

2022-09

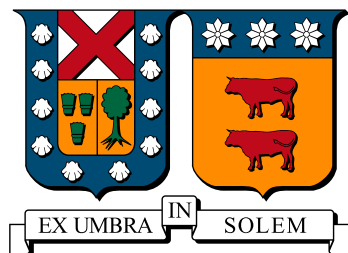
Diseño de plataforma IOT para monitoreo remoto y control de edificios inteligentes

Arriagada Chávez, Sebastián Alberto

<https://hdl.handle.net/11673/54053>

Repositorio Digital USM, UNIVERSIDAD TECNICA FEDERICO SANTA MARIA

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE ELECTRÓNICA
VALPARAÍSO – CHILE



**“DISEÑO DE PLATAFORMA IOT PARA
MONITOREO REMOTO Y CONTROL DE
EDIFICIOS INTELIGENTES”**

SEBASTIÁN ARRIAGADA CHÁVEZ

**MEMORIA DE TITULACIÓN PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL ELECTRÓNICO**

PROFESOR GUÍA: MOHAMED ABDELHAMID

PROFESOR CORREFERENTE: JOSÉ MANUEL MARTÍNEZ

SEPTIEMBRE 2022

Agradecimientos

Aprovecho este espacio para agradecer de todo corazón a todos a quienes me han apoyado en este proceso. Agradezco a mi profesor guía Mohamed Abdelhamid por toda su paciencia y apoyo, a mi esposa Nicol Cremer por todo su amor y apoyo, a todos mis amigos por su apoyo incondicional en especial en los tiempos difíciles y a todos los que fueron parte de este proceso difícil pero gratificante de aprendizaje.

También agradezco a los profesores del departamento de Ingeniería civil Electrónica por todo su apoyo y dedicación.

Y me gustaría dar un agradecimiento especial a la Agencia Nacional de Investigación y desarrollo (ANID), Proyecto Fondecyt de Iniciación en Investigación 2020, Proyecto ID 11200178, por apoyar este proyecto.

¡Muchas gracias!

Resumen

Con la creciente necesidad de obtener datos de los consumos energéticos dentro de un edificio es que se hace necesaria la investigación de la obtención, orden y visualización de dicha información.

Entonces, si se considera un edificio inteligente como aquel que permite obtener los datos desde dispositivos de medición energética y sensores, basta con tener acceso a dicho tipo de edificio para obtener la información buscada.

Lamentablemente, ya que no todos los edificios poseen esta característica, se propone una serie de dispositivos de medición que permitan recabar la información de uso energético. Pero, con los dispositivos instalados se requiere de una plataforma que permita su fácil visualización por el usuario final.

Para ello se propone una plataforma de arquitectura IoT que tome los datos de consumo energético, los almacene y los muestre al usuario que puede conectarse a ella desde internet. La implementación de los dispositivos de medición se hace pensando en la implementación en 3 escenarios característicos de un edificio (que se sabe además que forma parte de conjunto de estos llamado Campus) que son: una oficina, una sala de clases y un laboratorio. Y por otro lado se realiza la implementación en un banco de pruebas que permite tener acceso a todas las cargas planteadas para probar dicha plataforma.

Abstract

With the growing need to obtain data of energy consumption in buildings, it is necessary to investigate the obtaining, ordering and visualization of said information.

Then, if an intelligent building is considered as one that allows obtaining data from energy measurement devices and sensors, it is enough to have access to that type of building to obtain the that information .

Unfortunately, since not all buildings have this characteristic. This work proposes a series of measurement devices that allow information on energy use to be collected. With the devices installed, a platform is required that allows easy viewing the information by the end user.

For this, an IoT platform is proposed. This platform takes the energy consumption data, stores it and shows it to the user who can connect to it from the internet.

The application of the measurement devices must be done considering the implementation in three characteristic scenarios of a building (which is also known to be a part of a set of these buildings called Campus) which are: an office, a classroom and a laboratory. On the other hand, the implementation must be carried out in a test bench that allows access to all the loads proposed to test said platform.

Índice de Contenidos

Introducción	1
1. Definición del Problema	2
1.1. Descripción del problema	2
1.2. Objetivos del trabajo	3
1.3. Trabajo a desarrollar	3
1.4. Evaluaciones a realizar	4
1.5. Resultados esperados	5
2. Estado del Arte	6
2.1. Temas relevantes a tratar	6
2.2. Trabajos relacionados	7
3. Propuesta	14
3.1. Análisis de las alternativas posibles de solución	14
3.2. Arquitectura	14
3.2.1. Jerarquía en arquitectura para monitoreo y control	16
3.2.2. Modelamiento de un edificio inteligente	17
3.2.3. Capas propuestas	18
3.3. Escenarios de implementación	36

3.3.1. Oficina	36
3.3.2. Laboratorio	37
3.3.3. Sala de clases	37
3.4. Alternativas seleccionadas en dispositivos, tecnologías y servicios	38
3.4.1. Capa de poder	38
3.4.2. Capa de adquisición	39
3.4.3. Capa de comunicación	41
3.4.4. Capa Niebla	43
3.4.5. Capa de aplicación	45
4. Implementación	46
4.1. Capa de Adquisición de Datos	47
4.1.1. Monitoreo energético de equipos conectados a la red	47
4.1.2. Monitoreo de las condiciones ambientales en ambientes interiores	55
4.1.3. Monitoreo de la energía total	56
4.1.4. Monitoreo del sistema de paneles fotovoltaicos	57
4.1.5. Manejo de datos de estacion meteorológica	58
4.2. Capa de comunicación	60
4.3. Capa Niebla	65
4.4. Capa de Aplicación	66
Conclusiones	78
Bibliografía	80

Índice de cuadros

2.1. Resumen de las capas mencionadas en los principales proyectos desarrollados en el área.	13
3.1. Dispositivos de medición para sistemas fotovoltaicos.	19
3.2. Características de dispositivos de medición de energía.	20
3.3. Características de interruptores inteligentes.	25
3.4. Características de enchufes inteligentes.	25
3.5. Características de sensores ambientales.	27
3.6. Características de sensores magnéticos.	28
3.7. Características de sensores PIR.	28
3.8. Características de las Gateway LoRa.	29
3.9. Características de enchufes inteligentes.	40
3.10. Evaluación de las características de dispositivos de medición de energía. . .	40
3.11. Características de las Gateway LoRa.	42
3.12. Principales características de los protocolos de comunicación con la nube. .	43
3.13. Comparación porcentual de los principales protocolos de comunicación con la nube.	43

3.14. Comparación de las principales características de los dispositivos de computo propuestos para este proyecto.	44
3.15. Comparación porcentual de los dispositivos de computo propuestos para este proyecto	44
4.1. Parámetros de conexión WIFI de SonOff Pow R2 con ESPurna.	54
4.2. Descripción de tópicos de los dispositivos conectados por MQTT.	62

Índice de figuras

2.1. Esquema de capas de la tecnología IoT para edificios inteligentes.	7
2.2. Esquema de conexión de un BEM en la nube con los dispositivos en un edificio usando un <i>Gateway IoT</i>	8
2.3. Diseño de la <i>Gateway IoT</i> desarrollada.	9
2.4. Diagrama de Arquitectura IoT propuesta incluyendo <i>capa Fog</i>	9
2.5. Diagrama de Arquitectura IoT de 3 capas planteada para edificios inteligentes	10
2.6. Diagrama de las tecnologías utilizadas en implementaciones IoT para edificios inteligentes.	11
2.7. Visión general de las variables a considerar en un edificio inteligente. . . .	12
2.8. Dispositivos utilizados en sistemas de manejo de energía en edificios inteligentes.	12
3.1. Esquema de capas de la tecnología IoT para edificios inteligentes.	16
3.2. Esquema de edificios inteligentes formando un campus inteligente	16
3.3. Diagrama del manejo de datos para un sistema de manejo de energía en un edificio inteligente.	17
3.4. Diagrama de la implementación de un sistema de monitoreo de energía en un edificio inteligente.	18

3.5. Estación meteorológica utilizada en el edificio en estudio (Modelo Davis 6152).	19
3.6. Diagrama eléctrico de un sistema solar híbrido.	21
3.7. Diseño propuesto para la medición del bus de corriente continua en un sistema fotovoltaico híbrido.	22
3.8. Equipos propuestos para adquisición de datos desde el tablero eléctrico general.	23
3.9. Interruptores inteligentes propuestos.	23
3.10. Enchufes inteligentes propuestos.	24
3.11. Sensores ambientales.	24
3.12. Sensores magnéticos.	26
3.13. Sensores PIR.	26
3.14. Gateways LoRa.	29
3.15. Dispositivo de cómputo.	32
3.16. <i>Middlewares</i> propuestos para mantener la información en la nube	34
3.17. Mapa de edificios en el campus Valparaíso de la UTFSM y escenarios planteados para la implementación.	36
3.18. Equipos seleccionados para la adquisición de datos.	39
3.19. Alternativa seleccionada para el <i>gateway</i> LoRa	42
4.1. Diagrama de implementación de dispositivos de medición conectados al servidor planteado	48
4.2. Banco de pruebas armado donde se ha segmentado con colores diferentes partes de la implementación.	49
4.3. Imagen de la conexión para configuración de dispositivo Sonoff Pow R2.	51

4.4. Vistas de las configuraciones y vistas disponibles luego de la instalación del <i>firmware</i> ESPurna	53
4.5. Diagrama de conexión para configuración de dispositivo LAQ4	55
4.6. Diagrama de conexión del dispositivo de medición energética pzem004t-100A.	56
4.7. Conexión del módulo pzem004t-100A para la lectura general de energía. . .	57
4.8. Prototipo de equipo de medición de paneles solares	58
4.9. Vista general de los datos de la estación meteorológica obtenida desde el <i>WhetherLink Computer Software</i>	59
4.10. Ventana de activación y configuración de la comunicación de los enchufes inteligentes usando MQTT	61
4.11. Ventana se estado de la conexión entre la <i>gateway</i> LoRa y el servidor en <i>The Things Network</i>	63
4.12. Ventana se estado de la conexión entre el dispositivo final LAQ4 y el servidor en <i>The Things Network</i>	64
4.13. Ventana de configuración de envío de datos por MQTT en <i>The Things Network</i>	64
4.14. Esquema implementado para el reenvío de los datos recibidos de los enchufes inteligentes conectados	66
4.15. Esquema implementado para el reenvío de los datos recibidos del dispositivo de medición de energía general	67
4.16. Esquema implementado para el reenvío de los datos recibidos del dispositivo de medición de corriente continua	67
4.17. Ventana de configuración de <i>Droplet</i> en <i>Digital Ocean</i>	68

