



UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA DE COMPUTACIÓN

HERRAMIENTA DE AUTOMATIZACIÓN,
MONITOREO Y ANÁLISIS DE
COMPONENTES Y ARTEFACTOS BASADOS
EN EL INTERNET DE LAS COSAS

PEDRO LUIS BOLL LUGO
CÉDULA: 20.173.376

TUTOR:
PROF. ANTONIO RUSSONIELLO

CARACAS, OCTUBRE 2023

Agradecimientos

Primero agradezco a mi Dios todopoderoso, que en su infinita gloria me ha dado todo lo que he necesitado para emprender esta y todas las empresas que me he propuesto, siguiendo su plan y siendo la inspiración y participe de todas mis acciones más valiosas. Agradezco a mi familia, fuente de paz, apoyo, alegría y consuelo, especialmente a mi tía Mine, quien me inculca la profesionalidad en las cosas que hago y a mi tía Charo por sostenerme en esos detalles pequeños que la vida nos pone.

A mis profesores todos, quienes me han brindado su conocimiento, su experiencia y su ejemplo sin el cual llegar hasta aquí hubiese sido plenamente imposible. Especialmente agradezco a mi tutor, el profesor Antonio Russoniello por ayudarme en esta investigación de principio a fin. Para ustedes solo tengo palabras de agradecimiento y firme pensamiento de *nanos gigantum humeris insidentes*.

A los familia Stempel quienes me ayudaron a tener todo lo que necesitaba para comenzar este Trabajo Especial de Grado. También al Dr. Rafael Apitz y a Javier Bavío por prestarme un par de microcomputadores para hacer dispositivos IoT y a Victor Chiaramonte quien se ofreció para hacerme un logo para la aplicación web. También a mis amigos, los de toda la vida y los que hice durante mi carrera, porque me enseñaron tanto, me soportaron, a la hora de reír y en la hora de llorar y aun así permanecen aun a mi lado.

Por sobre todo a mis padres, por ser mis ejemplos, apoyos y fuente de mi propia vida. A mi padre por ser mi mayor ejemplo de virtud, el esfuerzo y el valor de ser una mano amiga. A mi madre por enseñarme tantos valores, a mejorar día tras día, de buscar el merito y la excelencia con ingenio y constancia. Por su amor, por su cariño, por los sacrificios que hicieron por mi y aunque ya no estén físicamente conmigo, me quedaré siempre corto al momento de darles las gracias por tanto.

A ti lector de este trabajo, también te agradezco por tomarte tu tiempo y querer comprender esta investigación, esperando que el conocimiento aquí depositado sea útil como lo ha sido este viaje para mi.

“Los cielos cuentan la gloria de Dios y el firmamento anuncia la obra de sus manos.”

Salmos 19:1

Resumen

La ubicuidad de una creciente cantidad de dispositivos inteligentes que se pueden conectar e interactuar con otros sistemas y dispositivos, aunado a la presencia de cada vez más sensores capaces de leer múltiples variables como de actuadores que se accionen siguiendo un comportamiento preestablecido han generado la necesidad de poder tratar con la complejidad subyacente tanto del volumen de datos que se generan en el proceso, así como también de formas adecuadas de presentar la información capturada de dichos procesos, sin dejar de lado la gestión de dichos procesos automatizados de una manera sencilla.

A esta situación donde se unen las capacidades de cómputo, de captura de información y de actuar sobre el ambiente, en conjunto con la posibilidad de conectarse a las redes es lo que ahora llamamos al Internet de las cosas. El potencial que representa el Internet de las cosas en la actualidad aunque amplio puede verse incompleto sin sistemas que sean capaces de brindar de manera adecuada información útil a los usuarios y la posibilidad de brindar la capacidad de controlar flujos automáticos de manera simple y centralizada, ya sea que este desencadene una serie de actividades en nuestro hogar o sea un complejo conjunto de operaciones críticas en una industria.

A continuación se presenta una propuesta para comenzar a llenar ese vacío fundamental entre la observabilidad y la gestión de dichos dispositivos IoT, mostrando el desarrollo e investigación realizado para crear una herramienta de visualización de datos de sensores, así como también la posibilidad de poder controlar los elementos configurables (actuadores) a través de la creación de flujos de automatización. Por tales objetivos este software desarrollado lleva el nombre de “HAMACA” que no es más que el acrónimo de “Herramienta de Automatización, Monitoreo y Análisis de Componentes y Artefactos”, como tanto nos gusta a los desarrolladores usar para nombrar nuestros proyectos.

Palabras Claves: Internet, Computador, Internet de las Cosas, Sistemas automatizados, Monitoreo, Visualización.

Índice general

Agradecimientos	2
Resumen	3
Lista de figuras	7
Lista de tablas	9
1. Introducción	10
2. Problema de Investigación	12
2.1. Planteamiento del Problema	12
2.1.1. Justificación	12
2.1.2. Alcance	13
2.1.3. Objetivos	13
2.1.3.1. Objetivos Generales	13
2.1.3.2. Objetivos Específicos	13
I Marco Teórico	15
3. Internet de las Cosas	16
3.1. Definición	16
3.2. Modelos de Comunicación	17
3.2.1. Comunicación Dispositivo a Dispositivo	17
3.2.2. Comunicación Dispositivo a la Nube	17
3.2.3. Comunicación Dispositivo a Puerta de Enlace	17
3.2.4. Comunicación Dispositivo a Intercambio de Datos en Back-end .	18
3.3. Aplicaciones del Internet de las Cosas	19
3.3.1. Hogares	19
3.3.2. Industrias	20
3.3.3. Transporte	21
3.3.4. Comercio y Logística	21
3.3.5. Tecnologías Vestibles	22
3.3.6. Medicina y Salud	22
3.3.7. Ciudades Inteligentes	22
3.4. Interoperatividad entre Infraestructuras y Dispositivos	23

3.4.1. Ecosistemas	23
3.4.2. Riesgos	24
3.4.3. Sistemas Heredados	24
3.4.4. Configuración de dispositivos	24
3.5. Protocolos y Estándares	25
3.5.1. Protocolos	25
3.5.1.1. HTTP	25
3.5.1.2. MQTT	27
3.5.1.3. IPv4 e IPv6	29
3.5.2. Estándares	30
3.5.2.1. Bluetooth	30
3.5.2.2. Redes Celulares	31
3.5.2.3. NFC	32
3.5.2.4. WiFi (802.11)	32
3.5.2.5. Zigbee	33
3.5.2.6. Z-Wave	33
3.6. Seguridad	34
4. Herramientas de Monitoreo, Visualización y Control	37
4.1. Herramientas de Visualización	38
4.1.1. Métodos utilizados para visualizar datos	39
4.2. Herramientas de Monitoreo	40
4.3. Herramientas de Control	40
5. Placas Programables	43
5.1. Definición	43
5.2. Placas Arduino	43
5.3. Placas Raspberry Pi	45
II Marco Aplicativo	48
6. Metodología General de Trabajo	49
6.1. Scrum	49
6.2. Control de Versiones de Software	50
6.2.1. Git	51
6.2.1.1. Git-Flow	51
6.3. Herramientas de Desarrollo	52
6.3.1. Herramientas de Hardware	52
6.4. Herramientas de Desarrollo	53
6.4.1. Python	54
6.4.2. Django	54
6.4.3. Bases de datos	55
6.4.3.1. Postgresql	55
6.4.3.2. InfluxDB	55
6.4.4. Eclipse Mosquitto	55
6.4.5. Grafana	56

6.4.6. Node-Red	56
6.4.7. Docker	57
6.5. Adaptación del Marco Metodológico	57
7. Diseño e Implementación	59
7.1. Diseño	59
7.1.1. Software de Visualización, Monitoreo y Control	61
7.2. Implementación de la solución	64
7.2.1. Implementación de los prototipos de dispositivos IoT	64
7.2.2. Implementación de la Aplicación Web HAMACA	67
7.2.2.1. Infraestructura	67
7.2.2.2. Configuración de Entorno	68
7.2.2.3. Desarrollo de la Aplicación Web HAMACA	68
8. Entorno de Pruebas	72
8.1. Escenarios de pruebas	72
8.1.1. Prototipos de Dispositivos IoT	72
8.1.1.1. Variedad en los Dispositivos	73
8.1.1.2. Variedad en los sensores	74
8.1.2. Aplicación HAMACA	75
8.1.2.1. Captura de Información	75
8.1.2.2. Visualización de datos	76
8.1.2.3. Monitoreo de Dispositivos	77
8.1.2.4. Control de Dispositivos	79
9. Casos de Uso	80
9.1. Infraestructura de Comunicación	80
9.2. Aplicación Web	81
10. Resultados, Limitaciones y Trabajos Futuros	84
10.1. Resultados	84
10.2. Limitaciones	86
10.3. Contribuciones	86
10.4. Trabajos Futuros	86
11. Conclusión	88
Bibliografía	89

Índice de figuras

3.1. Modelo dispositivo a dispositivo	17
3.2. Modelo dispositivo a la nube	17
3.3. Modelo dispositivo a puerta de enlace	18
3.4. Modelo dispositivo a intercambio de datos en back-end	18
3.5. Formato de un mensaje de petición HTTP	27
3.6. Formato de un mensaje de respuesta HTTP	27
3.7. Diagrama de funcionamiento del protocolo MQTT	28
3.8. Formato de mensaje del protocolo MQTT	29
5.1. Logo del proyecto Arduino	44
5.2. Arduino Uno R3	44
5.3. Logo de la marca Raspberry Pi	45
5.4. Raspberry Pi 4 modelo B	46
5.5. Raspberry Pi 4 modelo B	47
6.1. Etapas de desarrollo de la metodología Scrum	50
6.2. Ramas de desarrollo usando la metodología Git-flow	52
6.3. Logo de Python	54
6.4. Logo de Django	54
6.5. Logo de PostgreSQL	55
6.6. Logo de InfluxDB	56
6.7. Logo de Eclipse Mosquitto	56
6.8. Logo de Grafana	56
6.9. Logo de NodeRed	57
6.10. Logo de Docker	57
7.1. Prototipo de dispositivo exterior usando Arduino Uno	60
7.2. Prototipo de dispositivo interior usando un Raspberry Pi 3 modelo B	61
7.3. Prototipo de dispositivo de seguridad usando un Raspberry Zero	61
7.4. Prototipo de dispositivo de cámara de inteligente usando un Raspberry Pi 3 modelo B	62
7.5. Componentes sugeridos para la herramienta de visualización, monitoreo y control HAMACA	63
7.6. Arquitectura de la solución propuesta	64
7.7. Elementos para prototipado de dispositivos	64
7.8. Arduino Uno conectado a sensores y al iBook G4	65
7.9. Sensores y actuadores en el exterior	65

7.10. Sensor IoT en el interior del hogar antes y despues de su validación	66
7.11. Sensor IoT en el interior del hogar antes y después de su validación	67
7.12. Modelo autogenerado por el framework Django	69
7.13. Interfaz de configuración de la aplicación	69
7.14. Interfaz de la aplicación para la configuración y visualización de métricas de red	70
7.15. Modelo de la base de datos HAMACA	71
8.1. Identificación de rostros usando un modelo de IA desplegado en un Raspberry Pi 3 modelo B	74
8.2. A la izquierda el sensor de temperatura y humedad DHT11 y a la derecha el sensor de temperatura DS1820	75
8.3. Captura de datos por parte del backend de la aplicación web	76
8.4. Estadísticas de mensajería del broker MQTT	76
8.5. Consulta para construcción de gráficos usando la integración de Grafana	77
8.6. Cuadro de Mando con indicadores de la integración con Grafana	78
8.7. Cuadro de Mando con indicadores e interfaces de control de dispositivos	78
8.8. Flujos creados dentro de la integración de Node-Red para automatizar tareas	79
9.1. Diagrama de caso de uso para la comunicación vía MQTT	80
9.2. Diagrama de caso de uso para la aplicación web HAMACA	82

Índice de tablas

3.1. Dispositivos de Internet de las Cosas basados en su campo de uso según McKinsey Global Institute [1]	20
6.1. Lista de sensores y actuadores utilizados en los prototipos de dispositivos IoT	53
6.2. Actividades sugeridas para el desarrollo del trabajo especial de grado	58

Capítulo 1

Introducción

En general la adopción de nuevas tecnologías suele ocurrir de manera dispar. En algunas ocasiones la adopción es lenta y paulatina lo cual permite que pueda madurar en los diversos entornos en donde se implantan así como también permite crear formas ordenadas y planificadas de crecimiento de los elementos que se encuentran involucrados. Por otro lado también existen tecnologías que debido a su rápido crecimiento hacen que las personas y organizaciones deban adaptarse y ser flexibles en la manera en que se piensa se deben usar los avances, así como también la gama completa de oportunidades y debilidades que representa su uso.

Una de esas tecnologías que ha cambiado la manera en que los seres humanos actuamos es el computador personal. Con el acceso a una plataforma tan poderosa, la capacidad de poder automatizar elementos de la vida cotidiana y de procesos complejos en las industrias, se tiene la receta para ser una de las herramientas más importantes que haya creado el hombre.

Otra tecnología que ha cambiado al mundo es la capacidad de acceder y compartir información a través de una red. Su evolución a lo largo del tiempo a lo que ahora es el Internet ha sido uno de los avances cruciales en la historia. No es una tecnología reciente, pero se ha masificado y democratizado su acceso de tal forma que es un aspecto omnipresente para mas de la mitad de la población mundial. Las diversas plataformas que se apoyan en el Internet nos han ayudado a masificar la adopción de otro conjunto enorme de otras tecnologías, pues su flexibilidad y la madurez de los procesos que involucran la capacidad de conexión es catalizador de oportunidades para resolver problemas.

Si juntamos los aspectos de cómputo y conectividad vemos que de manera disruptiva actualmente se tiene la oportunidad de mejorar y automatizar muchos de los procesos que antes por costo, logística o complejidad eran difíciles de llevar a cabo. El crecimiento en la información y en masificación de artefactos y elementos que obtienen datos de su entorno proveen a los involucrados de una nueva visión del funcionamiento de las cosas que no era posible. Es esta revolución de la información la llamamos “Internet de las cosas”(también conocido en inglés como Internet of Things o IoT por sus siglas) y es una tendencia en la tecnología en pleno crecimiento y que seguirá creciendo de manera activa en los años por venir.

Sin embargo como en casi todo nuevo avance en la tecnología, no ha venido sin presentar retos y dificultades propias. El gran volumen de información generada de manera automatizada, el control y monitoreo de dispositivos y artefactos a lo largo y ancho de complejos sistemas y nuevos flujos automáticos donde antes no eran posibles de realizar hacen cada vez mas difícil el poder tener un panorama claro de las operaciones de estos sistemas por lo que se requieren de infraestructuras, plataformas y desarrollos nuevos para poder mejorar los aspectos de adopción mas ordenada de una forma tan nueva de hacer las cosas. Es así como nace la propuesta de comenzar a realizar la integración de tecnologías probadas que juntas puedan dar un mejor panorama en la observación y control de elementos de las operaciones y acciones que llevamos a cabo de manera automatizada en nuestro día a día.

En el siguiente trabajo de investigación se presenta una propuesta de un software que sea capaz de brindar la capacidad de mostrar datos en tiempo real e histórica provenientes de sensores y actuadores de dispositivos basados en el Internet de las cosas, así como también la capacidad de crear flujos de automatización y control de los mismos, de forma que sea una plataforma centralizada para la gestión de los dispositivos IoT dentro de un ambiente en específico.

En el capítulo dos presentamos de una manera detallada el problema de investigación, los antecedentes y motivaciones que llevan a examinar este problema desde el punto de vista investigativo, así como cuales son los principios que justifican indagarlo. Con ese conocimiento podemos presentar desde el punto de vista metodológico una propuesta de solución a través de un software, planteando objetivos generales y específicos dentro de un alcance que ha sido determinado bajo el contexto.

En los capítulos tres y cuatro se presentan los conceptos teóricos requeridos para abarcar los procesos mencionados previamente, en un primer momento tomando el concepto de Internet de las cosas de manera clara y las ventajas y desventajas que ha tenido la adopción de este tipo de tecnologías en la sociedad y luego presentando un panorama sobre las herramientas de visualización y de control existentes y como el enfoque adecuado puede ayudar a cerrar la brecha entre los datos que se van generando y la gestión de dispositivos que son cada vez mas omnipresentes como complejos.

Para los capítulos cinco, seis y siete presentamos el marco metodológico utilizado para diseñar, crear, probar y validar el funcionamiento integral de los componentes desarrollados con el fin de poder cumplir con los objetivos planteados, incluyendo los posibles escenarios donde este proyecto puede dar un valor agregado a las estructuras existentes.

Por último, los capítulos ocho y nueve nos dan la presentación de los resultados finales tras el desarrolló, entendiendo las circunstancias con la que se desarrollo el problema y sugiriendo una serie de trabajos futuros pueden llevarse a cabo a partir de esta base. Ademas se establecen las conclusiones a las que se llegó después de todo el trabajo realizado.

Capítulo 2

Problema de Investigación

2.1. Planteamiento del Problema

El evidente incremento actual en la cantidad de dispositivos inteligentes en los últimos años ha generado que estemos rodeados de sensores de toda índole, lo que representa una oportunidades de capturar información a la cual antes no se tenía acceso. Si a esa omnipresencia le agregamos el factor de conectividad presentes en dichos dispositivos vemos que estamos ante un tipo de tecnología disruptiva con el potencial para cambiar el entorno en los que todos hacemos vida.

El Internet de las cosas, nombre que se le da a esta combinación entre dispositivos la capacidad de conexión establece que seremos capaces de automatizar muchas más tareas complejas basándonos en micro-decisiones tomadas por los mismos dispositivos, haciendo uso de la información que recolectan y actuando en consonancia con ciertos criterios, haciendo más eficientes dichos procesos, consumiendo menos recursos, alertando sobre potenciales problemas, entre otras ventajas.

Sin embargo la cantidad de dispositivos, el volumen de información y la capacidad de cómputo requerido se incrementa a pasos agigantados haciendo que la capacidad de observar cada flujo de información de procesos y el control tanto automatizado como manual de dichos dispositivos se vuelva una tarea compleja. A pesar de la existencia de múltiples herramientas que permiten dar respuesta a esa problemática, son pocas las opciones que cumplen tanto la función de visualización de datos, de monitoreo de recursos así como también de la controlar complejos flujos de acciones de múltiples dispositivos de manera centralizada, ademas de no existir alternativas flexibles para quienes desean adaptar determinado aspecto de la automatizaciones.

2.1.1. Justificación

Dada ese situación donde es cada vez mas complejo poseer aspectos de gestión y administración de los flujos de automatización en los que se encuentran dichos dispositivos que a la larga es necesario tener a la mano una o más herramientas que permitan rescatar la observabilidad de uno de esos aspectos funcionales de las automatizaciones de procesos basados en el IoT, ya sea en el monitoreo de los sensores, en la gestión de

data e información en tiempo real o histórica y por último el control de dichos dispositivos bajo rutinas predeterminadas.

Esto también significa que para quienes se encarguen de realizar inteligencia de negocios sobre dichos datos masivos se encuentran en una posición difícil para tratar de entender el contexto de los mismos y que información pueden aportar sin tener que ser exhaustivos o aplicando métodos complejos para tratar la información. Incluso como usuarios finales en la vida cotidiana se nos hace cada vez más complejo observar todos los datos que se capturan de nuestros dispositivos y verlos de manera centralizada.

Es allí donde se requieren una nueva generación de herramientas de monitoreo, control, análisis y con la capacidad de poder sustentar dichos complejos procesos automáticos y presentarlos de manera fácil e intuitiva a las personas.

2.1.2. Alcance

Para comenzar a subsanar esa brecha en las herramientas se propone la creación de un desarrollo que en principio contemple una forma centralizada de monitorear y controlar dispositivos IoT, contando con la capacidad de observar los datos en tiempo real e histórico de los sensores y actuadores que se registren usando uno de los múltiples protocolos abiertos de conexión y con la capacidad de presentar información simplificada acordes a las necesidades que presente el usuario final para observar dichos datos. A su vez este trabajo debe servir de framework de desarrollos futuros en el área del IoT en general.

2.1.3. Objetivos

2.1.3.1. Objetivos Generales

Se plantea el desarrollo de un software que permita observar y controlar dispositivos de forma transparente, intuitiva y flexible en la forma en que se presente información en tiempo real como histórica de los sensores y actuadores de dispositivos que se registren en el. Este desarrollo también debe poseer la capacidad de extender mediante módulos futuros casos de uso para implementar o extender sus capacidades iniciales a un rango mayor de dispositivos y estándares.

2.1.3.2. Objetivos Específicos

Para alcanzar el objetivo general estipulado se plantean los siguientes objetivos específicos:

- Establecer la arquitectura para una solución modular adecuada para dar soporte a un sistema basado en el monitoreo/control de dispositivos IoT de manera centralizada.
- Construir prototipos funcionales de uno o más dispositivos IoT para la captura de datos de las variables sobre los procesos que simulen de manera realista un escenario para validar el monitoreo y gestión a través del proyecto.

- Seleccionar y utilizar herramientas diseñadas para trabajar con estándares abiertos y que permita la integración de múltiples herramientas de diversas índole a lo largo del proyecto, para procesar de mejor forma la información obtenida de los dispositivos.
- Utilizar e integrar herramientas de visualización de datos de forma intuitiva, así como también que permitan el control de artefactos y dispositivos.

Parte I

Marco Teórico

Capítulo 3

Internet de las Cosas

3.1. Definición

El Internet de las Cosas, también conocido como IoT por sus siglas en inglés es el término utilizado para designar al conjunto de artefactos y dispositivos que poseen la capacidad de conectarse entre ellos o a otras redes como el Internet de forma que pueden transmitir y recibir datos e información. De manera formal no existe una definición estandarizada sobre el concepto de IoT, pues dependiendo de la organización puede considerarse el concepto desde el punto de vista desde el cual se observe el concepto, sea desde la perspectiva de las redes, desde el punto de vista de los dispositivos o bien desde el punto de vista de los sistemas automatizados.

La primera aparición del término fue realizada en la conferencia “Congressional Black Caucus Foundation 15th Annual Legislative Weekend” en Washington, D. C. en Septiembre del año 1985 por parte de Peter Lewis [2] en donde define que “El Internet de las cosas, o IoT, es la integración de personas, procesos y tecnología con dispositivos y sensores conectables para permitir el monitoreo, estado, manipulación y evaluación remota de las tendencias de dichos dispositivos” [3].

Sin embargo este concepto fue olvidado hasta el año 1999 cuando Kevin Ashton independientemente lo utilizó para ilustrar el poder de conectar Etiquetas de Identificación por Radio Frecuencia (RFID por las siglas en inglés de Radio Frequency Identification) usadas en las cadenas de suministro corporativas a Internet para contar y rastrear mercancías sin la necesidad de intervención humana [4].

Para fines prácticos, durante esta investigación se toma el concepto original de Peter Lewis, al ser una propuesta genérica e independiente del aspecto funcional examinado. Sin embargo es importante recalcar el hecho que las diversas definiciones de IoT no necesariamente están en desacuerdo, sino que enfatizan diferentes aspectos de las tecnologías aplicadas sobre los dispositivos IoT desde diferentes puntos focales y casos de uso [4].

3.2. Modelos de Comunicación

Desde el punto de vista teórico, los dispositivos IoT pueden interconectarse de varias formas. Estos siguen el marco de desarrollo planteado por el estandar RFC-7452 [5] en el que se plantean 4 modelos de comunicación con características propias. Esos modelos son:

3.2.1. Comunicación Dispositivo a Dispositivo

Este modelo de comunicación es el más simple de todos los paradigmas y consiste básicamente en poder conectar directamente los dispositivos independientemente del medio usado los dispositivos se comunican usando alguno de los protocolos y estándares disponibles que sean capaces de comprender. En el ámbito del IoT esta comunicación se realiza de manera inalámbrica y donde los datos o instrucciones suelen ser bastante pequeños o poco frecuentes (figura 3.1). En grandes cantidades estaríamos en presencia de un modelo netamente distribuido.

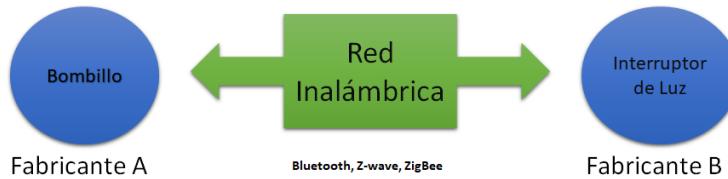


Figura 3.1: Modelo dispositivo a dispositivo

3.2.2. Comunicación Dispositivo a la Nube

En el modelo de comunicación dispositivo a la nube, la conexión del dispositivo se conecta directamente a una nube (propia ó federada) usando un proveedor de servicio (figura 3.2). Este enfoque frecuentemente se aprovecha de los mecanismos de comunicación como redes celulares ó la infraestructura de procesamiento de una organización de manera directa para establecer la conexión entre el dispositivo y el servicio en la nube.



Figura 3.2: Modelo dispositivo a la nube

3.2.3. Comunicación Dispositivo a Puerta de Enlace

El modelo de comunicación dispositivo a puerta de enlace establece una dispositivo o capa intermedia que concentre todas las comunicaciones (hub o broker) entre los

dispositivos y de allí de ser necesario a otros fragmentos de la red o a Internet (figura 3.3). La ventaja de este enfoque es la capacidad de operar de manera centralizada parte de las comunicaciones de los dispositivos. Muchos protocolos están basados en el principio del paradigma de cliente-servidor por lo que este se adapta de manera natural al modelo.

