servicio de Node-RED y la siguiente vez que se acceda a la URL, pedirá el usuario y contraseña como se aprecia en la figura 5.6:

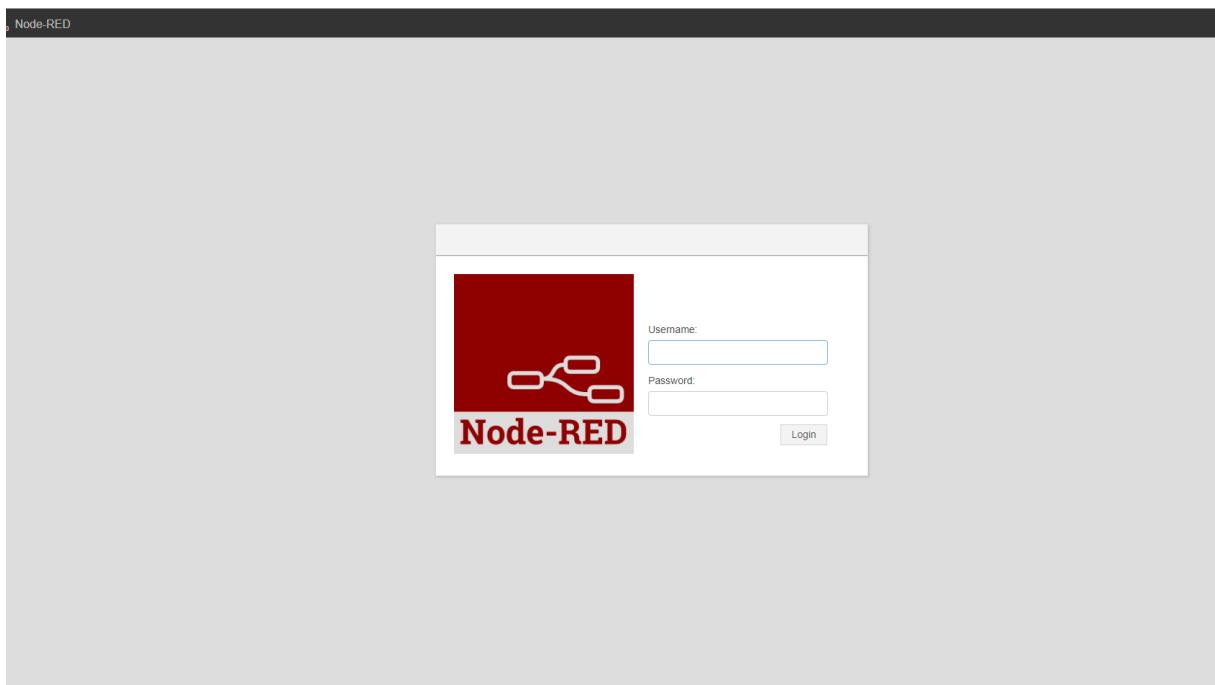


Figura 5.6: Login en Node-RED

Cambiar el puerto por defecto

Otra medida importante de seguridad es la de cambiar el puerto por defecto que tiene Node-RED. Es un riesgo importante de seguridad el mantener el puerto por

defecto que traen algunas aplicaciones y más si se va a exponer a Internet. Muchos ataques informáticos se enfocan a aplicaciones que tienen un puerto determinado por defecto y mantienen todas las configuraciones iniciales, estos ataques suelen ser automatizados por lo que tienen un alto grado de conseguir su propósito.

El puerto por defecto que tiene Node-RED es el 1880, en el proyecto este puerto será cambiado por el 1300. Para ello lo primero que se debe hacer es acceder al archivo de configuración de node-red igual que se ha hecho anteriormente.

Dentro del archivo se debe buscar el parámetro *uiPort* y se cambiará el 1880 por el 1300 quedando de la siguiente manera:

```
$ uiPort: process.env.PORT || 1300,
```

Después de modificar el archivo y guardarlo, se debe reiniciar el servicio Node-RED y habilitar el puerto en el Firewall si antes se encontraba deshabilitado.

Al finalizar, se volverá a ejecutar el Node-RED pero esta vez se accederá a la URL con el puerto 1300 para comprobar que todo funciona correctamente.

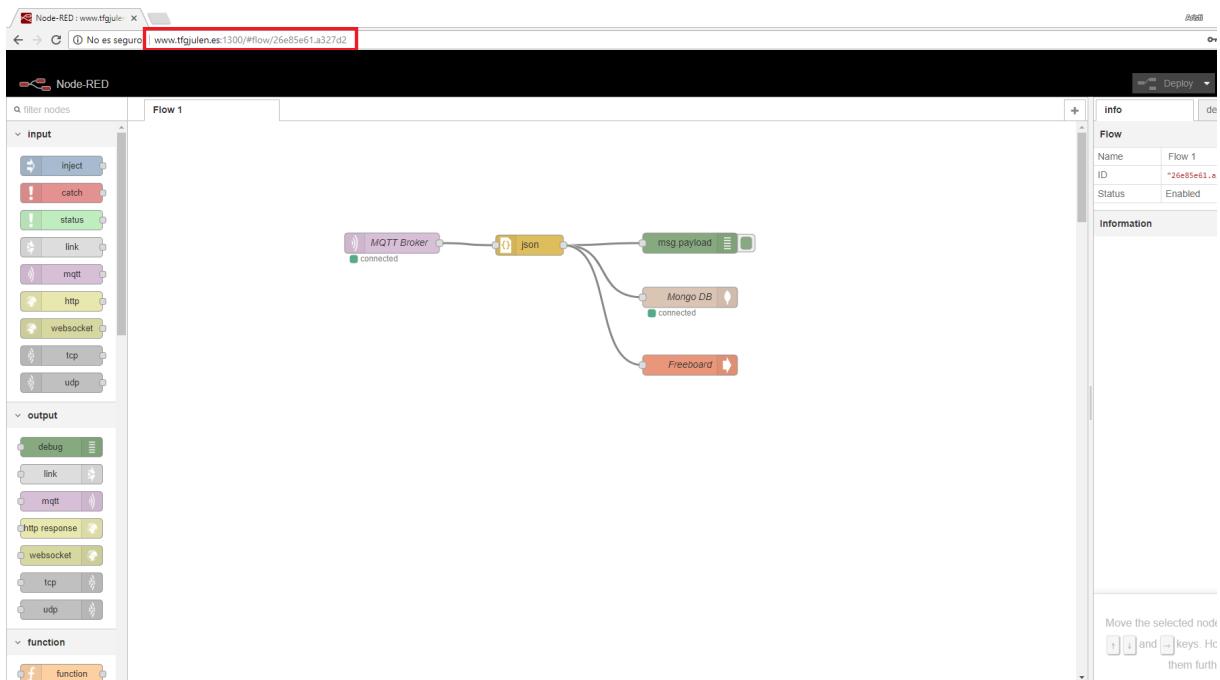


Figura 5.7: Node-RED sobre el puerto 1300

5.2.4. Recomendaciones de seguridad para MQTT

Proteger el bróker con usuario y contraseña

Hasta ahora cualquier usuario que conociese la IP de nuestro broker podía enviar mensajes o conectarse a él. En este apartado de proyecto se van a implementar usu-

rios y contraseñas para proteger el acceso al bróker y forzar así la autenticación de los clientes.

Para empezar se debe acceder a la carpeta principal de mosquitto:

```
$ cd /etc/mosquitto
```

Seguidamente, se utilizará una herramienta especial para generar el archivo de contraseñas que tiene incluido mosquitto. Después de ejecutar el siguiente comando, se deberá introducir la contraseña que se desee:

```
$ sudo mosquitto_passwd -c passwordfile Nuevo-Usuario
```

Después de crearse el archivo de contraseñas, se debe editar el archivo de configuración de mosquitto y añadir las dos siguientes líneas al final, para editar el archivo de configuración escribimos el siguiente comando:

```
$ sudo nano mosquitto.conf
```

Se deben añadir estas dos líneas al archivo:

```
allow_anonymous false  
password_file /etc/mosquitto/passwordfile
```

Al terminar de editar el documento, se debe reiniciar el servicio de mosquitto para que los cambios empiecen a funcionar. Después de reiniciar el servicio, se tendrá que tener en cuenta que cualquier conexión que se realice al bróker deberá ser autenticada antes de la conexión con el usuario y contraseña introducidos, si no la conexión será rechazada continuamente arrojando un error de autenticación.

En la aplicación este cambio afecta a la programación el Script del cliente MQTT ya que se deberán introducir líneas de código para establecer una conexión con usuario y contraseña.

```
client.username_pw_set("myUser", "myPassword")
```

Por la parte del Node-RED se deberá modificar la seguridad del nodo del bróker introduciendo también el usuario y la contraseña.