4.18. Ventana configuración de flujos para la visualización de datos de los enchufes inteligentes.	71
4.19. Ventana configuración de flujos para la visualización de datos del medidor general de tablero eléctrico.	71
4.20. Ventana configuración de flujos para la visualización de datos del medidor del panel fotovoltaico.	72
4.21. Ventana configuración de flujos para la visualización de datos ambientales. .	72
4.22. Ventana configuración de flujos para la activación de los enchufes inteligentes utilizados.	72
4.23. Configuraciones de nodos característicos para el almacenamiento de datos en MySQL	73
4.24. Configuraciones de nodos característicos para la visualización de datos recibidos por MQTT.	74
4.25. Captura de pantalla de la visualización de los datos de los enchufes inteligentes obtenidos.	75
4.26. Captura de pantalla de la visualización de los datos de medición del tablero eléctrico.	75
4.27. Captura de pantalla de la visualización de los datos del sistema de paneles solares.	76
4.28. Captura de pantalla de la visualización de los datos de energía consumida en el tablero eléctrico.	76
4.29. Captura de pantalla de la visualización de los datos ambientales obtenidos. .	77
4.30. Capturas de pantalla de los parámetros guardados en cada tabla en MySQL.	77

Glosario

- **BMS:** Del inglés *Building Management System*, se refiere al sistema que permite el manejo de las condiciones de un edificio, es decir, que tiene acceso a todas las variables medidas en dicho edificio.
- **Broker MQTT:** Programa que funciona como servidor central para los mensajes enviados por el protocolo MQTT.
- **ESPurna:** Firmware desarrollado principalmente por Xose Pérez que puede ser cargado en diferentes dispositivos de manera de obtener todos los datos provenientes de la capa física de cada dispositivo compatible y mostrarlos de manera clara por su interfaz web o enviarlos a través de internet por alguno de los protocolos definidos.
- **Firmware:** Software que maneja directamente el hardware donde está cargado y permite la interacción del equipo con el usuario.
- **Gateway:** Dispositivo que funciona como una interfaz de comunicación entre 2 o más tecnologías que son incompatibles entre sí.
- **IoT:** Del inglés *Internet of things* y que define la interconexión entre equipos (cosas) que han sido dotadas de algún medio de comunicación informática con el exterior.
- **LoRa:** Tecnología de comunicación inalámbrica que se basa en la modulación en frecuencia patentada que se destaca por su largo alcance y bajo ancho de banda.
- **LORAWAN:** Protocolo de red diseñado para la comunicación de equipos con tecnología LoRa.

- **Smart Building:** Edificios que poseen equipamiento y dispositivos que permite la gestión y control automatizada.
- **Smart Plug:** Enchufe inteligente que independiente del tipo de conexión que posea a la red, permite el monitoreo y/o accionamiento de las cargas conectadas a él.

Introducción

Considerando que a nivel mundial los edificios consumen un 60 % de la energía eléctrica producida [1] es que cobra real importancia la información del uso de la energía en este sector.

En particular, para aquellos usuarios que están a cargo de la administración y cuidado de un edificio, saber que equipos presentan un consumo fuera de los rangos normales es de vital importancia para controlar los gastos asociados a la operación del lugar.

Actualmente, existe la tendencia de incluir sistemas de generación eléctrica como paneles solares o mini estaciones eólicas entre otros sistemas de energías renovables a los edificios, pero esto no resuelve completamente el problema ya que incluso al diseñar el proyecto de un sistema de generación energética se hace un estudio del consumo general y no se contemplan los valores diarios reales de uso de energía y los potenciales tendencias de aumento en el uso de esta.

Es por ello que se requiere incluir cierta inteligencia (vista como la capacidad de transmisión de la información de estado y variables internas relevantes) a la instalación de equipos eléctricos, incluyendo dispositivos de monitoreo energético que se puedan conectar a la nube para observar los datos a la distancia. En particular se requiere de una plataforma que permita reunir la información de estos dispositivos inteligentes y mostrarla al usuario final para que tome las acciones que crea necesarias para el correcto funcionamiento del edificio.

El presente trabajo pretende proponer e instalar los equipos de medición y, diseñar y ejecutar dicha plataforma para el monitoreo energético del edificio.

Capítulo 1

Definición del Problema

1.1. Descripción del problema

Hoy en día los edificios a nivel mundial consumen el 60 % de la energía producida [1] por lo que se vuelve relevante realizar el monitoreo de la energía consumida en cada sistema de estos.

En Chile el 8 de febrero del 2021 se promulga la ley 21.305 de eficiencia energética en donde se establece, entre otras cosas, que el ministerio de energía debe generar periódicamente un plan nacional de eficiencia energética donde se definen las principales directrices del consumo de este tipo de recursos. Esta ley faculta al ministerio de energía para pedir datos de consumos energéticos específicos a los responsables de edificios e industrias según el rango del consumo en que se encuentren.

La incorporación de fuentes de energía renovables como pueden ser sistemas fotovoltaicos o pequeñas centrales eólicas permiten disminuir los gastos de electricidad suministrada por las compañías distribuidoras, pero no resuelven por sí mismos el problema de mejorar la eficiencia energética de los mismos. Para monitorear el consumo energético se deben instalar dispositivos inteligentes de medición y sensores que permitan obtener información de la utilización de los recursos utilizados.

En un edificio que posee tecnología inteligente es imprescindible contar con una tecnología que permita interactuar de manera directa con dichos dispositivos, ya que es un requisito

mínimo para ser considerado un edificio inteligente.

En particular, un edificio inteligente es aquel cuyas instalaciones y sistemas permiten una gestión y control integrado y automatizado, con el fin de aumentar la eficiencia energética, la seguridad, la usabilidad y la accesibilidad [2].

Una tecnología habilitadora para la realización de un edificio inteligente es el “internet de las cosas” o por sus siglas en inglés IoT, tecnología que propone una arquitectura orientada a la conexión de todos los dispositivos inteligentes dispuestos en un lugar.

Las tecnologías de comunicación de los dispositivos inteligentes son variadas, pueden ser cableadas o inalámbricas y en general muchos de estos dispositivos pueden interactuar con el usuario a través de aplicaciones dadas por el fabricante.

Por tanto, se requiere de una plataforma IoT que permita centralizar toda la información proveniente de dispositivos y sensores inteligentes de manera de mostrarla de forma clara al usuario y permitir su interacción con dichos dispositivos. Dicha plataforma debe ser de bajo costo y mínimamente invasiva en el entorno del edificio a instalarse.

1.2. Objetivos del trabajo

Diseñar una plataforma que permita integrar la información de sistemas fotovoltaicos y diferentes sensores inteligentes conectados a una red de comunicación para la gestión de energía en edificios inteligentes. Todos los datos deben ser recolectados y procesados para entregar la información relevante al usuario final quien podrá realizar acciones a distancia referente a dicha información.

1.3. Trabajo a desarrollar

- Definir los requerimientos del sistema para edificios inteligentes, estableciendo una priorización de estos requerimientos para el desarrollo de la solución incluyendo los datos de los sensores y los dispositivos de medición.
- Armar la capa de adquisición de la plataforma IoT para edificios inteligentes, la cual

incluye el diseño de un sistema de monitoreo de los dispositivos del edificio, del sistema fotovoltaico y de los diferentes dispositivos IoT y diseñar la red de comunicación cableada e inalámbrica.

- Implementación *back-end* y *front-end* del sistema: la fase *back-end* de la implementación involucra el desarrollo de la red de sensores y dispositivos de medición, la integración de la unidad de procesamiento y las bases de datos. La fase de implementación de *front-end* involucra el desarrollo de una interfaz de usuarios para la interacción, visualización y análisis del sistema.
- Implementación del sistema y validación, en donde el prototipo diseñado debe ser instalado y testeado en el contexto de una aplicación real para evaluar la funcionalidad y usabilidad de la solución

1.4. Evaluaciones a realizar

- Verificación de los requerimientos encontrados en base a los dispositivos que pueden ser implementados en el entorno en estudio.
- Pruebas de comunicación de los dispositivos y el envío de información uno a uno.
- Revisión de los procesos realizados en el *back-end* en base a la información proveniente de los sensores y dispositivos inteligentes.
- Revisión de la visualización de los datos desde el *front-end* de manera clara y ordenada.
- Implementación y pruebas de funcionalidad en 3 lugares característicos dentro del departamento de electrónica de la universidad como lo son una oficina, un laboratorio y una sala de clases

1.5. Resultados esperados

Finalmente, se espera tener un sistema funcional que permita su implementación en al menos 3 dependencias del departamento de electrónica de la universidad técnica Federico Santa María que sea capaz de reunir la información proveniente de los sensores y dispositivos inteligentes implementados además de mostrarla de manera práctica y clara por el operador quién podrá interactuar con el sistema a través de la aplicación. El sistema debe ser capaz de realizar el manejo de energía de los lugares donde se implemente.

Capítulo 2

Estado del Arte

2.1. Temas relevantes a tratar

La crisis energética global hace imprescindible la implementación de distintas tecnologías para que la utilización de la energía se realice de forma óptima [3], y en particular en los edificios se busca mantener esta eficiencia energética sin afectar mayormente la usabilidad y el confort en estos.

Los edificios inteligentes son aquellas construcciones que han sido diseñadas y construidas para ser eficientes, además pueden ser aquellos que utilizan diferentes tecnologías para optimizar sus funciones.

Aunque los edificios inteligentes incluyen tecnología de monitoreo y actuación sobre los dispositivos inteligentes dispuestos en éstos, es importante contar con una plataforma que permita reunir toda la información generada para automatizar los procesos de manera de alcanzar la tan anhelada eficiencia energética. Para lograr que la información sea visible para un operador que puede o no estar presente en el edificio se requiere que los datos que se vayan capturando dentro del edificio se suban a una plataforma que pueda ser accedida desde el exterior, es decir, que los datos se almacenen en la nube. Por otro lado se espera que, en base a dichos datos, se puedan utilizar algoritmos de control para los principales sistemas dentro del edificio.

Una de las tecnologías habilitadora para lograr la utilización de los datos provenientes de

dispositivos y sensores instalados en un edificio inteligente para su posterior visualización, análisis y almacenamiento en la nube es la tecnología del IoT (del inglés “*Internet Of Things*”).

Dado que el IoT es un concepto amplio es que se realizará el desarrollo de la plataforma en base a las capas mencionadas en [4], las cuales son: *capa de poder*, *capa de adquisición de datos*, *capa de comunicación de red* y *capa de aplicación*. El esquema de estas capas se muestra en la figura 2.1. Dicha figura fue tomada de [4]. Por este motivo, el estudio de los trabajos relacionados se analiza en cuanto a que capa se enfoca mayoritariamente.