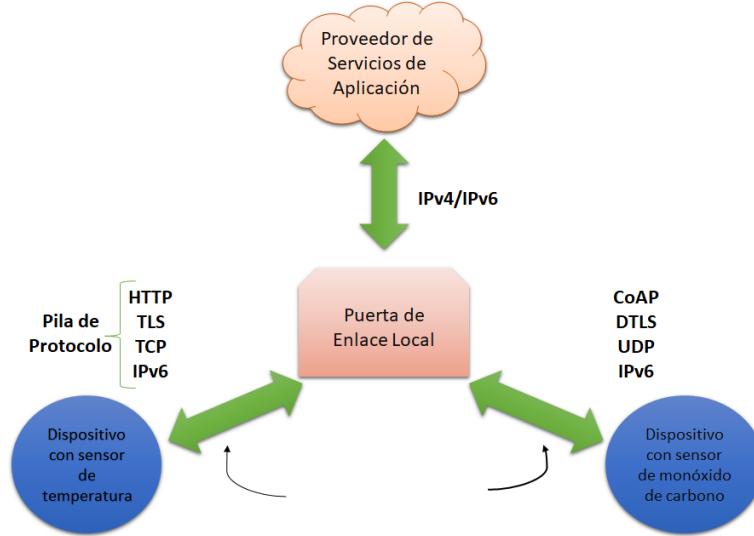


Figura 3.3: Modelo dispositivo a puerta de enlace

3.2.4. Comunicación Dispositivo a Intercambio de Datos en Back-end

Este modelo es una forma automatizada de conexiones, en donde el dispositivo envía los datos a una o más APIs para de manera transparente, haciendo que este pueda intercambiar la información entre servicios que no necesariamente están estructurados o que pertenecen a un tercero (figura 3.4). Particularmente este modelo es útil cuando se requiere que la información sea fácilmente accesible a través de múltiples plataformas ó sistemas independientes.

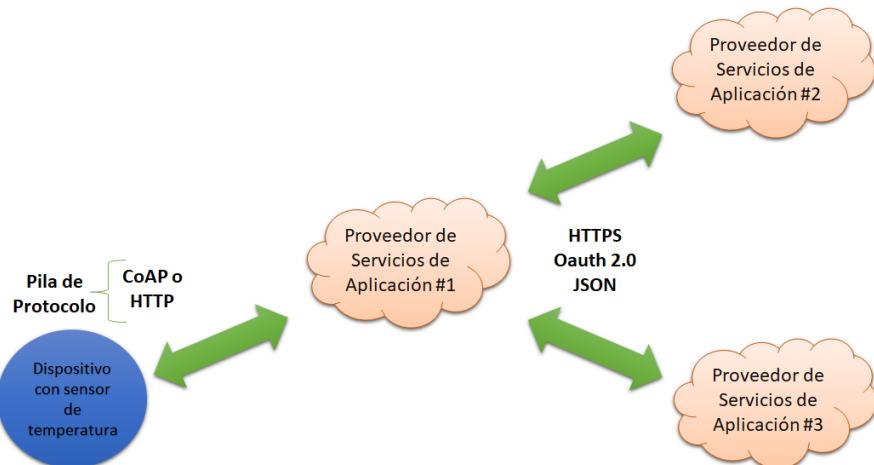


Figura 3.4: Modelo dispositivo a intercambio de datos en back-end

3.3. Aplicaciones del Internet de las Cosas

Si observamos la variedad de dispositivos distintos que encontramos tanto en la vida diaria como si pensamos en términos de los sistemas altamente industrializados, se hace evidente que la cantidad de distintos nichos donde el tener sensores o elementos actuadores hace que muchos procesos se vuelvan tanto dinámicos como reactivos. Las posibilidades de adopción dispositivos interconectados en flujos para automatizar procesos es algo que se está proporcionando la base de lo que llamamos computación ubicua.

Podemos categorizar según las formas de acción y su características, varios campos donde el Internet de las cosas tiene y tendrá un impacto significativo para las personas y organizaciones. En la tabla 3.1 podemos observar alguna de esas categorías y ejemplos de tecnologías acordes [1].

3.3.1. Hogares

En los ambientes domésticos, tecnologías como el IoT son de rápida adopción. Ya podemos observar una amplia gama de distintos dispositivos que van desde bombillos inteligentes, pasando por artefactos de linea blanca hasta elementos estructurales como sensores en tuberías que permiten tener un nivel no solo de automatización sobre labores que tradicionalmente son tediosas o difíciles de realizar.

Las tecnologías IoT dentro de los hogares permiten tener una amplia posibilidad de personalización y comodidad, además que una de las promesas es que al operar obteniendo datos ambientales y de los recursos utilizados puede ajustar y optimizar su utilización lo que representa un nivel de eficiencia que antes no era posible verificar a tan alto detalle. También es posible ahora generar una serie de tareas programadas o rutinas diarias de los habitantes, que van desde aprovechar la luz solar en las habitaciones, limpieza de robots de las habitaciones ó incluso ciertos aspectos de cuidados de mascotas también son posible ahora de administrar gracias a dispositivos inteligentes.

Hablando del manejo y monitoreo de recursos también ahora se abre la posibilidad de atender fenómenos recurrentes o críticos. Por ejemplo, sensores en tuberías pueden determinar si el flujo y la calidad de agua es adecuado para el consumo de la vivienda, pero también determinar si existe alguna anomalía como filtraciones. Otro caso donde puede verse uso son los sistemas de calefacción y aire acondicionados basados en termostatos inteligentes que se adaptan a la temperaturas existentes junto con otros sensores como los de movimientos dentro de una habitación para determinar la exigencia de tener una temperatura adecuada pero teniendo en cuenta el ahorro energético.

Aspectos como la seguridad también se hacen presentes en esta categoría. Los ya tradicionales sistemas de circuitos cerrados ahora son capaces de orquestarse junto con sensores de movimiento para poder determinar un escenario de intrusión o daños a la propiedad. También mediciones como los niveles de CO₂ y otros gases pueden alertar a los interesados sobre condiciones potencialmente peligrosas para los habitantes, incluso con la capacidad de alertar a las autoridades sobre el hecho anormal.

Dispositivos de Internet de las Cosas		
Uso	Descripción	Ejemplos
Personas	Dispositivos acarreados o implantados en el cuerpo humano	Dispositivos (tecnologías vestibles o digeribles) para monitorear y mantener la salud y el bienestar. Administración de tratamientos, incremento en desempeño del ejercicio físico
Hogar	Casas y edificios	Control de objetos y recursos del hogar y sistemas de seguridad
Ventas	Tiendas y distribuidores a minoristas	Tiendas, bancos, restaurantes, estadios, cualquier lugar donde un consumidor obtenga un producto o un servicio. Ofertas en tiendas, optimización de inventarios
Oficinas	Espacios de trabajo y de generación de conocimiento	Administración de recursos, seguridad en las oficinas. Productividad mejorada
Fábricas	Entornos de producción estandarizados	Lugares donde tengan rutinas repetitivas de trabajo, incluyendo granjas o líneas de producción. Operaciones eficientes, uso de equipamiento optimizado y de inventarios
Lugares de Construcción y extracción de material	Producción de materias primas	Infraestructuras en Minas, Petróleo y gas. Mantenimiento predictivo; seguridad laboral
Vehículos	Sistemas dentro de vehículos	Vehículos incluyendo carros, camiones, barcos, aviones y trenes. Mantenimiento basado en condiciones del sistema. Diseños basados en uso. Analítica preventiva

Tabla 3.1: Dispositivos de Internet de las Cosas basados en su campo de uso según McKinsey Global Institute [1]

3.3.2. Industrias

Uno de los sectores con experiencia en la adopción de dispositivos con amplia variedad de sensores y actuadores son las industrias. Desde procesos robóticos cuya tolerancia sea mínima o flujos de tareas críticas para las operaciones se han aprovechado de esto durante años. Sin embargo la llamada industria 4.0 ahora agrega el factor de la interconexión de los sistemas con los dispositivos para presentar estados e información en tiempo real de los procesos automatizados.

Con la información obtenida de todos sensores implicados en los procesos automatizados los gerentes pueden entender de mejor forma se debe proceder a realizar una o más tareas, además de ayudar a optimizar y encontrar puntos de mejora en todas

las etapas del proceso de producción, desde el suministro, el desarrollo y la creación, ahorrando tiempo y dinero al mismo tiempo [6].

La retroalimentación que se va generando en las automatizaciones debido a la existencia de estos dispositivos a manera de diagnóstico ayudan a poder hacer ajustes de manera inmediata en cadenas de producción y líneas de ensamblaje. Otro aspecto positivo es tener la capacidad de monitoreo y funcionamiento óptimo de maquinaria, de forma de realizar mantenimientos preventivos, lo que a la larga representa una reducción de costos y una mejora en la vida útil de las herramientas.

3.3.3. Transporte

En el caso del transporte, tanto público como privado las oportunidades que representa la capacidad de agregar sensores de manera transversal a la infraestructura existente es uno de los focos que tanto fabricantes, reguladores y gobiernos desean alcanzar. La revolución de los automóviles con capacidades de conducción autónoma es un ejemplo claro de como los sensores son capaces de brindar los datos necesarios para la toma de decisiones de los sistemas de control del automóvil sin asistencia y esa información recolectada es también útil para entrenar modelos de inteligencia artificial que permitan elevar el nivel de autonomía en la conducción de futuros modelos [7].

Otro punto que se benefician es en la cadena de suministros de repuestos. Un medio de transporte que es capaz de diagnosticar anomalías hace posible que se genere una cadena de producción directa a las necesidades requeridas, desde la fábrica de la parte hasta la solicitud de chequeo por parte del usuario.

En los sistemas públicos de transporte se benefician de poder traducir los datos en predicciones y en alertas para los usuarios, para ajustar itinerarios y también para poder masificar los servicios a más personas, adaptándolo a las necesidades del entorno o del momento.

3.3.4. Comercio y Logística

El comercio de bienes y servicios se ajusta a los modelos de Internet de las cosas al tener una observabilidad que no se tenía previamente. Desde sensores que dan estado de un producto desde su producción hasta el momento en que es adquirido y garantizan su calidad hasta datos de localización para la vigilancia de mercancías son tecnologías que ya en la actualidad se utilizan trayendo beneficio a productores como a consumidores en general, haciendo más transparente la cadena de producción, buscando eficiencia y sostenibilidad como factores importantes en la actualidad.

Por el lado de la logística el rastreo de bienes, el intercambio de información de inventarios de manera automática basados en la identificación de productos entre proveedores, minoristas y consumidores finales genera confianza y representa nuevos niveles de seguridad, sobre todo al momento de hacer entregas de mercancías.

3.3.5. Tecnologías Vestibles

Los wearables o tecnologías vestibles es uno de esos ámbitos en donde poco a poco está encontrando un nicho entre las personas. Desde relojes inteligentes, lentes para realidad aumentada y realidad virtual, prendas de vestir que vigilan nuestros signos vitales, entre otros muchos ejemplos son la forma que el Internet de las cosas toma forma para esta categoría. La miniaturización de sensores y de elementos de cómputo está haciendo posible poder crear dichos elementos y que sean cada vez más comunes en la vida cotidiana a pesar de retos como el consumo energético de los sensores y de las comunicaciones que estos realizan [8].

Accesorios como relojes inteligentes que son capaces de leer múltiples variables fisiológicas de forma de brindar un panorama general de la salud y otras características personales o trajes especiales, utilizados sobre todo en los deportes donde se busca tener una observación más marcada del desempeño del deportista, los dispositivos ayudan a brindar una mejor perspectiva de cara a entrenamientos y de encontrar métodos para alcanzar más rendimiento brindando métricas en tiempo real como de forma histórica.

3.3.6. Medicina y Salud

Muy a la par de las tecnologías vestibles, el área de la salud en general se ve beneficiada de la capacidad de poder obtener lecturas de variables fisiológicas así como de los signos vitales que puedan presentar los pacientes tanto en tiempo real como a lo largo de una intervención o de un tratamiento, así poder adaptar procedimientos, medicinas, dietas de los pacientes en búsqueda de una más rápida evolución y mejora de las personas.

También podemos ver un avance en la efectividad de tratamientos médicos suministrados gracias a la capacidad de obtener la información de manera más rápida y global por la presencia de esos sensores en los dispositivos que lleven los pacientes, sumando a la tendencia de poder brindar cada vez más tratamientos personalizados [9]. Además la conectividad del sistema de atención de la salud a través del Internet de las cosas, hace hincapié en las necesidades del paciente, es decir, tratamientos proactivos, precisión mejorada cuando se trata de diagnóstico, la intervención oportuna por parte del personal de salud, en miras de una medicina cada vez más preventiva en vez de una que sea reactiva.

3.3.7. Ciudades Inteligentes

Las ciudades en el futuro se perfilan en el uso de los dispositivos IoT como parte fundamental de la infraestructura que deben tener para brindar a sus ciudadanos la mayor cantidad de satisfacción y calidad de vida posible. Desde el punto ya tocado del transporte como el monitoreo y control de recursos vitales como es el caso del acceso a agua potable, el reciclaje de materiales, los gobiernos ya diseñan en función de la capacidad de poder contar con datos de sensores para poder evaluar mejores políticas ciudadanas, así como también brindar transparencia a los procesos de gestión de recursos [10].

3.4. Interoperatividad entre Infraestructuras y Dispositivos

Para las redes de computadoras, la interoperatividad es un elemento fundamental, incluso desde su mismo diseño, pues la idea de dispositivos conectados también requiere que se puedan comprender en un “lenguaje” común entre las partes. Para ello existen numerosos estándares y protocolos que han surgido de forma que sean ese marco común de entendimiento entre plataformas, dispositivos y arquitecturas.

Más allá de los aspectos técnicos, la interoperatividad tiene una influencia significativa en el impacto económico potencial de IoT. Aquellos marcos comunes, estándares y protocolos que tienen bien definido pueden fomentar la innovación y proporcionar a los fabricantes de dispositivos IoT valor agregado gracias a esas características y a su vez, aumentar el valor económico general del mercado.

Además, la aplicación de las normativas y el desarrollo de nuevos estándares abiertos, cuando sea necesario, ayudan a reducir las barreras de entrada, facilitan nuevos modelos de negocio y se convierten en opciones para el usuario final así como a las organizaciones quienes las regulan o las usan. Facilita la posibilidad de elegir dispositivos con las mejores características al mejor precio e integrarlos para que trabajen juntos. Los compradores pueden ser reacios a adquirir productos y servicios IoT si hay inflexibilidad en la integración, alta complejidad de uso, la propiedad final de información obtenida, la preocupación por el bloqueo del proveedor ó temor a la obsolescencia debido al cambio de los estándares [4].

3.4.1. Ecosistemas

Los ecosistemas de dispositivos IoT no son más que la consideración planificada bajo el paraguas de uno ó más estándares y/o protocolos para ofrecer características que den al usuario final un valor sobre su elección. Existen dos perspectivas sobre los ecosistemas de los dispositivos:

- Ecosistemas abiertos: Generalmente creados por una organización bajo el apoyo de múltiples empresas u otras organizaciones con la idea de acordar métodos de operación que sean estandarizados a nivel de las partes interesadas. En el mundo del Internet de las cosas se ve bajo la serie de tecnologías y protocolos que han sido adaptados y propuestos para poder realizar desde las tareas de comunicación, así como también en las características que sensores y actuadores envían, procesan y almacenan la información. De propósito libre y con un esquema en el cual los aportes de las comunidades y personas son escuchados tienen un ciclo de desarrollo muy rápido y una alta adopción por su flexibilidad. El principal problema que posee éste enfoque es que los esfuerzos de estandarizar ó el interés de organizaciones por sobre otras de utilizar las características de un protocolo o estándar pueden dispersar los esfuerzos de adopción de fabricantes por la cantidad de posibilidades necesarias para validar un funcionamiento correcto de los dispositivos ante múltiples protocolos o estándares y el costo que conlleva.

- Jardines mallados: De manera contraria a los ecosistemas abiertos, los jardines mallados son desarrollados por las organizaciones para poder controlar de manera cercana la visión de los estándares y protocolos para el funcionamiento de los dispositivos. Esta visión permite que exista una integración vertical amplia y a su vez, se beneficia de un manejo muy óptimo de los recursos disponibles para las tareas relacionadas. Así todos los dispositivos que están certificados para funcionar dentro de un ecosistema de estas características probablemente funcione de una manera mucho mejor de lo que lo haría en un ecosistema abierto. La contra de este enfoque es la poca flexibilidad tanto de fabricantes como de los usuarios finales para obtener características que den valor agregado a los dispositivos que desea obtener [4]. Además es común que los procesos de certificación y el coste de licenciamiento de dichos estándares y protocolos cerrados se traduzca en un precio mayor tanto para los fabricantes como para los usuarios finales.

3.4.2. Riesgos

Con respecto al tema de riesgos, podemos separarlos en dos espacios:

- Riesgos Programados: Los riesgos programados son aquellos riesgos que asumen los fabricantes, empresas y organizaciones al momento de diseñar, crear y ofrecer al público un producto. En el ámbito del Internet de las cosas este riesgo viene por la decisión de tomar uno o más estándares en búsqueda de interoperatividad. También se puede representar por el hecho de que se busque poner en el mercado un producto que no cumpla completamente un estándar con la probabilidad de no poder operar correctamente en un ecosistema.
- Riesgos Técnicos: Los riesgos técnicos van presentes en las consideraciones de diseños, arquitectura y finalmente la producción de los dispositivos en si mismos. La complejidad del funcionamiento de una o más características puede verse afectada si en algunos de esas etapas existe un fallo o también por algún defecto en los componentes.

3.4.3. Sistemas Heredados

Una arista de la interoperatividad que debe considerarse en todo momento tanto como por los fabricantes, como por los usuarios finales es la capacidad de lidiar con sistemas antiguos, también llamados heredados o legados. Esto se debe a que aunque existan estándares y protocolos que son suficientemente robustos pero no están diseñados con la idea de ser extendidos hacen que dispositivos que lleven esos estándares tengan dificultades para poder cumplir con las expectativas de las industrias. Del otro lado, el tener que crear dispositivos nuevos que tengan que cumplir con estándares y protocolos desactualizados puede hacer que los costes sean mayores, así como la dificultad de brindar soporte a los mismos.

3.4.4. Configuración de dispositivos

La configuración de dispositivos es un proceso que muchas veces es similar en aspectos como la parte de la conectividad. Sin embargo el problema se puede observar

cuando la cantidad de dispositivos a configurar es alto. Esta perspectiva puede ser desalentadora para los usuarios finales quienes tienen que asumir el tiempo para hacerlo o el coste de contratación de profesionales para poder realizar todo el proceso.

3.5. Protocolos y Estándares

Como se pudo observar en los puntos anteriores la existencia de estándares y de protocolos son parte fundamental para garantizar el correcto funcionamiento de cualquier dispositivo que entre dentro de cualquiera de las categorías del Internet de las cosas. El lado de manejo de las redes como pilar de la tecnología requiere conocer y elegir el método adecuado de comunicarse. Es por ello que tanto los modelos planteados al principio de este capítulo [5] como el protocolo adecuado puede cambiar diametralmente el desempeño del dispositivo.

Sin embargo, hay un valor considerable en la extensión de la capacidad de conectarse a Internet a los dispositivos más restringidos en término de recursos que a menudo necesitan versiones optimizadas o uso especial de estos protocolos. A continuación se presentan algunos de los protocolos y estándares actuales más importantes empleados por los dispositivos del Internet de las Cosas y que han sido considerados para la investigación y el desarrollo de este trabajo.

3.5.1. Protocolos

3.5.1.1. HTTP

El protocolo de transferencia de hipertexto o mejor conocido como HTTP [11], es un protocolo de la capa de aplicación para la transmisión de documentos hipermedia, como HTML. Fue diseñado para la comunicación entre los navegadores y servidores web, aunque puede ser utilizado para otros propósitos también. Sigue el patrón modelo-vista-controlador, en el que se establece una conexión, realizando una petición a un servidor y espera una respuesta del mismo. Se trata de un protocolo sin estado, lo que significa que el servidor no guarda ningún dato (estado) entre dos peticiones. Aunque la mayoría de los casos se basa en una conexión del tipo TCP /IP, puede usarse con cualquier otro protocolo del nivel de la capa de transporte orientado a conexión. [12]

Las principales características clave del protocolo son las siguientes:

- Sencillez: HTTP está pensado y desarrollado para ser leído y fácilmente interpretado por las personas, haciendo de esta manera más fácil la depuración de errores, y reduciendo la curva de aprendizaje para las personas que empiezan a trabajar con él.
- Extensible: Presentadas en la versión HTTP/1.0, las cabeceras de HTTP, han hecho que este protocolo sea fácil de ampliar y de experimentar con él. Funcionalidades nuevas pueden desarrollarse, sin más que un cliente y su servidor, comprendan la misma semántica sobre las cabeceras de HTTP.

- Basado en sesiones, sin estados: HTTP es un protocolo sin estado, es decir, no guarda ningún dato entre dos peticiones en la misma sesión. Esto plante la problemática, en caso de que los usuarios requieran interactuar con determinadas páginas Web, de forma ordenada y coherente. Pero mientras HTTP ciertamente es un protocolo sin estado, el uso de HTTP cookies, si permite guardar datos con respecto a la sesión de comunicación. Usando la capacidad de ampliación del protocolo HTTP, las cookies permiten crear un contexto común para cada sesión de comunicación.
- Orientado a Conexión: Una conexión se gestiona al nivel de la capa de transporte, y por tanto queda fuera del alcance del protocolo HTTP. Aún con este factor, HTTP no necesita que el protocolo que lo sustenta mantenga una conexión continua entre los participantes en la comunicación, solamente necesita que sea un protocolo fiable, o que no pierda mensajes (como mínimo, en todo caso, un protocolo que sea capaz de detectar que se ha pedido un mensaje y reporte un error).

HTTP es un protocolo que funciona en dos fases fundamentales, una fase de petición y una fase de respuesta. Esto se hace de la siguiente manera:

- En la fase de petición:
 1. Se abre una conexión: Se utiliza un protocolo de conexión, generalmente TCP, que se usará para hacer una o una serie de peticiones y recibir una respuesta. El cliente puede abrir una conexión nueva, reusar una existente, o abrir varias a la vez hacia el servidor.
 2. Se hace una petición HTTP: Los mensajes HTTP (previos a HTTP/2) son legibles en texto plano. A partir de la versión del protocolo HTTP/2, los mensajes se encapsulan en franjas, haciendo que no sean directamente interpretables, aunque el principio de operación es el mismo.
- En la fase de respuesta:
 1. Se recibe y lee la respuesta enviada por el servidor.
 2. Se cierra o recicla la conexión para futuras peticiones.

Existen dos tipos de mensajes HTTP: peticiones y respuestas, cada uno sigue su propio formato.

- Petición HTTP: Consta de un método, que generalmente un verbo como GET ó POST que define la operación que el cliente desea realizar, la dirección del recurso pedido en formato de URL, la versión del protocolo y el cuerpo del mensaje (figura 3.5). Opcionalmente puede llevar cabeceras que pueden aportar más información.
- Respuesta HTTP: Formados por los campos de versión del protocolo, el código del estado indicando si la petición fue exitosa o no y el porque, una descripción del estado, cabeceras HTTP y por último y dependiendo de lo requerido, un recurso (figura 3.6).

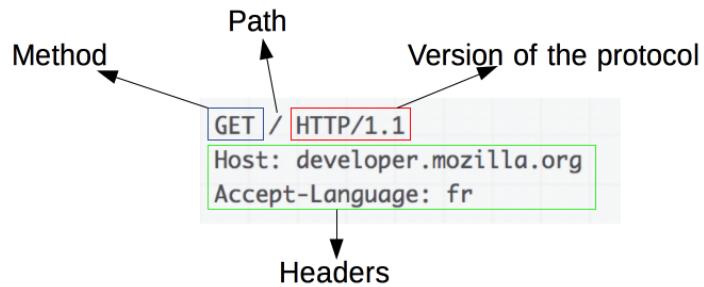


Figura 3.5: Formato de un mensaje de petición HTTP

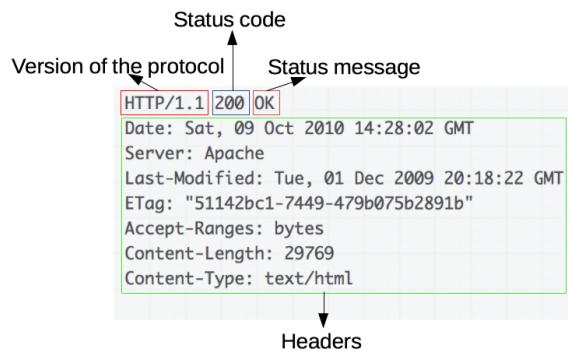


Figura 3.6: Formato de un mensaje de respuesta HTTP

3.5.1.2. MQTT

MQTT son las siglas de Message Queue Telemetry Transport [13] es un protocolo de mensajería basado en colas que como su nombre indica, su propósito principal es la telemetría, o el monitoreo remoto. Su objetivo es recopilar datos de muchos dispositivos y transportarlos a la infraestructura de servicios en red. Se dirige a grandes redes de pequeños dispositivos que necesitan ser monitoreados o controlados desde la nube. [14]

IBM creó el protocolo MQTT para poder establecer comunicación vía satélite con sensores en campos petroleros. Fue diseñado para que las comunicaciones a través de este protocolo fuesen confiables y utilizando la menor cantidad de energía posible. MQTT utiliza un modelo de publicación-suscripción y requiere un corredor MQTT central (Broker) para administrar y enrutar mensajes entre los nodos de una red MQTT. Utiliza TCP en la capa transporte, que se caracteriza por ser confiable, ordenado y con comprobación de errores. [15]

El protocolo MQTT tiene como fortalezas:

- **Modelo de publicación-suscripción:** El modelo de publicación-suscripción es un modelo que no tiene problemas al escalar la cantidad de nodos en la red y además hace de un uso eficiente de la energía. Los Brokers y los nodos publican información y otros se suscriben según el contenido, el tipo o el tema del mensaje. (Estos son términos estándar de MQTT.) Generalmente, el Broker se suscribe a todos los mensajes y luego administra el flujo de información a sus nodos.