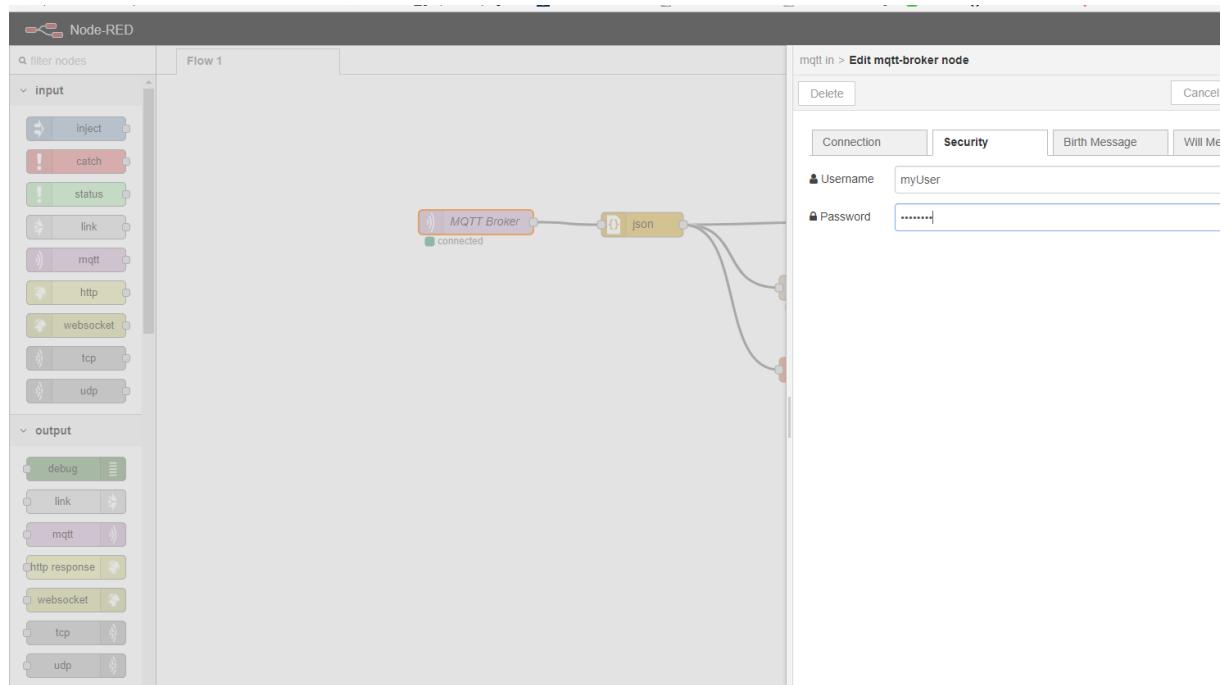


Figura 5.8: Seguridad en el nodo Broker MQTT

5.2.5. Quick Check Guide

Pregunta	Si	No
Has cambiado el usuario y contraseña por defecto de la Raspberry?	✓	
Has deshabilitado el usuario root?	✓	
Has modificado la configuración por defecto para evitar redirecciones y la aceptación de paquetes como si fuésemos un router?	✓	
Has deshabilitado el inicio de sesión root mediante SSH?	✓	
Has establecido un máximo de intentos de inicio de sesión y sesiones permitidas?	✓	
Has deshabilitado las contraseñas vacías?	✓	
Has deshabilitado el uso de PAM?	✓	
Has instalado un Firewall para la Raspberry?	✓	
Has creado un usuario nuevo y deshabilitado el usuario root en el VPS?	✓	
Has utilizado contraseñas fuertes para todos los usuarios?	✓	
Has establecido una política para mantener el software actualizado?	✓	
Has configurado el software correctamente?	✓	
Utilizarás SFTP en vez de FTP?	✓	
Tendrás cuidado con la descarga de archivos de sitios no oficiales?	✓	
Añadirás herramientas como un IDS o un scanner de malware?	✓	
Has incluido usuario y contraseña para proteger Node-RED?	✓	
Has cambiado el puerto por defecto de Node-RED?	✓	
Has protegido el bróker mqtt con usuario y contraseña?	✓	
Has deshabilitado la conexión de clientes anónimos con el bróker?	✓	
Has formado al personal para evitar ataques de ingeniería social?	✓	
Has establecido una política de copias de seguridad?	✓	
Has optado por instalar un UPS?	✓	

Cuadro 5.1: Quick Check Guide

5.3. Riesgos

La seguridad de la aplicación a aumentado en gran medida después de implantar las recomendaciones de seguridad vistas anteriormente, pero el sistema como todos los sistemas del mundo tiene riesgos.

El riesgo más importante que existe es que los mensajes enviados en la conexión MQTT no están cifrados, por lo que si alguien intercepta la conexión podría hacerse con el control de los datos. Para ello sería necesario el uso de protocolos de comunicación seguros como SSL o TLS. Este punto quedará reflejado en el apartado de trabajo futuro para su desarrollo en otro posible proyecto de fin de grado.

Parte IV

Caso de estudio

CAPÍTULO 6

MANUAL DE USUARIO DEL PROTOTIPO

En este apartado del proyecto se va a especificar el funcionamiento del prototipo creado.

Diagrama de funcionamiento

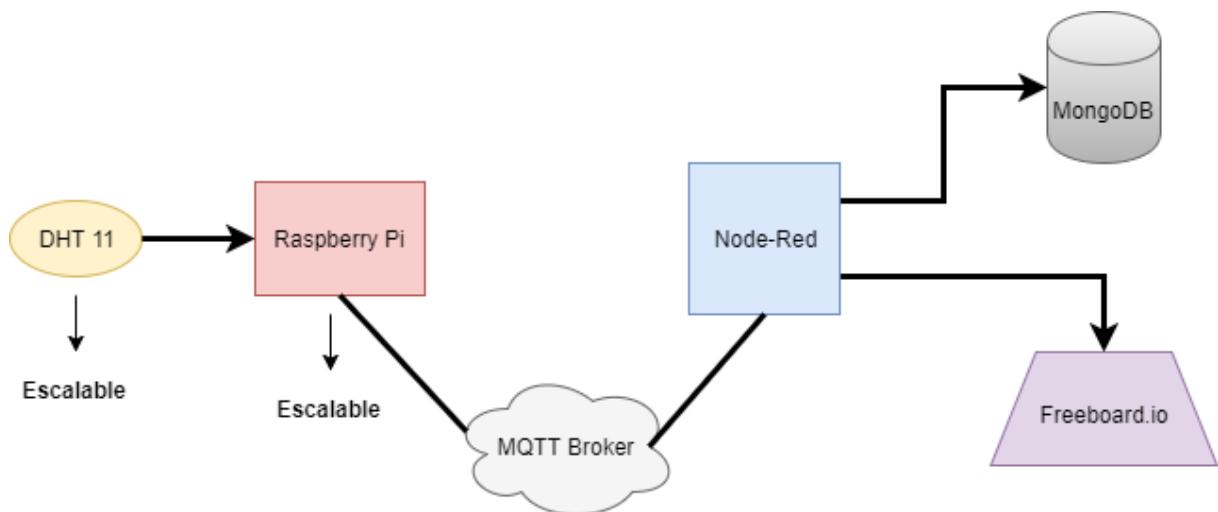


Figura 6.1: Diagrama de funcionamiento del prototipo

Como se puede observar en la Figura 6.1, el flujo de la aplicación comienza cuando el cliente MQTT programado dentro de la Raspberry Pi establece una conexión segura con el bróker y acto seguido recibe las medidas de temperatura y humedad desde el sensor DHT11. Este cliente crea objetos JSON que contienen la medida de temperatura, humedad y por último la fecha de la medición. Estos mensajes son enviados al bróker cada 15 segundos bajo el tópico "*temp/DHT11data*". Además de esto, el cliente también posee un tratamiento de errores simple.

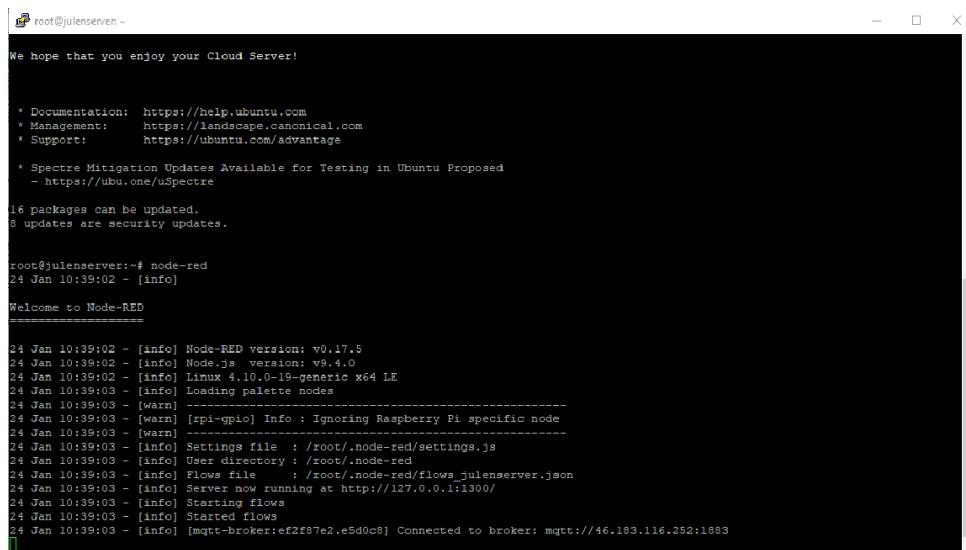
Por otro lado, en el VPS, Node-RED tiene que haber sido ejecutado para haber podido establecer la conexión con el cliente de la Raspberry. El bróker está siempre a la espera de recibir mensajes y redireccionarlos por lo que en la paleta de Node-RED se ha creado un nodo cliente MQTT que está suscrito al tópico antes mencionado.

Cuando el cliente alojado en Node-RED recibe los datos, se encarga de almacenarlos en la base de datos MongoDB que se ha creado anteriormente y los envía a la plataforma Freeboard.io. Estos datos llegan a Mongo y Freeboard en formato JSON para facilitar su integridad.

Por último, accediendo a la URL que ha creado Freeboard para nuestra paleta, se podrán observar los datos de las medidas en tiempo real haciendo que sea más sencillo para el usuario su visualización.

Flujo real del prototipo

Para que la aplicación comience su flujo deben estar Node-RED ejecutándose en el VPS y el script del cliente preparado en la Raspberry Pi.



```

root@julenserver:~-
We hope that you enjoy your Cloud Server!

* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage

* Spectre Mitigation Updates Available for Testing in Ubuntu Proposed
- https://ubu.one/uSpectre

16 packages can be updated.
0 updates are security updates.

root@julenserver:~# node-red
24 Jan 10:39:02 - [info] Welcome to Node-RED
=====
24 Jan 10:39:02 - [info] Node-RED version: v0.17.5
24 Jan 10:39:02 - [info] Node.js version: v9.4.0
24 Jan 10:39:02 - [info] Linux 4.10.0-19-generic x64 LE
24 Jan 10:39:03 - [info] Loading palette nodes
24 Jan 10:39:03 - [warn] -----
24 Jan 10:39:03 - [warn] [rpi-gpio] Info : Ignoring Raspberry Pi specific node
24 Jan 10:39:03 - [warn] -----
24 Jan 10:39:03 - [info] Settings file : /root/.node-red/settings.json
24 Jan 10:39:03 - [info] User directory : /root/.node-red
24 Jan 10:39:03 - [info] Flows file : /root/.node-red/flows_julenserver.json
24 Jan 10:39:03 - [info] Server now running at http://127.0.0.1:1300/
24 Jan 10:39:03 - [info] Starting flows
24 Jan 10:39:03 - [info] Started flows
24 Jan 10:39:03 - [info] [mqtt-broker:ef2f07e2.e5d0c8] Connected to broker: mqtt://46.183.116.252:1883

```

Figura 6.2: Comando para ejecutar Node-RED

Una vez el bróker de MQTT y Node-RED están a la espera, se ejecuta el script del cliente. Como se puede ver en la figura 6.3 se ha establecido la conexión correctamente, de lo contrario el script devolvería algún tipo de error.