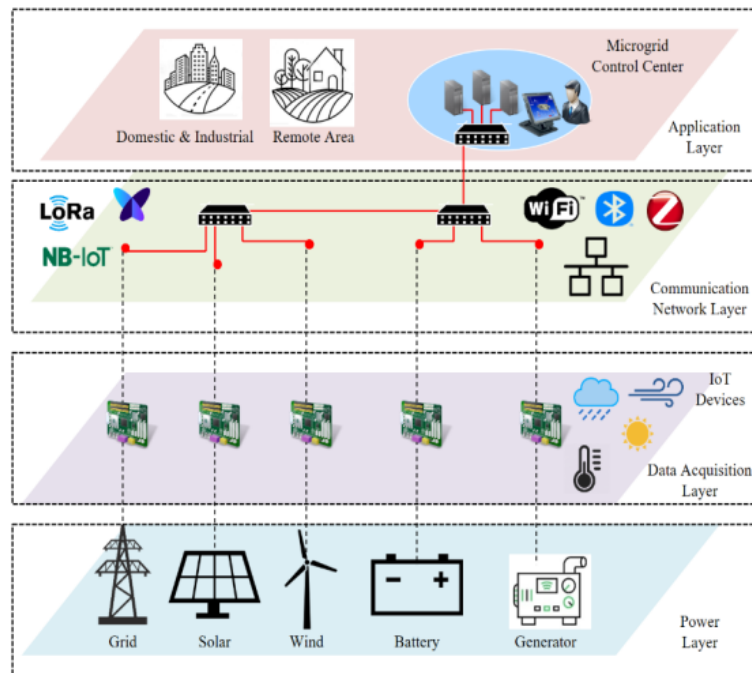


Figura 2.1: Esquema de capas de la tecnología IoT para edificios inteligentes.

2.2. Trabajos relacionados

En [3] se observa el gran potencial del software encargado del manejo de la energía o **BEM** (por sus siglas en inglés *Building Energy Management*) que se ejecuta en la nube ya que mejora aspectos como: escalabilidad, acceso remoto, fácil implementación, recuperación ante

fallas, análisis y almacenamiento de datos y conexión inteligente a la red. Para lograr la conexión entre el software **BEM** instalado en la nube y los sensores y dispositivos inteligentes en el edificio se requiere que cada uno de estos dispositivos tenga una capa adicional que le permita tener una configuración NAT (*Network Address Translation*) que haga alcanzable sus datos desde internet y propone que esta capa faltante sea incluida en un dispositivo nuevo denominado **IoT Gateway** la cual debe permitir la conexión de un gran número de dispositivos y soportar los protocolos más usados en IoT.

En la figura 2.2 se observa el diagrama de la conexión del **BEM** que se ejecuta en la nube y los dispositivos que controla en el edificio con ayuda del **IoT Gateway** propuesto.

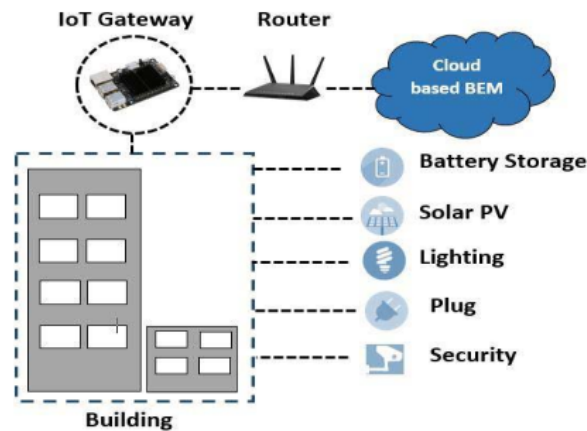


Figura 2.2: Esquema de conexión de un BEM en la nube con los dispositivos en un edificio usando un *Gateway IoT*.

El diseño del software de la **IoT Gateway** propuesta se muestra en la figura 2.3, en esta se ven los protocolos soportados para la conexión con los dispositivos en el edificio.

Este trabajo se enfoca mayoritariamente en las capas de *adquisición de datos* y de *comunicación*.

En [5] se plantea una nueva arquitectura de capas para la comunicación IoT en un edificio, donde se propone agregar una nueva capa al *stack* IoT la cual corresponde a una etapa de “preprocesamiento” y que se denomina *capa niebla* haciendo alusión a la gran capacidad de procesar datos de la nube pero en este caso en a mucho menor escala.

En la figura 2.4, tomada de [5], se observa que, claramente, la capa de niebla se ubica inmediatamente antes del envío de los datos a la nube.

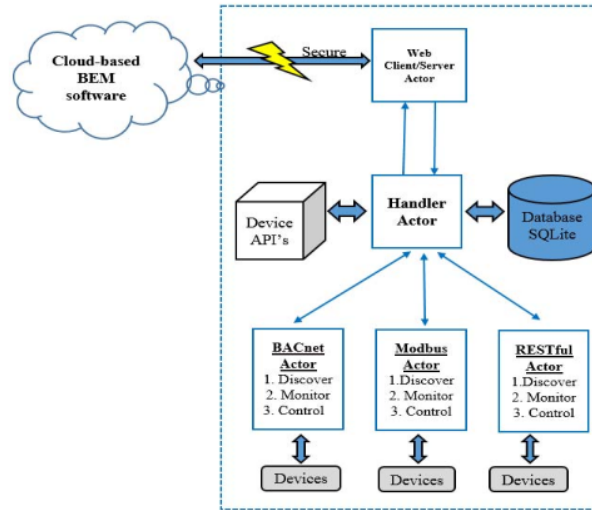


Figura 2.3: Diseño de la *Gateway IoT* desarrollada.

Dado que la *capa niebla* es una capa que maneja los datos se considera que se enfoca en la *capa de aplicación* y de *comunicación* del modelo propuesto en [4].

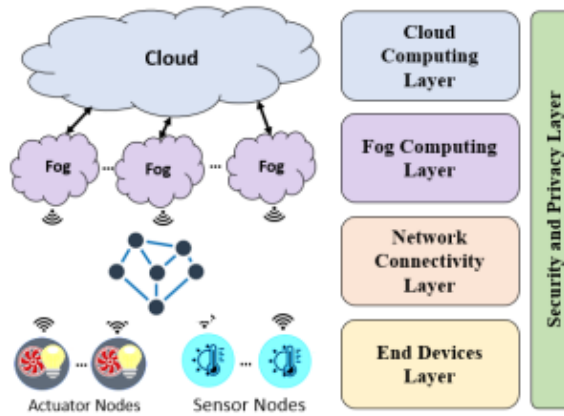


Figura 2.4: Diagrama de Arquitectura IoT propuesta incluyendo *capa Fog*.

En el trabajo desarrollado en [6] se aborda la adopción de tecnología IoT en el desarrollo de edificios inteligentes, se resumen las tecnologías habilitadoras en cada una de las 3 capas convencionales de la arquitectura IoT, a saber, capa de percepción, capa de red y la capa de aplicación como se muestra en la figura 2.5.

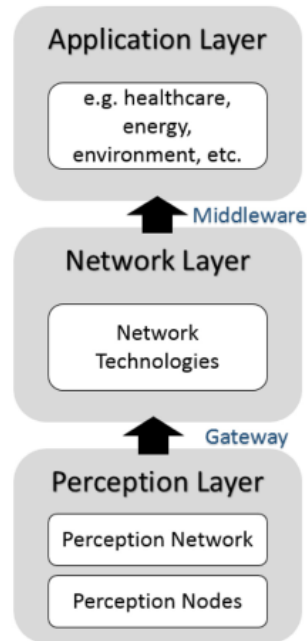


Figura 2.5: Diagrama de Arquitectura IoT de 3 capas planteada para edificios inteligentes

Por otro lado, se explican las diferentes capas de la tecnología IoT enfocada a edificios inteligentes y las tecnologías habilitadoras en cada una de éstas, las cuales se pueden apreciar en la figura 2.6 que muestra en forma esquemática los protocolos usados en edificios inteligentes.

En [6] se hace referencia a las principales tecnologías de *IoT* para edificios inteligentes correspondientes a las 3 primeras capas mencionadas en [4].

En [7] se estudian las aplicaciones básicas usadas en edificios inteligentes. En particular, se estudian aplicaciones para el manejo de imágenes y videos de seguridad, control avanzado de la iluminación, monitoreo del uso del agua, control de los sistemas de aire acondicionado y detección de incendios. En la figura 2.7 se muestra un ejemplo de las funciones que buscan las principales aplicaciones en edificios inteligentes presentada en [7] y en la figura 2.8 se muestran los sistemas a automatizar para el control de la energía presentados en [7].

Este trabajo se enfoca en las *capas de poder y aplicación* del modelo antes mencionado.

Adicionalmente, se estudian trabajos relacionados con la implementación de sistemas de

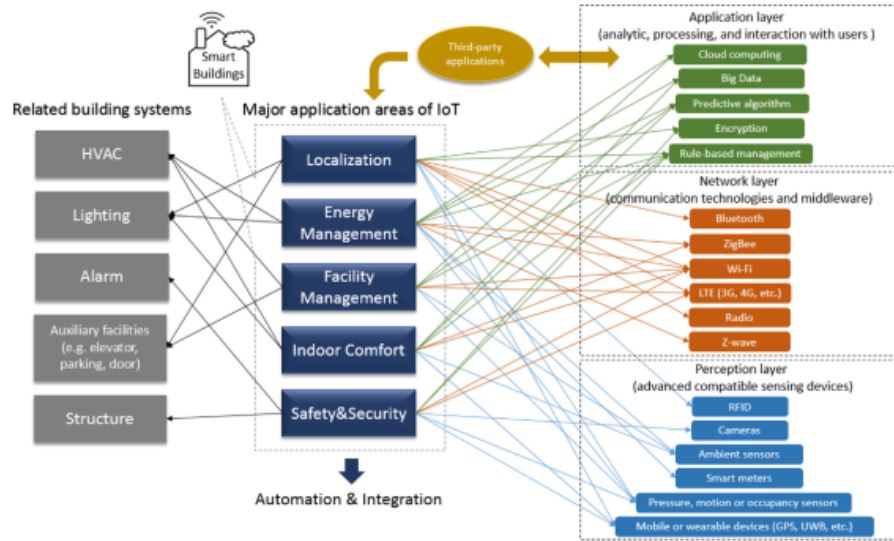


Figura 2.6: Diagrama de las tecnologías utilizadas en implementaciones IoT para edificios inteligentes.