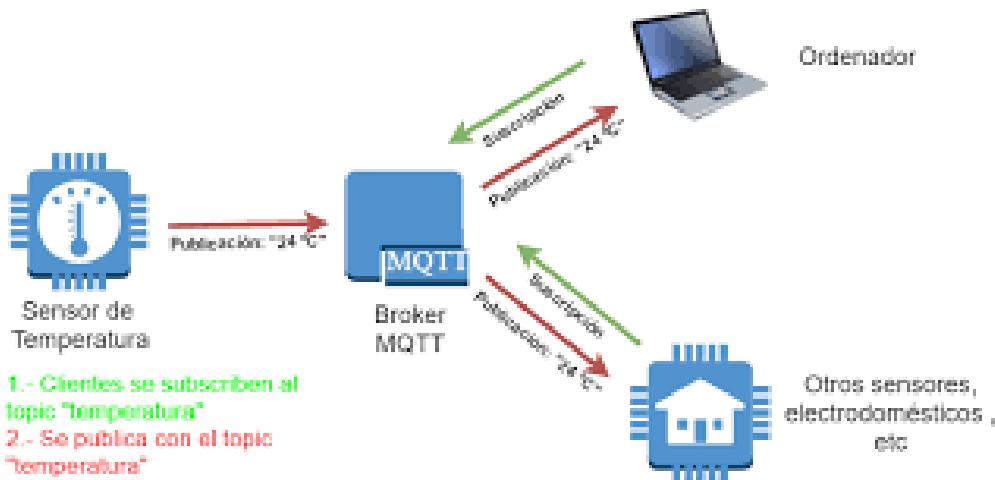


Figura 3.7: Diagrama de funcionamiento del protocolo MQTT

- Seguridad: Al utilizar TCP como protocolo en la capa de transporte, Los paquetes MQTT no están encriptados por defecto. Pero puede utilizar la seguridad de Internet TLS/SSL [13]. TLS es un método muy seguro para cifrar el tráfico, pero también es un recurso intensivo para clientes ligeros debido a su apretón de manos requerido y el aumento del overhead de los paquetes. Para las redes donde la energía es una prioridad muy alta y la seguridad mucho menos, cifrar sólo la carga útil del paquete puede ser suficiente.
- Soporte para Calidad de Servicio (QoS): En MQTT, los niveles QoS 0, 1 y 2 describen niveles crecientes de entrega garantizada de mensajes. EL nivel 0 no repite paquetes, es decir, es un nivel de “apunta y dispara”. El nivel 1 intenta garantizar que un mensaje sea recibido al menos una vez por el destinatario previsto. Una vez que un mensaje publicado es recibido y comprendido por el destinatario deseado, reconoce el mensaje con un mensaje de acuse de recibo (PUBACK) dirigido al nodo de publicación. Por último el nivel 2 intenta garantizar que el mensaje sea recibido y descodificado por el destinatario deseado. Este es el nivel de QoS MQTT más seguro y confiable.
- Testamento y ultima voluntad (LWT): MQTT proporciona un mensaje de “último testamento” que se puede almacenar en el Broker MQTT en caso de que un nodo se desconecte inesperadamente de la red. Este LWT conserva el estado y propósito del nodo, incluyendo los tipos de comandos que publicó y sus suscripciones. Si el nodo desaparece, el intermediario notifica a todos los suscriptores del LWT del nodo. Y si el nodo vuelve, el corredor lo notifica de su estado anterior. Esta característica acomoda redes con alto porcentaje o probabilidades de pérdidas.
- Suscripciones flexibles de temas: Un nodo MQTT puede suscribirse a todos los mensajes dentro de una funcionalidad determinada. Por ejemplo, en un cocina un nodo horno puede suscribirse a todos los mensajes de “cocina/horno/+”, con el “+” como comodín. Esto permite una cantidad mínima de código. Hay otros comodines MQTT igualmente útiles para reducir la cantidad del código utilizado y por lo tanto el tamaño y el coste de la memoria.

Por otro lado MQTT tiene tres desventajas evidentes:

- El Broker: Como intermediario central puede ser un inconveniente para los sistemas IoT distribuidos. En entornos donde existe poca presencia de sensores y/o actuadores o donde no crece la cantidad de los dispositivos, la presencia de un Broker genera latencia dentro de la red. Por otro lado el Broker puede convertirse en un único punto de fallo para la red completa.
- El uso de TCP: TCP fue diseñado originalmente para dispositivos con más memoria y recursos de procesamiento que los que pueden estar disponibles en una red de dispositivos IoT. Por ejemplo, el protocolo TCP requiere que las conexiones se establezcan en un proceso de apretón de manos de varios pasos antes de intercambiar cualquier mensaje. Esto aumenta los tiempos de activación y de comunicación y reduce la duración de la batería a largo plazo.
- Retrasos en la transmisión: Una vez más, el uso de TCP sin persistencia de sesión puede requerir un tiempo de transmisión incremental para el establecimiento de la conexión. Para nodos con tráfico periódico y repetitivo, ésto puede afectar el consumo energético.

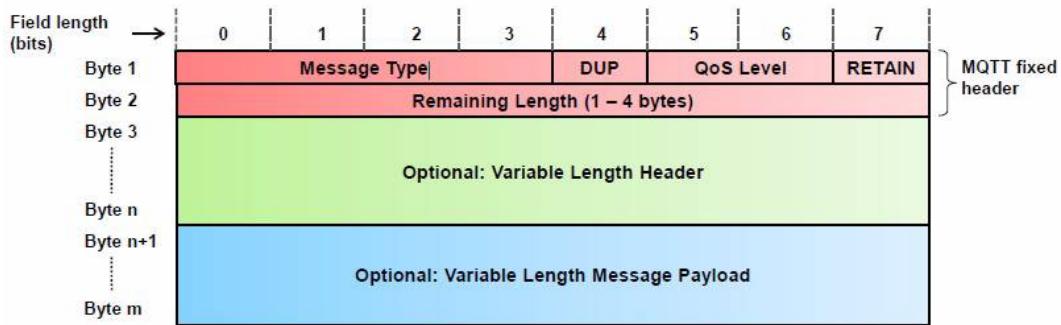


Figura 3.8: Formato de mensaje del protocolo MQTT

3.5.1.3. IPv4 e IPv6

Internet Protocol o IP es un protocolo de capa de red, no orientado a conexión, con una política de mejor esfuerzo en la entrega de los paquetes y cuya principal tarea es permitir la comunicación bidireccional en la red haciendo uso de direcciones a las cuales se pueden enrutar convenientemente. IP no provee ningún mecanismo para determinar si un paquete alcanza o no su destino y únicamente proporciona seguridad (mediante checksums o sumas de comprobación) de sus cabeceras y no de los datos transmitidos.

Este protocolo tiene como pilares fundamentales:

- El direccionamiento: Se refiere a la capacidad de poder asignar una dirección lógica a una interfaz de red y a su vez conocer la manera en que se divide la red a la que pertenece. Para el direccionamiento efectivo, el protocolo hace uso de direcciones IP, un sistema de identificación lógica y jerárquicamente una interfaz de red.

- EL enrutamiento: el enrutamiento permite redistribuir de manera eficiente el tráfico de las comunicaciones entre nodos de manera que estos puedan ir de su origen a su destino.

El protocolo IP en la actualidad conviven dos versiones del protocolo, IPv4 establecido en el RFC-791 [16] e IPv6 establecida en el RFC-2460 [17], ambos basados en los mismos principios pero con diferencias en la implementación.

3.5.2. Estándares

3.5.2.1. Bluetooth

Bluetooth es uno de los primeros estándares de comunicación creados pensando en dispositivos móviles. Desarrollado por la compañía sueca Ericsson estableció un protocolo de comunicación inalámbrica en áreas personal, con una tasa de transmisión inicial de 1 Mbps, que luego pasó a ser parte del estándar IEEE 802.15.1 [18]. Los dispositivos Bluetooth utilizan señales de radio en la frecuencia de 2.4 GHz para establecer dicha conexión. Con este estándar se busca que los dispositivos puedan:

- Establecer conexiones con poco gasto energético.
- Establecer enlaces por lo general de corta duración.
- Otorgar seguridad mediante diversas maneras de cifrado de datos, además de exigir el uso de un PIN para establecer conexiones entre equipos.
- Soportar tráfico de voz y datos.
- Tener un bajo costo de producción e implementación.

En la actualidad el estándar Bluetooth se compone de un núcleo o también llamado Bluetooth Core que contiene las directivas de comunicaciones generales e inherentes al estándar y de dos extensiones llamadas Bluetooth BR/EDR (Basic Rate/Enhanced Data Rate) optimizado para hacer uso de intercambio de streams de datos, como audio y el Bluetooth Low Energy diseñado para hacer un uso muy eficiente de la energía utilizada en la comunicación, en casos donde el intercambio de datos es pequeño y que ha sido implementado desde la versión 4.0 del estándar. Los dispositivos tienen la posibilidad de implementar ya sea Bluetooth BR/EDR o Bluetooth Low Energy o ambas de forma mixta.

La gran ventaja de este estándar es que gran cantidad de dispositivos cuentan con la capacidad de comunicarse a través de este estándar, además de que permite una comunicación con un modelo dispositivo a dispositivo fácil de establecer para el usuario final. Sin embargo estas comunicaciones son de corto alcance cosa que puede afectar el desempeño entre dispositivos. El Bluetooth Low Energy abre las puertas a dispositivos que requieren manejar de manera eficiente el consumo energético de las comunicaciones pero no es recomendado su uso para el intercambio exhaustivo de información. Por otro lado, es recomendable utilizar el Bluetooth BR/EDR cuando la aplicación que se usa requiere enviar una cantidad de información importante o continuamente, pero tiene la desventaja que el consumo energético es mayor afectando negativamente la duración de los dispositivos si estos operan con una fuente estable de energía.

3.5.2.2. Redes Celulares

Los estándares celulares son las más utilizadas en cuanto a redes inalámbricas. Extendidas al rededor del mundo proveen a los dispositivos una manera sencilla de poder conectarse a servicios de voz y datos.

Las tecnologías de 3G son la respuesta a la especificación IMT-2000 de la Unión Internacional de Telecomunicaciones [19]. En Europa y Japón se seleccionó el estándar UMTS (Universal Mobile Telecommunication System) [20], basado en la tecnología W-CDMA [21]. UMTS está gestionado por la organización 3GPP, también responsable de GSM, GPRS y EDGE. Para el continente americano, para África y para el resto del mundo se establecieron los estándares High-Speed Packet Access (HSPA) [22] es una fusión de dos protocolos móviles, High Speed Downlink Packet Access (HSDPA) [22] y High Speed Uplink Packet Access (HSUPA) [22] que extiende y mejora el rendimiento de las redes de telecomunicaciones móviles de tercera generación (3G), como son el 3.5G o HSDPA y 3.5G Plus, 3.75G o HSUPA existentes utilizando los protocolos WCDMA.

La cuarta generación de tecnología en telecomunicaciones móviles, abreviado comúnmente como 4G, es el estándar aplicado al mercado móvil de la actualidad. Para que una tecnología pueda ser considerada como parte del 4G, esta debe cumplir con ciertos requisitos para ser considerado dentro del estándar. Entre las tecnologías que se encuentran dentro de dicho estándar están: LTE-Advanced [23], WiMAX móvil(IEEE 802.16e) [24] y WiMAX Release 2 (IEEE 802.16m). [24]

Los requerimientos para que una red sea considerada 4G están expuestos en la IMT-Advanced de la ITU y son:

- Alto grado de uniformidad de funciones en todo el mundo, manteniendo al mismo tiempo la flexibilidad de admitir una amplia gama de servicios y aplicaciones rentables.
- Compatibilidad de servicios con las IMT y las redes fijas.
- Interoperabilidad con otros sistemas de acceso radioeléctrico.
- Servicios móviles con soporte para calidad de servicio.
- Capacidad de itinerancia mundial.
- Velocidad de 100 Mbits para una movilidad alta y de 1 Gbits para una movilidad baja.

Plenamente diseñados para hacer su uso en dispositivos de carácter móvil y aprovechando la infraestructura existente, hacen de los estándares de comunicación celular una de las opciones más fáciles y convenientes de implementar en soluciones de comunicación inalámbrica, sobre todo cuando no se puede implementar una red dedicada para el tráfico de los nodos. Hacen un uso eficiente de la energía del dispositivo, pues los celulares y teléfonos inteligentes, pues es el recurso crítico para las operaciones.

3.5.2.3. NFC

La Comunicación de Campo Cercano o NFC por sus siglas en inglés, es un estándar que permite transferencia bidireccional de datos entre dispositivos. Su rango de transmisión es corto de máximo 10 cm de distancia, con tasas de transferencia máxima de 420kbps. El conjunto de tecnologías que permiten esta comunicación están enmarcados en el estándar ISO/IEC 18000-3. [25]

NFC puede funcionar en dos modos:

- Activo, en el que ambos equipos con chip NFC generan un campo electromagnético e intercambian datos.
- Pasivo, en el que solo hay un dispositivo activo y el otro aprovecha ese campo para intercambiar la información.

En cuanto a la seguridad, NFC utiliza su corta distancia de transmisión como un mecanismo de protección, pues cualquier ataque deberá realizarse cerca de los dispositivos que realizan la transmisión.

3.5.2.4. WiFi (802.11)

WiFi es el estándar de comunicación inalámbrico más utilizado en el mundo después de las redes celulares y ha sido una selección obvia al momento de integrar comunicación dispositivos con una red. El estándar WiFi está basado en el estadar de la IEEE 802.11 [26] y ha atravesado una serie de versiones que han permitido mejorar su cualidades de despliegue en términos de distancia, velocidad de transmisión, seguridad e interoperatividad.

El estándar 802.11 consta de una serie de técnicas de modulación half duplex inalámbrica que utilizan el mismo protocolo básico. Las versiones 802.11b y 802.11g utilizan la banda ISM de 2,4 GHz. La dificultad que representa usar esta banda es que los dispositivos pueden sufrir interferencias con electrodomésticos tan comunes como el microondas o el horno o con dispositivos Bluetooth. Es por que eso que deben controlar dicha susceptibilidad a las interferencias mediante métodos de señalización de espectro ensanchado por secuencia directa (DSSS, Direct-Sequence Spread Spectrum) y de multiplexación por división de frecuencia ortogonal (OFDM, OFDM, Orthogonal Frequency Division Multiplexing), respectivamente.

Por otro lado la versión 802.11a y la versión 802.11ac utilizan la banda de 5 GHz. También hay versiones que pueden usar ambas bandas, como 802.11n pueden utilizar las bandas de 2,4 GHz o la de 5 GHz. El alcance de este estandar depende de la potencia del dispositivo, de la versión de WiFi y de las bandas utilizadas, al igual que la velocidad de transmisión pudiendo pasar de 54 Mbps a 1 Gbps.

A pesar de ser uno de los estándares más utilizados, WiFi posee como inconvenientes la complejidad del hardware y software requerido así como también la cantidad de energía requerida para establecer y mantener la comunicación, convirtiéndose en prohibitivo para dispositivos con capacidades limitadas. En las revisiones más nuevas del

estándar se está buscando la manera de que se mantengan las ventajas de transmisión y de espectros utilizados, sin que esto signifique un impacto severo en el consumo energético de los dispositivos.

3.5.2.5. Zigbee

Zigbee es otro estándar de comunicación inalámbrica que agrupa un conjunto de protocolos de comunicación basado en el estándar IEEE 802.15.4 [27] y que está diseñado para usar la banda de 2,4GHz en intercambios de paquetes poco frecuentes. Opera en un rango no mayor a 100 metros.

Zigbee tiene como características un funcionamiento con operación en bajo consumo energético, ofreciendo seguridad, robustez y escalabilidad de altas cantidades de nodos y dispositivos conectados y está pensado para ser utilizado para el intercambio de paquetes bajo un modelo dispositivo a dispositivo. [28]

Una de las principales ventajas de Zigbee es lo sencillo y el bajo coste que supone para la empresa producir dispositivos con esta tecnología de comunicación ya que es mucho más sencillo que Bluetooth por ejemplo. [29] Existen tres distintos roles que puede cumplir un nodo o dispositivo que hace uso del estándar Zigbee:

- El Coordinador Zigbee: es el nodo más completo y se encarga de controlar toda la red y los caminos para su comunicación. [29] Es de carácter obligatorio para la comunicación de los dispositivos.
- El Router Zigbee: interconecta los nodos para poder ejecutar código del usuario, es decir, ofrece un nivel de aplicación dentro de la pila de protocolos. [29] Puede ser el mismo nodo que es Coordinador.
- El Dispositivo: Los nodos finales de ZigBee sólo reciben información y se comunican únicamente con el nodo padre (coordinador y/o router). [29]

Zigbee es altamente utilizado en entornos industriales, sin embargo no es tan ampliamente utilizado como su rival más cercano el estándar Bluetooth, que posee más compañías asociadas brindadole apoyo.

3.5.2.6. Z-Wave

El estándar Z-Wave es una tecnología de comunicación inalámbrica de baja potencia diseñada principalmente para la automatización del hogar en productos como lámparas, sensores y muchos otros. Esta optimizado para ser estable, de baja latencia y con velocidad de datos de hasta 100kbps. [28]

A diferencia de otros estándares , Z-Wave no utiliza la banda de los 2.4 GHz, sino que utiliza la banda de los 900MHz, pues con ello proporciona un rendimiento superior por dos motivos: menos interferencias (por funcionar a baja frecuencia) con otros dispositivos y mayor penetración de las ondas en paredes, pisos y muebles (al tener mayor longitud de onda). [30]

Una característica importante de este estándar es que implementa redes malladas (Mesh) para la comunicación entre nodos, permitiendo una cantidad alta de dispositivos intercambiando en la red y posibilitando un amplio rango en el despliegue de la misma. Los tres principales problemas que posee el estándar son:

- Al utilizar un red de mallado, la comunicación entre dispositivos se maneja en cada nodo, por lo que este requiere tener un poder de procesamiento acorde para ello, además del gasto de energía que pueda suponer el comunicarse con múltiples puntos de la red.
- Los chips avalados para trabajar con este estándar son fabricados por una sola compañía (Sigma Designs), lo cual afecta la producción de dispositivos que manejen el estándar por la distribución de los chips.
- Al usar la banda de los 900MHz dependiendo de la región en donde se planee utilizar los dispositivos, hace falta en muchos casos homologarlos, pues la banda puede ser utilizada para otros propósitos.

3.6. Seguridad

La seguridad de los dispositivos diseñados para formar parte del Internet de las cosas, deben estar centrados no solamente en la recolección de datos, el despliegue de información o acciones sobre el ambiente, sino en todos los eslabones que hagan parte del conjunto de tecnologías utilizadas para tales fines.

Los aspectos fundamentales de seguridad deben ser de carácter transversal a toda la infraestructura. También debe ser abordada durante todo el ciclo de vida de los dispositivos, desde el diseño inicial, pasando por las pruebas requeridas para comprobar su correcto funcionamiento, hasta la implantación en el entorno deseado y que incluye todo lo concerniente a la comunicación, al acceso y administración de los dispositivos, equipos y servicios de la arquitectura utilizada, a los datos que pueden proveer, o todo recurso que los haga un blanco ideal para los piratas informáticos.

Podemos separar en categorías las diferentes medidas aplicables para establecer la seguridad de la siguiente forma:

- Arquitecturas: Las soluciones tecnológicas deben contemplar las diversas amenazas y vulnerabilidades que puedan presentar el software, el hardware y el middleware en cada elemento de la arquitectura y que deben de ser contemplados desde la etapa de diseño. La evaluación de riesgos debe ser parte fundamental en cada escenario de uso y en cada momento de la puesta a punto de los infraestructura y validando los resultados a través del uso de pruebas, adquisición de certificados de entes competentes y reguladores y permitiendo auditorías periódicas de los resultados obtenidos.
- Dispositivos o equipos: Para todos los equipos utilizados en los procesos en los que interviene el Internet de las cosas, es importante que estos tengan durante su ciclo

de vida, actualizaciones regulares que corrijan errores y malos funcionamientos, tanto a nivel del sistema operativo, firmware de dispositivos y funcionalidades previstas a nivel de software y hardware. También es importante controlar el acceso a estos dispositivos y equipos por parte de otros equipos o servicios utilizando servicios de listas blancas y listas negras.

- Protocolos de seguridad en Internet: La seguridad que se puede brindar los protocolos de red es parte importante de las consideraciones en el diseño y funcionamiento de los dispositivos. Se establece haciendo uso de las características de encriptado, del uso de certificados de confianza y aprovechando las ventajas que pueda proveer cada protocolo en particular.
- Redes y Gateways: Desde el punto de vista de las redes, los dispositivos enruteadores y las puertas de enlace, la seguridad viene de la mano de los mecanismos provistos para las comunicaciones, haciendo uso de cortafuegos, logs y previniendo el acceso no autorizado a los equipos de red de manera física o remota y siempre utilizando métodos estrictos de autenticación de usuarios.
- Protocolos de comunicación: Los protocolos de comunicación son claves para mantener a salvo la información generada y en transmisión. La clave radica en que además de cumplirse con los estándares de calidad y seguridad establecidos por normativas o regulaciones, también se establezcan las políticas de seguridad extras con el fin de mantener a los nodos seguros y evitando las intrusiones de terceros.
- Consumidores y aplicaciones: Los dispositivos y equipos deben ser capaces de darle la opción a los usuarios de definir el nivel de seguridad requerido en los niveles de toda la infraestructura tecnológica utilizada, haciendo uso de interfaces de control o monitoreo. La educación de los usuarios, en cuanto a las mejores prácticas y el manejo adecuado de los dispositivos, plataformas y la información generada o almacenada, es también parte fundamental para mantener la seguridad en todo momento.
- Identificación y Privacidad: Los diversos usuarios y servicios (incluyendo las terceras partes) que pueden interactuar con los sistemas, tanto para el despliegue como para la recolección de datos e información, necesitan regirse según modelos de gestión, identificación y permisología adecuados, de manera que la privacidad, los privilegios y las integraciones y de todas las acciones que ocurren en los sistemas, estén siendo manejados de manera responsable, cumpliendo con los estándares de seguridad requeridos según el contexto.

La acelerada proliferación y adopción de dispositivos conectados ha despertado la preocupación general por las consecuencias que podrían llegar a tener los fallos de seguridad de los mismos. Para evitar que se realicen ataques que aprovechen las vulnerabilidades de las tecnologías implicadas en el futuro hace falta realizar una serie de esfuerzos [31] que ayuden a mitigar las amenazas posibles entre los cuales se sugieren:

- Coordinar a través de los departamentos y agencias gubernamentales para involucrarse con las partes interesadas en el aspecto de IoT y explorar conjuntamente formas de mitigar los riesgos planteados anteriormente.

- Crear conciencia de los riesgos asociados al uso de los dispositivos entre los grupos de interés. Es importante que las partes interesadas sean conscientes de los riesgos para que puedan posicionarse y abordarlos.
- Identificar y promover incentivos para incorporar la seguridad. Quienes formulan políticas, los entes legisladores y las partes interesadas deben considerar formas de incentivar los esfuerzos para mejorar la seguridad de los productos. En el entorno actual, a menudo no está claro quién tiene la responsabilidad de la seguridad de un producto o sistema dado. Además, los costos de la escasa seguridad a menudo no son soportados por los que están mejor posicionados para aumentar este aspecto. Los interesados deben considerar cómo la responsabilidad civil, la legislación, la reglamentación, la gestión voluntaria de la certificación, las iniciativas de establecimiento de normas y las iniciativas voluntarias al nivel de la industria y otros mecanismos podrían mejorar la seguridad y al mismo tiempo fomentar la actividad económica y la innovación innovadora.
- Contribuir a los procesos de desarrollo de estándares internacionales para éste ámbito. El Internet de las cosas es parte de un ecosistema global, y los países y organizaciones están comenzando a evaluar muchas de estas mismas consideraciones de seguridad. Es importante que las actividades relacionadas con ello no se dividan en conjuntos inconsistentes de normas o reglas.

Capítulo 4

Herramientas de Monitoreo, Visualización y Control

Si el Internet de las cosas es todo sobre la capacidad de conectar dispositivos con sensores y actuadores a otros dispositivos o Internet para lograr uno ó más fines, el sentido con el que operan estos dispositivos es gracias a la generación de datos, información y de conocimiento de manera transversal a las partes implicadas. ¿Pero como podemos realmente aprovechar a los dispositivos y las ventajas que nos traen si no somos capaces de entender que datos nos proveen?.