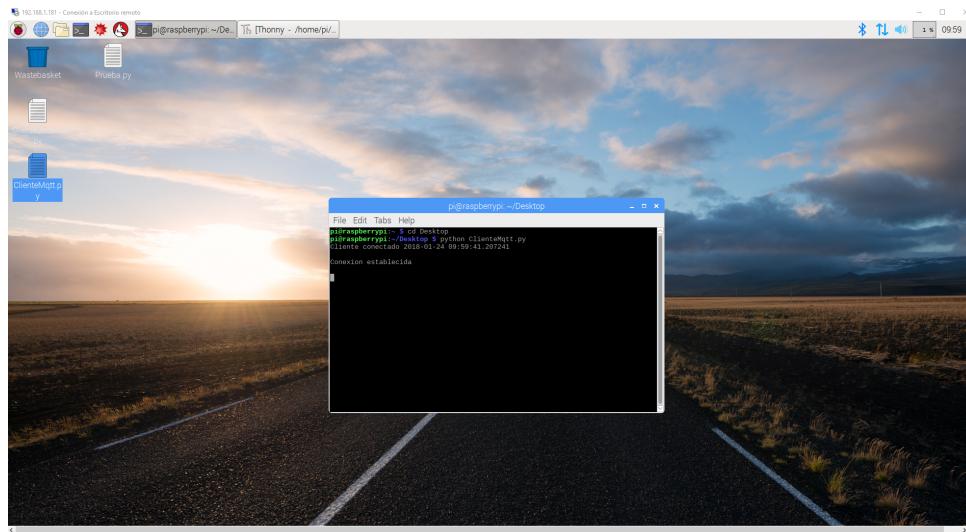


Figura 6.3: Ejecución del cliente MQTT

Los mensajes comienzan su transmisión hacia el bróker el cual se encuentra en espera para redirigir esos mensajes hacia los suscriptores. Si todo transcurre correctamente empezarán a llegar las medidas al cliente alojado en Node-RED como se puede observar en la figura 6.4, el nodo de "debug" empieza a mostrar en pantalla los mensajes que está recibiendo.

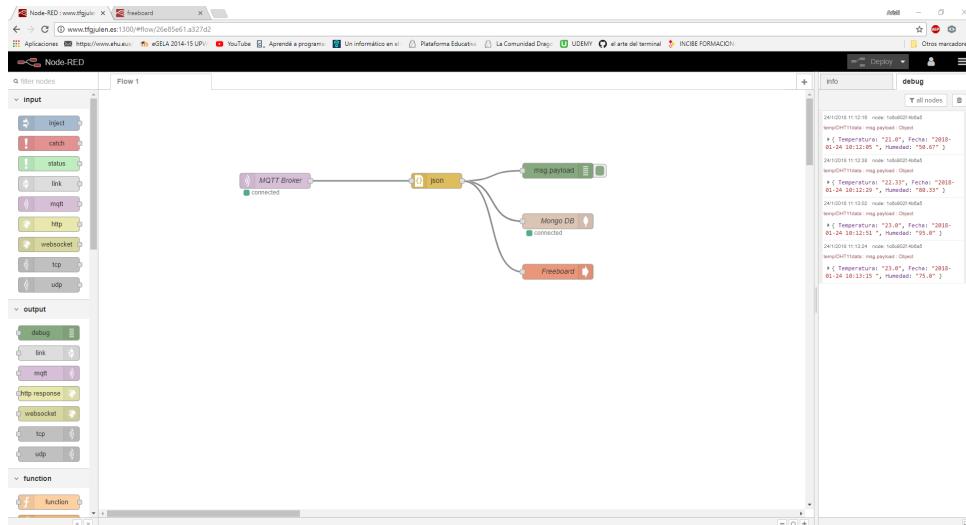


Figura 6.4: Mensajes recibidos por el cliente de Node-RED

Estos mensajes son almacenados en la base de datos de MongoDB, la cual realiza la conexión con el Nodo que se ha implementado. Este nodo contiene la IP del servidor más el puerto de MongoDB, además de la colección en la que queremos guardar los datos y la operación de insertar. En la figura 6.5 se comprueba que los datos han sido guardados correctamente.

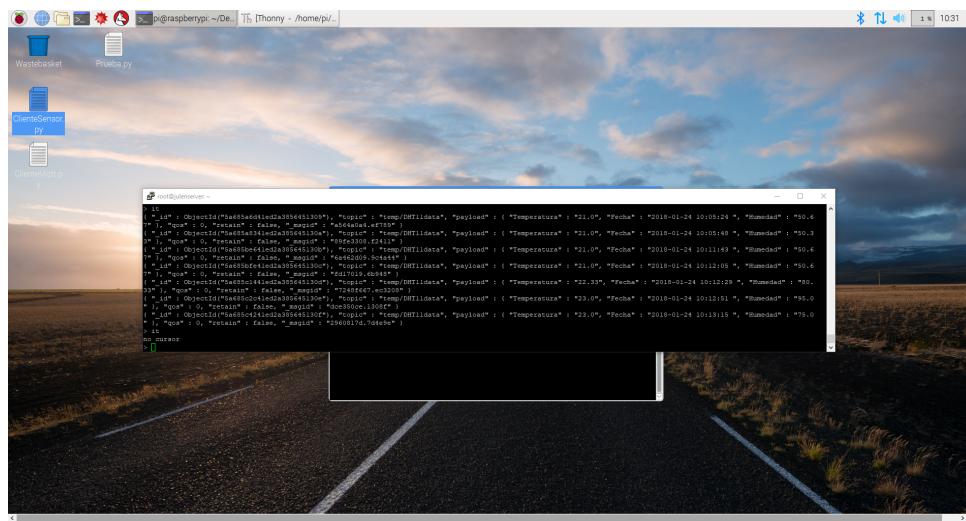


Figura 6.5: Últimas medidas guardadas en MongoDB

Por último, los mensajes son enviados a la plataforma Freeboard.io la cual contiene diferentes gráficos encargados de mostrar los datos de las últimas medidas. A continuación, en la figura 6.6 se observa la paleta creada con las medidas correctas en tiempo real.

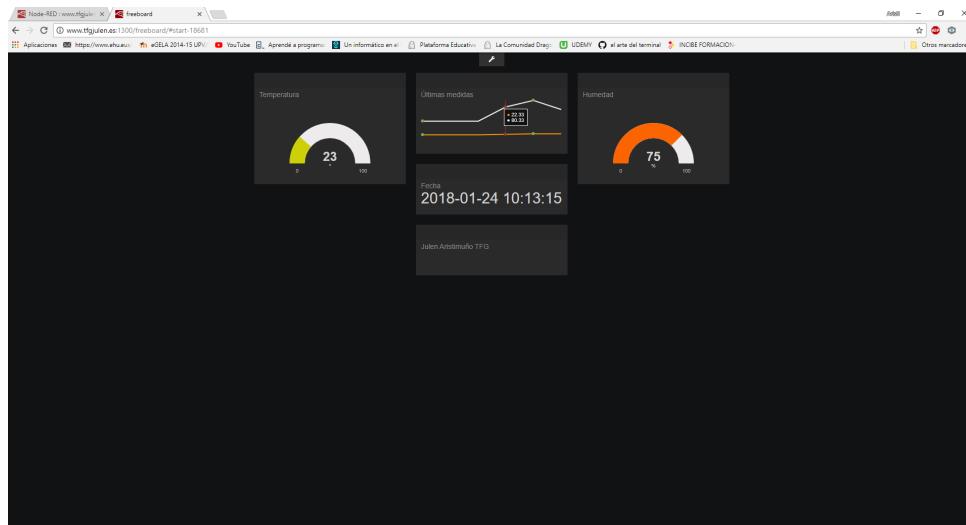


Figura 6.6: Visualización de las últimas medidas en Freeboard.io

Parte V

Análisis del trabajo

CAPÍTULO 7

CONCLUSIONES

En primer lugar, el trabajo realizado tiene como objetivo el de realizar una aplicación de envío de mensajes simulando un área de trabajo relacionado con el Internet de las Cosas donde miles de sensores son controlados por máquinas con el objetivo de recoger la mayor cantidad de datos posibles.

En nuestro caso hemos desarrollado una aplicación con únicamente un sensor y una placa base pero en el ámbito empresarial esto podría escalar hasta donde se quisiese.

Por otro lado, el objetivo más importante era el de empezar a añadir las bases de seguridad a aplicaciones de este tipo intentando erradicar uno de los peligros más grandes que amenaza hoy en día la informática dentro de la industria y se trata de los ataques informáticos que sufren las empresas. Cada vez son más y más los cibерdelincuentes que se aprovechan de la debilidad que poseen algunas empresas en el área de la seguridad y esto puede conllevar tanto robos de información como perder el control de las operaciones que realiza dicha empresa.

En este proyecto se han intentado enseñar las vulnerabilidades más populares hoy en día y como realizando unas operaciones de protección puede llegar a mitigarse.

En segundo lugar, se ha trabajado con protocolos de comunicación ligeros diseñados específicamente para el área elegida para el proyecto. Tecnologías ligeras como mosquitto tienen un gran futuro por la facilidad de integración que poseen. Otro punto importante es la sencillez que nos ha ofrecido Node-RED, la curva de aprendizaje es grande pero una vez adquieres conocimientos sobre los nodos y como gestionarlos las posibilidades que ofrece son enormes. Sin embargo, los desarrolladores tienen que aumentar la seguridad de estas plataformas para asegurar todas las conexiones y su cifrado.

En conclusión, este proyecto consta de dos aportes fundamentales en el área de las aplicaciones del Internet de las Cosas y de la seguridad informática. Por un lado el desarrollo de una arquitectura de transmisión de datos entre múltiples dispositivos con tecnologías ligeras. Por otro lado, se ha desarrollado una check guide (tabla 5.1) donde

se recogen la mayoría de las recomendaciones dadas para implantar unas bases de seguridad dentro de la aplicación y de la organización encargada de trabajar con ella.

CAPÍTULO 8

TRABAJO FUTURO

Este trabajo tiene un gran potencial de desarrollo, con el paso del tiempo nuevas tecnologías utilizables en este ámbito surgirán y con ello, nuevas vulnerabilidades en los sistemas.

El trabajo futuro podría dividirse en los siguientes puntos:

- Implementar SSL/TLS sobre la comunicación de mosquitto. Realmente veo este punto muy interesante para mejorar la seguridad de la aplicación aún más.
- Securizar MongoDB
- Integrar IPTables
- Desarrollar un sistema de claves publicas y privadas para iniciar las sesiones SSH
- Mejorar el sistema añadiendo nuevos sensores y tópicos con más clientes publicadores y suscriptores para acercarse aún más al ámbito empresarial.
- Mejorar la implantación del sistema de detección de intrusos utilizado en el proyecto ya que no funciona demasiado bien. Se podría añadir un nuevo sistema o mejorar el existente.
- Añadir nuevas vulnerabilidades que hayan aparecido a lo largo del tiempo.
- Mejorar la seguridad de la aplicación con nuevas técnicas o más avanzadas.

CAPÍTULO 9

VIABILIDAD

En este capítulo se analizan los requisitos del trabajo para cumplir los objetivos y se define una planificación de los recursos humanos, de tiempo y económicos. Además se estudian los riesgos que pueden dificultar, retrasar o impedir la realización de los objetivos.