control y utilización de los datos obtenidos por los equipos instalados en un edificio inteligente, lo que coincide con la capa de aplicación por el uso de los datos recibidos y con la capa de poder por los sensores necesarios para su desarrollo, en particular, en [8] se propone un *framework* de aprendizaje del modelo térmico de un edificio inteligente en base a datos históricos obtenidos con termostatos inteligentes dispuestos en el edificio en estudio, mientras que en [9] se realiza el análisis de tecnologías de procesamiento de datos como lo son *Big Data*, *Cloud Computing* y *Monitorización* en base a datos obtenidos por redes de sensores instalados y que utilizan el sistema operativo **Contiki OS** y en [10] se propone un modelo de *Deep Learning* para el control de este tipo de sistemas en base a la estimación del número de personas en una cierta área dentro de un edificio inteligente. En [11] se hace énfasis en el control de las luces dentro de un edificio inteligente y en particular se propone una API para la obtención de datos del sistema de iluminación para el control de éste. En [12] se habla del interés en los edificios inteligentes en la unión europea y propone los principales sistemas y sensores a implementar en un edificio inteligente para mejorar la eficiencia energética, además de proponer estrategias para el control energético en base a los datos recibidos en la nube y, finalmente, en [13] se hace el análisis de la tecnología IoT utilizada en las ciudades inteligentes y se hace una clasificación de las tecnologías integradas, los tipos de redes,

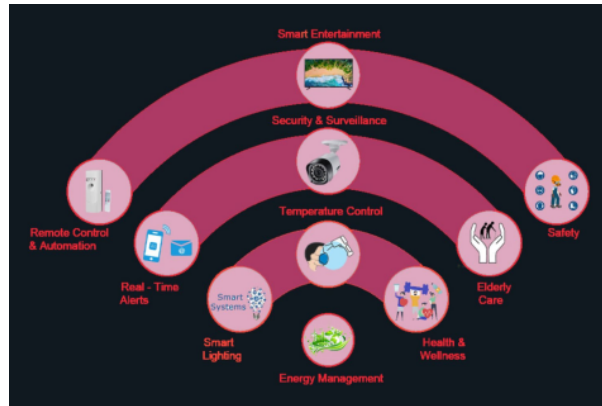


Figura 2.7: Visión general de las variables a considerar en un edificio inteligente.

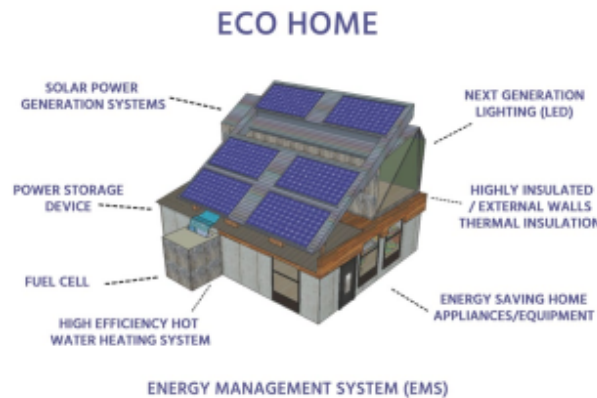


Figura 2.8: Dispositivos utilizados en sistemas de manejo de energía en edificios inteligentes.

posibles oportunidades y requisitos principales para la interconexión de los distintos entes incluidos en la ciudad para que sea considerada inteligente.

El presente proyecto pretende diseñar una plataforma completa debiendo desarrollar todas las capas antes mencionadas con el afán de entregar al operador toda la información necesaria para el monitoreo eficiente de las instalaciones en cuestión, además de almacenar dicha información de manera que una aplicación haga uso de ella para mejorar la eficiencia energética.

Finalmente, se incluye un resumen en la tabla 2.1 donde se comparan trabajos previos en cuanto a arquitecturas IoT empleadas. Como se puede apreciar, el trabajo descrito en esta tesis abarcará 4 capas en su estructura, a diferencia de la mayoría de trabajos estudiados.

Cuadro 2.1: Resumen de las capas mencionadas en los principales proyectos desarrollados en el área.

Referencia	Capas			
	Poder	Adquisición de datos	Red de comunicación	Aplicación
[3]	No	Si	Si	No
[5]	No	No	Si	Si
[6]	Si	Si	Si	No
[7]	Si	No	No	Si
[8]	Si	No	No	Si
[9]	Si	No	No	Si
[10]	No	No	No	Si
[11]	Si	No	No	Si
[12]	Si	No	No	Si
[13]	Si	Si	Si	No
Proyecto Propuesto	Si	Si	Si	Si

Capítulo 3

Propuesta

3.1. Análisis de las alternativas posibles de solución

A continuación se analizan las alternativas de solución separadas por las capas propuestas anteriormente.

3.2. Arquitectura

La arquitectura planteada se muestra en la figura 3.1 y consta de 4 capas de implementación las cuales se describen a continuación:

- Capa de poder: cubre los dispositivos de generación energética y las cargas. Entre la inyección de energía al edificio se encuentra la conexión a la red de distribución energética (provista por Chilquinta), dispositivos como paneles solares, turbinas eólicas y se incluyen las baterías. Mientras que en el lado del consumo se encuentran todos los electrodomésticos del edificio, equipos de ventilación y aire acondicionado, luces, vehículos eléctricos que se carguen conectados a la red del edificio, entre otros.
- Capa de adquisición de datos: es la encargada de captar todos los datos provenientes de

la capa de poder e incluso datos asociados a dichos dispositivos para tomar decisiones con respecto a estos. Ejemplo de esto son sensores de luz, temperatura, medición de energía entre otros.

- Capa de comunicación: En esta capa se definen tecnologías y protocolos de comunicación de los dispositivos de adquisición de datos con la capa de aplicación. Puesto que esta es la capa encargada de recibir los datos de los sensores y enviarlos hasta la capa de aplicación, es que podría necesitar enviar los datos a través de varias redes como pueden ser redes locales (*Local Area Network*), redes del edificio (*Building Area Network*), entre otras. Entre las tecnologías más usadas en la comunicación entre dispositivos IoT se encuentra el WiFi, LoRa, Zigbee, bluetooth y otros, y entre los protocolos de comunicación con la nube se encuentra MQTT, CoAp, Web Socket.
- Capa de aplicación: En esta capa se recibe toda la información de monitoreo y estado de los dispositivos utilizados en la capa de adquisición. Su mayor función es el monitoreo en tiempo real y el control según los diferentes enfoques buscados, como puede ser: manejo energético, seguridad, comodidad de los usuarios, manejo de sistemas de aire acondicionado, calefacción y ventilación. En esta capa también se reconoce el servicio de *middleware* que permite guardar e interactuar con los datos en la nube.

En la figura 3.1 se observa un diagrama de la arquitectura propuesta para el presente trabajo, la cual se plantea en [4] y se modifica para los fines de la implementación de este proyecto.

Se considera que un campus inteligente consiste en un grupo de edificios (como se observa en el esquema mostrado en la figura 3.2), donde cada edificio puede ser visto como una micro red conectada a la red de distribución energética, con sistemas de energía renovable (como el sistema de paneles fotovoltaicos) y dispositivos que consumen energía (cargas).

La implementación de este proyecto se realiza en el campus Valparaíso de la universidad técnica Federico Santa María, específicamente el edificio B y se consideran 3 casos de estudio (una oficina, un laboratorio y una sala de clases).

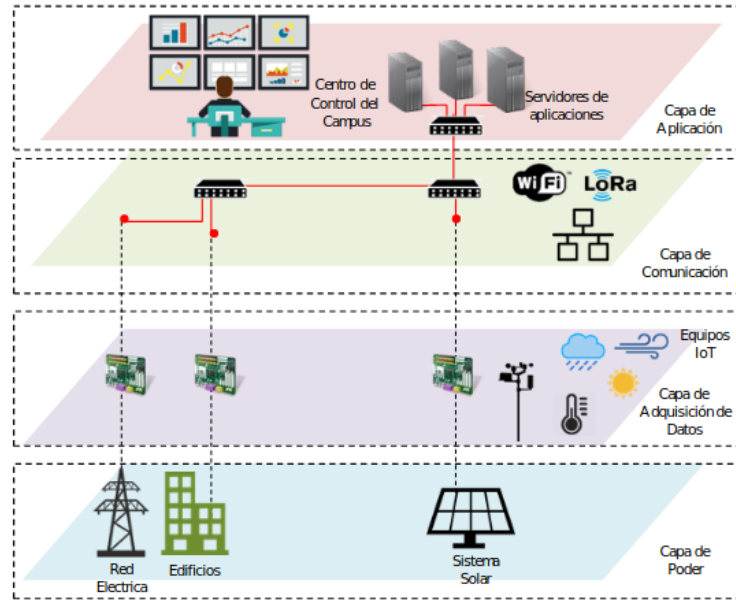


Figura 3.1: Esquema de capas de la tecnología IoT para edificios inteligentes.



Figura 3.2: Esquema de edificios inteligentes formando un campus inteligente

3.2.1. Jerarquía en arquitectura para monitoreo y control

Dentro del sistema de monitoreo y control de la energía se encuentra una clara jerarquía que incluye el sistema de manejo de energía del campus, de un edificio y de las oficinas, laboratorios y salas de clases dentro de este. Esta jerarquía se describe a continuación.

- Sistema de energía en el campus (Campus Energy Management System, CEMS): Este sistema recibe los datos de consumo de energía de cada edificio y estima el consumo y la generación futuros en base a los datos históricos.

- Sistema de manejo de energía de edificios (Building Energy Management System, BEMS): Es el encargado de la recolección de datos de energía y otros datos recolectados por los sensores ubicados en el edificio. En un edificio podemos encontrar oficinas, laboratorios y salas de clases.

En la figura 3.2 se observa de manera esquemática la constitución del campus como un grupo de edificios con sus principales variables energéticas monitorizadas.