A principios de este milenio, en el mundo de la tecnología se empezó a gestar uno de los paradigmas más importantes en el área: Con los datos cada vez más fáciles de obtener, quienes fuesen capaces de interpretarlos de mejor manera y convertirlos en información útil serían quienes llevarían la punta de lanza de los nuevos desarrollos por venir. En el año 2006, Clive Humby, considerado como uno de los primeros científicos de datos, acuño la frase: “los datos son el nuevo petróleo”. Luego Micheal Palmer completo ésta idea diciendo “Los datos son valiosos, pero si no están refinados, en realidad no se pueden usar [32].

Si examinamos detenidamente el flujo de cualquier proceso de creación de conocimiento, esta afirmación se ve respaldada plenamente. Deben existir procesos intermedios en donde estos datos se conviertan en información útil para quienes observan dichos procesos esperando responder o resolver problema, necesidad o inclusive encontrar las oportunidades que otros no han visto, ventajas que en el mundo competitivo de hoy son fundamentales. Una manera de hacer esto es a través de ordenar los datos para mostrarse basados en una lógica. Este paradigma de tratamiento de la información es natural para el ser humano, por lo que saber aprovechar de manera correcta el como presentar datos a contrapartes para respaldar una hipótesis es una forma muy efectiva de generar valor a los datos sobre un particular.

Ahora supongamos que los datos que observamos presentan un panorama en tiempo real de características de fenómenos y que estos de momento no persiguen actividades pasadas. El representar esos datos para hacer seguimiento del fenómeno adquiere una dimensión de vigilancia. Lamentablemente los seres humanos no son aptos para ser

participe de vigilancia de datos e información a través del tiempo, pero teniendo herramientas de visualización adecuados así como ciertas rutinas establecidas según se demande en el caso caemos en el concepto de monitoreo. Esta actividad cuya faceta es en muchos aspectos pasiva puede llegar a ser tediosa por lo que contar con indicadores que ayuden a ver los cambios de estados de los fenómenos observados es vital en muchos ámbitos.

Pero, ¿que sucede si ese monitoreo nos remite a una situación en donde debamos actuar?, ¿como podemos cambiar las características de lo que está ocurriendo cuando muchas veces donde ocurre no es siquiera el mismo lugar geográfico donde se encuentra el fenómeno en sí? Es allí donde surgen herramientas cuyo fin es poder operar uno o múltiples flujos de tareas acordes a recetas existentes o requiriendo la intervención humana para establecer la consecución de un objetivo sobre la situación establecida. A estas herramientas las llamamos herramientas de control.

Muchas veces encontramos herramientas que cumplen muy bien uno o incluso dos de los aspectos mencionados anteriormente. Por ejemplo, es muy común que las herramientas de visualización permitan en alguna medida el monitoreo de datos en tiempo real, o herramientas de control permitan tener la capacidad de monitorear algunos procesos de manera general.

La importancia de reconocer cual herramienta es la correcta para un caso u otro en particular significa que la obtención de conocimiento, la vigilancia y la gestión de procesos se realmente eficiente, aumentando el valor de las tareas que se realizan, independientemente del área al cual se aplique. A continuación se presentan de manera más detallada las características de estas herramientas y como estos son utilizados en la actualidad en consonancia con el problema de investigación planteado.

4.1. Herramientas de Visualización

Una de las maneras en que se puede inferir un conocimiento nuevo es a través de la información que se puede desplegar de manera visual. Los seres humanos están acostumbrados a que de ésta forma podamos entender mejor nuestro ambiente por lo que es natural pensar que a partir de un conjunto de datos que se presenten de manera ordenada y lógica podemos ser capaces de comprender elementos que no necesariamente se pueden ver a simple vista. Así podemos definir como la visualización de datos como la representación gráfica de información y datos. Al utilizar elementos visuales como cuadros, gráficos y mapas [33].

Es así como el ser humano desde la antigüedad ha buscado las mejores formas de representar esos datos e información. Disciplinas como la matemática han permitido a las personas atravesar la barrera de datos inconexos a útil y accionable. En la actualidad la gran disponibilidad de los datos hace que cada vez más necesitemos de herramientas que no permitan resumir de forma adecuada y gráfica la descripción de uno o múltiples fenómenos de los cuales se plantea la extracción de la información. Es como las herramientas de visualización de datos proporcionan una manera accesible de

ver y comprender tendencias, valores atípicos y patrones en los datos.

Desde el punto de vista computacional las herramientas de visualización son cada vez más importante para darle sentido a las billones de filas de datos que se generan cada día. La visualización de datos ayuda a contar historias seleccionando los datos en una forma más fácil de entender [33] siempre recordando que el objetivo final es poder transmitir el análisis de dichos elementos a un tercero.

Para el caso de la investigación se ha de tener en cuenta que buscamos una forma de poder observar datos que ingresan de múltiples fuentes y características, teniendo en cuenta que los datos de los sensores de dispositivos IoT en realidad no son homogéneos y por lo tanto, los usuarios que deseen ver esos datos deben ser capaces de poder comprenderlos haciendo uso de cualquiera de los métodos existentes para visualizarlos.

4.1.1. Métodos utilizados para visualizar datos

Independientemente de la herramientas de visualización, existen un conjunto enorme de formas de representar los datos, dependiendo de las características que estos posean. Es importante señalar el aspecto de poder seleccionar de manera correcta el método adecuado para representarlos es una elección crítica, pues en una etapa de análisis, equivocarse puede representar una conclusión errónea.

Entre los diversos métodos de visualización tenemos:

- Gráficos: Las visualizaciones gráficas informan de comparaciones, relaciones y tendencias. Resaltan y hacen más claras las cifras. [34] La representación gráfica permite establecer valores que no se han obtenido experimentalmente sino mediante la interpolación (lectura entre puntos) y la extrapolación (valores fuera del intervalo experimental). Entre los mas comunes tipos de gráficos están los gráficos en columnas, gráficos en barras, gráficos de linea, gráficos de área, gráficos circulares (torta o donut), entre otros.
- Tablas: Las tablas son formas ordenadas de datos en los cuales se busca categorizarlos basados en su posición por filas, por columnas o por ambas.
- Mapas: Un mapa no es más que la representación geográfica simplificada dando información relativa al plano observado. Generalmente a dos dimensiones se suelen representar las diferencias entre características usando colores.
- Infografías: Las infografías son diagramas visuales complejos que aprovechan recursos como imágenes, texto, diagramas, logos, etc, para poder establecer de manera didáctica una idea de una manera dinámica.
- Cuadros de mando: Es un conjunto de indicadores representados de manera agrupada generalmente usados para representar resultados (parciales o totales) sobre datos a los que ya se las ha realizado un análisis previo. Busca representar solo la información indispensable para responder una serie de preguntas específicas.

4.2. Herramientas de Monitoreo

Las herramientas de monitoreo son aquellas que se basan en el proceso continuo y sistemático en el cual se verifican la eficiencia y la eficacia de parámetros de uno o más elemento mediante la identificación de comportamientos o de la presentación de datos para detectar eventuales anomalías.

Para el proceso de monitoreo es importante entender que la vigilancia del fenómeno observado es plenamente identificable con los datos recibidos, y muchos veces estos son representados con información gráfica de manera similar a la manera que lo hacen las herramientas de visualización de datos. Sin embargo es importante entender que la visualización de datos aunque se puede dar en tiempo real, esta mas pensando para entender el pasado en base los datos mismos. El monitoreo entiende que existe previamente un conocimiento con el cual se debe regir la observación para identificar las anomalías antes mencionadas.

En general las herramientas de monitoreo también sirven para automatizar situaciones críticas o realizar alertas basadas en comportamientos inesperados en la data que esta ingresando. Dichas alertas pueden verse representadas en diversas formas como indicadores visuales en un panel, correos electrónicos, mensajería instantánea o cualquier otro método que este integrado y disponible para llamar la atención de los responsables e incluso permitiendo que otros sistemas puedan actuar de manera automática en respuesta a la eventualidad.

En el marco de esta investigación se requiere que la herramienta sea capaz de poder hacer el seguimiento continuo de múltiples variables sobre sensores y actuadores registrados. Esta información puede ser usada para crear alertas automáticas basadas en los patrones reconocidos anteriormente en el contexto del dispositivo IoT.

4.3. Herramientas de Control

Las herramientas de control difieren de las herramientas anteriores porque estos no buscan mostrar datos e información al usuario final, sino que buscan establecer formas de gestión de procesos y modificar basado en una o una serie de reglas el comportamiento de otro artefacto. Estas herramientas permiten planificar tareas y procedimientos de manera previa para un escenario dado, así como también permiten que las personas intervengan en escenarios donde se requiere del conocimiento experto no programado de una automatización.

En particular los software de control son todos aquellos sistemas ó programas que se encuentran desplegados a lo largo o ancho de las operaciones de un flujo de trabajo y son omnipresentes en la actualidad de cualquier industria o empresa y buscan la menor intervención posible de las personas para evitar introducir el componente del error humano, así como también orquestar las acciones de procesos tediosos o costosos en recursos.

Una herramienta de control robusta debe ser capaz no solo llevar los procesos como

fueron establecidos y automatizados sino también dar la oportunidad a los usuarios finales de tomar el mando en cualquier momento y en el mejor de los escenarios, ser lo suficientemente flexible para poder generar nuevos flujos o modificar los existentes de manera que todo el proceso sea parametrizable.

Finalmente, para el caso de los dispositivos IoT una herramienta de control debe soportar estándares y protocolos con los cuales este garantizada la interoperatividad de los sensores y actuadores, de forma que estos sean capaces de recibir ordenes de una forma centralizada acorde a las necesidades del usuario. También el fin de poder controlar los dispositivos es también generar procesos automatizados complejos a partir de programación que no necesariamente existe como regla de negocio dentro del dispositivo sino que se pueda dar a partir de los requerimientos de un usuario final.

Capítulo 5

Placas Programables

5.1. Definición

Las placas programables, también llamadas placas de programación o desarrollo son dispositivos básicos que cuentan con un microcontrolador programable con el cual se pueden ejecutar diferentes instrucciones. En muchos escenarios estamos en presencia de una computadora de recursos muy reducidos (conocidos como Single Board Computer o SBC por sus siglas en inglés), pues siguen al pie de la letra con la arquitectura del computador de John von Nuemann [35].

Estas computadoras de una sola placa se hicieron como sistemas de demostración o desarrollo, para sistemas educativos o para su uso como controladores integrados de computación los que las hace especialmente útiles para poder crear prototipos funcionales de dispositivos de toda clase, desde robots hasta dispositivos de Internet de las cosas. Para poder crear las rutinas se hacen uso de lenguajes de programación que establecen el control directo contra la microprocesador o incluso contra los sensores y actuadores que puedan estar conectados al dispositivo.

Algunas placas incluso poseen la capacidad de soportar un sistema operativo completo los cuales los hacen inmensamente flexibles a la hora de realizar tareas, de agregar nuevos sensores o controlar otros dispositivos a través de esa placa. Existen gran cantidad de modelos distintos en el mercado. Sin embargo en este trabajo de investigación nos centraremos en dos tipos de placas en específico:

5.2. Placas Arduino

Arduino (Genuino a nivel internacional es una plataforma electrónica en hardware y software libres, fáciles de usar. Está pensado para cualquier persona que haga proyectos interactivos. [36](figura 5.1) Las placas Arduino tiene la capacidad detectar el ambiente recibiendo entradas de variedad de sensores, y afecta su entorno controlando luces, motores y otros actuadores. La lógica de micro controlador Arduino deberá hacerse escribiendo código en el lenguaje de programación Arduino (basado en C++) y utilizando el entorno de desarrollo de Arduino.



Figura 5.1: Logo del proyecto Arduino

Arduino nació en el Ivrea Interaction Design Institute como una herramienta fácil para el prototipado rápido, dirigido a estudiantes sin formación en electrónica y programación. Tan pronto como llegó a una comunidad más amplia, la junta de Arduino comenzó a cambiar para adaptarse a las nuevas necesidades y desafíos, diferenciando su oferta de simples placas de 8 bits a productos para aplicaciones IoT, portátiles, impresión 3D y dispositivos embebidos. Todas las placas de Arduino son totalmente de hardware libre, lo que permite a los usuarios construirlas independientemente y eventualmente adaptarlas a sus necesidades particulares. El software, también es de código abierto y está creciendo a través de las contribuciones de los usuarios de todo el mundo.

Las placas Arduino están disponibles de dos formas: ensambladas o en forma de kits “Hágalo tú mismo”. Los esquemas de diseño del hardware están disponibles bajo licencia Libre, con lo que se permite que cualquier persona pueda crear su propia placa Arduino sin necesidad de comprar una prefabricada. La primera placa Arduino fue introducida en 2005 llamada Arduino Uno (5.2), ofrecida a un bajo costo y facilidad de uso para novatos y profesionales. Se buscaba desarrollar proyectos interactivos con su entorno mediante el uso de actuadores y sensores. A partir de octubre de 2012, se incorporaron nuevos modelos de placas de desarrollo que usan micro controladores Cortex M3, ARM de 32 bits, que coexisten con los originales modelos que integran micro controladores AVR de 8 bits.

En la actualidad, existen 29 distintas placas de micro controladores, 7 módulos complementarios, 15 Shields (placas de expansión), con lo cual se posee un ecosistema rico de diversos micro controladores para cada tarea o cada ambiente de desarrollo.



Figura 5.2: Arduino Uno R3

5.3. Placas Raspberry Pi

Raspberry Pi es una marca (figura 5.3) de micro computadores de hardware libre, que comenzó como proyecto para llevar a la educación de escolares ingleses conceptos de la computación, de la programación y de la electrónica, y que luego la comunidad Maker y algunas compañías adoptaron, pues vieron el potencial para crear dispositivos, sensores, actuadores, robots entre otros y para poder llevar a cabo prototipos funcionales de dispositivos y de productos muy rápidamente. El relativo bajo costo (que va desde \$10 por la placa mas sencilla hasta \$80 por el modelo más avanzado) y facilidad de uso, han facilitado la adopción de este micro computador por personas en todo el mundo para toda clase de proyectos amateurs y profesionales.

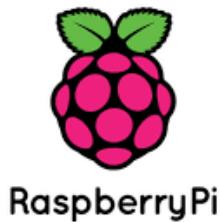


Figura 5.3: Logo de la marca Raspberry Pi

Este es un proyecto llevado a cabo con la Raspberry Pi Foundation [37] desde el año 2012 ofreciendo hasta el día de hoy siete modelos distintos de placas, con procesadores ARM, memoria RAM que va desde los 256MB hasta 1GB y cuya memoria es de carácter externo usando tarjetas MicroSD en la mayoría de los modelos y cuyo consumo no es mayor al de 2,5 amperios. Los modelos son:

- Raspberry Pi Modelo A: Fue el primer modelo de Raspberry Pi en salir al mercado, en el año 2012. Basado en un SoC Broadcom BCM28235, cuyo procesador es un ARM11 32 bits a 700MHz, Gráficas Broadcom VideoCore IV, con 256MB de memoria RAM, un puerto USB, una salida HDMI, un conector RCA, una entrada CSI para un modulo de cámara, y sin características de conectividad alguna por defecto. Para el almacenamiento se usan tarjetas SD.
- Raspberry Pi Modelo B/B+: También del año 2012, es una variante del Modelo A, trajo consigo diversas mejoras, como la inclusión del doble de memoria RAM, pasando de 256MB a 512MB. Trajo consigo un puerto USB más y un conector Ethernet (RJ-45). Se mantuvo tanto su tamaño como su coste. No hubo variaciones en el procesador ni en la gráfica. Tiempo después se lanzó el Modelo B+, que incluyó 4 puertos USB y pasó de usar una SD a una MicroSD.
- Raspberry Pi 2 Modelo B: Lanzada en 2014 es el primer modelo que no incluye el mismo procesador usado en los tres anteriores: se sustituye por uno de la misma marca, pero de modelo BCM2836 con lo cual pasa de ser de un núcleo a cuatro, y de 700MHz a 900MHz, no obstante emplea la misma gráfica, la VideoCore IV. Dobra la cantidad de memoria RAM, pasando de 512MB a 1GB de memoria (también compartida con la gráfica). También incluye 40 pines GPIO, y mantiene los cuatro puertos USB. Suprime la conexión RCA.

- Raspberry Pi 3 Modelo B: Sale al mercado en el año 2016, renovando procesador, una vez más de la compañía Broadcom, siendo un procesador de cuatro núcleos al igual que el modelo anterior, pero pasa de 900MHz a 1.20GHz, manteniendo la RAM en 1GB. Su mayor novedad fue la inclusión de WiFi y Bluetooth (4.1 Low Energy) sin necesidad de adaptadores.
- Raspberry Pi Zero: Fue el primer modelo miniaturizado de las Raspberry Pi, teniendo un tamaño un poco mayor a un Pen Drive. Lanzado en 2015 con un coste de 5 dólares, es una 40 % más potente que el primer modelo de Raspberry. Tiene un CPU Broadcom BCM2835, que funciona a 1GHz con dos núcleos. Posee 512MB de RAM, y comparte la gráfica VideoCore IV. Debido a su tamaño sustituye el puerto HDMI por MiniHDMI, manteniendo así las prestaciones. Tampoco usa USB estándar, sino que tiene dos MicroUSB, uno de alimentación y otro de datos. Posee salida RCA, pero en vez de por clavija son solo dos conectores integrados en la placa. Usa MicroSD como sistema de almacenamiento.(figura 5.4)

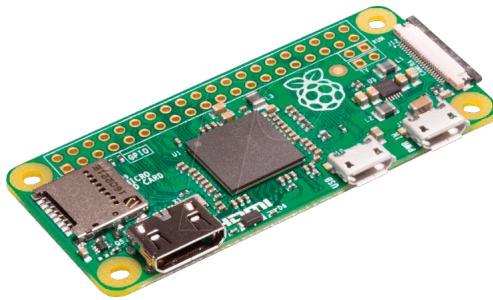


Figura 5.4: Raspberry Pi 4 modelo B

- Raspberry Pi Zero W: Es la sucesora de la Raspberry Pi Zero. la W es por Wireless, ya que la única novedad de esta placa con respecto a su antecesora es la inclusión de WiFi y Bluetooth, por lo que su precio ascendió a 11 dólares.
- Raspberry Pi 3 Modelo B+: Del mismo modo que para el Raspberry Pi Modelo B, esta es una actualización en términos de rendimiento sobre la Raspeberry Pi 3 Modelo B. Pasa a ser un poco más rápida que su antecesora y ofrece la capacidad de conectarse a Wifi doble banda. Incorpora Bluetooth 4.2 y la velocidad de su puerto ethernet pasa de 100 Mbits/seg a 300 Mbits/seg.
- Raspberry Pi 3 Modelo A: Los modelos A+ presentan menores prestaciones a un menor precio. Cuenta con 512 MB de RAM (compartidos con la GPU VideoCore IV), un solo puerto USB y sin puerto de conexión de red ethernet.
- Raspberry Pi 4 modelo B: Esta versión viene a sustituir a la versiones 3 y cambia los puertos HDMI de tamaño completo por dos puertos microHDMI. Cuenta con la capacidad de manejar una pantalla a 4K a 60 Hz, o dos pantallas 4K a 30 Hz. Se ha incluido por primera vez USB 3.0, y el puerto Ethernet ya no está limitado a 300 Mbps. Tiene un procesador Broadcom nuevo. Es la primera serie que posee 3 variantes disponibles en los que cambia la cantidad de memoria RAM de 2GB, 4GB, y de 8GB (figura 5.5).



Figura 5.5: Raspberry Pi 4 modelo B

- Raspberry Pi 400: Cuenta con una placa personalizada que se deriva de la Raspberry Pi 4 existente, específicamente remodelada para incluirla en un teclado derivado del Raspberry Pi Keyboard. Posee una solución de enfriamiento mas robusta que su contraparte 4 modelo B al poseer una placa de metal ancha para disipar el calor y un conmutador actualizado para la fuente de alimentación que es un poco mas alta que la del Raspberry Pi 4. La computadora cuenta con 4GB de memoria RAM LPDDR4.
- Raspberry Pi Pico: Anunciada en el 2021, es una placa pequeña y versátil construida con RP2040, un nuevo chip microcontrolador diseñado por Raspberry Pi Fundation. Este modelo está gobernada por un pequeño SoC que cuenta con un procesador dual core ARM Cortex M0+ funcionando a 133 MHz, acompañado de 264 KB de RAM y 2 MB de almacenamiento integrado.
- Raspberry Pi 5: La mas nueva de las versiones del Raspberry Pi existentes al momento de la escritura de esta investigación representa el salto de rendimiento sobre el modelo anterior de Raspberry Pi 4. Con un procesador Broadcom BCM2712 casi duplica la frecuencia del CPU con respecto a su predecesora, además de contar ahora con una nueva GPU VideoCore VII mucho más rápida con una frecuencia de 800 MHz. Entre otras mejoras.

Existen gran variedad de sistemas operativos que pueden ser usados con este micro computador, la mayoría de ellos, basados en Linux, pero también con la posibilidad de instalar Windows 10 (Iot Core) o Android (Android Things).

Una característica principal de estas placas es la existencia de 40 pines GPIO, los cuales proveen entradas y salidas digitales para el micro computador con lo que se le puede expandir sus funcionalidades con el uso de sensores (HATs o a través de circuitos eléctricos tradicionales), cuya lógica puede ser creada en lenguajes de programación como Python, C++, Java, entre otros.

El proyecto ha ganado una comunidad muy amplia alrededor del mundo, que han probado la versatilidad que posee este micro computador en proyectos de toda índole, dando a las personas la capacidad de crear soluciones basadas en IoT con una relación coste-rendimiento positivo.

Parte II

Marco Aplicativo

Capítulo 6

Metodología General de Trabajo

6.1. Scrum

Scrum es una metodología ágil y flexible para gestionar el desarrollo de software. Se basa en construir primero la funcionalidad de mayor valor para el cliente y en los principios de inspección continua, adaptación, auto-gestión e innovación [38]. En Scrum se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente y obtener el mejor resultado posible de un proyecto.

Es parte de la filosofía de Scrum el poder realizar entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados rápidamente, con requisitos cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales.

En Scrum un proyecto se ejecuta en bloques temporales cortos y fijos (iteraciones) que normalmente son de 2 semanas, aunque en algunos equipos son de 3 y hasta 4 semanas, límite recomendado para proveer de un buen feedback y reflexión sobre lo realizado. Cada iteración tiene que proporcionar un resultado completo, un incremento de producto final que sea susceptible de ser entregado con el mínimo esfuerzo al cliente cuando lo solicite. El proceso parte de la lista de objetivos/requisitos priorizada del producto, que actúa como plan del proyecto. En esta lista el cliente prioriza los objetivos balanceando el valor que le aportan respecto a su coste y quedan repartidos en iteraciones y entregas. Finalmente el proceso se lleva a cabo tantas veces como sea requerido para poder finiquitar el proyecto (véase figura:6.1).

Las actividades que se llevan a cabo bajo la metodología de Scrum son las siguientes:

1. Planificación de la iteración: En el primer día de cada sprint, el equipo hace reunión de planificación de la iteración. Esta etapa consiste en una primera fase de selección de requisitos, en la cual el cliente presenta al equipo la lista de requisitos priorizada del proyecto y se responden a las dudas que surgen sobre el dominio de las tareas a realizar y una segunda fase de planificación de la iteración, donde el equipo elabora la lista de tareas necesarias para desarrollar lo que se ha requerido

para la iteración, ademas de la estimación de esfuerzos por parte de los miembros del equipo.

2. Ejecución de la iteración: Cada día se realiza una reunión con todo el equipo, cuya duración es de 15 minutos como máximo para inspeccionar el avance en la realización de las tareas, dependencias u actividades bloqueantes y/o insumos pendientes para su culminación. Durante la iteración el Scrum Master (facilitador) se encarga de que el equipo pueda articularse y cumplir los compromisos adquiridos por el equipo.
3. Inspección y adaptación: El ultimo día de la iteración se realiza la reunión de revisión de la iteración. Esta se conforma de dos partes, la demostración en donde el equipo presenta al cliente el/los entregable(s) con los requisitos mínimos acordados y la retrospectiva en la que el equipo analiza si su manera de trabajar ha sido la adecuada, los problemas que han surgido, su solución, de forma que se pueda mejorar de manera continua la productividad.