9.1. Requisitos funcionales del trabajo

Estos son los requisitos que el proyecto debía cumplir:

- Desarrollo de una aplicación del sector IoT
- Uso de tecnologías ligeras modernas
- Comunicar diferentes máquinas
- Aplicar bases de seguridad en la aplicación
- Realizar un análisis de las diferentes vulnerabilidades existentes
- Realizar un estudio de las diferentes tecnologías candidatas para el prototipo
- Realizar una guía sobre principios básicos de seguridad para cualquier usuario

9.2. Planificación del tiempo

La planificación del tiempo del proyecto se realiza descomponiendo el trabajo en fases, tareas y entregables.

9.2.1. Estructura de descomposición del trabajo

El propósito de la EDT es organizar y definir el alcance del proyecto. La EDT establece una jerarquía de los distintos entregables y tareas que componen el trabajo.

La planificación del proyecto se establece siguiendo las figuras 9.1 y 9.2

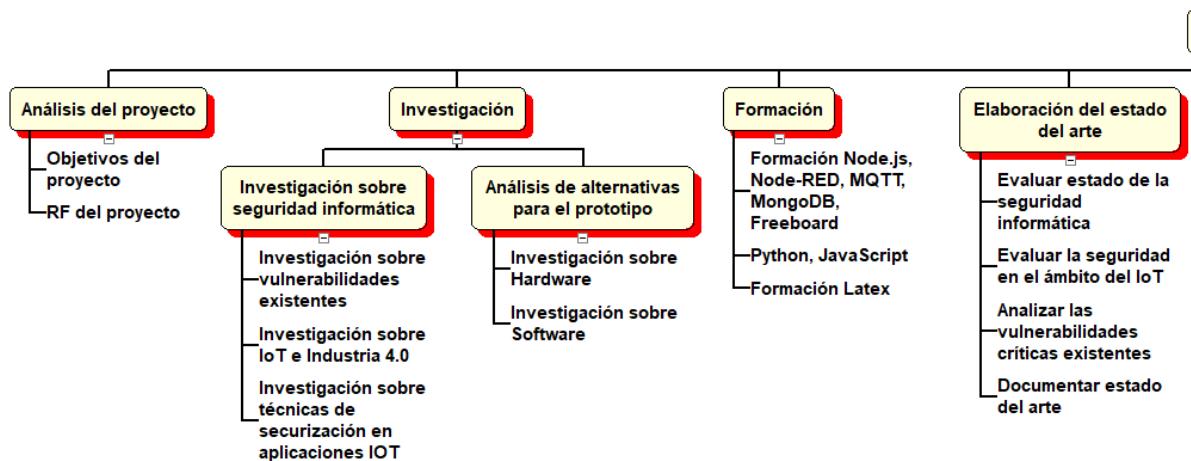


Figura 9.1: Estructura de Descomposición del Trabajo 1

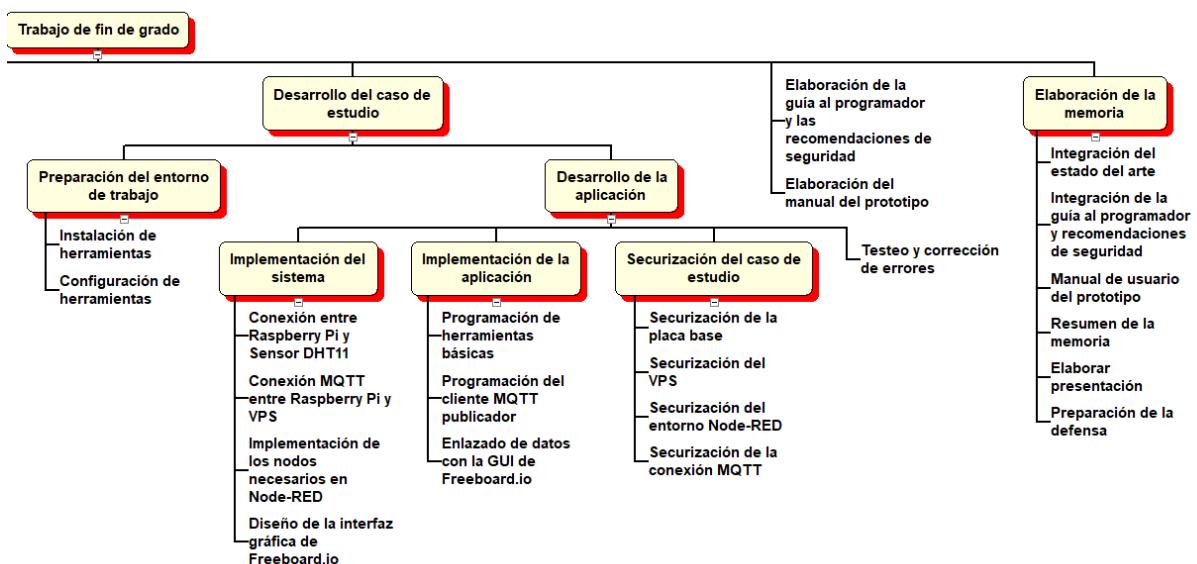


Figura 9.2: Estructura de Descomposición del Trabajo 2

9.2.2. Tareas

En esta sección se muestran y describen las tareas y entregables planificados en la realización del proyecto.

A continuación se muestra la lista de tareas que dispone el trabajo:

0. Análisis del proyecto

- 0.1. Objetivos del proyecto
- 0.2. RF del proyecto

1. Investigación

- 1.1. Investigación sobre seguridad informática
 - 1.1.1. Investigación sobre vulnerabilidades existentes
 - 1.1.2. Investigación sobre IoT e Industria 4.0
 - 1.1.3. Investigación sobre técnicas de securización en aplicaciones IOT
- 1.2. Análisis de alternativas para el prototipo
 - 1.2.1. Investigación sobre Hardware
 - 1.2.2. Investigación sobre Software

2. Formación

- 2.1. Formación Node.js, Node-RED, MQTT, MongoDB, Freeboard
- 2.2. Formación Python, JavaScript
- 2.3. Formación L^AT_EX

3. Elaboración del estado del arte

- 3.1. Evaluar estado de la seguridad informática
- 3.2. Evaluar la seguridad en el ámbito del IoT
- 3.3. Analizar las vulnerabilidades críticas existentes
- 3.4. Documentar estado del arte

4. Desarrollo del caso de estudio

- 4.1. Preparación del entorno de trabajo
 - 4.1.1. Instalación de herramientas
 - 4.1.2. Configuración de herramientas
- 4.2. Desarrollo de la aplicación
 - 4.2.1. Implementación del sistema
 - 4.2.1.1. Conexión entre Raspberry Pi y Sensor DHT11
 - 4.2.1.2. Conexión MQTT entre Raspberry Pi y VPS

- 4.2.1.3. Implementación de los nodos necesarios en Node-RED
 - 4.2.1.4. Diseño de la interfaz gráfica de Freeboard.io
 - 4.2.2. Implementación de la aplicación
 - 4.2.2.1. Programación de herramientas básicas
 - 4.2.2.2. Programación del cliente MQTT publicador
 - 4.2.2.3. Enlazado de datos con la GUI de Freeboard.io
 - 4.2.3. Securización del caso de estudio
 - 4.2.3.1. Securización de la placa base
 - 4.2.3.2. Securización del VPS
 - 4.2.3.3. Securización del entorno Node-RED
 - 4.2.3.4. Securización de la conexión MQTT
 - 4.2.4. Testeo y corrección de errores
5. **Elaboración de la guía al programador y las recomendaciones de seguridad**
 6. **Elaboración del manual del prototipo**
 7. **Elaboración de la memoria**
 - 7.1. Integración del estado del arte
 - 7.2. Integración de la guía al programador y recomendaciones de seguridad
 - 7.3. Manual de usuario del prototipo
 - 7.4. Resumen de la memoria
 - 7.5. Elaborar presentación
 - 7.6. Preparación de la defensa
 8. **Reuniones periódicas**

A partir de aquí, se explicará brevemente en qué consiste cada tarea:

Número: 0.1.

Nombre: Objetivos del proyecto.

Descripción: Definir los objetivos que tiene que cumplir el TFG.

Trabajo estimado: 6 horas.

Número: 0.2.

Nombre: RF del proyecto.

Descripción: Definir los Requisitos Funcionales del proyecto.

Trabajo estimado: 4 horas.

Número: 1.1.1.

Nombre: Investigación sobre vulnerabilidades existentes.

Descripción: Investigar las diferentes vulnerabilidades existentes.

Trabajo estimado: 30 horas.

Número: 1.1.2.

Nombre: Investigación sobre IoT e Industria 4.0.

Descripción: Investigar el estado del sector de la seguridad informática en la Industria 4.0 e IoT.

Trabajo estimado: 10 horas.

Número: 1.1.3.

Nombre: Investigación sobre técnicas de securización en aplicaciones IOT.

Descripción: Investigar diferentes técnicas de seguridad para aplicaciones similares al caso de estudio.

Trabajo estimado: 30 horas.

Número: 1.2.1.

Nombre: Investigación sobre Hardware.

Descripción: Investigar los mejores componentes de Hardware para desarrollar el caso de estudio.

Trabajo estimado: 20 horas.

Número: 1.2.2.

Nombre: Investigación sobre Software.

Descripción: Investigar los mejores componentes de Software para desarrollar el caso de estudio.

Trabajo estimado: 20 horas.

Número: 2.1.

Nombre: Formación Node.js, Node-RED, MQTT, MongoDB, Freeboard.

Descripción: Formación en los componentes elegidos para desarrollar la aplicación.

Trabajo estimado: 20 horas.

Número: 2.2.

Nombre: Formación Python, JavaScript.

Descripción: Formación en los lenguajes de programación que se van a utilizar para programar el caso de estudio.

Trabajo estimado: 8 horas.

Número: 2.3.

Nombre: Formación en LaTex.

Descripción: Formación en LaTex para realizar la memoria.

Trabajo estimado: 4 horas.

Número: 3.1.

Nombre: Evaluar estado de la seguridad informática.

Descripción: Redactar una visión general de la seguridad informática en la actualidad.

Trabajo estimado: 4 horas.

Número: 3.2.

Nombre: Evaluar la seguridad en el ámbito del IoT.

Descripción: Redactar una visión general de el sector del IoT.

Trabajo estimado: 4 horas.

Número: 3.3.

Nombre: Analizar las vulnerabilidades críticas existentes.

Descripción: Redactar de manera organizada las diferentes vulnerabilidades existentes.

Trabajo estimado: 4 horas.

Número: 3.4.

Nombre: Documentar estado del arte.