3.2.2. Modelamiento de un edificio inteligente

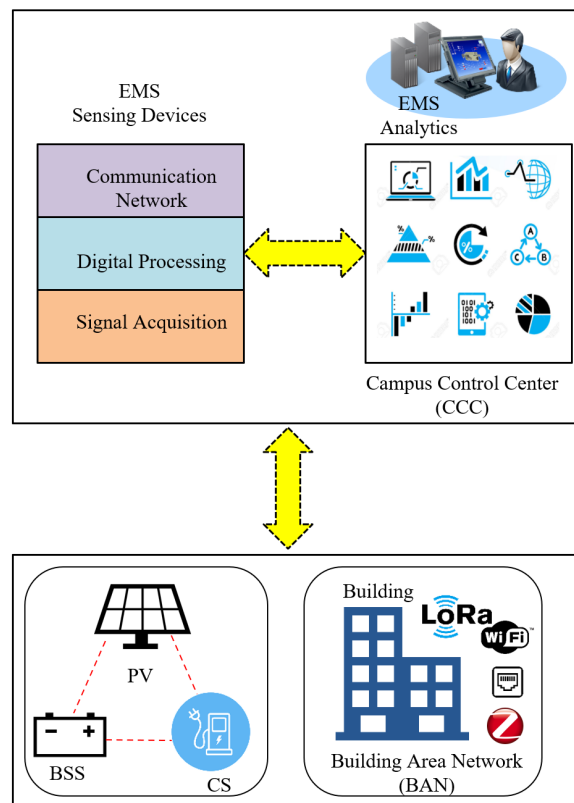


Figura 3.3: Diagrama del manejo de datos para un sistema de manejo de energía en un edificio inteligente.

En la figura 3.3 se observa la utilización de los datos de un sistema de manejo de energía para un edificio inteligente, en esta se observa el trabajo con los datos en 2 partes: Los dispositivos de medición del sistema de manejo de energía (*EMS Sensing Devices*) y las Analíticas

del mismo sistema (*EMS Analytics*). Es precisamente en las analíticas que se provee toda la información al usuario final para la toma de decisiones.

Las soluciones para la implementación de sistemas de manejo de energía en edificios inteligentes son diferentes, pero en términos genéricos se pueden resumir como en la figura 3.4. Cabe mencionar que las cargas pueden ser variadas y pueden ser controladas con los sensores propuestos más adelante.

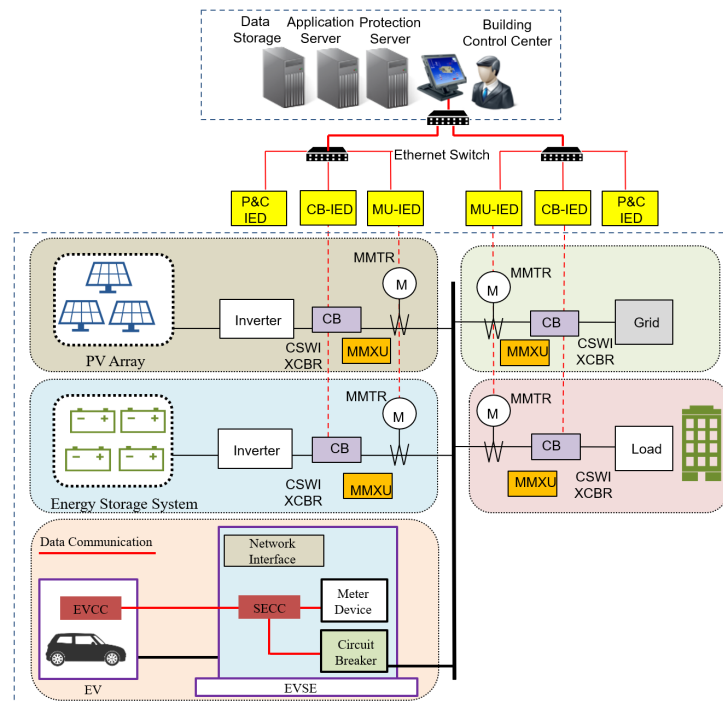


Figura 3.4: Diagrama de la implementación de un sistema de monitoreo de energía en un edificio inteligente.

3.2.3. Capas propuestas

Dado que en todos los trabajos relacionados con la integración de tecnologías *IoT* se trabaja en capas es que se divide el trabajo en estas capas, las cuales, se detallan a continuación.

Capa de poder

En esta capa se encuentran cargas presentes en el edificio como son equipos de iluminación (mayoritariamente luces como tubos fluorescentes y LED), computadores, proyectores, impresoras y enchufes que permiten conectar dispositivos móviles personales.

Por otro lado se considera la red eléctrica provista por la empresa de distribución Chilquinta y una fuente de energía renovable que es un arreglo fotovoltaico.

En particular, la información relevante para el manejo de sistemas fotovoltaicos se muestran



Figura 3.5: Estación meteorológica utilizada en el edificio en estudio (Modelo Davis 6152).

en el cuadro 3.1.

Cuadro 3.1: Dispositivos de medición para sistemas fotovoltaicos.

Equipo	Parte	Dispositivo de medición
Sistema Fotovoltaico	Arreglo de paneles	Voltaje, corriente, potencia, temperatura, ángulo de inclinación y de acimut
	Red eléctrica	Voltaje corriente hacia y desde la red y potencia hacia y desde la red
	Estación meteorológica	radiación solar, temperatura del aire, velocidad del viento, dirección del viento.

Capa de adquisición

Las principales funciones que debe cumplir esta capa son: tomar los datos de la capa de poder y convertirlos digitalmente para poder enviarlos de manera inalámbrica al sistema y recibir datos de actuación desde el sistema para poder ejecutar acciones[5]. Además, para todos los dispositivos planteados se considera su tecnología de comunicación (que sea compatible con el dispositivo de adquisición de datos) y se le da gran importancia a los equipos que sean vendidos en el país puesto que la implementación del presente proyecto se plantea para edificios que ya tendrían equipos inteligentes incluidos y que podrían haber sido adquiridos en el país.

Por otro lado, para definir las tecnologías y dispositivos a utilizar en esta capa es necesario recordar los objetivos buscados, es decir, monitorear los parámetros para la gestión de la energía en un edificio inteligente. Luego, para este fin se requiere monitorear la energía consumida de una habitación completa, tener control **on/off** de los dispositivos conectados a la red, el uso de la iluminación y condiciones ambientales básicas como temperatura y humedad ambiental.

Para monitorear la energía consumida por un circuito completo como puede ser una habitación o un piso en un edificio se utilizan *Smart Meter*, como se muestra en trabajos como en [14], [15] y [6]. Los detalles de los principales equipos de medición de energía con comunicación inalámbrica se muestran en el cuadro 3.2.

Cuadro 3.2: Características de dispositivos de medición de energía.

Modelo	Sonoff POWR3	Vbestlife ZMAi-90	PZEM-004T-100A
Voltaje de entrada/-salida	100 a 240 V	90 a 250 V	80 a 260 V
Corriente Máxima	25 A	60 A	100 A
Dimensiones	162 x 122 x 45,5 mm	80 x 66 x 36 mm	73 x 28 x 12 mm
Precio de Referencia	\$36.215	\$42.214	\$13.153

Para el caso de la medición de datos del sistema fotovoltaico se considera que este tipo de

sistema incluye varios subsistemas interiores, pero para acotar la discusión de estos se considera un **Sistema Fotovoltaico Híbrido** el que puede ser descrito según el diagrama eléctrico mostrado en 3.6, donde los principales objetivos son la medición del bus de corriente alterna (que puede ser medido con el medidor AC propuesto para tableros eléctricos), la radiación solar (que puede ser medida con la estación meteorológica propuesta) y la medición del bus corriente continua. Esta última medición puede ser obtenida con un sensor transformador de corriente y enviada con un equipo que puede ser fácilmente diseñado como prototipo, por lo que se propone el diseño mostrado en 3.7.

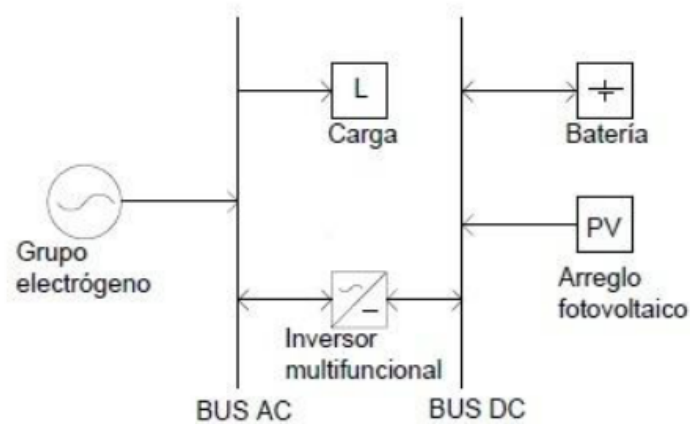


Figura 3.6: Diagrama eléctrico de un sistema solar híbrido.

Otro gran consumo de energía eléctrica en edificios es el proveniente de la iluminación de los espacios y en [14] se usan *smart light switches* (interruptores de luz inteligentes) por lo que el siguiente equipo necesario para la gestión de la energía es el interruptor inteligente y los principales dispositivos encontrados se muestran en el cuadro 3.3. junto con este se propone agregar enchufes inteligentes (o *smart plug*) ya que estos permiten conectar equipos no inteligentes, permitir el encendido/apagado de estos además del monitoreo de consumo energético individual (en el caso de los enchufes inteligentes que poseen esta característica).

Dado que los mayores consumos de energía eléctrica en un edificio provienen de dispositivos de control de temperatura, ventilación y aire acondicionado (**HVAC**, por sus siglas en inglés *Heating – Ventilation - Air conditioning*), en [15] se utilizan sensores de temperatura y

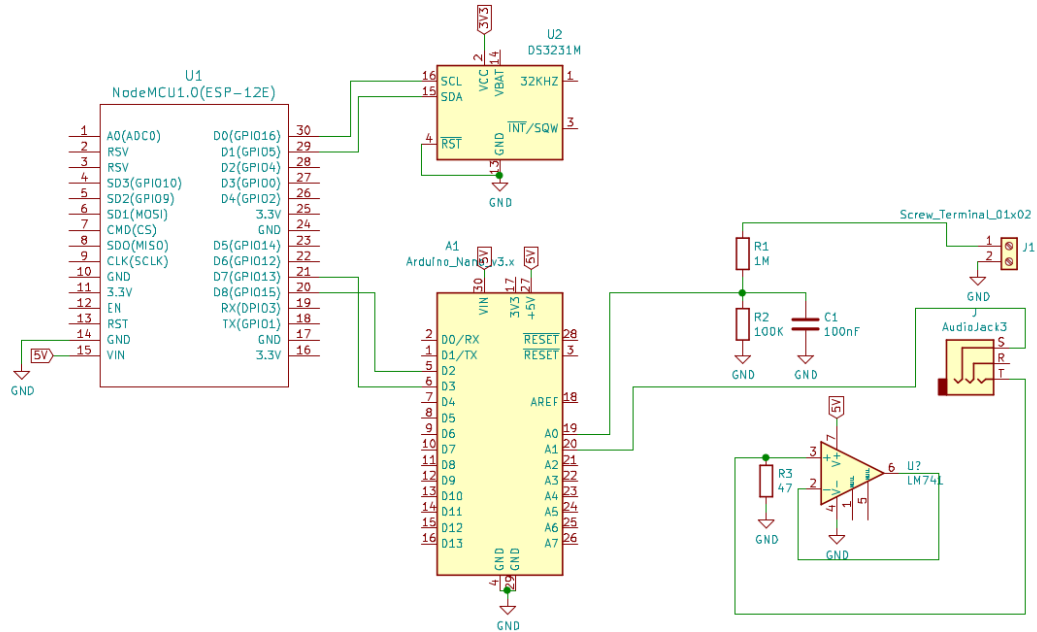


Figura 3.7: Diseño propuesto para la medición del bus de corriente continua en un sistema fotovoltaico híbrido.

humedad. En caso de los trabajos en que se tiene un sistema centralizado de actuación sobre calefactores y aires acondicionado, se utilizan termostatos inteligentes [8].