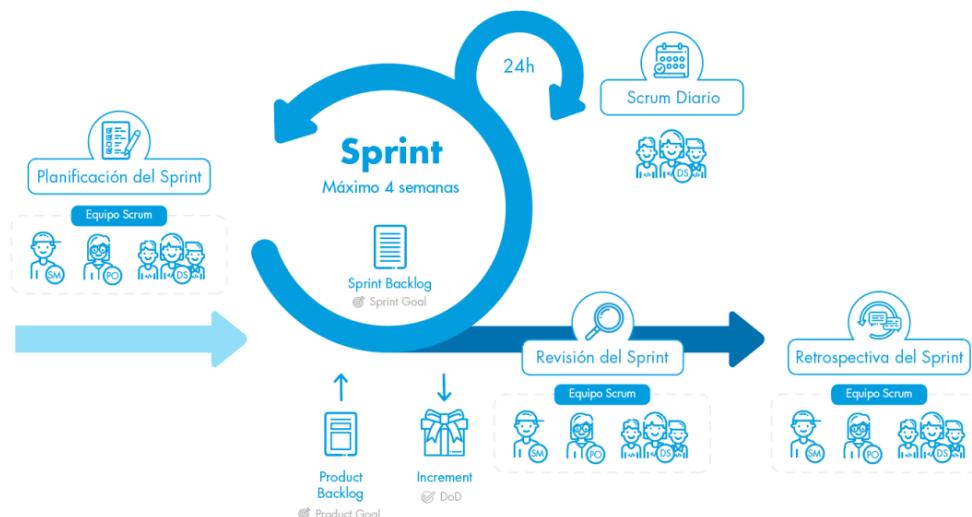


Figura 6.1: Etapas de desarrollo de la metodología Scrum

6.2. Control de Versiones de Software

Un software de control de versiones es un aquél que permite registrar y gestionar cambios a nivel del código fuente de manera histórica permitiendo retornar a versiones anteriores o comparar diferencias entre ellas.

Esto permite tener:

- Flujos de trabajo organizados: A través del uso de versiones se pueden crear o modificar características siguiendo metodologías de desarrollo sin que esto suponga inconvenientes para el proyecto.

- Descripción de versiones: Cada versión viene acompañada de una descripción que permite reconocer el cambio realizado en el proyecto.
- Colaboración: Permite que las personas puedan trabajar de manera independiente en el mismo proyecto sin temor de interferir en el trabajo de otros colaboradores.
- Historial de cambios: Se tiene una lista completa de los cambios realizados a través del tiempo en el proyecto permitiendo en cualquier momento volver a versiones previas.

6.2.1. Git

Git es un sistema de control de versiones moderno, distribuido y seguro que fue desarrollado originalmente por Linus Torvalds en el año 2005. Entre sus características principales se destacan:

- Arquitectura distribuida: Cada copia del proyecto se sincroniza con el resto de manera distribuida.
- Flexibilidad: Permite adaptar la manera en que se hacen los cambios según la metodología de desarrollo que sigue el proyecto.
- Seguridad: Permite gestionar el tipo de control que tienen los colaboradores sobre los proyectos. Además da la posibilidad de identificar quien o quienes realizan cambios sobre el proyecto.
- Rendimiento: Solo los cambios realizados son sincronizados a través de las copias del proyecto por lo que es eficiente en el envío y recepción de los datos en la red.

6.2.1.1. Git-Flow

Git Flow es una metodología para usar Git en la que se combina la capacidad de git de trabajar versiones a través de ramas (figura:6.2). Estas, se separan entre ramas de función y ramas principales. Fue popularizado por Vincent Driessen en 2010. Los aspectos clave del modelo Git Flow son:

- Ramas principales y de desarrollo: En lugar de tener una única rama principal, Git Flow utiliza dos ramas principales:
 1. Main (o Master): Es la rama con el proyecto en su estado de producción. Almacena el historial oficial del proyecto hasta la actualidad.
 2. Develop: Sirve como rama de integración para las funciones que se van desarrollando. También es utilizado para realizar pruebas y comprobar que los requerimientos nuevos y antiguos funcionen adecuadamente.
- Ramas de función: Los desarrolladores crean ramas de función para trabajar en características específicas. Estas ramas se fusionan con la rama develop cuando la función está completa.
- Ramas de publicación y corrección de errores:

1. Release Branches: Se crean a partir de develop para preparar una nueva versión. Aquí se realizan pruebas finales y se corrigen errores antes de la publicación.
 2. Hotfix Branches: Se utilizan para corregir errores críticos en la versión actual de producción. Se crean a partir de main y se fusionan tanto con main como con develop.
- Versionado y etiquetado: Git Flow simplifica la gestión de versiones al asignar números de versión a las confirmaciones en main y etiquetarlas.

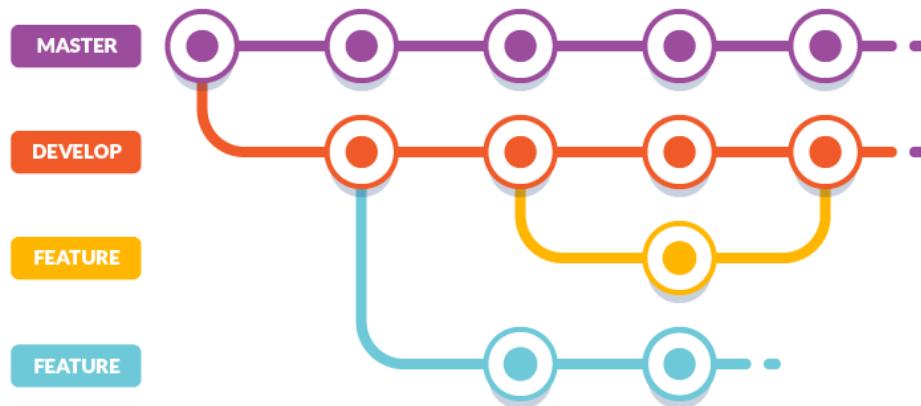


Figura 6.2: Ramas de desarrollo usando la metodología Git-flow

6.3. Herramientas de Desarrollo

A continuación se detalla aquellas herramientas de hardware y de software utiizados durante el desarrollo de este trabajo de investigación.

6.3.1. Herramientas de Hardware

Dado el enfoque de tener agrupados dispositivos IoT que pudiesen obtener data real del entorno distintos en donde fuesen desplegados, se procedió a obtener un conjunto de placas programables así como también de sensores y actuadores para poder crear prototipos funcionales. Se contó con cuatro placas programables de distinta índole:

- Dos Raspberry Pi modelos 3 B.
- Un Raspberry Pi Zero.
- Un Arduino Uno R3.

Entre los sensores y actuadores a integrar a esos dispositivos se ha contado con los elementos de la tabla 6.1

Sensor	Tipo de medición	Cantidad
PIR HC-SR501	Movimiento	Dos (2)
Robodo Sen18	Nivel de Agua	Uno (1)
DS1820	Temperatura	Uno (1)
Leds RGB	Leds (iluminación)	Dos (2)
LDR	Fotorresistencia (iluminación)	Uno (1)
DHT11	Temperatura y Humedad	Uno (1)
Leds color verde	Leds (iluminación)	Dos (2)
Leds color rojo	Leds (iluminación)	Dos (2)
Leds color blanco	Leds (iluminación)	Seis (6)
TSL2561	Intensidad lumínica (iluminación)	Uno (1)
Buzzer HW-508	Buzzer (sonido)	Uno (1)
LCD 16x2 I2C Hd44780	Pantalla LCD	Uno (1)
RFID RC522	Lector de radio frecuencia	Uno (1)
RFID 100	Llavero RFID	Uno (1)
RFID Card	Tarjeta RFID	Uno (1)
RaspiCam V1	Camara	Uno (1)

Tabla 6.1: Lista de sensores y actuadores utilizados en los prototipos de dispositivos IoT

Por otro lado para el desarrollo de software se hizo uso de un computador con las siguientes características:

- CPU Intel Core Duo a 2,0 GHz
- 4 GB RAM DDR2
- 256 GB HDD Sata
- Sistema operativo Mac OS X Yosemite

Finalmente el software desarrollado fue desplegado en uno de los dispositivos Raspberry Pi 3 modelo B cuyas características son:

- CPU ARM Cortex-A53 Quad core a 1,2 GHz
- 1 GB de RAM LPDDR2
- 64 GB microSD
- Sistema operativo Raspbian (Basado en Debian 8)

6.4. Herramientas de Desarrollo

A continuación se detalla aquellas herramientas de hardware y de software utiizados durante el desarrollo de este trabajo de investigación.

6.4.1. Python

Python es un lenguaje de programación de alto nivel, multiplataforma, débilmente tipado de propósito general, multiparadigma e interpretado [39], creado por Guido Van Rossum. Su filosofía se basa en el poder crear código que sea muy legible, con una sintaxis simple, utilizando indentaciones para delimitar bloques de código, más corto y potente que en otros lenguajes de programación. [40]

Se dice que Python es un lenguaje que viene con "pilas puestas" [41], es decir, que de por si, posee un conjunto de funciones amplio para afrontar cualquier tipo de situación dentro de las librería estándar del lenguaje y destaca en su facilidad para aprender y alta portabilidad y es usado en una gran cantidad de aplicaciones, que van desde el área web, pasando por la ciencia de datos, la inteligencia artificial y la programación de dispositivos.

La escogencia sobre este lenguaje de programación para el desarrollo recae en la facilidad del mismo, sin que esto afecte el rendimiento o legibilidad al momento de realizar el código. Su característica multipropósito y la portabilidad lo hace especialmente útil debido a que muchos de los sensores y actuadores poseen drivers nativos sobre este lenguaje. De cara al desarrollo web, Python también es uno de los lenguajes de programación más utilizados dadas la estabilidad y potencia de las diversas librerías y frameworks de desarrollos existentes.



Figura 6.3: Logo de Python

6.4.2. Django

Django es un framework web de alto nivel que permite el desarrollo rápido de sitios web seguros y mantenibles. Desarrollado por programadores experimentados, Django se encarga de gran parte de las complicaciones del desarrollo web, por lo que puedes concentrarte en escribir tu aplicación sin necesidad de reinventar la rueda. Es gratuito y de código abierto por lo que es fácil de construir y modificar elementos sobre el. Fue escogido por el poco esfuerzo requerido para poder tener un ambiente web completo, la gran cantidad de plugins y por la experiencia del desarrollado con el mismo.



Figura 6.4: Logo de Django

6.4.3. Bases de datos

Para el desarrollo de la aplicación web se desplegaron dos bases de datos para poder aprovechar las características de los datos y flujos involucrados:

- Una base de datos Postgresql para la gestión de los datos y elementos de la aplicación web en sí misma. Su robustez y madurez, sumado a la facilidad de despliegue y utilización fueron las razones de su escogencia.
- Una base de datos InfluxDB para almacenamiento de la información registrada por la operación de los sensores y actuadores de los dispositivos IoT. Esta base de datos fue seleccionada por su rol de los datos de sensores estrechamente relacionados con el tiempo de su registro, así como su facilidad de consulta.

6.4.3.1. Postgresql

Es un sistema de gestión de bases de datos relacional orientado a objeto que utiliza el lenguaje estándar de consulta (SQL por sus siglas en inglés) y de código abierto. Sus principales características son:

- Alta concurrencia.
- Amplia cantidad de tipos nativos.



Figura 6.5: Logo de PostgreSQL

6.4.3.2. InfluxDB

Es una base de datos de series temporales de código abierto desarrollada por InfluxData, utilizada para almacenar y recuperar datos de series temporales en áreas como monitoreo de operaciones, métricas de aplicaciones, datos de sensores de Internet de las Cosas y análisis en tiempo real. Está escrita en el lenguaje de programación Rust.



Figura 6.6: Logo de InfluxDB

6.4.4. Eclipse Mosquitto

Eclipse Mosquitto es un broker de mensajes de código abierto que implementa las versiones 5.0, 3.1.1 y 3.1 del protocolo MQTT. Al ser asíncrono y ligero ademas de poder ser usado en toda clase dispositivos, desde computadoras de placa única de baja potencia hasta servidores completos lo hacen especialmente útil para enviar información de dispositivos IoT, por lo que fue elegido como la mejor alternativa para el desarrollo de este proyecto. Esta escrito en el lenguaje de programación C pero posee librerías para interactuar con gran variedad de lenguajes de programación.



Figura 6.7: Logo de Eclipse Mosquitto

6.4.5. Grafana

Grafana es una plataforma de visualización de código abierto que permite generar métricas, registros y trazas de datos de otras aplicaciones a través del uso de graficos y tablas organizados en dashboards. Permite consultar, visualizar, configurar alertas y comprender tus métricas, sin importar dónde se almacenen. Su versatilidad, facilidad de uso y solidez lo convirtieron en la opción de facto para la integración de visualización de datos en el desarrollo del trabajo de investigación.



Figura 6.8: Logo de Grafana

6.4.6. Node-Red

Node-RED es una herramienta de programación que te permite conectar, integrar y automatizar dispositivos de hardware, APIs y servicios en línea. Su editor basado en paneles dentro del navegador facilita la creación de flujos mediante una amplia variedad de nodos, que luego se pueden implementar en su entorno de ejecución al mismo tiempo de manera sencilla. Se escogió como la integración de control y automatización de tareas por su amplio catalogo de instrucciones y plugins disponibles.

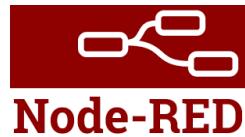


Figura 6.9: Logo de NodeRed

6.4.7. Docker

Docker es una plataforma de código abierto que simplifica y automatiza el proceso de construir, desplegar y gestionar contenedores. Los contenedores son componentes estandarizados y ejecutables que combinan el código de la aplicación con las dependencias del sistema operativo.



Figura 6.10: Logo de Docker

6.5. Adaptación del Marco Metodológico

Para llevar a cabo la investigación propuesta, se decidió adaptar la metodología de forma que está favoreciera el desarrollo de los diversos elementos que se requerían. De esta manera se acordó hacer uso de lo siguiente:

- Sprints de dos semanas de duración.
- Reuniones mensuales para realizar retrospectiva, sprint review y planificación de actividades para los siguientes sprints. Reunión en formato daily cada semana para comentar avances, bloqueos y recursos requeridos.
- Hacer entregas incrementales basadas en la adición de features y correcciones de código cada final de sprint.
- Kick off del proyecto para el mes de septiembre del 2018.
- Duración estimada del desarrollo de los dispositivos y de la aplicación web de 9 sprints (18 semanas).
- El profesor Antonio Russoniello en calidad de tutor del trabajo especial de grado, asume el rol de Scrum Master.
- El bachiller Pedro Boll asume el desarrollo.

Con ello definido junto con la selección de las tecnologías para el desarrollo, se presentan las siguientes actividades para el desarrollo y culminación del proyecto según los objetivos generales y específicos en la tabla 6.2, junto con su duración aproximada.

Actividades de desarrollo		
Actividad	Tiempo Estimado	Sprint
Levantamiento de requerimientos	Una semana	Sprint 1
Instalación y despliegue de Infraestructura	Una semana	Sprint 1
Diseño y prototipado de dispositivos	Una semana	Sprint 2
Desarrollo de scripts para sensores, actuadores y comunicación de datos	Dos semanas	Sprint 2 y sprint 3
Desarrollo de back-end del proceso de obtención de datos	Una semana	Sprint 3
Captura de datos de dispositivos	Doce semanas	Sprint 4 a sprint 10
Documentación de proceso de obtención de datos	Una semana	Sprint 4
Diseño de aplicación web	Una semana	Sprint 4
Implementación base de aplicación web (módulo inicial, templates base)	Una semana	Sprint 5
Desarrollo de módulo de gestión de aplicación	Una semana	Sprint 5
Desarrollo de módulo de visualización de datos	Una semana	Sprint 6
Desarrollo de módulo de control y automatización de flujos de dispositivos	Una semana	Sprint 6
Desarrollo de módulo de comunicación y redes	Una semana	Sprint 7
Documentación de desarrollo de aplicación web	Dos semanas	Sprint 7 y sprint 8
Pruebas end to end de todos los flujos	Tres semanas	Sprint 8 y sprint 9
Recolección de resultados del proyecto de investigación	Dos semanas	Sprint 10

Tabla 6.2: Actividades sugeridas para el desarrollo del trabajo especial de grado

Capítulo 7

Diseño e Implementación

En este capítulo se presenta a partir del problema planteado una solución basada en una aplicación web que es capaz de poder tomar los datos de dispositivos IoT entre sensores y actuadores con los que interactuar, a través de la integración de diversas herramientas de visualización y control, incluyendo la posibilidad tanto de presentar indicadores de datos en tiempo real, así como también interactuar con data histórica para su análisis.

Todas las tecnologías aquí presentadas son software y hardware libre por lo que pueden ser adaptadas rápidamente según los requerimientos funcionales del problema y cumplen con legislativa local para poder ser exploradas a fondo.

7.1. Diseño

El diseño propuesto para examinar la solución al problema planteado pasa por dos etapas:

- Prototipos de uno o más dispositivos IoT que sean capaces de generar información real de su entorno, explorando los métodos más adecuados para poder llevar a cabo la tarea de transmitir información, así como explorar las posibilidades de procesamiento de cada dispositivo.
- Aplicación web que incluya la posibilidad de visualizar los datos, monitorear y controlar los dispositivos IoT que se encuentren registrados. Para ello se integraran una serie de herramientas de software que aprovechen las mejores características de su diseño para poder cumplir con los objetivos planteados.

Dado que los dispositivos Raspberry Pi pueden usar un sistema operativo completo estos pueden operar de manera independiente a otros dispositivos, no siendo el caso del dispositivo Arduino Uno se complementa su integración haciendo uso de una computadora Apple iBook G4 para poder recibir y enviar la data de manera inalámbrica.

Siendo la idea examinar variables y comportamientos de dispositivos en un ambiente de un hogar inteligente junto con la variedad de placas programables, sensores y actuadores disponible se sugieren la creación de 4 prototipos de dispositivos IoT distintos:

- Un prototipo de dispositivo IoT usando Arduino Uno (junto al computador iBook G4) para representar un artefacto que se coloque en exteriores para obtener y representar variables ambientales y de recursos comunes en esas áreas. Para ello se destinan un sensor PIR HC-501, un sensor de test de nivel de agua Robodo Sen18, un sensor de temperatura DS1820, un led RGB para representar de manera visual la temperatura y una fotorresistencia LDR de 10K (véase figura: 7.1).

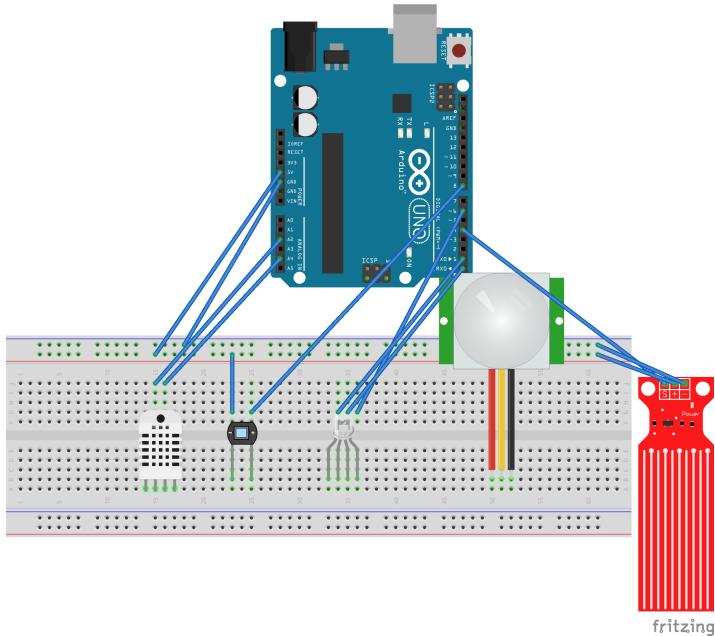


Figura 7.1: Prototipo de dispositivo exterior usando Arduino Uno

- Un prototipo de dispositivo IoT usando un Raspberry Pi 3 Modelo B para representar un artefacto que se coloca en un espacio interior para obtener y representar variables ambientales y recursos comunes en el área. Para ello se harán uso de un sensor de temperatura y humedad DHT11, un Led RGB para representar de manera visual la temperatura, un sensor de medición de distancia usando ultrasonido HC-SR504, un led rojo y un led verde operacionales para indicar presencia de movimiento con el sensor de movimiento, un un buzzer y un servomotor que dada una cierta distancia con el sensor de movimiento se accionaran y a su vez tocará una melodía, un conjunto de 3 leds blancos para representar bombillos en una habitación y por último un sensor de intensidad lumínica TSL2561 para medir la intensidad de la luz en la habitación.(véase figura: 7.2).
- Un prototipo de dispositivo IoT usando un Raspberry Pi zero para representar un artefacto pensado para en variables de seguridad y acceso que se coloca en un espacio interior. Para ello se harán uso de un lector RFID RC522, una pantalla LCD para representar la hora y darle la bienvenida al usuario, un led rojo y un led verde operacionales para indicar existencia de movimiento gracias a un sensor PIR HC-SR501, 3 leds blancos para representar bombillos (que se activen con movimiento).(véase figura:7.3).
- Un prototipo de dispositivo IoT usando un Rapsberry Pi 3 Modelo B para representar una cámara inteligente capaz de detectar rostros de personas registradas y

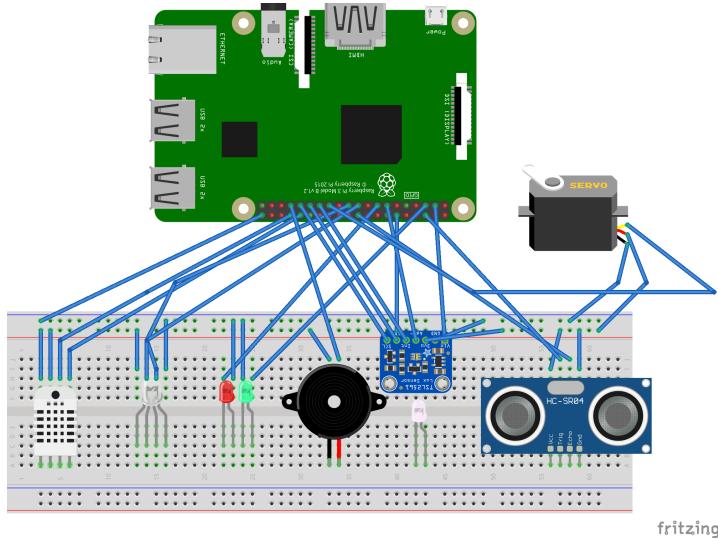


Figura 7.2: Prototipo de dispositivo interior usando un Raspberry Pi 3 modelo B

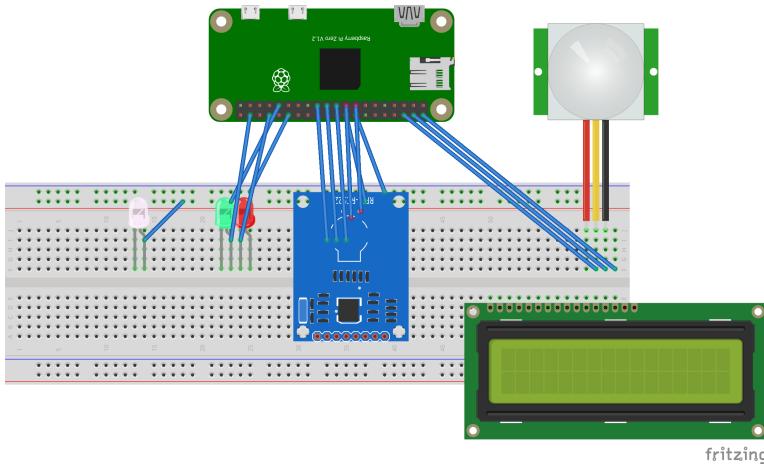


Figura 7.3: Prototipo de dispositivo de seguridad usando un Raspberry Zero

de desconocidos para alertar una posible intrusión usando un modelo de inteligencia artificial para ello. Se destina la cámara RaspiCam v1 para esta tarea.(véase figura:7.4). También dadas las capacidades computacionales de esta placa programable se establece que pueda usarse como punto de acceso para los otros prototipos y lugar de despliegue de la herramienta web para visualización, monitoreo y control de los prototipos.

Con esta gama de distintos prototipos planteados para escenarios diferentes se hará obtención de data y generar acciones automáticas dependiendo del contexto que estos datos provean.

7.1.1. Software de Visualización, Monitoreo y Control

La parte mas importante del planteamiento del problema se traduce en la falta de soluciones que permitan visualizar datos históricos y en tiempo real, junto con la capacidad de monitorear el estado de los dispositivos y la posibilidad final de controlarlos y

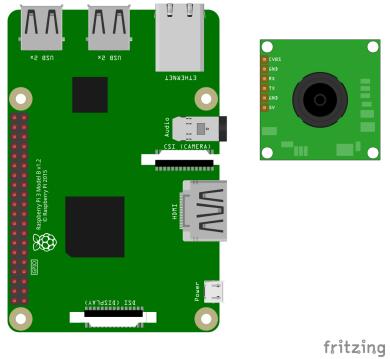


Figura 7.4: Prototipo de dispositivo de cámara de inteligente usando un Raspberry Pi 3 modelo B

establecer rutinas mas allá de su programación inicial.