Descripción: Elaborar un estado del arte en base a toda la información recogida.

Trabajo estimado: 4 horas.

Número: 4.1.1.

Nombre: Instalación de herramientas.

Descripción: Instalar todas las herramientas necesarias para realizar la aplicación.

Trabajo estimado: 2 horas.

Número: 4.1.2.

Nombre: Configuración de herramientas.

Descripción: Configurar todas las herramientas necesarias para realizar la aplicación.

Trabajo estimado: 3 horas.

Número: 4.2.1.1.

Nombre: Conexión entre Raspberry Pi y Sensor DHT11.

Descripción: Realizar la conexión entre la Raspberry Pi y el sensor.

Trabajo estimado: 1 horas.

Número: 4.2.1.2.

Nombre: Conexión MQTT entre Raspberry Pi y VPS.

Descripción: Realizar la conexión entre la Raspberry Pi y el VPS.

Trabajo estimado: 8 horas.

Número: 4.2.1.3.

Nombre: Implementación de los nodos necesarios en Node-RED.

Descripción: Crear la paleta de Node-RED con los nodos necesarios.

Trabajo estimado: 4 horas.

Número: 4.2.1.4.

Nombre: Diseño de la interfaz gráfica de Freeboard.io.

Descripción: Diseñar la interfaz de Freeboard.io añadiendo los elementos necesarios.

Trabajo estimado: 2 horas.

Número: 4.2.2.1.

Nombre: Programación de herramientas básicas.

Descripción: Programar las herramientas necesarias para realizar la aplicación.

Trabajo estimado: 8 horas.

Número: 4.2.2.2.

Nombre: Programación del cliente MQTT publicador.

Descripción: Programar el cliente encargado de transmitir los mensajes desde la Raspberry Pi.

Trabajo estimado: 6 horas.

Número: 4.2.2.3.

Nombre: Enlazado de datos con la GUI de Freeboard.io.

Descripción: Recibir los datos en Freeboard.io y gestionarlos para poder visualizarlos.

Trabajo estimado: 4 horas.

Número: 4.2.3.1.

Nombre: Securización de la placa base.

Descripción: Aplicar técnicas básicas de seguridad en la Raspberry Pi.

Trabajo estimado: 12 horas.

Número: 4.2.3.2.

Nombre: Securización del VPS.

Descripción: Aplicar técnicas básicas de seguridad en el VPS.

Trabajo estimado: 12 horas.

Número: 4.2.3.3.

Nombre: Securización del entorno Node-RED.

Descripción: Aplicar técnicas básicas de seguridad en la plataforma Node-RED.

Trabajo estimado: 12 horas.

Número: 4.2.3.4.

Nombre: Securización de la conexión MQTT.

Descripción: Aplicar técnicas básicas de seguridad en el bróker MQTT.

Trabajo estimado: 16 horas.

Número: 4.2.4.

Nombre: Testeo y corrección de errores.

Descripción: Realizar teses y correcciones para asegurar el correcto funcionamiento de la aplicación.

Trabajo estimado: 8 horas.

Número: 5.

Nombre: Elaboración de la guía al programador y las recomendaciones de seguridad.

Descripción: Redactar un apartado con la guía al programador para realizar la aplicación junto a las recomendaciones de seguridad.

Trabajo estimado: 8 horas.

Número: 6.

Nombre: Elaboración del manual del prototipo.

Descripción: Redactar un apartado con el manual del prototipo donde se detalla el funcionamiento de la aplicación.

Trabajo estimado: 8 horas.

Número: 7.1.

Nombre: Integración del estado del arte.

Descripción: Integrar el estado del arte en la memoria del proyecto.

Trabajo estimado: 4 horas.

Número: 7.2.

Nombre: Integración de la guía al programador y recomendaciones de seguridad.

Descripción: Integrar la guía al programador y las recomendaciones a la memoria.

Trabajo estimado: 2 horas.

Número: 7.3.

Nombre: Manual de usuario del prototipo.

Descripción: Integrar el manual de usuario a la memoria.

Trabajo estimado: 2 horas.

Número: 7.4.

Nombre: Resumen de la memoria.

Descripción: integrar el resumen y los anexos a la memoria.

Trabajo estimado: 2 horas.

Número: 7.5.

Nombre: Elaborar presentación.

Descripción: Elaborar la presentación del proyecto.

Trabajo estimado: 6 horas.

Número: 7.6.

Nombre: Preparación de la defensa.

Descripción: Preparar la defensa del proyecto.

Trabajo estimado: 6 horas.

9.2.3. Entregables

Al finalizar el proyecto se entregará el software desarrollado junto con un manual técnico y de uso que detallará de forma sencilla y concisa la compilación y ejecución del trabajo junto a todas las mejoras de seguridad.

9.2.4. Agenda del proyecto

Para realizar el proyecto se trabajará una media jornada, es decir, 4 horas diarias de Lunes a Viernes y se seguirá el calendario laboral oficial del BOE para Álava (Tabla 9.1). El desarrollador del proyecto podrá realizarlo junto su jornada laboral ya que se ha planificado de esta forma. Así se estima que el trabajo se realizará desde el día Lunes 16 de Octubre de 2017 al Viernes 2 de Febrero del 2018.

Fecha	Evento
12 de Octubre	Fiesta Nacional de España
6 de Diciembre	Día de la Constitución
8 de Diciembre	Inmaculada Concepción
25 de Diciembre	Navidad
1 de Enero	Año Nuevo
6 de Enero	Día de Reyes

Cuadro 9.1: Calendario de días festivos

Se estima lo siguiente:

- Trabajo total estimado: 304 horas
- Duración total estimada: 304 horas
- Costo estimado: 5.261,00 €

La gestión del costo será realizada en uno de los siguientes apartados.

9.2.5. Cronograma

El diagrama de Gantt expone el tiempo de duración previsto para las diferentes tareas a lo largo del tiempo total determinado para el proyecto. En las figuras 9.3 y 9.4 se puede observar el tiempo de duración de cada tarea.

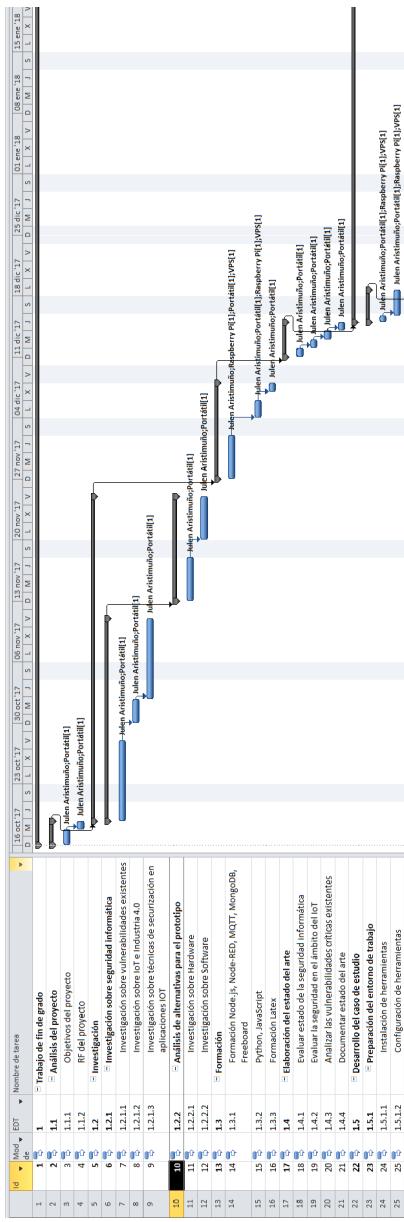


Figura 9.3: Diagrama de Gantt 1

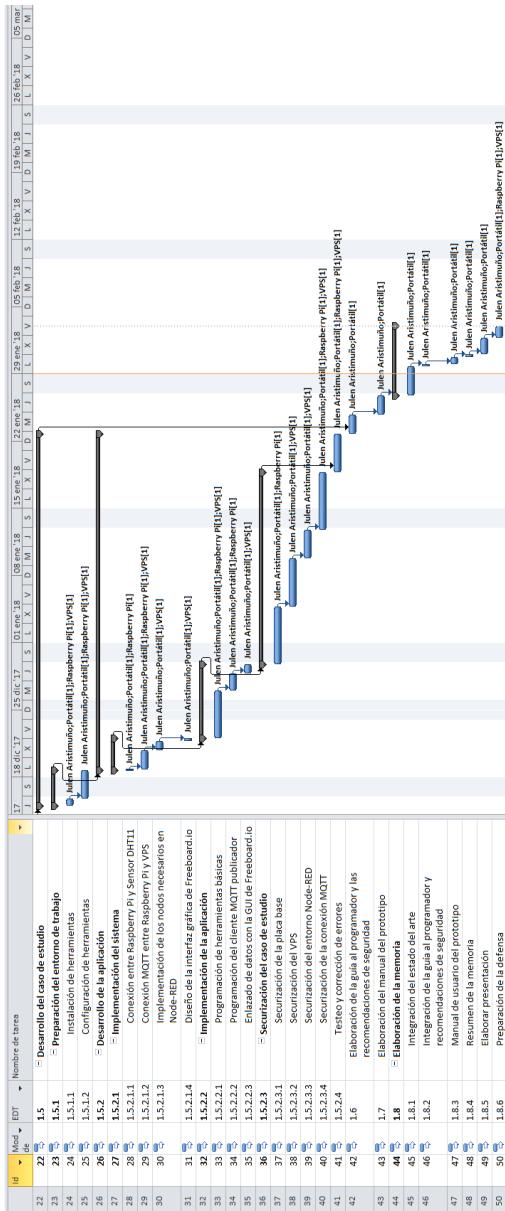


Figura 9.4: Diagrama de Gantt 2

9.3. Gestión de costos

En esta sección se muestran los recursos materiales y humanos que se utilizarán en la realización del trabajo.

	Nombre del recurso	Tipo	Etiqueta de	Iniciales	Grupo	Capacidad	Tasa	Tasa horas extra	Costo/Uso	Acumular	Calendario
1	Julen Aristimuño	Trabajo		J		100%	17,00 €/hora	0,00 €/hora	0,00 €	Prorratoe	Estándar
2	Portátil	Material		P			0,00 €		1,50 €	Prorratoe	
3	Raspberry Pi	Material		R			0,00 €		1,50 €	Prorratoe	
4	VPS	Material		V			0,00 €		1,50 €	Prorratoe	

Figura 9.5: Tabla de recursos

9.3.1. Presupuesto

Para realizar la estimación del presupuesto se han tenido en cuenta los datos que aparecen de las siete siguientes tablas. En ellas se indica lo que cobran los recursos humanos, que tienen una tasa estándar por hora correspondiente a 17 €.