Puesto que en el edificio donde se llevan a cabo las pruebas del sistema planteado no cuenta con equipos de aire acondicionado, ventilación o calefacción centralizados es que no se usarán termostatos inteligentes y como sustituto se consideran sensores de temperatura y humedad que permitan monitorear este tipo de condiciones para permitir al usuario definir el uso de artefactos no inteligentes para cubrir estas necesidades (los cuales podrían ser aires acondicionados portátiles o estufas). En el cuadro 3.5 se muestran los sensores encontrados con sus principales características.

Otra variable importante a considerar es la ocupación de las habitaciones del edificio y para ello se utiliza una mezcla de la información entre sensores de movimiento y sensores magnéticos ubicados en puertas como se hace en [5]. En los cuadros 3.6 y 3.7 se muestran las principales alternativas propuestas y sus características esenciales.



(a) Sonoff Pow R3



(b) zmai-90



(c) PZEM-004T-100A

Figura 3.8: Equipos propuestos para adquisición de datos desde el tablero eléctrico general.



(a) Interruptor Sonoff
WIFI



(b) Interruptor Sonoff
WIFI RF



(c) Interruptor Arkifi

Figura 3.9: Interruptores inteligentes propuestos.

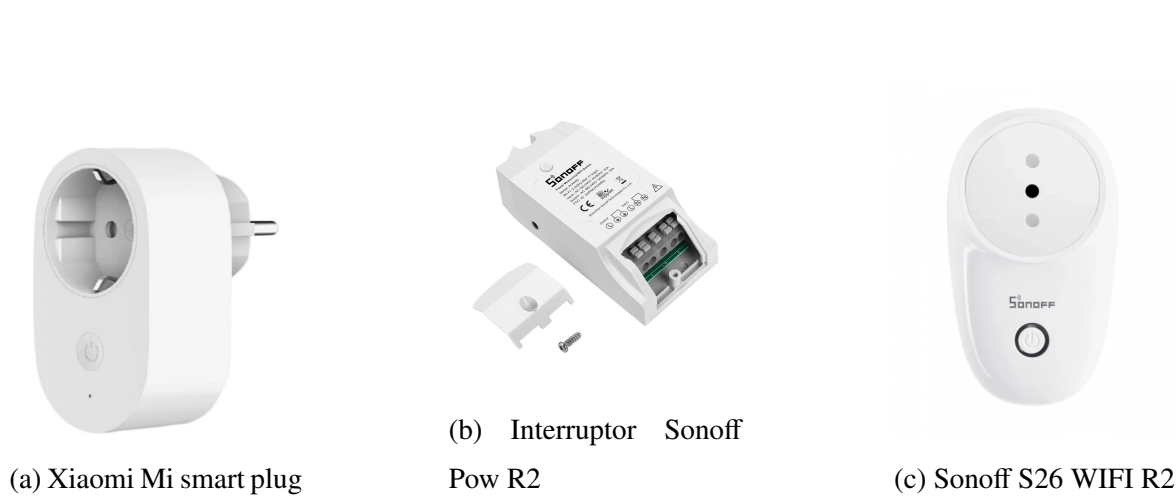


Figura 3.10: Enchufes inteligentes propuestos.

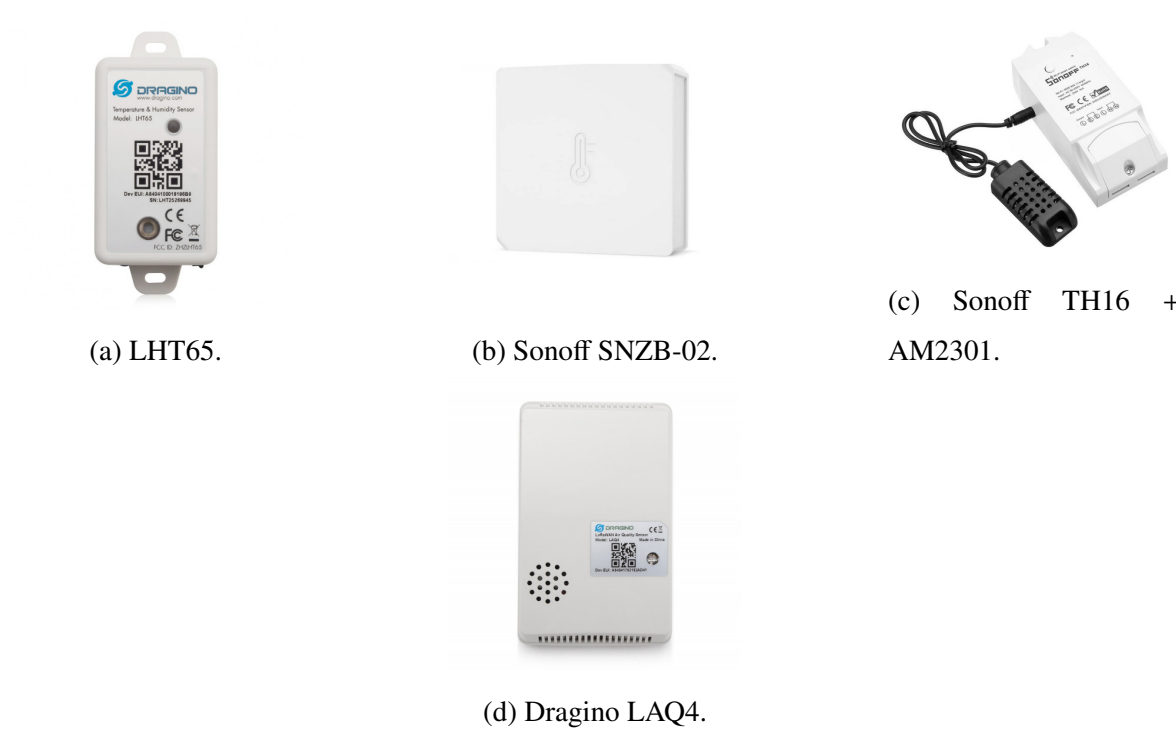


Figura 3.11: Sensores ambientales.

Cuadro 3.3: Características de interruptores inteligentes.

Modelo	Interruptor sonoff WiFi	Interruptor sonoff WiFi RF	Arkifi Interruptor inteligente
Corriente máxima	2 A	2 A	2,5 A
Conectividad	Wifi 2,4 GHz	Wifi 2,4 GHz, RF 433 MHz	Wifi 2,4 GHz
Certificación	CE, RoHS	CE, RoHS	Sin certificación
Precio de Referencia	\$17.790	\$20.990	\$18.900

Cuadro 3.4: Características de enchufes inteligentes.

Modelo	Xiaomi Mi Smart Plug	Sonoff Pow R2	Sonoff S26 WiFi R2
Corriente máxima	16 A	16 A	16 A
Monitoreo de energía	No	Si	No
Conectividad	Wifi 2,4 GHz	Wifi 2,4 GHz	Wifi 2,4 GHz
Precio de Referencia	\$15.990	\$15.690	\$12.590

Capa comunicación

Esta capa es la encargada de comunicarse con los nodos por lo que debe tener la tecnología para ello. A continuación, se muestran las principales tecnologías de comunicación utilizadas por las redes de sensores.

- **WIFI:** Es una tecnología de comunicación ampliamente usada en computadores personales, *smartphones*, *tablets* entre otros dispositivos de uso diario. Cumple con el estándar IEEE 802.11 y las frecuencias más comunes en las que opera es 2.4 GHz y 5 GHz, teniendo un alcance en interiores de 45 m (a 2.4 GHz) y de 15 m (a 5 GHz).
- **Bluetooth:** Tecnología de comunicación inalámbrica de corto alcance (alrededor de



(a) Sonoff DW2 - WIFI.



(b) Vivitar WT06.



(c) Sensor magnético Tuya Smart.



(d) LDS02-AU915.

Figura 3.12: Sensores magnéticos.



(a) Detector PIR Smart Life.



(b) sensor PIR Fersontec.

Figura 3.13: Sensores PIR.

Cuadro 3.5: Características de sensores ambientales.

Modelo	LHT65	Sonoff SNZB-02	Sonoff TH16 + AM2301	Dragino LAQ4
Parámetros medidos	Temperatura y humedad	Temperatura y humedad	Temperatura y humedad	Compuesto orgánico volátil total, CO2 equivalente, Temperatura, Humedad Relativa del Aire
Alimentación	Batería 2400 mAh	CR2450 – 3 V	Conectado a la red	4000mAh Li-SOCI2 (Duración 7 a 10 meses)
Conexión internet	Si	Si	Si	Si
Conectividad	LoRa	Zigbee	Wifi 2,4 GHz	LoRa
Precio de Referencia	\$34.390	\$6.990	\$14.990	\$51.899

10 m). Al igual que la tecnología WIFI, requiere que los nodos se conecten en estrella como se plantea en [6].

- **Zigbee:** Es una tecnología de comunicación inalámbrica que cumple con el estándar IEEE 802.25.4 diseñada para comunicación de corto alcance (propone una distancia entre nodos de entre 10 y 20 m) y trabaja a una frecuencia de 2.4 GHz. Este protocolo permite que la conexión se realice en estrella o en malla y en este caso la comunicación puede recorrer algunos nodos antes de llegar al *gateway* de comunicación a internet como se describe en [6].
- **LTE:** Sus siglas vienen del nombre en inglés *Long-Term Evolution* y fue diseñado para comunicación de alta velocidad basada en las tecnologías inalámbricas GSM/EDGE y UMTS/HSxPA. Este servicio es provisto por las operadoras de telefonía celular.

Cuadro 3.6: Características de sensores magnéticos.