Para ello y examinando las diversas opciones de tecnologías estudiadas e investigadas durante el marco teórico del seminario se propone la creación de un software que funcione de la siguiente manera:

- Una infraestructura básica que permita conectar a los dispositivos de manera centralizada, capaz de funcionar usando protocolos y estándares existentes, a la vez que este diseñado para su utilización en dispositivos IoT. Basados en la investigación previa por sus ventajas, robustez y simplicidad de uso se opta por utilizar el protocolo MQTT [14] bajo la implementación Mosquitto [42].
- Dada la naturaleza no estructurada de los datos enviados por los dispositivos IoT, se requiere una base de datos que pueda almacenar fácilmente dichos valores. Se tiene que tener en cuenta que esta base de datos debería estar orientada a la utilización de datos históricos y en tiempo real (dimensión de tiempo). Es por ello que basados en la investigación previa realizada y teniendo en cuenta la naturaleza de los datos y las tareas que se esperan del sistema manejador de base de datos, se toma la decisión de usar InfluxDB [43], que es una base de datos orientada a series de tiempo con alta integración con otras herramientas disponibles en el mercado.
- Una herramienta web que sea capaz de desplegar la información de sensores, actuadores tanto de manera histórica como en tiempo real, además de gestionar y centralizar el control de dispositivos y rutinas extras, con la posibilidad de tener acceso a ella basado en registro de usuarios y que estos sean capaces de crear cuadros de mando adaptados a las necesidades de sus dispositivos y requerimientos. Esta parte de la propuesta busca no crear todo una suite desde 0 sino aprovechar herramientas ya existentes para ello. Teniendo en cuenta eso, se decide en primer lugar crear una webapp usando el lenguaje de programación Python [39] en su versión 3 con el framework Django [44] bajo una base de datos Postgresql [45] para almacenamientos de los datos referentes a esta aplicación. En cuanto al tema de visualización y monitoreo se establece el uso integrado de la aplicación web Grafana [46] y del mismo modo, para el control de dispositivos y de flujos de trabajos se acepta finalmente el uso de la herramienta web Node-Red [47]

Con estos componentes definidos y después de discutir sobre la mejor manera de desplegar este conjunto de tecnologías, buscando facilidad de operación y manutención así como la portabilidad y la independencia de plataformas se decide utilizar la mayor cantidad de componentes posibles como contenedores. Para ello el uso de docker [48], comenzando por las integraciones y otros componentes esenciales (figura 7.5). Finalmente con estos componentes se establecen las siguientes directrices:



Figura 7.5: Componentes sugeridos para la herramienta de visualización, monitoreo y control HAMACA

- Se usara Mosquitto como Broker MQTT para las comunicaciones entre los dispositivos y aquel elemento que lleve la información a la base de datos. Este estará centralizado en el mismo dispositivo donde se alojará la aplicación web, aunque se recalca el hecho que pueden existir multiples brokers bajo la arquitectura de MQTT, así como también que el broker puede ser configurado de manera externa.
- La aplicación web HAMACA seguirá el paradigma cliente-servidor, con un front-end marcado y distinto a su back-end. La aplicación web solo dependerá de la base de datos PostgreSQL para almacenar variables de configuración, estadísticas de uso y aquellas que den soporte a la autenticación y gestión de usuarios.
- La Base de datos InfluxDB solo ha de ser consultada por la integración a Grafana para la visualización y monitoreo óptimo de los datos haciendo uso de los dashboards configurables dentro de la aplicación. Grafana será embebido en la aplicación web HAMACA en su frontend. Al igual forma que la implementación de MQTT este puede existir en otro punto de red pero se sugiere el despliegue en el mismo dispositivo.
- La herramienta de control y de gestión de flujo de tareas automatizado Node-Red, al igual que Grafana será embebida sobre el frontend de la aplicación web HAMACA. Este componente podrá interactuar también con el broker MQTT para poder transmitir (y en algunos casos, también capturar directamente) los sensores y actuadores de los prototipos de dispositivos IoT.

De esta forma se presenta el diseño de la arquitectura de la solución para el problema de investigación como se puede notar en la figura 7.6

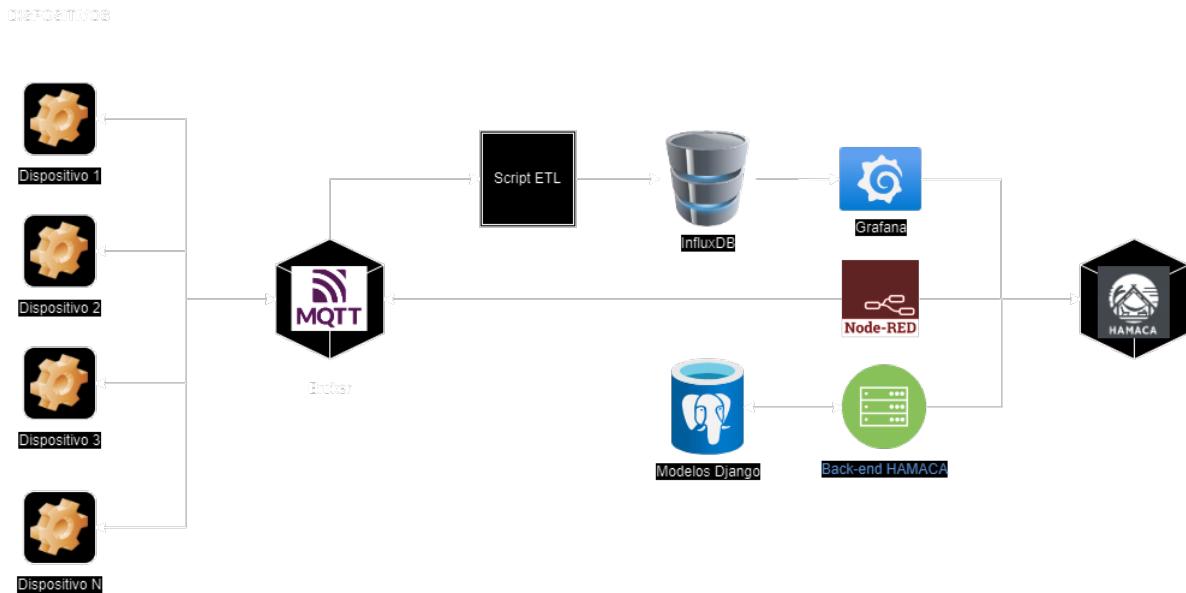


Figura 7.6: Arquitectura de la solución propuesta

7.2. Implementación de la solución

Siguiendo lo convenido en la sección anterior se dispuso la misma estructura organizativa de llevar por separado la implementación de los prototipos de dispositivos IoT por un lado y la aplicación web HAMACA por otro.

7.2.1. Implementación de los prototipos de dispositivos IoT

El proceso partió por obtener y organizar los elementos requeridos para la creación de los prototipos funcionales, es decir, las placas programables, los sensores, los actuadores, elementos de construcción de electrónicos (cables, resistencias, estaño, dispositivos de almacenamiento, protoboard, etc) y las herramientas físicas (pela cables, piqueta, soldador, tester, etc) de manera de tener en recaudo todo lo concerniente con el desarrollo físico de los dispositivos (véase figura ??).



Figura 7.7: Elementos para prototipado de dispositivos

Una vez organizados, se construyó uno a uno cada prototipo de dispositivo, comenzando por el prototipo de dispositivo para exteriores. Tomado el Arduino Uno y los sensores conectados a él, junto con la computadora iBook G4 como interfaz de despliegue del código a la placa programable y para captura y envío de los datos se procedió a desarrollar dos scripts:

- Un primer script en el lenguaje de programación Arduino con los drivers y la lógica de control de la placa programable para los sensores y actuadores que fueron tomados según el diseño convenido anteriormente (figuras 7.8 y 7.9).

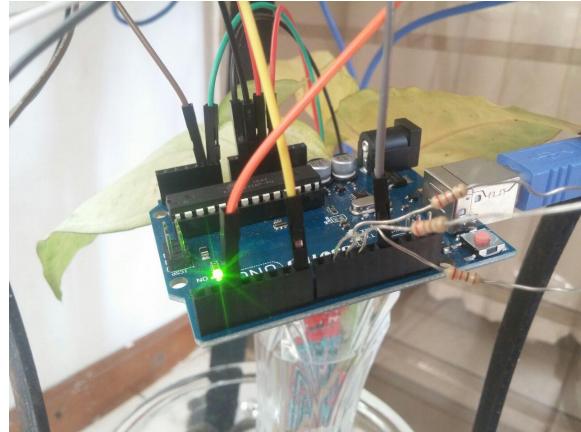


Figura 7.8: Arduino Uno conectado a sensores y al iBook G4

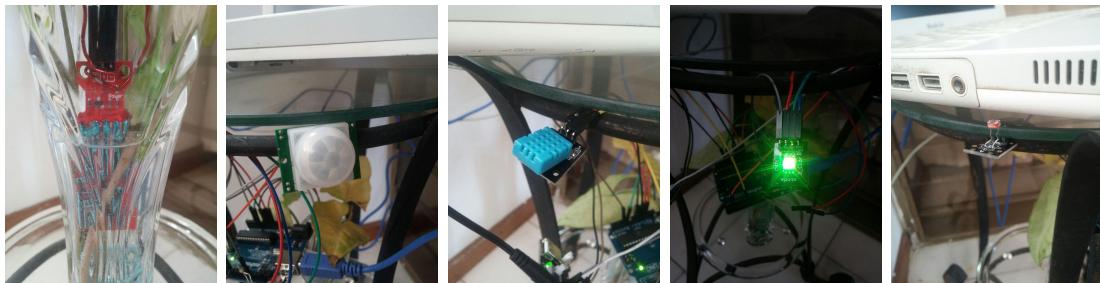


Figura 7.9: Sensores y actuadores en el exterior

- Un segundo script en el lenguaje de programación Python con la capacidad de leer los datos seriales provenientes del Arduino Uno y con la capacidad de enviarlos a través del protocolo MQTT al broker.

Para tener estandarizada la comunicación de los dispositivos, se decidió que el mensaje a enviar debe tener el siguiente formato:

```
{
  ''value'': <'Valor de la medición'>,
  ''sensor'': <'Nombre del sensor'>,
  ''host_device'': <'Nombre del dispositivo'>,
  ''measurement_type'': <'Nombre del tipo de lectura realizada'>,
  ''location'': <'Ubicación'>,
```

```

    ''date'':< 'Fecha en formato YYYY-MM-DDTHH:MM:SS :FZ' >
}

```

Finalmente el dispositivo y el computador fueron colocados geográficamente en el balcón del apartamento para simular con la mayor precisión posible un ambiente exterior propiamente dicho.

El siguiente paso de manera similar consistió en construir el prototipo funcional del dispositivo IoT para ser interior del hogar. El dispositivo elegido para esta tarea fue uno de los dos Raspberry Pi 3 modelo B disponibles, así como la lista de sensores asociados para completar la tarea. De esta forma se ejecutaron las siguientes labores:

- Colocación de todos los sensores en un protoboard y de allí fueron conectados al Raspberry Pi.
- Instalación del sistema operativo Raspbian al dispositivo y se dejó preparado para utilizar una configuración headless, es decir, sin uso de monitor, solo acceso remoto.
- Scripts en el lenguaje de programación Python por cada sensor y actuador utilizado.
- Construcción de una carcasa alrededor del dispositivo y el protoboard con la ayuda de bloques de lego como se puede observar en la figura 7.10. Su disposición final geográficamente fue situado en el centro del apartamento.

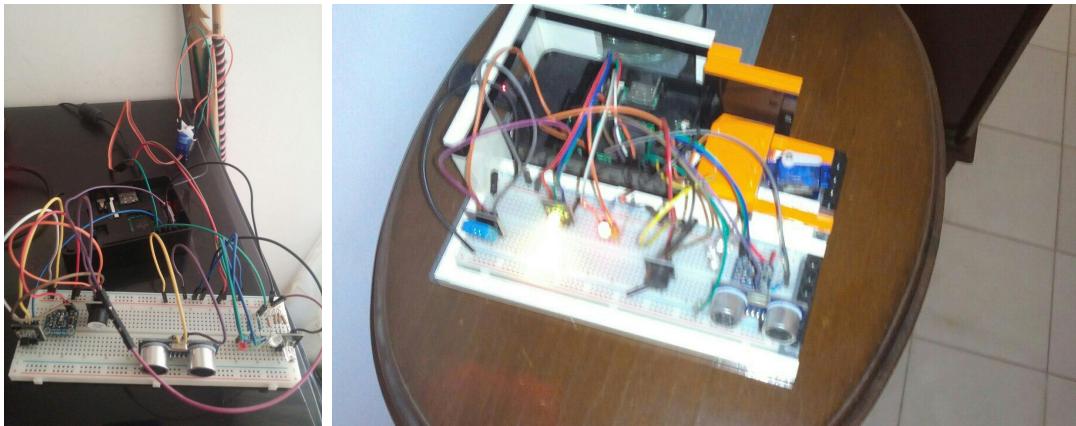


Figura 7.10: Sensor IoT en el interior del hogar antes y después de su validación

Luego, se continuó con los dos prototipos que representan elementos de seguridad. Para ellos se utilizaron el Raspberry Pi Zero y el Raspberry Pi 3 modelo B restante. Para poder dejar activos los elementos correspondientes a ellos se siguieron los siguientes pasos:

- Colocación de todos los sensores y actuadores en un protoboard y al Raspberry Pi Zero.
- Instalación del sistema operativo Raspbian al dispositivo y se dejó preparado para utilizar una configuración headless, es decir, sin uso de monitor, solo acceso remoto.

- Conexión directa del Raspberry Pi Zero al Raspberry Pi 3 modelo B para proveer de conectividad a la red al dispositivo.
- Desarrollo de scripts en el lenguaje de programación Python por cada sensor y actuador utilizado en Raspberry Pi zero.
- Desarrollo de un modelo de inteligencia artificial en el lenguaje de programación Python para usar la cámara acoplada al Raspberry Pi 3 modelo B de forma que reconociera las caras de los habitantes que entrasen al hogar.
- Se instalaron los elementos necesarios para dar futuro soporte al despliegue de la aplicación web HAMACA en el Raspberry Pi 3 modelo B, así como también se creó un punto de acceso para los dispositivos se pudieran conectar a él de manera independiente de la red del hogar.

Al finalizar el desarrollo del prototipo de dispositivo este fue situado validado y colocado en la entrada del hogar (figura 7.11), cuestión que fuera centro neuralgico de los sensores de seguridad. Aparte la capacidad de cómputo del Raspberry Pi 3 modelo B también estaría a disposición siendo el elemento de cómputo definido para la siguiente etapa de la implementación.



Figura 7.11: Sensor IoT en el interior del hogar antes y después de su validación

7.2.2. Implementación de la Aplicación Web HAMACA

Por el lado de la aplicación web HAMACA, el proceso de implementación fue llevado a cabo siguiendo los lineamientos de las metodologías ágiles, con entregables incrementales, rápidas iteraciones con nuevas características y corrección de errores. Se estimó inicialmente un período de desarrollo de aproximadamente un mes y medio para tener todos los aspectos necesarios para cubrir los objetivos planteados y dar una propuesta de solución al problema de investigación.

Los pasos que se siguieron para poder desarrollar y desplegar la aplicación web fueron los siguientes:

7.2.2.1. Infraestructura

Por el lado de la infraestructura, al tener como elemento de cómputo para despliegue un Raspberry Pi 3 Modelo B, el sistema operativo base utilizado fue raspbian, un flavor

de Debian, una opción popular y fácil de utilizar para los desarrollos de software.

Directamente sobre el sistema operativo se instalaron todos y cada uno de los paquetes, base de datos (InfluxDB y PostgreSQL), herramientas a integrar (Grafana y Node-Red), la implementación de MQTT, así como también el ambiente de Python 3 para manejar la webapp. Este proceso cambió para luego dejar la mayor cantidad de elementos posibles para ser utilizados bajo el esquema de contenedores docker, siendo migrados a esa forma la base de datos InfluxDB, y las herramientas Grafana y Node-Red. Lo cual permite un despliegue mucho más rápido en cualquier ambiente nuevo, así como modularizar el proyecto de una mejor forma.

7.2.2.2. Configuración de Entorno

Una vez instalados los diversos paquetes y herramientas se procedió con la configuración de todos los elementos previos a la parte de desarrollo.

- La configuración asociada a la comunicación entre los dispositivos basados en el broker MQTT instalado.
- La creación de una base de datos en InfluxDB, junto con un usuario y métodos de acceso de forma de almacenar a través del backend de la aplicación toda la data relativa a los dispositivos.
- Despliegue de la herramienta de visualización y monitoreo Grafana. Configuraciones asociadas para la conexión a la fuente de datos (base de datos InfluxDB) así como también el acceso y privilegios de gestión de cuadros de mando sin la necesidad de un usuario logeado y con privilegios asociados.
- Despliegue de la herramienta Node-Red para el control de dispositivos. Se le armó configuraciones para poder crear interfaces basadas en cuadros de mando para poder realizar tareas de control sobre sensores y actuadores de los dispositivos. También se probaron y validaron la posibilidad de generar flujos de tareas para los dispositivos a través de esta herramienta.

7.2.2.3. Desarrollo de la Aplicación Web HAMACA

Finalmente con el entorno configurado se procedió con el desarrollo de la aplicación web HAMACA. Fue diseñada como una aplicación web liviana y sencilla haciendo uso del framework de desarrollo Django para el lenguaje de programación Python en su versión 3.

Al comenzar el proyecto se creó la aplicación a partir de la plantilla de instrucciones que provee el framework, junto con la definición de utilizar PostgreSQL como base de datos, puesto que es una respuesta más robusta que SQLite (base de datos por defecto) tanto para el soporte mismo de la aplicación como de sus futuros procesos. Es así como luego de configurada la base y siguiendo las instrucciones se generó en la base de datos de HAMACA los modelos que podemos observar en la figura 7.12.

En el argot de Django, este primer paso se le llama migración. Con ello se comienza con el desarrollo del primer módulo de control y acceso a la aplicación basado en los

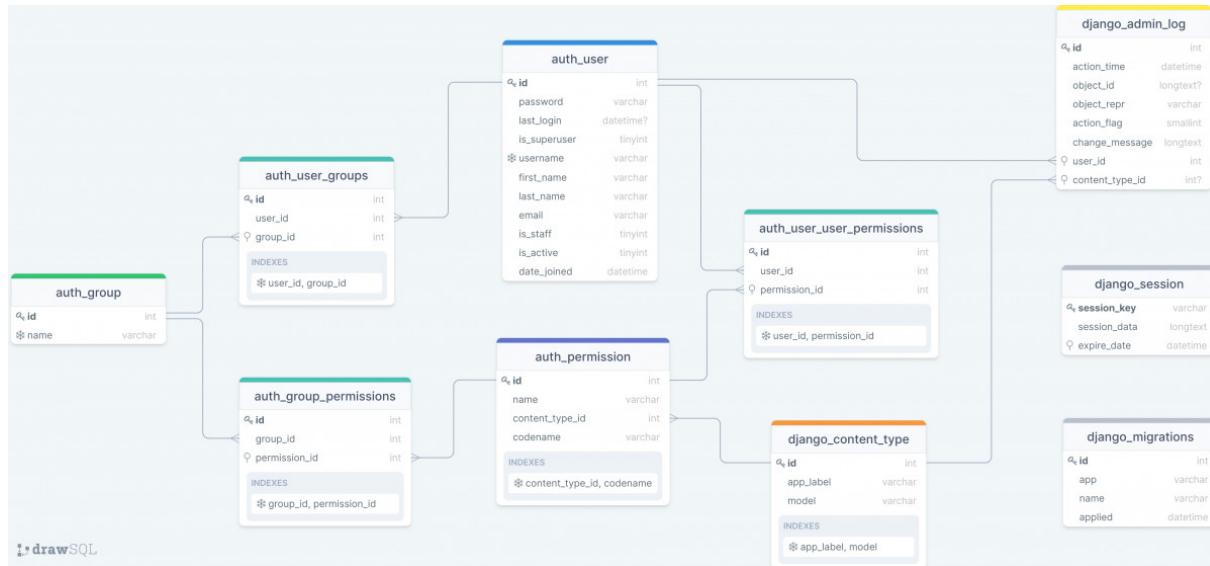


Figura 7.12: Modelo autogenerado por el framework Django

aspectos de roles y privilegios del que se dispone en el framework.

Es en este punto donde también se estructuran las bases para un desarrollo modular en el que se fueron integrando progresiva e incrementalmente nuevos elementos entre los cuales se encuentran las librerías comunes, plantillas para los aspectos visuales, los elementos estáticos como las fuentes para los textos, gráficos y colores.

Se comenzó con un primer módulo desarrollado para el acceso a la aplicación junto con las de configuraciones (figura 7.13), el cual contiene las herramientas de gestión de la aplicación, como la administración de usuarios, así como la capacidad de cambiar la ubicación de las herramientas integradas, sea que se encuentren de manera local o en algún lugar que pueda ser accedido por este desarrollo.

Este es un screenshot de la interfaz web de la aplicación HAMACA. El sidebar izquierdo tiene los siguientes íconos y opciones:

- Hogar
- Monitoreo
- Control
- Redes
- Configuración
- Cerrar Sesión

El contenido principal se titula "Configuración de la Aplicación".

Configuración de servicios

- Ubicación del servicio de Grafana: http://localhost:3000
- Ubicación del servicio de Node-RED: http://localhost:1880

Configuración de Usuarios

Botón "Crear nuevo usuario".

Lista de usuarios

Usuario	Nombre	Apellido	Correo	Opciones
pedro			pedro.boll22@gmail.com	

Página 1 de 1

Figura 7.13: Interfaz de configuración de la aplicación
69

A partir del punto anterior se creó un modulo de Monitoreo el cual contiene la lógica para integrar de manera adecuada la herramienta Grafana para visualización y monitoreo de datos, de forma que los usuarios puedan crear sus cuadros de mando de una manera sencilla y directa.

De manera análoga se creó un modulo de control y un home que contienen la lógica de la integración de la herramienta Node-Red para llevar a cabo las labores de controlar y gestionar flujos de trabajo complejos con los dispositivos.

El home de la aplicación particularmente se convierte en un punto donde se muestran los cuadros de mando creados para manipular (e incluso observar) sensores y actuadores de aquellos dispositivos que registremos en los flujos para mostrar. Por último pero no menos importante se creó un apartado de redes en el cual se pueden observar los registros de las comunicaciones realizadas entre los dispositivos y el broker, así como otras métricas asociadas a ello (figura 7.14).

The screenshot shows the HAMACA application interface. The left sidebar has a dark theme with icons for Hogar, Monitoreo, Control, Redes, and Configuración, and a 'Cerrar Sesión' button. The main title is 'Configuración de Redes'. Under 'Estadísticas de red', it shows 'Estadísticas de mensajes MQTT' with metrics: Mensajes Recibidos: 125538, Mensajes/min: 58; Cantidad de topicos: 4, Topicos activos: 3. Below this is a 'Detalles' section with a table:

Topico	Número de Mensajes	Activo	Mensajes/min	Hora	Ultimo Mensaje
Indoor/Temperatura	62738	Activo	28	6 de Octubre de 2019 a las 19:55	DHT11: Temp = 23.0 C - Hora: 2019-10-06 19:55:42.177400
Indoor/Humedad	62791	Activo	28	6 de Octubre de 2019 a las 19:55	DHT11: HR = 40.0% - Hora: 2019-10-06 19:55:42.177542
Outdoor/Temperatura	6	Inactivo	1	4 de Octubre de 2019 a las 14:46	DHT11: Temp = 21.0 C - Hora: 04-10-2019 14:56:45
python/mqtt	3	Activo	1	10 de Octubre de 2023 a las 17:48	mensajes: 1

Página 1 de 1

Figura 7.14: Interfaz de la aplicación para la configuración y visualización de métricas de red

Luego del desarrollo de cada modulo y aunque no todos los módulos interactuan con la base de datos se realizó nuevamente el proceso de migración, de forma que la base de datos se adaptara a los requerimientos nuevos de los módulos existentes. Así la base de datos se le agregaron dos nuevas tablas como se pueden observar en la figura 7.15

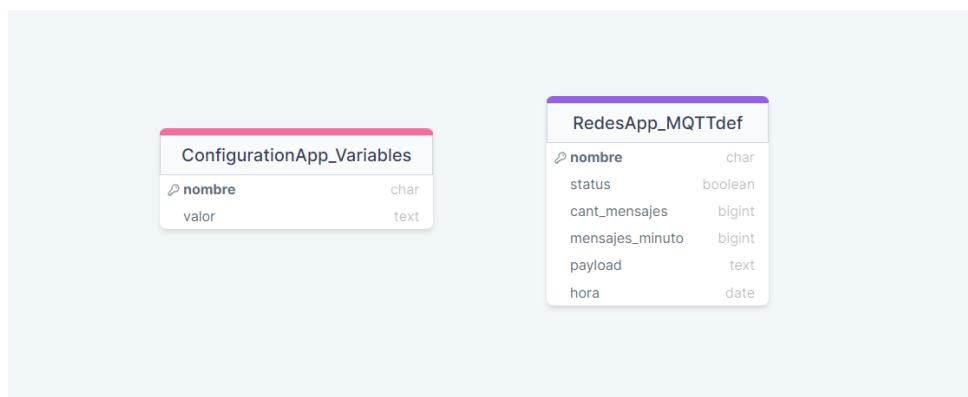


Figura 7.15: Modelo de la base de datos HAMACA

Capítulo 8

Entorno de Pruebas

8.1. Escenarios de pruebas

En aras de validar que la propuesta de solución tiene respaldos suficientes para cumplir los objetivos planteados, se realizaron un conjunto sistemático de pruebas en las que se pudo comprobar la eficacia de la solución, así como los aspectos funcionales.

Desde el diseño, se pensó que la mejor manera hacer esto es llevando tanto prototipos de dispositivos como la aplicación al entorno lo más realista posible, dejando por fuera perspectivas como la de simulación de datos, poca variedad placas programables, en sensores y actuadores o en poco volumen de información.