Los recursos materiales se consideran con un coste por uso de 1,50 € en concepto de luz y otros gastos que se cobran independiente del tiempo de uso.

Los datos de tiempo y trabajo han sido extraídos de las siguientes vistas de Microsoft Project. En la realización de este presupuesto se tiene en cuenta tanto los recursos materiales como las licencias de software necesarias para el desarrollo del trabajo. Al haber utilizado todo software libre a excepción del sistema operativo Windows y la licencia de Microsoft Project los gastos de materiales van a ser reducidos.

Para el cálculo de las amortizaciones se ha considerado un tiempo de amortización de 3 años. El cálculo del coste unitario de amortización es la división entre el coste unitario y el tiempo de amortización. Se considera el tiempo de amortización como 3 años . 200 días laborables . 8 horas = 4800 horas.

Concepto	Coste
Julen Aristimuño	17,00 €/hora

Cuadro 9.2: Coste de recursos trabajo

Concepto	Coste
PC Portátil	649,99 €
Raspberry Pi 3	38,00 €
Sensor DHT 11	4,50 €
VPS	40,00 €

Cuadro 9.3: Coste recursos materiales (Hardware)

Concepto	Coste
Windows 10	145,00 €
Microsoft Project 2013	1.369,00 €
Raspbian	0,00 €
Ubuntu Server 17.04	0,00 €
MQTT	0,00 €
Node.js	0,00 €
Node-RED	0,00 €
MongoDB	0,00 €
Freeboard.io	0,00 €

Cuadro 9.4: Coste recursos materiales (Software)

Concepto	Trabajo (h)	Trabajo horas extra	Coste	Coste Horas extra	Importe
Julen Aristimuño	304	0	17,00 €/h	0	5168,00 €
TOTAL					5168,00 €

Cuadro 9.5: Costo recursos trabajo

Concepto	Unidades	Coste	Importe
PC Portátil	1	1,50 €	36 x 1,50 €= 54 €
Raspberry Pi 3	1	1,50 €	12 x 1,50 €= 18 €
VPS	1	1,50 €	14 x 1,50 €= 21 €
TOTAL			93,00 €

Cuadro 9.6: Coste recursos materiales

Concepto	Coste unitario	T. Amortización	C.U. Amortización	T. Uso	Importe
PC Portátil	649,99 €	4800 h	0,135414 €/h	304h	41,16 €
Raspberry Pi 3	38,00 €	4800 h	0,007917 €/h	98h	0,77 €
VPS	40,00 €	4800 h	0,008333 €/h	113h	0,94 €
Windows 10	145,00 €	4800 h	0,030208 €/h	304h	9,18 €
Microsoft Project 2013	1.369,00 €	4800 h	0,285208 €/h	19h	5,41 €
Raspbian	0,00 €	4800 h	0,000000 €/h	98h	0,00 €
Ubuntu Server 17.04	0,00 €	4800 h	0,000000 €/h	113h	0,00 €
MQTT	0,00 €	4800 h	0,000000 €/h	63h	0,00 €
Node-RED	0,00 €	4800 h	0,000000 €/h	49h	0,00 €
MongoDB	0,00 €	4800 h	0,000000 €/h	33h	0,00 €
Freeboard.io	0,00 €	4800 h	0,000000 €/h	39h	0,00 €
TOTAL					57,46 €

Cuadro 9.7: Amortizaciones de Hardware y Software

Concepto	Importe
Recursos de trabajo (R.T.)	5.168,00 €
Recursos Materiales (R.M.)	93,00 €
Costo fijo	0,00 €
Amortizaciones	57,46 €
SUMA	5.318,46 €
Gastos generales (10 %)	531,85 €
Beneficio (15 %)	797,77 €
SUBTOTAL	6.648,08 €
IVA (21 %)	1.396,10 €
TOTAL	8.044,18 €

Cuadro 9.8: Total presupuesto

Tal y como se puede apreciar en la tabla 9.8, el coste total del proyecto asciende a la cifra de **ochenta mil cuarenta y cuatro con dieciocho euros** (8.044,18 €)

9.4. Gestión de riesgos

En este apartado se hace un análisis sobre las amenazas que pueden perjudicar el desarrollo del trabajo. Para ello se han evaluado una serie de riesgos y se ha valorado la gravedad del impacto en el trabajo.

Riesgo	Peligro
Dedicación no exclusiva al trabajo	Media
Incidencias con los materiales	Media
Cambios o ampliación de requisitos	Media
Estancamiento a la hora de programar	Media
Enfermedades	Alta
Pérdida de información	Alta
Planificación muy optimista	Alta
Elección equivocada de tecnología	Alta

Cuadro 9.9: Riesgos

9.4.1. Explicación de los riesgos y plan de contingencia

A continuación se describe con mas detalle los riesgos de la tabla 9.9:

- Dedicación no exclusiva al trabajo
 - Descripción: Podría suceder que el desarrollador no pudiese dedicarse exclusivamente al trabajo.
 - Probabilidad: 20 %
 - Peligrosidad: Media
 - Medidas preventivas: No previsible.
 - Medidas correctoras: Se elevará el numero de horas al día dedicadas al trabajo para eliminar el retraso. Si es necesario, se ampliaría el plazo de entrega en el calendario.
- Incidencias con los materiales
 - Descripción: Podría suceder que sucediese una incidencia con los recursos materiales.
 - Probabilidad: 15 %
 - Peligrosidad: Media
 - Medidas preventivas: Mantener todos y cada uno de los materiales. Tener acceso a materiales que puedan suplir a los principales si se da el caso.
 - Medidas correctoras: Buscar una solución ya sea reparando los materiales o adquiriendo unos nuevos.
- Cambios o ampliación de requisitos
 - Descripción: Podría darse el caso de que los requisitos no cubran las necesidades establecidas o se descubran nuevos según avanza el proyecto.

- Probabilidad: 30 %
 - Peligrosidad: Media
 - Medidas preventivas: No previsible
 - Medidas correctoras: Adaptar el desarrollo de estos requisitos ampliando la jornada laboral o modificando el calendario.
- Estancamiento a la hora de programar
 - Descripción: Podría llegar a darse algún problema de difícil solución que paralizaría el avance del proyecto.
 - Probabilidad: 35 %
 - Peligrosidad: Media
 - Medidas preventivas: Se estudiará con detalle las herramientas que se usarán.
 - Medidas correctoras: Dedicar tiempo en la formación modificando alguna otra tarea que en principio sea de menor dificultad.
 - Enfermedades
 - Descripción: Podría darse el caso de que el desarrollador padeciera alguna enfermedad o sufriese un accidente.
 - Probabilidad: 5 %
 - Peligrosidad: Alta
 - Medidas preventivas: No previsible.
 - Medidas correctoras: Dependiendo de la causa de la baja continuar si se puede. En caso de no ser posible, parar el proyecto y ampliar el calendario.
 - Pérdida de información
 - Descripción: Podría suceder que la información del proyecto desapareciese o se borrase.
 - Probabilidad: 5 %
 - Peligrosidad: Alta
 - Medidas preventivas: Establecer una política de copias de seguridad para mantener la información guardada en diferentes puntos.
 - Medidas correctoras: Extraer la información de una copia de seguridad anterior.
 - Planificación muy optimista
 - Descripción: Podría darse el caso de que la planificación inicial fuera completamente errónea.
 - Probabilidad: 25 %

- Peligrosidad: Alta
 - Medidas preventivas: Intentar establecer una planificación algo pesimista contando con algún retraso.
 - Medidas correctoras: Modificar la planificación procurando no retrasas mucho la entrega prevista.
- Elección equivocada de tecnología
- Descripción: Es posible que alguna de las tecnologías escogidas no sea la más adecuada para realizar el trabajo.
 - Probabilidad: 40 %
 - Peligrosidad: Alta
 - Medidas preventivas: Realizar un análisis de las alternativas para poder estar seguros de que herramienta se va a seleccionar.
 - Medidas correctoras: Modificar la planificación integrando la nueva tecnología lo antes posible.

Parte VI

Apéndices y bibliografía

APÉNDICE A

GLOSARIO

Address Resolution Protocol (ARP) El protocolo de resolución de direcciones es responsable de convertir las dirección de protocolo de alto nivel a direcciones de red físicas. 17–19

Advanced Encryption Standard (AES) Es un esquema de cifrado por bloques adoptado como un estándar de cifrado por el gobierno de los Estados Unidos. 28

Autonomous System (AS) Un Sistema Autónomo se define como un grupo de redes IP que poseen una política de rutas propia e independiente. 20

Binary JSON (BSON) BSON es un formato de intercambio de datos usado principalmente para su almacenamiento y transferencia en la base de datos MongoDB. 52

Border Gateway Protocol (BGP) Es el protocolo encargado de establecer las rutas que deben seguir los paquetes dentro de las comunicaciones de Internet. 20

Central Processing Unit (CPU) La Unidad Central de Procesamiento, es la parte encargada de procesar todas las instrucciones y datos del software y del hardware, motivo por el cual constituye el elemento más importante del computador. 16

Cookie Una cookie es una pequeña información enviada por un sitio web y almacenada en el navegador del usuario, de manera que el sitio web puede consultar la actividad previa del usuario. 20, 21

Cyber Physical System (CPS) Un Sistema ciberfísico es un mecanismo controlado o monitoreado por algoritmos basados en computación y estrechamente integrados con internet. 7

Denial of Service (DoS) Denegación de Servicio. 15, 16, 18, 19

Distributed Denial of Service (DDoS) Denegación de Servicio Distribuido. 16

Driver Un controlador de dispositivo o driver es un programa informático que permite al sistema operativo interaccionar con un periférico, haciendo una abstracción del hardware y proporcionando una interfaz para utilizar el dispositivo. 37

File Transfer Protocol (FTP) Es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP, basado en la arquitectura cliente-servidor. 77

Grupo hacktivista Hacktivismo se entiende normalmente como la utilización no-violenta de herramientas digitales ilegales o legalmente ambiguas persiguiendo fines políticos. 16

Hash Los hash o funciones de resumen son algoritmos que consiguen crear a partir de una entrada una salida alfanumérica de longitud normalmente fija que representa un resumen de toda la información que se le ha dado. 26, 27

Information and Communication Technology (ICT) Técnologías de la información y la comunicación. 7

Internet of Things (IoT) El Internet de las Cosas. 7, 9, 13, 14, 16, 54, 55, 57, 58

Internet Service Provider (ISP) El proveedor de servicios de Internet es la empresa que brinda conexión a Internet a sus clientes. 16, 20

Intrusion Detection System (IDS) Es un programa de detección de accesos no autorizados a un computador o a una red. 75, 77

JavaScript Object Notation (JSON) Es un formato de texto ligero para el intercambio de datos. 52, 59, 66