Modelo	Sonoff DW2 - WiFi	Vivitar WT06	“Sensor magnético tuyasmart”	LDS02-AU915
Conectividad	Wifi 2,4 <i>GHz</i>	Wifi 2,4 <i>GHz</i>	Wifi 2,4 <i>GHz</i>	LoRa AU 915 <i>MHz</i>
Alimentación	batería 2 x AAA 1.5V	batería 2 x AAA 1.5V	batería 2 x AAA 1.5V	batería 2 x AAA 1.5V
Precio de Referencia	\$6.290	\$18.990	\$9.990	\$21.980

Cuadro 3.7: Características de sensores PIR.

Modelo	“Detector de movimiento PIR Smart life”	”Fersontec sensor PIR de movimiento”
Conectividad	Wifi 2,4 <i>GHz</i>	Wifi 2,4 <i>GHz</i>
Alimentación	batería 3 x AAA 1.5V	batería 3 x AAA 1.5V
Ángulo de sensibilidad	120°	120°
Alcance	5 <i>m</i>	12 <i>m</i>
Precio de Referencia	\$12.990	\$19.990

- **LoRa:** En un estándar para comunicaciones de largo alcance y bajo consumo. Tiene un alcance máximo de unos 10 *Km*, consume baja energía y permite comunicarse a largas distancias. Permite una velocidad máxima de transmisión de datos de unos 50 *Kbps* según se expone en [6].
- **Otros:** Tecnologías menos usadas para este tipo de aplicaciones son Z-Wave, RFID, WAVE, IrDA y USB, descritas en [6].

En [5] y [15] se usa el protocolo de comunicación WIFI con una topología de estrella para realizar la conexión de los sensores y actuadores.



(a) Dragino LG308.



(b) Laird RG191.



(c) LIG16-915

Figura 3.14: Gateways LoRa.

Un importante dispositivo a considerar es una *Gateway LoRa*, puesto que estos equipos permiten la conexión a *Internet* de los dispositivos que se comuniquen con la tecnología LoRa, tecnología que permite la comunicación de nodos a distancias de unos cientos de metros, característica muy útil en caso de medir parámetros en lugares donde la señal de **WIFI** no es estable. Estos dispositivos se muestran en el cuadro 3.8.

Cuadro 3.8: Características de las Gateway LoRa.

Modelo	Dragino LG308	Laird RG191	LIG16-915
Basado en el chip	SX1308 / SX1276	SX1301 / SX1257	SX1302
Frecuencia LoRa	915 <i>MHz</i>	915 <i>MHz</i>	915 <i>MHz</i>
Conexión Ethernet	Si	Si	Si
Conectividad	Wifi 2,4 <i>GHz</i>	Wifi 2,4 <i>GHz</i>	Wifi 2,4 <i>GHz</i>
Precio de Referencia	\$249.900	\$479.990	\$154.690

También, esta capa es la responsable de la implementación de los protocolos de mensajería de aplicación (*application messaging protocols*) [5] que se describen a continuación:

- **MQTT:** Sus siglas son por su nombre en inglés *Message Queue Telemetry Transport* y su mayor ventaja es su adaptabilidad con varias plataformas para conectar dispositivos a internet [6], también su facilidad de implementación, extrema ligereza y tener un protocolo de ahorro de energía [5].

- **RESTFUL:** Protocolo ampliamente usado por dispositivos inteligentes [3] ya que se basa en HTTP.
- **CoAP:** basado en REST, una funcionalidad en la cima de HTTP. Este protocolo se construyó para dispositivos IoT de recursos limitados que no requieren la complejidad de HTTP para que puedan tener usar las interacciones RESTful [6].
- **Web Socket:** Es un protocolo que permite comunicación vía TCP, permite la arquitectura Req/Res (de petición y respuesta) y Pub/Sub (de publicación y subscripción), además permite usar los protocolos de seguridad TLS y SSL[3].
- **AMPQ, DDS, SMQTT, XMPP.**

En [5] se usa el protocolo MQTT para enviar los datos de los sensores a la red, para ello se usa el *broker Mosquito MQTT* (que es un broker de código abierto) y se ejecuta en una **Raspberry Pi 3B** para comunicarse con los sensores.

En [3] se requiere que el software de BEM (*Building Energy Management*) que se encuentra ubicado en la nube solicite información y esta sea respondida por los dispositivos del edificio y por ende utiliza **Web Socket** para su desarrollo en esta capa. Además este protocolo permite tener una conexión "siempre abierta" entre la *gateway* y el software *BEM* en la nube. Por otro lado, para mantener la comunicación con el software BEM, se requiere que se encapsule el paquete que se encuentra bajo algunos de los protocolos soportados por el BEM (como puede ser BACnet, Modbus o HTTP Restful) y por la arquitectura de la nube en que se trabaja.

Capa Niebla (*Fog Computing Layer*)

Esta capa se presenta con los propósitos de extender las capacidades de la nube de almacenar datos al interior del edificio y acortar los tiempos para tomas de decisiones locales ya que se evita tener que enviar todos los datos a la nube, que sean procesados y recibir retroalimentación desde ésta [5].

En [5] se implementa esta capa en una Raspberry Pi 3B usando el *framework* de código abierto llamado Node-RED (JavaScript) para la interfaz de monitoreo.

La capa niebla debe ser montada en un dispositivo de cómputo cuya tecnología de comunicación sea compatible con los nodos que componen las redes de sensores y actuadores utilizadas. Por ejemplo, en [14] se usa *Raspberry Pi*.

Entre las alternativas que se pueden encontrar en el país se observan:

■ **Raspberri Pi 4B:**

- Procesador: Broadcom BCM2711, Cortex-A72 de cuatro núcleos (ARM v8) SoC de 64 bits a 1,5 GHz
- Memoria: 2 - 4 - 8 GB LPDDR4
- Conectividad: LAN inalámbrica IEEE 802.11b/g/n/ac de 2,4 *GHz* y 5,0 *GHz*, Bluetooth 5,0, BLE, Gigabit Ethernet, 2 x USB 3,0, 2 x USB 2,0
- GPIO: estándar de 40 pines
- Video y sonido: 2 puertos micro HDMI (hasta 4Kp60 admitidos)
- Puertos: DSI para pantalla, CSI para cámara
- Soporte de tarjeta SD
- Potencia de entrada: 5V DC a través del conector USB-C (mínimo 3A)
- Temperatura de funcionamiento 0–50°C.
- Dimensiones: (88 x 58 x 18,5mm)

■ **Odroid-xu4:**

- Procesador: Samsung Exynos5422 de 8 núcleos 2Ghz
- Gráficos: Mali-T628 MP6
- Memoria: 2GB DD3
- Almacenamiento: Flash integrado en la palca
- Puertos: 2 USB 3.0 / 1 USB 2.0
- Tarjeta de red: Gigabit Ethernet 10/100/1000



(a) Raspberry Pi 4B.



(b) Odroid-xu4.



(c) Jetson Nano NVIDIA.

Figura 3.15: Dispositivo de cómputo.

- Salida de Video: HDMI

■ jetson Nano NVIDIA:

- Módulo Jetson Nano Modelo B01
- GPU: NVIDIA Maxwell TM de 128 núcleos
- CPU: ARM® A57 de cuatro núcleos
- Memoria RAM: 4 GB de LPDDR4 de 64 bits
- Almacenamiento microSD
- Video Encode: 4K @ 30 — 4x 1080p @ 30 — 9x 720p @ 30 (H.264/H.265)
- USB: 4x USB 3.0, USB 2.0 Micro-B
- HDMI / DisplayPort
- Conectividad: Gigabit Ethernet 10/100/1000
- Dos conectores de cámara MIPI-CSI

Capa de Aplicación

Dentro de esta capa se encuentra el *middleware* que es el sistema que se encarga de almacenar y procesar los datos en la nube. Las principales plataformas de *middleware* del mercado son:

- **Amazon Web Services:** Es la Plataforma Cloud más grande del mercado y que permite la comunicación por protocolos MQTT y HTTP(S), además de permitir lenguajes de programación como Java, JS, C++, Python y C Embebido [16]. Dentro de sus mayores beneficios se encuentra la escalabilidad y la flexibilidad en su uso [17].
- **Microsoft Azure:** Plataforma que, al igual que la anterior, permite la comunicación por protocolos MQTT y HTTP(S) y admite los lenguajes .NET, C, Java, NodeJS y Python [16] . Permite el uso híbrido, es decir, en un servidor local o en *multicloud* [18].
- **Google Cloud Platform:** Admite la comunicación por protocolos MQTT y HTTP(S) de comunicación, pero en este caso permite programarse en C#, Java, NodeJS, Go, PHP, Python y Ruby [16] [19].
- **Digital Ocean:** Permite la utilización de una maquina virtual que puede servir como servidor para alojar los sistemas necesarios para la implementación requerida. Los recursos contratados son fácilmente escalables pudiendo contratar más recursos y los costos asociados son fijo y dependientes de los recursos contratados [20].

Cabe destacar que las 3 primeras plataformas poseen sistemas de manejo de dispositivos, almacenamiento habitual y almacenamiento frío, protocolos de seguridad para los datos que almacenan y integración con sus propios servicios de computo. Pero además, al igual que la ultima opción, poseen servicios de equipos virtuales de computo (maquinas virtuales) con una ip fija, donde se puede instalar todo el software necesario para la ejecución de un proyecto.

Dentro de esta capa también se encuentran las aplicaciones que utilizan los datos tanto para tomar decisiones como para mostrar dichos datos de una manera ordenada. Los datos obtenidos dependen de los sensores, actuadores y dispositivos conectados según el sistema a automatizar, en particular, los principales sistemas a automatizar en edificios inteligentes son[7]:

- **Control de Seguridad**



(a) Amazon Web Services.



(b) Microsoft Azure.



(c) Google Cloud Platform.



(d) Digital Ocean.

Figura 3.16: *Middlewares* propuestos para mantener la información en la nube

- **Manejo de Energía**
- **Monitoreo y Control de sistemas de climatización**
- **Manejo de Agua**
- **Sistema avanzado de iluminación**
- **Detección de Incendios**
- **Sistemas de Salud y cuidado de ancianos**

Esta capa puede contener interfaces de aplicación de programas (*application program interfaces, APIs*) que permiten que los datos sean alcanzados por cualquier dispositivo con conexión de red autorizado [5]. Pero principalmente se propone definir el *dashboard* para la correcta visualización de datos.

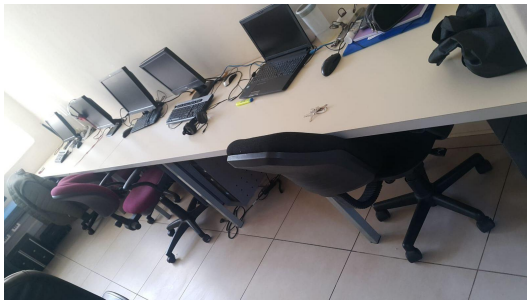
También en esta capa se pueden definir los protocolos de seguridad con que se manejarán los datos en la nube.