Durante los 8 meses que estuvieron operativos los dispositivos se logró capturar cerca de 2.6 GB de información no estructurada, 3.1 GB de imágenes y 1.7 GB de videos. Con ello se buscaba observar el comportamiento de generación de datos por su volumen es una arista a tomar en cuenta en cualquier desarrollo basado en tecnologías del Internet de las cosas.

Por el lado de la aplicación web se buscó medir la eficiencia y la eficacia al momento de poder realizar tareas de visualización de datos para facilitar su análisis, monitoreo y control de dispositivos así como también el desempeño de los componentes integrados y en qué medida fueron o no capaces de complementarse.

Con este panorama de pruebas a la mayor escala posible se prestó atención a los elementos y así se evaluaron el cumplimiento o no de los objetivos planteados.

8.1.1. Prototipos de Dispositivos IoT

Correspondiente al prototipado de dispositivos de IoT, el entorno de pruebas se basó en la concepción inicial de tener la mayor cantidad de escenarios posibles medidos. Estos son las variables que se consideraron desde el diseño:

8.1.1.1. Variedad en los Dispositivos

Los dispositivos IoT en el mundo real no son homogéneos, por lo que aunque tengan las mismas funciones, estos pueden lograrlo de diversas maneras. Factores como la marca, los estándares, los servicios ofrecidos, la calidad y diversidad de los datos, versiones de un mismo dispositivo, entre otros hacen que la propuesta de tener múltiples instancias de dispositivos similares apoya el sentido de considerarlo al momento de desarrollo del prototipo, así como también en etapas posteriores.

En el marco de esta investigación probar esta hipótesis se logró de haciendo uso de las variedad de placas programables, sensores y actuadores de forma que en dos escenarios similares pudiesen cumplir un conjunto de mismas funciones, en entornos distintos, como por ejemplo, capacidad de medir temperatura y luminosidad de las habitaciones donde se encuentran los prototipos, uno construido con la placa programable Arduino uno y otro haciendo uso de la placa Raspberry Pi 3 modelo B.

Al final se pudo constatar que dependiendo del escenario la utilización de un placa con respecto a la otra puede brindar ventajas operacionales sobre los dispositivos:

- El uso del Arduino se recomienda para labores que impliquen una limitada capacidad de cómputo y cuya intención de iterar incrementalmente su software sea relativamente baja pues requiere en todo caso de un ente externo para cambiar su programación inicial. Sin embargo es una plataforma robusta y al disponer de la capacidad de lectura tanto de sensores digitales como analógicos lo hacen ideal para poder experimentar con un amplio rango de sensores y actuadores de manera sencilla.
- Las placas Raspberry Pi Zero, al igual que las placas Raspberry Pi 3 Modelo B cuentan una capacidad de cómputo mayor y su arquitectura les permite utilizar sistemas operativos completos, lo que hace que sean herramientas potentes para pruebas de concepto y prototipos funcionales de dispositivos IoT.
- La placa Raspberry Pi Zero cuenta con un tamaño y capacidad menor que la Raspberry Pi modelo 3 B, pero es más versátil en consumo energético. Al igual que las placas Arduino, inicialmente requiere de otros periféricos para poder ser configurado de manera adecuada pero una vez hecho eso, la posibilidad y la flexibilidad de usar scripting para prototipar software y la portabilidad que ofrecen les da un extra al momento de ser usados como prueba de concepto.
- Las placas Raspberry Pi 3 Modelo B son las placas más grandes del grupo pero al ser micro computadores en todo el sentido de la palabra poseen numerosas formas de poder realizar conexión con sensores, actuadores y también de periféricos. Además posee la capacidad de conectarse a redes cableadas e inalámbricas de manera nativa e incluye la misma posibilidad de conexiones por pines que la placa Raspberry Pi Zero. También su capacidad de cómputo las hace ideales para crear prototipos que requieran capacidad de preprocesar data o de incluir tareas demandantes como modelos de inteligencia artificial (como la que se implementó para detectar rostros haciendo uso de la cámara RaspiCam v1 y se pueden observar

en la figura 8.1) o la integración de múltiples herramientas. La principal desventaja que presenta es que usan más recursos, como el hecho de su precio final al usuario.



Figura 8.1: Identificación de rostros usando un modelo de IA desplegado en un Raspberry Pi 3 modelo B

8.1.1.2. Variedad en los sensores

Dada la redundancia de algunos sensores que median el mismo tipo de variables ambientales y teniendo en cuenta el item anterior se quiso observar las diferencias en la representación, desempeño y facilidad de utilizar y mantener, se estableció un pequeño estudio comparativo, seleccionando las variables de temperatura y el registro de luminosidad de la habitación para entender como las diferencias entre sensores puede afectar la data generada.

Particularmente esto se llevó a cabo del siguiente modo: Haciendo uso de la placa programable Arduino Uno se decidió construir el prototipo de dispositivo IoT con el sensor de Temperatura DS1820 y la utilización de una fotorresistencia (LDR) para medir la intensidad de la luz. Por otro lado con una de las placas Raspberry Pi 3 Modelo B le fueron conectados un sensor de Temperatura (y humedad) DHT11, así como también un sensor de medición de intensidad TSL2561. Los resultados de la comparación fueron:

- En cuanto a los sensores de temperatura se vieron que ambo sensores de temperatura tanto el DHT11 como el DS1820 (figura 8.2) fueron capaz de hacer lecturas del ambiente bajo el margen de error de los fabricantes. Sin embargo es importante recalcar que el sensor DS1820 fue mucho más preciso en la medición que el sensor DHT11. Esto se puede explicar por las especificaciones de cada tipo de sensor. Otra diferencia es en la manera y velocidad que se reflejaban los datos de los mismos, siendo más rápido el sensor DS1820 al momento de hacer lecturas que el sensor DHT11, siendo su dato un número de punto flotante versus un tipo de dato complejo que se genera en el sensor DHT11, siendo esto explicado que este último también lee en el mismo margen de tiempo la humedad ambiental. Ambos sensores representaban la temperatura en grados centígrados.
- Con respecto a la variable de intensidad lumínica en la habitación se compararon una fotorresistencia (LDR) simple contra un sensor TSL2561. A pesar de que ambos buscan el mismo objetivo, la fotorresistencia arroja valores que deben ser interpretados, pues inicialmente este representa la resistencia de una corriente eléctrica y teniendo ese valor se puede calcular la cantidad de luz disponible en

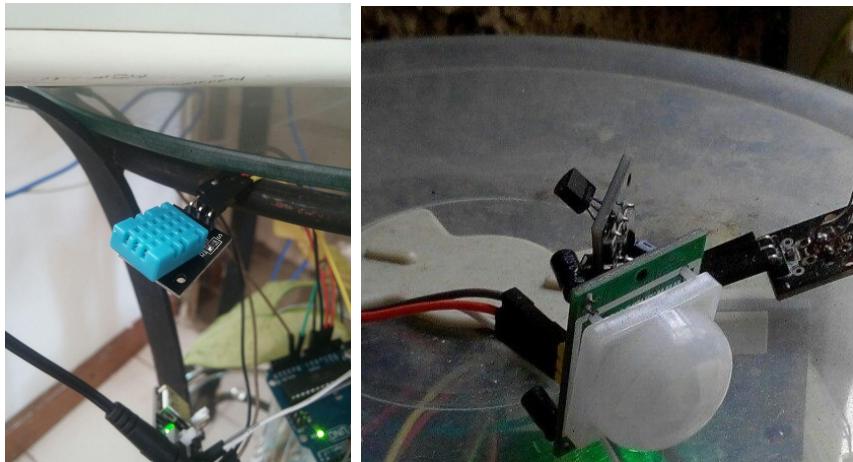


Figura 8.2: A la izquierda el sensor de temperatura y humedad DHT11 y a la derecha el sensor de temperatura DS1820

el ambiente, mientras que el sensor TSL2561 directamente registra un valor en lux, que es una unidad directa de nivel de iluminación. Ambos valores del tipo numérico con la diferencia que el sensor TSL2561 usa valores de punto flotante y la fotorresistencia registra valores entre 0 y 1023 en el dispositivo Arduino.

Otro punto importante dentro de la variedad de los sensores y actuadores es el formato de los distintos datos generados. Se pudo constatar que dependiendo de del tipo de sensor los formatos y los valores de los datos es bastante variado, lo suficiente como para concluir que es una potencial debilidad y que se deben utilizar métodos adecuados para poder tratar con ellos. Esto se vio reflejados en elecciones que se podrán observar en la siguiente parte de los escenarios de prueba.

8.1.2. Aplicación HAMACA

En el caso de la aplicación web HAMACA las pruebas consistieron en poder obtener gestionar los datos e información generados de dispositivos, y dar opciones de control y monitoreo de los mismos así como también en ofrecer a los usuarios la flexibilidad y facilidad de uso sobre las herramientas integradas, todo esto haciendo uso de una navegación sencilla sobre las interfaces.

8.1.2.1. Captura de Información

En términos de la captura de información y como parte del modulo de Redes desarrollado dentro de la aplicación se creo un script que es capaz de poder hacer las lecturas de tópicos para aquellos mensajes MQTT que llegacen al broker. Dichos mensajes generan dos elementos claves para la operación de esta solución:

- Guardar en la base de datos InfluxDB el dato que está siendo recibido para fines de visualización en tiempo real y almacenamiento histórico para su posterior análisis (figura 8.3).
- Guardar estadísticas de uso, tópicos y contenido de mensajes recibidos por el broker como parte de la aplicación en la base de datos Postgresql (figura 8.4).

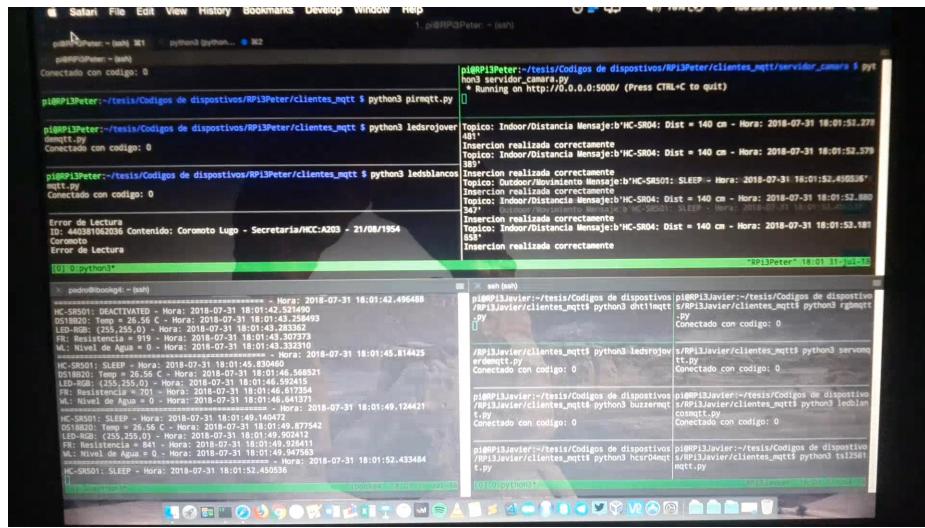


Figura 8.3: Captura de datos por parte del backend de la aplicación web

La captura de pantalla muestra la interfaz de usuario de la aplicación 'HAMACA' para la configuración de redes. En la sección 'Configuración de Redes', se visualizan estadísticas de red y mensajes MQTT. Las estadísticas de red muestran que se han recibido 125538 mensajes y hay 3 temas activos. La sección 'Mensajes MQTT' muestra una tabla con los últimos mensajes recibidos:

Topico	Número de Mensajes	Activo	Mensajes/min.	Hora	Último Mensaje
Indoor/Temperatura	62738	Activo	28	6 de Octubre de 2019 a las 19:55	DHT11; Temp = 23.0 C - Hora: 2019-10-06 19:55:42.177400
Indoor/Humedad	62791	Activo	28	6 de Octubre de 2019 a las 19:55	DHT11; HR = 40.0% - Hora: 2019-10-06 19:55:42.177542
Outdoor/Temperatura	6	Inactivo	1	4 de Octubre de 2019 a las 14:46	DHT11; Temp = 21.0 C - Hora: 04-10-2019 14:56:45
python/mqtt	3	Activo	1	10 de Octubre de 2023 a las 17:48	mensajes: 1

Figura 8.4: Estadísticas de mensajería del broker MQTT

Se pudo observar como en ambas instancias este script cuya función es hacer un proceso de extracción, transformación y carga (ETL), además de hacer la inserción en ambas bases de datos utilizadas, proceso crítico como parte de la ingestión de datos que luego se procesa en las herramientas integradas como en la propia aplicación.

8.1.2.2. Visualización de datos

En la parte de visualización de datos, la evaluación de la aplicación web correspondía si a las funcionalidades de la herramienta de visualización de datos que se integró (Grafana) era lo suficientemente robusta y genérica para poder cumplir con los objetivos buscados.

La primera funcionalidad es la rápida y fácil integración con la fuente de datos, es decir, con el sistema manejador de base de datos InfluxDB. Este proceso es bastante sencillo y la herramienta también presenta una gran variedad distinta de otras fuentes de datos posibles. Una vez configurado ese aspecto, se puede ir directamente sobre la

opción de crear un cuadro de mando o dashboard en el cual se puede agregar todo tipo de elementos que van desde múltiples tipos de gráficos, textos, representaciones visuales e incluso, la posibilidad de embeber otros elementos de otras fuentes. Así es posible generar cuadros de mando altamente personalizados de manera muy sencilla.

Al momento de crear un gráfico, Grafana nos provee con una interfaz en donde realizar la consulta a la fuente de datos configurada (en este caso, la base de datos InfluxDB) haciendo uso de las consultas nativas de la fuente como se puede observar en la figura 8.5.

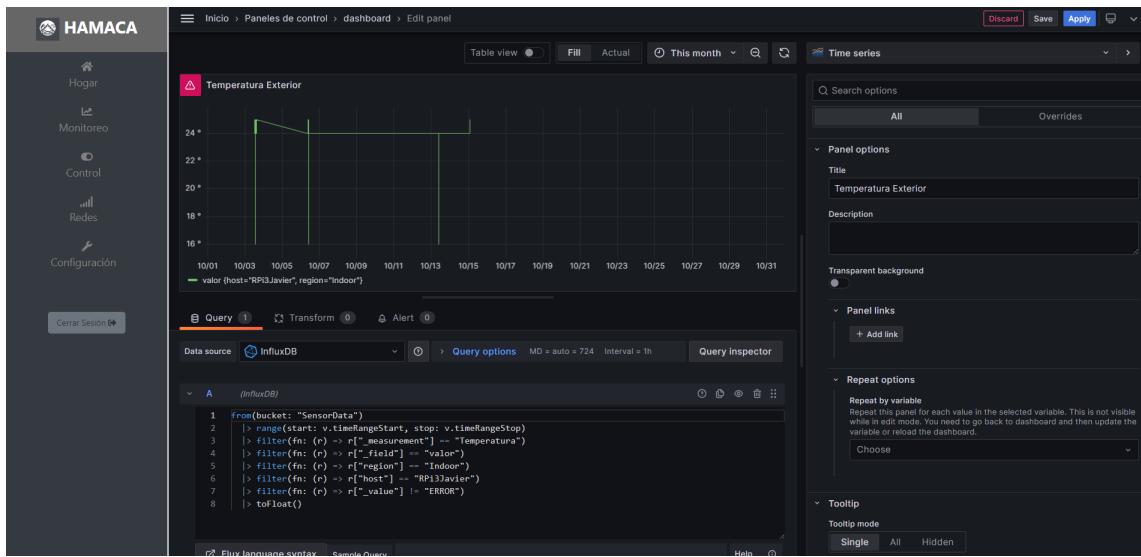


Figura 8.5: Consulta para construcción de gráficos usando la integración de Grafana

Este es el principal problema que posee la herramienta pues hace que el usuario conozca el como consultar los datos que quiera representar. Si no conoce la naturaleza de los mismos, la herramienta no tiene la posibilidad de ayudar a construir la información. Particularmente InfluxDB en su versión 2.0 cambio el lenguaje de consulta por uno llamado "flux" que lamentablemente es más complejo que su contraparte anterior lo cual puede elevar la curva de aprendizaje sobre la herramienta.

Sin embargo la potencia y flexibilidad ofrecida por Grafana la hace una herramienta idónea tanto para la vista de datos en tiempo real, como información histórica y junto con InfluxDB que al ser una base de datos basada en series de tiempo, definitivamente es una combinación que puede brindar información significativa y conocimiento accionable a través del análisis que se puede realizar sobre los diversos elementos que proveen los cuadros de mando como se puede ver en la figura 8.6.

8.1.2.3. Monitoreo de Dispositivos

Para el caso del monitoreo (y del control) la aplicación web ha hecho uso de la integración de la herramienta Node-Red, la cual fue diseñada para poder brindar flujos de automatización a dispositivos IoT más allá de su programación inicial. Con ellos se

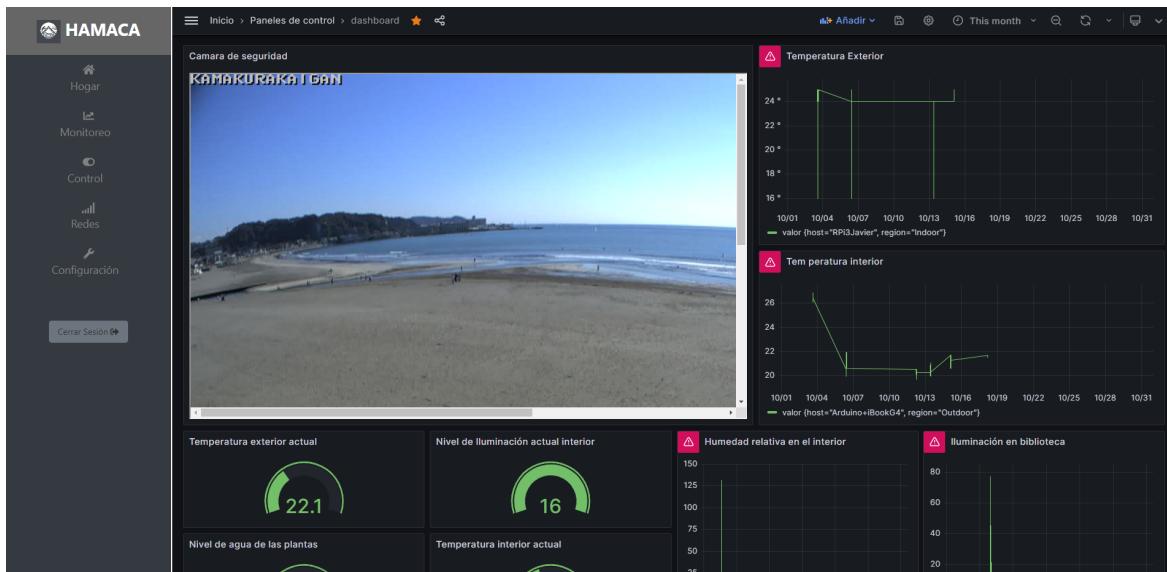


Figura 8.6: Cuadro de Mando con indicadores de la integración con Grafana

pueden crear “flujos” con los que poder extender comportamientos y automatizar una infinidad de aspectos sobre ellos.

Cuando se crean uno ó más “flujos”, estos pueden llevar a un cuadro de mando con interfaces que permiten generar controles sobre los dispositivos siguiendo la receta anteriormente creada. Estos se despliegan como parte del Home de la aplicación web donde se resume de manera intuitiva para los usuarios (figura 8.7), teniendo la posibilidad de asignar flujos a diferentes cuadros de mando que se pueden acceder desde esa opción.

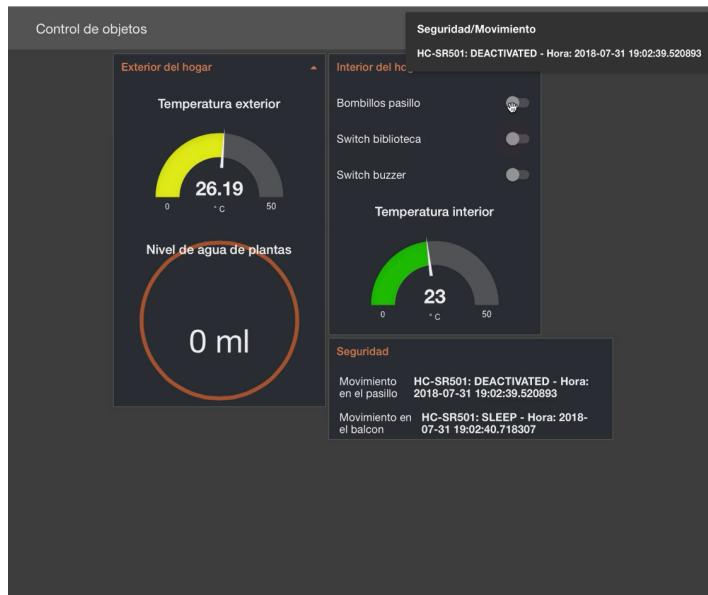


Figura 8.7: Cuadro de Mando con indicadores e interfaces de control de dispositivos

Se pudo comprobar que creando el flujo desde la interfaz del cuadro de mando hacia el broker MQTT (como se explicará en el siguiente apartado) que efectivamente se podían controlar los elementos accionables de los actuadores de los prototipos, logran-

do así que los usuarios puedan también bajo demanda utilizar las funcionalidades de dispositivos IoT.

8.1.2.4. Control de Dispositivos

Finalmente para el control de los dispositivos se estableció que aplicación (a través de la integración de Node-Red) fuese capaz de poder crear formas de control (manual o automatizada) de los prototipos que fueron creados.

Para tal objetivo se crearon una serie de flujos que no solo sirvieran para poder establecer la interfaz de monitoreo sino también para poder automatizar una serie de elementos representados por recursos de seguridad como lo fueron la presencia de movimiento en el exterior del hogar, el no reconocimiento de un rostro por parte del modelo de inteligencia artificial en la cámara del Raspberry Pi 3 modelo B, la notificación de una temperatura muy baja o muy alta para abrir o cerrar las ventanas o la notificación de la poca o nula presencia de agua por parte del sensor de nivel de agua en el exterior.

Como se puede notar en la figura 8.8 estos elementos son creados a partir de pasos visuales, que luego requieren cierta configuración interna para poder funcionar correctamente. En la mayoría de los casos no se requiere de ningún conocimiento previo, pero por otro lado, elementos puntuales de transformación de datos y el manejo de comunicación requiere tanto conocimientos de programación(en el lenguaje javascript) como de como funcionan los protocolos o la infraestructura en la cual se soportan dichos servicios, lo cual puede desanimar a la mayoría de los usuarios de realizar tareas de automatización más complejas.

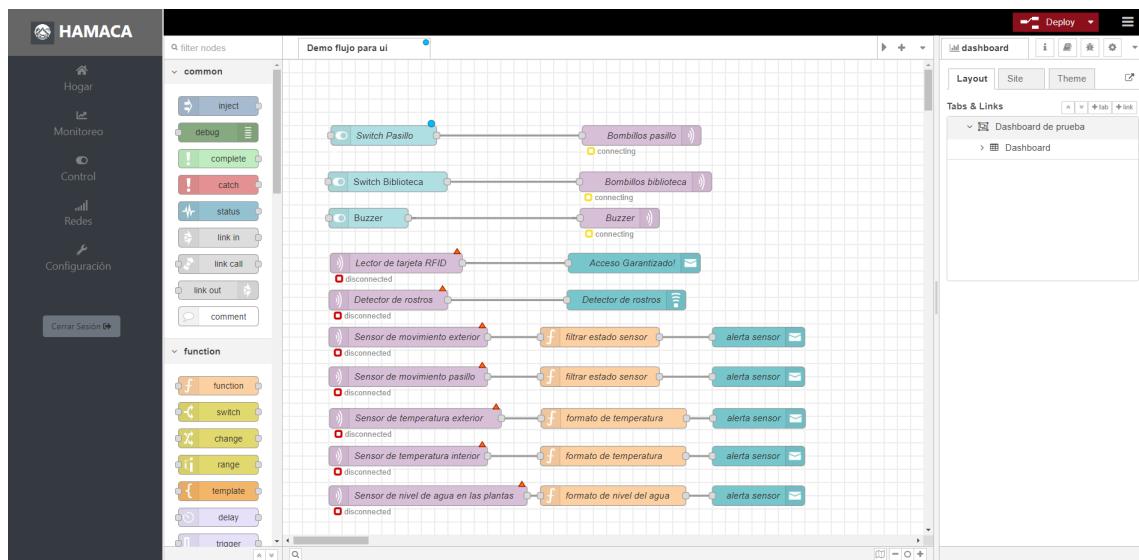


Figura 8.8: Flujos creados dentro de la integración de Node-Red para automatizar tareas

Capítulo 9

Casos de Uso

9.1. Infraestructura de Comunicación

En la solución prevista, específicamente hablando de los componentes que requieren comunicarse usando MQTT como protocolo y el broker, es decir, los dispositivos IoT y la aplicación web HAMACA se establece lo siguiente:

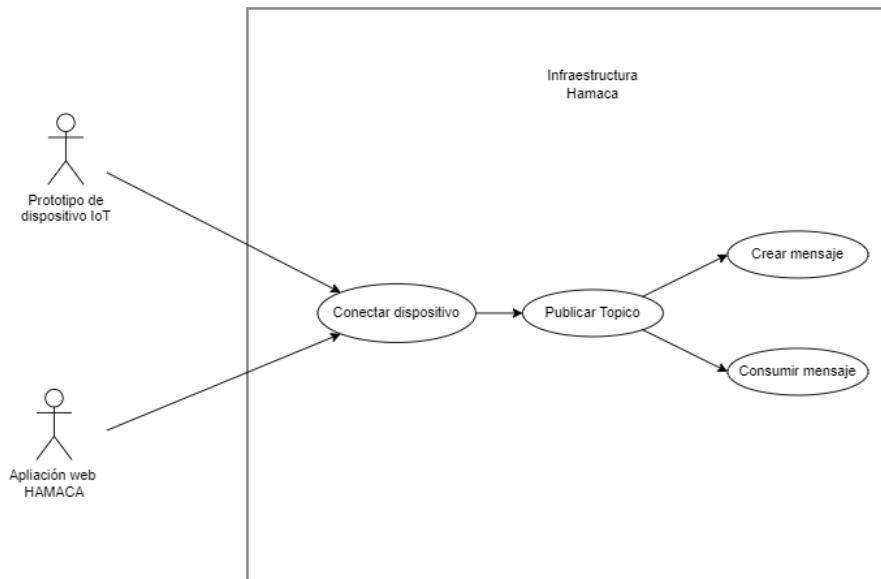


Figura 9.1: Diagrama de caso de uso para la comunicación vía MQTT

El detalle de cada caso es el siguiente:

- **Conectar dispositivo:** Este caso de uso hace referencia a negociar la conexión entre un dispositivo y el broker MQTT utilizado, en este caso, bajo la implementación Mosquitto. Para poder conectarse solo se necesita la dirección del servicio, se

encuentre o no de manera local en la red. Ambos actores pueden realizar esta acción.