Local Area Network (LAN) Una red de área local o LAN es una red de computadoras que abarca un área reducida a una casa, un departamento o un edificio. 19

Man in The Middle (MiTM) Ataque de Hombre en el Medio. 16, 17, 19

Open Web Application Security Project (OWASP) Es un proyecto de código abierto dedicado a determinar y combatir las causas que hacen que el software sea inseguro. 33

Proxy Un proxy es un equipo informático que hace de intermediario entre las conexiones de un cliente y un servidor de destino, filtrando todos los paquetes entre ambos. 44

Puerto En informática, un puerto es una interfaz a través de la cual se pueden enviar y recibir los diferentes tipos de datos. 79, 80

Representational State Transfer (REST) La transferencia de estado representacional o REST es un estilo de arquitectura software para sistemas hipermedia distribuidos como la World Wide Web. 55

Root Root es una cuenta de usuario que tiene un control absoluto de todo lo que ocurre en un sistema. 72–74, 76, 77

Secure Shell (SSH) Es el nombre de un protocolo y del programa que lo implementa, y sirve para acceder servidores privados a través de una puerta trasera o backdoor. 73, 74

Secure Sockets Layer (SSL) Son los protocolos de seguridad de uso común que establecen un canal seguro entre dos ordenadores conectados a través de Internet o de una red interna. 16, 21, 27, 28, 44, 54, 84, 96

SSH File Transfer Protocol (SFTP) Este protocolo tiene el mismo cometido que su versión no segura, pero utilizando el puerto 22 para realizar la conexión, añadiendo la característica de FTP a SSH. 77

Transport Layer Security (TLS) Son protocolos criptográficos que proporcionan comunicaciones seguras por una red. 16, 21, 44, 54, 84, 96

Uncomplicated Firewall (UFW) Es un cortafuegos diseñado para ser de fácil uso desarrollado por Ubuntu. Utiliza la línea de comandos para configurar las iptables usando un pequeño número de comandos simples. 76

Virtual Private Server (VPS) Es un método de particionar un servidor físico en varios servidores de tal forma que todo funcione como si se estuviese ejecutando en una única máquina. 59, 60, 77

APÉNDICE B

CONEXIÓN GPIO ENTRE RPI Y EL SENSOR DHT 11

En este apartado de los apéndices se explicará como se ha realizado la conexión entre la Raspberry Pi y el sensor DHT11.

Los materiales necesarios para realizar la conexión son los siguientes:

- Raspberry Pi 3 Model B
- Sensor DHT11
- Mini protoboard
- Jumpers de conexión

Antes de realizar la conexión se debe conocer los diferentes pines GPIO existentes en la Raspberry, para ello se puede observar la figura B.1 [57].

Raspberry Pi 3 GPIO Header			
Pin#	NAME	NAME	Pin#
01	3.3v DC Power	DC Power 5v	02
03	GPIO02 (SDA1 , I ² C)	DC Power 5v	04
05	GPIO03 (SCL1 , I ² C)	Ground	06
07	GPIO04 (GPIO_GCLK)	(TXD0) GPIO14	08
09	Ground	(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)	(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)	Ground	14
15	GPIO22 (GPIO_GEN3)	(GPIO_GEN4) GPIO23	16
17	3.3v DC Power	(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)	Ground	20
21	GPIO09 (SPI_MISO)	(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)	(SPI_CE0_N) GPIO08	24
25	Ground	(SPI_CE1_N) GPIO07	26
27	ID_SD (I ² C ID EEPROM)	(I ² C ID EEPROM) ID_SC	28
29	GPIO05	Ground	30
31	GPIO06	GPIO12	32
33	GPIO13	Ground	34
35	GPIO19	GPIO16	36
37	GPIO26	GPIO20	38
39	Ground	GPIO21	40

Rev. 2
29/02/2016

www.element14.com/RaspberryPi

Figura B.1: Pines GPIO en Raspberry Pi 3

Otro aspecto importante a conocer es el modelo de sensor que se va a utilizar. En el caso de estudio se ha utilizado un sensor DHT11 con 3 pines y una resistencia integrada. El DHT11 utilizado es el mismo modelo del que se aprecia en la figura B.2

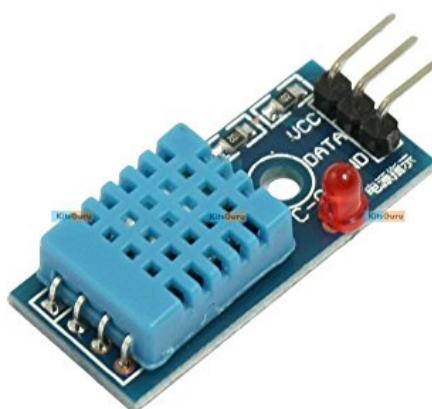


Figura B.2: Sensor DHT11

La conexión de los pines del sensor con la Raspberry se realizará siguiendo la siguiente tabla B.1:

DHT pin	Señal	Pi pin
1	Tierra	6
2	Data/Out	7 (GPIO04)
3	3.3V	1

Cuadro B.1: Conexión de pines

Después de realizar la conexión, el prototipo conectado queda como se puede apreciar en la figura B.3

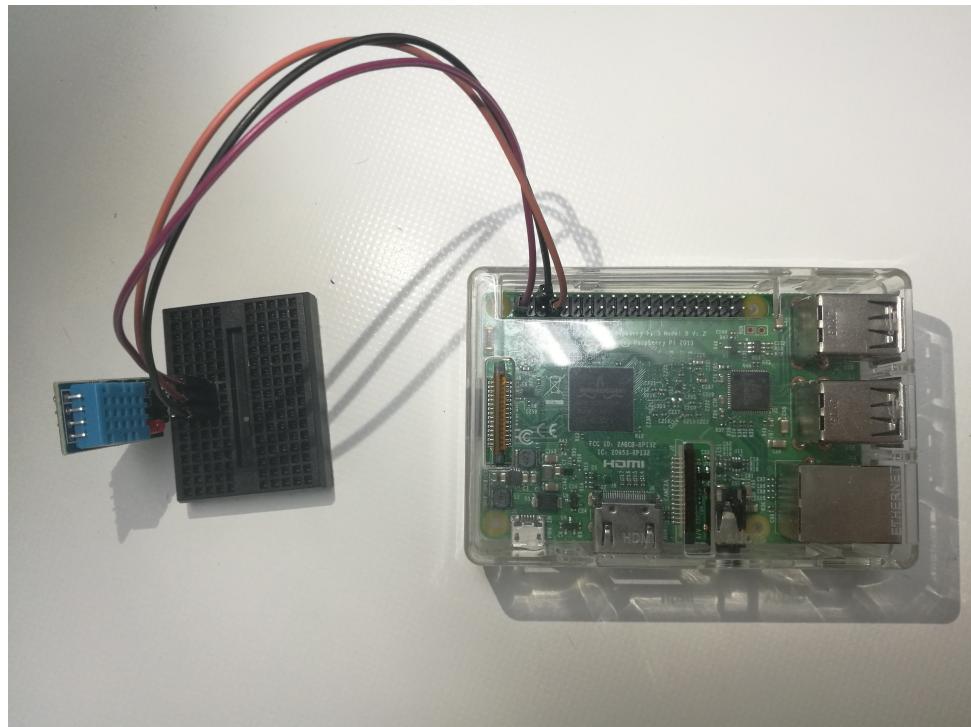


Figura B.3: Conexión completa del prototipo

APÉNDICE C

CREACIÓN Y MANEJO DE LA BASE DE DATOS MONGODB

En este apéndice se explicará como se ha creado la base de datos donde se guardan los datos de las medidas y cómo manejárla para ver los datos y realizar cualquier acción sobre ella [58].

Lo primero que se debe hacer es escribir el siguiente comando en la terminal:

```
$ mongo
```

Por defecto mongo se conectará al servidor que se encuentra escuchando en el puerto 27017.

Para ver las bases de datos que se encuentran en el servidor se debe utilizar el siguiente comando:

```
show dbs
```

Para crear una nueva base de datos o acceder a la base de datos que se necesite existe el comando que se observa a continuación:

```
use miNuevaBase
```

Con el comando anterior se crearía una base de datos llamada "miNuevaBase.^º en caso de estar creada se accedería a ella.

Para comprobar la base de datos que se está usando se utiliza el comando siguiente:

```
db
```

Una vez se haya establecido la conexión a la base de datos deseada el siguiente

paso es crear colecciones.

Para ello primero se debe crear una variable que posteriormente será introducida en la base. Por ejemplo, se creará una variable de la siguiente forma:

```
x = { name : "kali" }
```

A continuación se introduce dicha variable en una colección llamada "nuevaColección" de la siguiente manera:

```
db.nuevaColeccion.insert( x )
```

Al introducir la primera variable mongo creará la colección con el nombre que se le ha introducido.

Para verificar que la colección se ha creado sin errores se introducirá el siguiente comando:

```
show collections
```

Por último con el comando que se muestra a continuación se podrán observar las tuplas existentes dentro de la colección:

```
db.nuevaColeccion.find()
```

APÉNDICE D

SCRIPT CLIENTE PUBLICADOR MQTT

Se ha programado el siguiente Script que realiza la función de recibir los datos desde el sensor, realizar una conexión con el bróker y enviar los datos en formato JSON.

```
#!/usr/bin/env python

from __future__ import division

import Adafruit_DHT
import paho.mqtt.client as mqtt
import subprocess
import datetime
import time
import json

# Broker data
broker = brokerIP #En esta variable introducir la IP del broker a utilizar
broker_port = 1883
timeout_reconnect = 60
broker_topic = "#" #All topics have to be read

# Topic to send payload
topic = "temp/DHT11data"

def on_connect(client, userdata, flags, rc):
    if int(str(rc)) == 0:
        print("Conexion establecida")
        print ""
    else:
        print("Error result code: " + str(rc))

def on_message(client, userdata, msg):
    if msg.topic == "temp/test":
        if str(msg.payload) != "":
            print "Message on: " + str(datetime.datetime.now())
```

```
        + " - topic: " + msg.topic + ", payload: " +
        str(msg.payload)
    client.publish(topic, "Reset...", qos=0, retain=
                  False)
    pass

def on_publish(mosq, obj, mid):
    pass

def on_subscribed(mosq, obj, mid, granted_qos):
    print("Subscribed mid: " + str(mid) + ", qos: " + str(granted_qos))

def on_log(mosq, obj, mid, string):
    pass

client = mqtt.Client()

client.on_connect = on_connect
client.on_message = on_message
client.on_publish = on_publish
client.on_subscribed = on_subscribed
client.on_log = on_log

client.connect(broker, broker_port, timeout_reconnect)
client.subscribe(broker_topic, 0)

print "Cliente conectado " + str(datetime.datetime.now())
print ""

client.loop_start()

while True:

    try:

        def avg(data):
            ##      Dada una lista devuelve la media de las mediciones
            return sum(data) / len(data)

        # Configuracion basica del sensor
        sensor = Adafruit_DHT.DHT11
        gpio = 4
        count = 3
        date = time.gmtime()

        # Recogemos las mediciones que realiza el sensor
        humid_temp_data = [Adafruit_DHT.read_retry(sensor, gpio)
                           for _ in range(count)]

        # Separamos la temperatura de la humedad
        humid_stats = avg([humid for humid, _ in humid_temp_data])
        temp_stats = avg([temp for _, temp in humid_temp_data])
```

```
# Redondeamos los resultados a 2 decimales
humid_statsT = round(humid_stats,2)
temp_statsT = round(temp_stats,2)

# Pasamos los datos a formato String
humid = str(humid_statsT)
temp = str(temp_statsT)

# Establecemos el formato del objeto JSON
Medida = {
    'Fecha' : time.strftime("%Y- %m- %d %H: %M: %S ", date)
    ,
    'Temperatura' : temp,
    'Humedad' : humid
}
mensajeMQTT = json.dumps({"Fecha": time.strftime("%Y- %m- %d %H: %M: %S ", date), "Temperatura": temp, "Humedad": humid});

client.publish(topic, mensajeMQTT, qos=0, retain=False)
time.sleep(15)

except KeyboardInterrupt:
    break

client.disconnect()
print("Desconectado")
```


GNU FREE DOCUMENTATION LICENSE

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.