- Publicar Tópico: Para poder comenzar a comunicarse, se tiene que definir un tópico para que pueda ser escuchado/leído por cualquier dispositivo con acceso.
- Crear mensaje: Simplemente se envía un mensaje MQTT con el tópico definido y el contenido útil de dicho mensaje
- Consumir mensaje: MQTT sigue le paradigma de publicación, suscripción por lo que es posible que cualquiera que este suscrito a un tópico será capaz de leer el mensaje y este no estará más disponible.

9.2. Aplicación Web

Para el caso de la aplicación web HAMACA, se consideraron dos tipos de usuarios que interactuarán con ella. El primer es un usuario administrador que tiene privilegios por sobre todas las acciones disponibles y un usuario final cualquiera que solo podrá gestionar las acciones que tengan que ver con la visualización, monitoreo y control de dispositivos IoT, sus sensores, actuadores y la data asociada a ellos como se puede ver en el diagrama de la figura 9.2

A continuación se explica cada acción dentro del caso de uso:

- Gestionar usuario: Se refiere a la capacidad de administrar los aspectos concernientes a los usuarios de la aplicación. Solo puede ser accedido por los usuarios administradores.
- Crear usuario: Consiste en llenar el formulario con la información del usuario nuevo de la plataforma para ser guardado en base de datos.
- Editar usuario: Consiste en editar la información existente de un usuario en particular en cualquiera de los elementos disponibles. Se puede editar el nombre, apellido, correo electrónico o la contraseña. Lo realiza el usuario administrador.
- Eliminar usuario: Se borra de la aplicación un usuario. Solo puede borrar el usuario administrador.
- Configurar fuente de datos: Para la configuración inicial de Grafana es requerido crear una conexión a una fuente de datos. Esta acción es para solo aquellos que tiene el rol de admin del sistema.
- Controlar dispositivo: Se refiere a poder usar la interfaz de control para poder realizar una o mas acciones de control sobre uno o más dispositivos IoT. Esto lo puede hacer cualquier usuario.
- Gestionar cuadro de mando: Esta es la acción de administrar lo concerniente a los dashboards o cuadros de mando que posee la integración de Grafana. Puede ser realizado por cualquier usuario.



Figura 9.2: Diagrama de caso de uso para la aplicación web HAMACA

- Crear cuadro de mando: Representa la acción de poder crear un nuevo cuadro de mando desde cero. Puede ser realizado por cualquier rol de usuario.
- Modificar cuadro de mando: Esta acción conlleva editar cualquier cuadro de mando que ya haya sido creado. Puede ser realizado por cualquier usuario.
- Eliminar cuadro de mando: Esta acción simplemente borra un cuadro de mando de la integración. Puede realizarla cualquier usuario.
- Crear indicador: Esta acción consiste en crear un indicador (gráfico, texto, alerta, etc) sobre un cuadro de mando. Debe existir un cuadro de mando al menos para ello y esta acción puede ser llevada a cabo por cualquier rol de usuario.
- Editar indicador: Esta acción consiste en editar un indicador existente en un

cuadro de mando. Puede ser realizada por cualquier usuario.

- Eliminar indicador: Consiste en borrar un indicador sobre un cuadro de mando. Puede ser llevada a cabo por cualquier usuario.
- Gestionar flujo de automatización: Se refiere a la capacidad de administrar uno o mas flujos de automatización de la aplicación haciendo uso de la integración de la herramienta Node-Red. Esta acción puede ser llevada a cabo por cualquier tipo de usuario.
- Crear flujo de automatización: Esto es el poder crear un flujo nuevo desde cero para automatizar una o más tareas. Puede ser realizada por cualquier usuario.
- Desplegar flujo de automatización: Consiste en tomar un flujo de trabajo existe y activarlo para su funcionamiento. Esta acción la puede hacer cualquier usuario.
- Modificar flujo de automatización: Como su nombre lo dice, se refiere a la capacidad de los usuarios de editar un flujo de trabajo existente, sin importar que rol posean.
- Eliminar flujo de automatización: Cualquier usuario tiene la potestad de poder borrar un flujo de trabajo existe.
- Crear tarea: Esto es la capacidad de crear una tarea para automatizar sobre uno o mas sensores/actuadores de uno o más dispositivos. Se requiere que exista un flujo de trabajo sobre el cual hacer esto. Es una acción que puede realizar cualquier usuario.
- Editar tarea: Significa que se cambien características o instrucciones sobre una tarea existente. Esta acción la puede realizar cualquier usuario.
- Eliminar Tarea: Hace referencia a la capacidad de borrar una tarea existente sobre un flujo de trabajo. Esto es posible para cualquier usuario, independientemente de su rol.

Capítulo 10

Resultados, Limitaciones y Trabajos Futuros

10.1. Resultados

Como primer elemento representativo de este trabajo, se hicieron tres prototipos funcionales de dispositivos IoT utilizando diferentes placas programables, entre las que se encuentran el microcontrolador Arduino Uno, Raspberry Pi Zero y Raspberry Pi 3 Modelo B. A cada uno de ellos se le integró una serie de sensores capaces de hacer lecturas sobre el ambiente, así como tambien de elementos actuadores para reaccionar ante determinados eventos tanto de manera automática como de manera manual.

Para ello se crearon una serie de scripts en el lenguaje de programación Python y en el lenguaje de programación Arduino, para establecer la comunicación entre sensores y actuadores y las placas programables. Se estableció también MQTT como protocolo adecuado para realizar la comunicación entre los dispositivos, usando el paradigma de publicación/suscripción ideal para ambientes distribuidos y diseñado para ser liviano y eficiente, métricas que son de relevancia a la hora de trabajar con dispositivos electrónicos con limitaciones de recursos, tanto computacionales como energéticos.

La operación de dichos prototipos condujo a la obtención de 2.6 GB de datos no estructurados, 3.1 GB de imágenes y de 1.7 GB de vídeos, que ayudaron a validar la hipótesis de que aun dentro del ambiente de pruebas que se diseñó fue un escenario cercano a la realidad en donde se obtuvo volumen, diversidad de datos así como el modo de operaciones de cada uno de los distintos dispositivos para modificar el ambiente a través de los actuadores.

Esto también ayudó a validar la serie de dificultades técnicas que se debían abordar posteriormente para la implementar de manera correcta la solución que permitiese realizar observación, monitoreo y automatización de los dispositivos y afectó a la selección del software para almacenamiento de datos, las integraciones con herramientas de visualización, monitoreo y control disponibles.

Llegados al escenario del desarrollo del software que permitiese lograr los objetivos

anteriormente expuestos durante el planteamiento del problema se diseñó y desarrollo una solución cuya arquitectura contuviese los elementos necesarios para poder ser desplegado en cualquier plataforma. Esta arquitectura posee el componente de gestión de protocolo de comunicación MQTT bajo la implementación Mosquitto. Gracias al conocimiento adquirido de la naturaleza de los datos se tomó la decisión de usar la base de datos InfluxDB, la cual esta diseñado con el propósito de ser usado bajo el esquema de series de tiempo, dimensión común junto a los valores que proveen los sensores y actuadores a lo largo y ancho del proyecto.

Como se mencionó antes, la elección de las herramientas de visualización, monitoreo y control fueron influenciadas también por las características de los datos y de los prototipos. Por el lado de la herramienta de visualización se escogió el uso de Grafana como software de visualización por su sencillas y a la vez potencia para poder desplegar cuadro de mandos a partir de información en tiempo real e histórica de manera intuitiva y que apoye al análisis posterior de los datos que se van generando por parte del usuario final. Por el lado del monitoreo y control, la decisión tomada fue usar Node-Red, un software que permite crear automatizaciones sobre elementos virtuales y físicos de manera visual lo que la hace bastante sencilla de utilizar, ademas de permitir desplegar y realizar acciones en los dispositivos a través de flujos de tareas de forma automática o manual (en los paneles de control generados a partir de los mismos).

La aplicación web fue diseñada para poder integrar estas herramientas así como también para ser modular y fue construida usando el lenguaje de programación Python bajo el framework de desarrollo web Django. Se comenzó el desarrollo desde el módulo de gestión de las configuraciones de la aplicación web, pasando luego por los módulos que integrarían las herramientas de visualización de datos y de monitoreo y control de los dispositivos.

Con la data ya disponible se validó que las herramientas fueran capaces de poder hacer uso de los mismos lo cual se evidenció en la creación de en el caso de Grafana, de cuadros de mando con indicadores de las variables medidas por los sensores tanto de forma histórica como los datos en tiempo real y para el caso de Node-red, la creación de flujos de tareas que fuesen capaces de automatizar procesos que no estaban dentro de la lógica realizada para los prototipos de dispositivos IoT como la generación de interfaces de control para encender o apagar elementos actuadores, así como también la posibilidad de usar otro tipos de indicadores en los dashboards que se generan a partir de los flujos de tareas automatizadas (y capaces de generar alertas personalizadas). Por último con respecto a la red se validó que el proyecto fuese capaz de realizar el seguimiento estadístico de las comunicaciones realizadas por dispositivo y por los sensores y actuadores involucrados, llevando y mostrando un registro de las actividades actuales con base a la red.

De esta manera se logró realizar un software que es capaz de poder minimizar el impacto de no tener herramientas diseñadas específicamente para el entorno de administración y gestión de dispositivos IoT teniendo en mente el poder ver y actuar de manera centralizada que se encuentren desplegados dentro de un ambiente.

10.2. Limitaciones

A pesar de que se dio una respuesta para poder organizar y gestionar datos y procesos administración de dispositivos IoT, aun existen problemas para hacer de la experiencia de usuario algo más placentero e intuitivo. Las herramientas integradas (Grafana y Node-Red) si bien son bastante fáciles de usar, requieren de cierto conocimiento técnico para poder obtener todas las ventajas que pueden proveer, lo cual puede desalentar al usuario final del uso completo de toda la aplicación.

Por otro lado, las fuentes de datos (entiéndase los dispositivos que generan los datos) pueden cumplir con estándares que no están considerados en la actualidad dentro de esta solución por lo que de momento quedarían por fuera de las capacidades actuales del software desarrollado.

10.3. Contribuciones

Durante el presente trabajo de investigación se pudo realizar un desarrollo que de principio a fin, trato de examinar la problemática y sus características en un entorno que fuera lo más cercano a la realidad, teniendo como objetivo en mente que este software pueda ser desplegado con confianza en entornos donde se requiere la observación y gestión de dispositivos IoT.

En principio, el presente trabajo se consideró solo para el uso de este software dentro de ámbitos doméstico, es decir, en hogares con alta cantidad de artefactos inteligentes. Sin embargo, después de examinar detenidamente otros escenarios se pudo establecer que este software se puede utilizar en casi cualquier contexto que requiera de una forma de observar datos de dispositivos, así como poder hacer gestión de automatización de procesos, como bien puede ser el contexto de un laboratorio (bien sea de índole académico, industrial o comercial) y así como también en el desarrollo de dispositivos o la gestión de recursos usando objetos inteligentes en un ambiente.

Por último la esta herramienta se espera que pueda servir como prueba de concepto y como marco de desarrollo para en principio ser usado dentro del Laboratorio de Internet de las Cosas de la Escuela de Computación de la Facultad de Ciencias de la Universidad Central de Venezuela, en aras de permitir una experiencia más profunda en el mundo de los dispositivos IoT.

10.4. Trabajos Futuros

Este trabajo fue pensado y diseñado para ser modular, pues el mundo del Internet de las cosas es bastante amplio y cambiante, por lo que para que este software pueda crecer orgánicamente y adaptarse a las necesidades de los usuarios, siendo incluso pilar fundamental durante cada etapa de este trabajo la utilización de tecnologías de software libre, que permitan extender características mas allá de las originales si así se desea.

Uno de los aspectos de mejora que tiene el desarrollo es la capacidad de poder utilizar y conectarse a dispositivos que utilicen otros protocolos y estándares distintos a MQTT de forma de abarcar una cantidad mayor de dispositivos y que este sea un hub para casi cualquier tipo de dispositivo IoT futuro.

Otra característica del software donde se puede mejorar es en la interfaz gráfica de la aplicación web HAMACA. De momento, la aplicación cuenta con una serie de elementos que permiten su correcta visualización en el computador pero se sugiere que para el futuro se tengan en cuenta interfaces mejor adaptadas para pantallas móviles así como también dentro de las herramientas integradas.

Por otro lado, si bien es cierto que todos los elementos desarrollados tienen una serie de funcionalidades de seguridad, se pueden agregar nuevos elementos y estándares que vayan surgiendo en el futuro para blindar la aplicación, teniendo en cuenta que se utilizan datos que pueden ser sensibles o de alto valor y causar daños en manos inescrupulosas si estos no están bien asegurados. Se sugiere examinar las posibilidades de extender este apartado tanto desde el punto de vista de las redes como de dispositivos también, el como mitigar vulnerabilidades inherentes al uso de un sistema web.

Capítulo 11

Conclusión

Este trabajo de investigación permitió demostrar la factibilidad del desarrollo de una herramienta de visualización de datos, monitoreo y gestión de procesos concernientes a dispositivos de Internet de las cosas, partiendo desde el desarrollo de prototipos funcionales que pudiesen generar información para proveer de un contexto cercano sobre el como de desplegaría un futuro ambiente donde los artefactos inteligentes son ubicuos.

El desarrollo de dispositivos IoT permitió identificar las fortalezas y debilidades de los enfoques utilizados para medir y controlar estos artefactos de manera transparente. Se observaron como el fenómeno de la variedad entre sensores, actuadores y dispositivos en si pueden generar data que aunque representen lo mismo tiene matices de eficiencia y eficacia que pueden ser aprovechados por otros elementos en la búsqueda de capturar, mostrar y utilizar la información que capturamos de ellos.

Por el lado del desarrollo de la aplicación web se demostró que es posible integrar en un software que pueda orquestar e integrar herramientas existentes para tareas específicas como visualización de datos, automatización de procesos, monitoreo y control de dispositivos y que puede ser una alternativa viable para poder centralizar las operaciones inherentes a dispositivos IoT y dentro de un esquema flexible que permite no solo adaptarse a las necesidades de los usuarios y sino también con alta capacidad de crecimiento futuro, sea desde el punto de vista de protocolos y estándares, como de software análisis automatizado.

Finalmente, se considera que los objetivos planteados en el marco del planteamiento del problema fueron cumplidos con éxito y se espera que este software sea al final de utilidad en donde el contexto permita aprovechar las funcionalidades desarrolladas anteriormente, sea dentro del ámbito académico, investigativo, industrial o uso personal.

Bibliografía

- [1] James Manyika & otros. The internet of Things: Mapping the Value Beyond the Hype. http://www.mckinsey.com/insights/business_technology/the_internet_of_things_the_value_of_digitizing_the_physical_world, Junio 2015. [Accedido: 17-06-2017].
- [2] Chetan Sharma. Correcting The IoT History. <https://www.chetansharma.com/correcting-the-iot-history/>. [Accedido: 14-10-2023].
- [3] Kamlesh Lakhwani. *Internet of Things (IoT) : Principles, Paradigms and Applications of IoT*. BPB Publications, 2020.
- [4] Karen Rose & Scott Eldridge & Lyman Chapin. The Internet of Things: An Overview - Understanding the issues and Challenges of a More Connected World. Technical report, The Internet Society, Octubre 2015.
- [5] H. Tschofenig & others. Architectural Considerations in Smart Object Networking. Technical report, Internet Architecture Board, Marzo 2015.
- [6] Jen Clark. What is industry 4.0? <https://www.ibm.com/blogs/internet-of-things/industry-4-0/>, Octubre 2016. [Accedido: 18-06-2017].
- [7] Lynne Slowey. From automotive to mobility, the future of IoT for the car industry. <https://www.ibm.com/blogs/internet-of-things/automotive-future-iot/>, Febrero 2017. [Accedido: 17-06-2017].
- [8] Rahul. IoT applications spanning across industries. <https://www.ibm.com/blogs/internet-of-things/iot-applications-industries/>, Abril 2017. [Accedido: 17-06-2017].
- [9] Kevin Patel. 6 Benefits of IoT for Hospitals and Healthcare. <https://www.ibm.com/blogs/internet-of-things/6-benefits-of-iot-for-healthcare/>, Enero 2017. [Accedido: 17-06-2017].
- [10] Bill McCabe. Creating smart cities with IoT. <https://www.ibm.com/blogs/internet-of-things/creating-smart-cities-iot/>, Junio 2016. [Accedido: 17-06-2017].
- [11] Henrik Nielsen, Roy T. Fielding, and Tim Berners-Lee. Hypertext Transfer Protocol – HTTP/1.0. RFC 1945, May 1996.

- [12] Sergio Collado & otros. HTTP. <https://developer.mozilla.org/es/docs/Web/HTTP>, 2017 Mayo. [Accedido: 18-06-2017].
- [13] Cigdem Sengul and Anthony Kirby. Message Queuing Telemetry Transport (MQTT) and Transport Layer Security (TLS) Profile of Authentication and Authorization for Constrained Environments (ACE) Framework. RFC 9431, July 2023.
- [14] Stan Schneider. Understanding The Protocols Behind The Internet Of Things. <http://www.electronicdesign.com/iot/understanding-protocols-behind-internet-things>, Octubre 2013. [Accedido: 18-06-2017].
- [15] James Stansberry. The IoT communication protocols. <https://www.linkedin.com/pulse/iot-communication-protocols-james-stansberry>, Octubre 2015. [Accedido: 18-06-2017].
- [16] Information Sciences Institute of University of Southern California. INTERNET PROTOCOL. Technical report, Defense Advanced Research Projects Agency, Septiembre 1981.
- [17] Stephen Deering & Robert Hinden. Internet Protocol, Version 6 (IPv6). Technical report, The Internet Society, Diciembre 1998.
- [18] The Institute of Electrical and Inc. Electronics Engineers. Ieee 802.15 wpan task group 1 (tg1). <http://www.ieee802.org/15/pub/TG1.html>. [Accedido: 22-06-2017].
- [19] International Telecommunication Union. About mobile technology and IMT-2000. <https://www.itu.int/osg/spu/imt-2000/technology.html>. [Accedido: 22-06-2017].
- [20] UMTSWorld.com. 3G and UMTS Frequently Asked Questions. <http://www.umtsworld.com/umts/faq.htm#f1>. [Accedido: 22-06-2017].
- [21] 3GPP Initiative. W-CDMA. <http://www.3gpp.org/technologies/keywords-acronyms/104-w-cdma>. [Accedido: 22-06-2017].
- [22] 3GPP Initiative. HSPA. <http://www.3gpp.org/technologies/keywords-acronyms/99-hspa>. [Accedido: 22-06-2017].
- [23] 3GPP Initiative. LTE. <http://www.3gpp.org/technologies/keywords-acronyms/98-lte>. [Accedido: 22-06-2017].
- [24] The Institute of Electrical and Inc. Electronics Engineers. IEEE 802.16e Task Group (Mobile WirelessMAN). <http://www.ieee802.org/16/tge/>. [Accedido: 22-06-2017].
- [25] Organización Internacional de Estandarización. ISO/IEC 18000, Part 3, Information Technology AIDC Techniques - RFID for Item Management - Air Interface, Part 3 - Parameters for Air Interface Communications at 13.56 MHz. Technical report, Organización Internacional de Estandarización, Mayo 2002.

- [26] The Institute of Electrical and Inc. Electronics Engineers. IEEE 802.11 WIRELESS LOCAL AREA NETWORKS. <http://www.ieee802.org/11/>. [Accedido: 22-06-2017].
- [27] The Institute of Electrical and Inc. Electronics Engineers. Ieee 802.15 wpanTM task group 4 (tg4). <http://www.ieee802.org/15/pub/TG4.html>. [Accedido: 22-06-2017].
- [28] RS Components. 11 internet of things (iot) protocols you need to know about. <https://www.rs-online.com/designspark/eleven-internet-of-things-iot-protocols-you-need-to-know-about>, Abril 2015. [Accedido: 22-06-2017].
- [29] El androide libre. Todo sobre zigbee, la tecnología ultrabaratina para comunicación inalámbrica. <https://elandroidelibre.elespanol.com/2015/08/todo-sobre-zigbee-la-tecnologia-ultrabarata-para-comunicacion-inalambrica.html>, Agosto 2015. [Accedido: 22-06-2017].
- [30] Pentadom. LA DOMÓTICA INALÁMBRICA CON Z-WAVE, UNA REALIDAD PARA VIVIENDAS EXISTENTES. <http://pentadom.com/2014/03/la-domotica-inalambrica-con-z-wave-una-realidad-para-viviendas-ya-construidas/>, Marzo 2014. [Accedido: 22-06-2017].
- [31] U.S. Department of Homeland Security. STRATEGIC PRINCIPLES FOR SECURING THE INTERNET OF THINGS (IoT). Technical report, U.S. Department of Homeland Security, Noviembre 2015.
- [32] Javier Murillo. Los datos son el nuevo petróleo. <https://www.forbes.com.mx/red-forbes-los-datos-son-el-nuevo-petroleo/>. [Accedido: 14-10-2023].
- [33] LLC TABLEAU SOFTWARE. Guía de visualización de datos para principiantes: definición, ejemplos y recursos de aprendizaje. <https://www.tableau.com/es-mx/learn/articles/data-visualization#:~:text=La%20visualizaci%C3%B3n%20de%20los%20datos%20es,y%20patrones%20en%20los%20datos>. [Accedido: 14-10-2023].
- [34] IBM Cognos Analytics. Visualizaciones gráficas. https://www.ibm.com/docs/es/cognos-analytics/11.1.0?topic=SSEP7J_11.1.0/com.ibm.swg.ba.cognos.pp_usr_gd.doc/c_graphreports.htm, Niviembre 2022. [Accedido: 14-10-2023].
- [35] John von Neumann. First Draft of a Report on the EDVAC. Technical report, Moore School of Electrical Engineering, University of Pennsylvania, Junio 1945.
- [36] The Arduino Company. WHAT IS ARDUINO? <https://www.arduino.cc>. [Accedido: 08-07-2017].
- [37] Raspberry Pi Foundation. Raspberry Pi. <https://www.raspberrypi.org>. [Accedido: 08-07-2017].

- [38] SOFTENG. Metodología Scrum. <https://www.softeng.es/es-es/empresa/metodologias-de-trabajo/metodologia-scrum.html>. [Accedido: 17-07-2017].
- [39] Guido Van Rossum. What is Python, Executive Summary. <https://www.python.org/doc/essays/blurb/>, 1997. [Accedido: 11-06-2017].
- [40] Guido Van Rossum. Foreword for 'Programing Python' (1st ed.). <https://www.python.org/doc/essays/foreword/>, Mayo 1996. [Accedido: 11-06-2017].
- [41] A. M. Kuchling. Python Advanced Librery. <https://www.python.org/dev/peps/pep-0206/>. [Accedido: 11-06-2017].
- [42] Roger A Light. Mosquitto: server and client implementation of the MQTT protocol. *The Journal of Open Source Software*, 2(13), Mayo 2017.
- [43] influxdata. The Time Series Database Platform Optimized for Speed and Scale. <https://www.influxdata.com/solutions/>. [Accedido: 15-10-2023].
- [44] Meet Django. <https://www.djangoproject.com>. [Accedido 11-06-2017].
- [45] PostgreSQL: The World's Most Advanced Open Source Relational Database. <https://www.postgresql.org/>. [Accedido 15-10-2023].
- [46] Dashboard anything. Observe everything. <https://grafana.com/grafana>. [Accedido 15-10-2023].
- [47] Node-RED Low-code programming for event-driven applications. <https://nodered.org/>. [Accedido 15-10-2023].
- [48] Overview of the get started guide. <https://docs.docker.com/get-started/>. [Accedido 15-10-2023].