<<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document,
but changing it is not allowed.

Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “**Document**”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “**you**”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “**Modified Version**” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “**Secondary Section**” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “**Invariant Sections**” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “**Cover Texts**” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “**Transparent**” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “**Opaque**”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only

by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “**Title Page**” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

The “**publisher**” means any person or entity that distributes copies of the Document to the public.

A section “**Entitled XYZ**” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “**Acknowledgements**”, “**Dedications**”, “**Endorsements**”, or “**History**”.) To “**Preserve the Title**” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document’s license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of

these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is

the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

“Massive Multiauthor Collaboration Site” (or “MMC Site”) means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A “Massive Multiauthor Collaboration” (or “MMC”) contained in the site means any set of copyrightable works thus published on the MMC site.

“CC-BY-SA” means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

“Incorporate” means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is “eligible for relicensing” if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright © YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with . . . Texts.” line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

BIBLIOGRAFÍA

- [1] BBC Mundo. <http://www.bbc.com/mundo/noticias-37631834>, 2016.
- [2] I-SCOOP. <https://www.i-scoop.eu/industry-4-0/>, 2017.
- [3] Universidad de las Américas de Puebla. http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/jerez_l_ca/capitulo1.pdf, 2017.
- [4] Centro de Seguridad TIC de la Comunidad Valenciana. http://www.csirtcv.gva.es/sites/all/files/downloads/%5BCSIRT-CV%5D%20Informe-Internet_de_las_Cosas.pdf.
- [5] INCIBE. <https://www.incibe.es/protege-tu-empresa/blog/amenaza-vs-vulnerabilidad-sabes-se-diferencian>, 2017.
- [6] CERT de Seguridad e Industria. <https://www.certsi.es/blog/clasificacion-ataques-dos>, 2015.
- [7] Silicon. http://www.silicon.es/inseguridad-iot-ataques-ddos-2339466?inf_by=5a70a30c681db83c688b51ef, 2017.
- [8] Imperva Incapsula. <https://www.incapsula.com/ddos/ip-spoofing.html>.
- [9] Cloud Security Services. <https://hacking-etico.com/2010/08/26/hablemos-de-spoofing/>, 2010.
- [10] Medium. <https://medium.com/@marvin.soto/qu%C3%A9-es-el-envenenamiento-arp-o-ataque-arp-spoofing-y-c%C3%B3mo-funciona-7f1e174850f2>, 2016.
- [11] BG Protect. <http://www.bgprotect.com/ip-hijacks>.
- [12] IT Security Telelink. <http://itsecurity.telelink.com/session-hijacking/>.

- [13] **Checkmarx.** <https://www.checkmarx.com/knowledge/knowledgebase/session-hijacking>, 2017.
- [14] **Info Spyware.** <https://www.infospyware.com/articulos/que-son-los-malwares/>, 2013.
- [15] **W3II.** http://www.w3ii.com/es/cryptography/attacks_on_cryptosystems.html, 2017.
- [16] **BlackList Hackers.** <http://blacklisthackers.com/2016/05/29/types-of-password-attack/>, 2016.
- [17] **Cerstuperior.** <https://www.certsuperior.com/Blog/ataques-tecnologia-ssl>, 2016.
- [18] **We Live Security.** <https://www.welivesecurity.com/la-es/2017/02/09/ataques-al-dns/>, 2017.
- [19] **Area tecnología.** <http://www.areatecnologia.com/informatica/que-es-dns.html>.
- [20] **Association Franc Nommage Internet en Coop.** <https://www.afnic.fr/medias/documents/afnic-dns-attacks-security-guide-2009-06.pdf>, 2009.
- [21] **Los Indestructibles.** <https://losindestructibles.wordpress.com/2011/09/30/dns-poisoning-spoofing/>, 2011.
- [22] **INCIBE.** <https://www.incibe.es/protege-tu-empresa/blog/enfrentandonos-ransomware>, 2015.
- [23] **Diego Lázaro.** <https://diego.com.es/ataques-xss-cross-site-scripting-en-2017>.
- [24] **We Live Security.** <https://www.welivesecurity.com/la-es/2015/04/29/vulnerabilidad-xss-cross-site-scripting-sitios-web/>, 2015.
- [25] **Hostalia.** <https://pressroom.hostalia.com/white-papers/ataques-inyeccion-sql>, 2013.
- [26] **BitxBit.** <http://lam-h.blogspot.com.es/2011/06/sql-inyeccion-con-php-parte-iii.html>, 2013.
- [27] **Tecnologías Web Blogspot.** <http://tecnologiasweb.blogspot.com.es/2010/12/que-es-una-inyeccion-ldap.html>, 2012.
- [28] **International Journal of Advanced Engineering and Global Technology Angel Panesar.** <http://ijaegt.com/wp-content/uploads/2015/05/409512-pp-851-855-angel.pdf>, 2015.
- [29] **EcuRed.** https://www.ecured.cu/Desbordamiento_de_b%C3%BAfer.

- [30] CyberSeguridad.net. <https://cyberseguridad.net/index.php/453-falsificacion-de-peticiones-en-sitios-cruzados-cross-site-request-forgery-csrf/>, 2015.
- [31] DarkNet. <https://www.darknet.org.uk/2017/07/all-you-need-to-know-about-cross-site-request-forgery-csrf/>, 2017.
- [32] Mundo Informático. <https://www.darknet.org.uk/2017/07/all-you-need-to-know-about-cross-site-request-forgery-csrf/>, 2011.
- [33] Security at Work. <https://www.securityartwork.es/2010/03/30/owasp-top-10-iv-referencia-directa-a-objetos-insegura/>, 2010.
- [34] Security at Work. <https://www.securityartwork.es/2010/03/24/owasp-top-10-iii-perdida-de-autenticacion-y-gestion-de-sesiones/>, 2010.
- [35] Iván Ramos. http://es.seguridad-en-aplicaciones-web.wikia.com/wiki/Redirecciones_y_Reenv%C3%ADos_no_validados.
- [36] Norton. <https://mx.norton.com/importancia-de-las-actualizaciones/article>.
- [37] Alegsa. <http://www.alegsa.com.ar/Dic/ups.php>, 2016.
- [38] Hipertextual. <https://hipertextual.com/2013/11/contraseña-segura>, 2013.
- [39] Enter.co. <http://www.enter.co/guias/lleva-tu-negocio-a-internet/ingenieria-social/>, 2016.
- [40] My Broad Band. <https://mybroadband.co.za/news/hardware/195442-asus-tinker-board-vs-raspberry-pi-3-specifications-and-pricing.html>, 2017.
- [41] Techlila. <https://www.techlila.com/beaglebone-black-vs-raspberry-pi/>, 2017.
- [42] KickStarter. <https://www.kickstarter.com/projects/263025908/jaguarboard-x86-based-single-board-computer?lang=es>, 2017.
- [43] ComputerHoy. <http://computerhoy.com/noticias/hardware/nanopi-neo-plus2-alternativa-raspberry-pi-mas-pequena-barata-64576>, 2017.
- [44] Up Guard. <https://www.upguard.com/articles/apache-vs-nginx>, 2017.

- [45] **BlueHosting.** <https://docs.bluehosting.cl/tutoriales/servidores/apache-versus-nginx-cual-es-el-servidor-web-ideal.html>, 2017.
- [46] **Node-RED.** <https://nodered.org/docs/user-guide/>, 2018.
- [47] **EspBerry.** <https://espberry.wordpress.com/2016/03/07/instalar-mis-aplicaciones-en-raspbian/>, 2017.
- [48] **UnoCero.** <https://www.unocero.com/2012/08/02/raspbian-sistema-operativo-gratuito-para-la-raspberry-pi/>, 2017.
- [49] **AyudaLinux.** <https://ayudalinux.com/fedora-caracteristicas-distro-linux>, 2017.
- [50] **Ach Linux.** [https://wiki.archlinux.org/index.php/Arch_Linux_\(Espa%C3%B1ol\)](https://wiki.archlinux.org/index.php/Arch_Linux_(Espa%C3%B1ol)), 2017.
- [51] **Telefonica.** <https://iot.telefonica.com/blog/lenguajes-de-programacion-en-la-era-del-internet-de-las-cosas>, 2017.
- [52] **Alejandro Alfonso Pérez García.** <http://repositorio.upct.es/bitstream/handle/10317/179/pfc2475.pdf?sequence=1>, 2007.
- [53] **SG.** <https://sg.com.mx/revista/17/sqlite-la-base-datos-embebida#.WWXiB4jyiU16k>.
- [54] **Victor Robles.** <https://victorroblesweb.es/2017/10/14/diferencia-entre-mongodb-y-mysql-comparativa/>.
- [55] **Enredate.** <http://txorienredate.blogspot.com.es/2009/09/un-protocolo-de-comunicaciones-es-un.html>, 2009.
- [56] **PageFault Blog.** <https://pagefault.blog/2017/03/02/using-local-mqtt-broker-for-cloud-and-interprocess-communication/>, 2017.
- [57] **Element 14.** <https://www.element14.com/community/docs/DOC-73950/l/raspberry-pi-3-model-b-gpio-40-pin-block-pinout>, 2015.
- [58] **MongoDB.** <https://docs.mongodb.com/v2.4/tutorial/getting-started/>, 2017.