(a) Vista aérea del edificio B, para donde se planifica la implementación.



(b) Oficina B-349



(c) Laboratorio B-110



(d) Sala de clases B-213

Figura 3.17: Mapa de edificios en el campus Valparaíso de la UTFSM y escenarios planteados para la implementación.

3.3. Escenarios de implementación

A continuación, se presentan los escenarios a implementar la solución planteada, para ello se describen las necesidades de cada uno de los espacios

3.3.1. Oficina

Se trata de la oficina B-349 del profesor Mohamed Abdelhamid, la cual es representativa de otras oficinas distribuidas en el campus y cuyos mayores consumos incluyen un computador con uno o más monitores, una impresora, enchufes para cargar algunos dispositivos móviles personales y cuenta con iluminación fija accionada por un interruptor en la pared.

Por tratarse de un espacio relativamente pequeño no se considera la medición de energía de

la sala completa pero si la utilización de dispositivos como interruptores inteligentes (para controlar la iluminación), un enchufe inteligente por cada equipo conectado, un sensor de detección de movimiento y uno magnético para la puerta además de un sensor de temperatura. Para este escenario no se considerará la obtención de datos de los paneles solares o de las baterías puesto que estos datos solo se considerarán en uno de los escenarios para no tener información redundante.

3.3.2. Laboratorio

El segundo escenario es el laboratorio B-110 del departamento de electrónica. En este caso se cuenta con al menos 4 puestos de trabajo fijos donde se puede conectar un computador y un monitor externo por cada uno de estos puestos además se cuenta con el espacio para realizar diferentes proyectos por lo que en este caso se plantea la implementación de enchufes inteligentes para el monitoreo de computadores y otros equipo conectados a la red, la instalación de un interruptor inteligente para el control de las luces y la instalación de un sensor de movimiento y uno magnético de puerta para a futuro estimar la ocupación del laboratorio. Por otro lado dada la requerimientos energéticos del espacio se propone instalar el medidor energético propuesto en el tablero eléctrico para el monitoreo total de la energía en la habitación.

3.3.3. Sala de clases

El tercer escenario planteado es de una sala de clases (en este caso se utilizará la sala B-213 del departamento de electrónica) la cual cuenta con luminarias que serán controladas con interruptores inteligentes, uno o más proyectores que serán controlados con enchufes inteligentes y varios enchufes que permiten a alumnos y profesores conectar sus dispositivos personales. Es por este último punto que se utilizará un medidor energético en el tablero

eléctrico. Además se instalará el dispositivo medidor de parámetros ambientales para monitorear la calidad del aire al momento de la realización de las clases.

3.4. Alternativas seleccionadas en dispositivos, tecnologías y servicios

A continuación, se describen las alternativas seleccionadas de cada tipo de dispositivo o tecnología propuesta en donde se describen los factores tomados en cuenta para su elección. Para ello se hace nuevamente la división en capas de los dispositivos y tecnología que componen la solución del problema propuesto.

3.4.1. Capa de poder

En este caso se describen los principales equipos a monitorear y los cuales serán separados por los diferentes escenarios en estudio. Cabe destacar que los equipos pueden ser separados en *cargas fijas* que son equipos que permanecen fijos conectados a la red, el circuito de iluminación, enchufes de propósitos generales (donde se conecta cualquier equipo necesario en el uso diario, pero pueden ser desconectados para usar otros) y los tableros eléctricos instalados en algunos escenarios.

- Oficina: Aquí se encuentran cargas fijas (como computador, monitor, impresora), enchufes de uso general y el circuito de iluminación.
- Sala de Clases: Aquí se encuentra el proyector como carga fija, enchufes de uso general, circuito de iluminación y tablero general de la sala.
- Laboratorio: Donde las cargas fijas son en general computadores, enchufes de propósito general, el tablero eléctrico, el circuito de iluminación y el sistema de paneles solares.



(a) Raspberry Pi 4B



(b) Medidor de energía pzem-004t-100A



(c) Enchufe inteligente Sonoff Pow R2.



(d) Sensor de calidad de aire LAQ4.

Figura 3.18: Equipos seleccionados para la adquisición de datos.

3.4.2. Capa de adquisición

Capa que se encarga de la obtención de datos provenientes de la capa de poder y donde se propone la mayor adquisición de equipos, los cuales se describen a continuación.

Enchufe inteligente

En el cuadro 3.9 se observan las características de los principales enchufes inteligentes y debido a su gran similitud entre cada uno es que se elige el único que permite la medición de energía del dispositivo conectado a este por lo que se elige el **Sonoff Pow R2**, que se muestra en la figura 3.18c.

Por otro lado, de las alternativas mencionadas para la interacción con los circuitos de iluminación, se verifica que ninguno de los equipos planteados permite el monitoreo energético, por lo que se decide utilizar este mismo enchufe inteligente, pero conectado en serie al circuito de iluminación.

Cuadro 3.9: Características de enchufes inteligentes.

Modelo	Plug Xiaomi Mi Smart Plug	Sonoff Pow R2	Sonoff S26 WiFi R2
Corriente máxima	16 A	16 A	16 A
Monitoreo de energía	No	Si	No
Conectividad	Wifi 2,4 GHz	Wifi 2,4 GHz	Wifi 2,4 GHz
Precio de Referencia	\$15.990	\$15.690	\$12.590

Dispositivo de medición de energía en corriente alterna

Para este caso, se consideran las características de los equipos destinados a medir el consumo eléctrico desde el tablero eléctrico asignándole un porcentaje proporcional a la corriente máxima soportada, se le asigna un porcentaje dependiendo de lo invasivo de la medición y se agrega el tipo de conexión para la obtención de los datos. Estos datos se muestran en el cuadro 3.10. Con estos datos se elige directamente el equipo **pzem-004t-100a** que permite

Cuadro 3.10: Evaluación de las características de dispositivos de medición de energía.

Modelo	Sonoff POWR3	Vbestlife ZMAi-90	pzem-004t-100A
Corriente Máxima	25 %	60 %	100 %
Invasivo	100 %	100 %	50 %
Conexión envío de datos	Tasmota	Tasmota	Serial

su fácil incorporación en el tablero eléctrico a realizar la medición, permitiendo colocarse en tableros eléctricos de gran potencia de la manera menos invasiva posible.

Sensores de temperatura y humedad, de presencia y magnético para puerta

En el caso del dispositivo de medición de temperatura y humedad se observan grandes similitudes en las características de todas las alternativas propuestas pero en este caso el dispositivo de menor valor debe considerar la adquisición de un dispositivo adicional que permita la comunicación usando la tecnología *zigbee* por lo que se elige el equipo que lee la mayor cantidad de variables, es decir, el **Dragino LAQ4** cuya imagen se muestra en la figura 3.18d.

Para el caso del sensor magnético y el sensor PIR, se decide no implementarlos en el presente trabajo ya que no involucra directamente el monitoreo energético del edificio.

3.4.3. Capa de comunicación

Las tecnologías de comunicación cableadas e inalámbricas son muy importantes en la implementación de una arquitectura IoT porque manejan el intercambio de toda la información del sistema.

Para el caso de la tecnología de comunicación se decide usar la red WiFi ya que dentro del campus se tiene una buena señal en casi todo el campus, además para la comunicación a larga distancia se utilizará la tecnología LoRa y se definirá el protocolo de comunicación con la Nube.

Gateway LoRa

En el cuadro 3.11 se muestran algunas alternativas de *gateways* que permitan obtener los datos de los sensores LoRa, donde se observan diferencias menores entre cada uno de los dispositivos y dado que en este caso se tiene a disposición la *gateway* LORAWAN LG308 es que se utiliza ésta.

Cuadro 3.11: Características de las Gateway LoRa.

Modelo	Gateway Lorawan LG308	Laird RG191	LIG16-915
Basado en	SX1308 / SX1276	SX1301 / SX1257	SX1302
Frecuencia LoRa	915 MHz	915 MHz	915 MHz
Conexión Ethernet	Si	Si	Si
Conectividad	Wifi 2,4 GHz	Wifi 2,4 GHz	Wifi 2,4 GHz
Precio de Referencia	\$249.900	\$479.990	\$154.690



Figura 3.19: Alternativa seleccionada para el gateway LoRa

Protocolo de comunicación con la nube

En el cuadro 3.13 se presentan las principales características de los protocolos de comunicación utilizados para comunicación con la nube.

Se asigna un porcentaje de 100 % si el protocolo permite una conexión persistente (para asegurar, en caso de ser necesario, un flujo de datos constante), 100 % en caso de que admita asegurar la calidad de servicio (en caso necesitar procesar datos directamente desde al nube) y 100 % si posee una arquitectura de publicación/subscripción (para permitir enviar los datos a penas están disponibles a la nube). Los resultados de esta asignación se muestran en el cuadro 3.13.

Cuadro 3.12: Principales características de los protocolos de comunicación con la nube.

Protocolo	¿conexión persistente?	protocolo de trans- porte	Calidad de servicio (QoS)	Arquitectura	Seguridad
MQTT	Si	TCP	Si	Pub/Sub	TLS/SSL
CoAP	No	UDP	Si	Req/Res	DTLS
Web Socket	Si	TCP	No	Pub/Sub - Req/Res	TLS/SSL
RESTFUL	No	HTTP	No	Req/Res	HTTPS

Cuadro 3.13: Comparación porcentual de los principales protocolos de comunicación con la nube.

Protocolo	¿conexión persistente?	Calidad de servicio (QoS)	Arquitectura	Total
MQTT	100 %	100 %	100 %	300 %
CoAP	0 %	100 %	0 %	100 %
Web Socket	100 %	0 %	100 %	200 %
RESTFUL	0 %	0 %	0 %	0 %

Con los resultados expuestos en el cuadro 3.13 se obtiene que el protocolo usado para comunicar los datos locales con la nube es el protocolo **MQTT**.

3.4.4. Capa Niebla

Definición del dispositivo de cómputo local

Las alternativas propuestas para el dispositivo de computo local con sus principales características de *hardware* se presentan de forma resumida en el cuadro 3.14.

Cuadro 3.14: Comparación de las principales características de los dispositivos de computo propuestos para este proyecto.

Dispositivo	Núcleos	RAM Gb	Almacenamiento	Conectividad	Precio
Raspberry Pi 4B	4	4 (hasta 8)	SD	Ethernet Gigabit, WIFI, Bluetooth 5.0 BLE	\$62.990
Odroid-xu4	8	2	Flash integrado a la placa	Ethernet Gigabit	\$105.990
Jetson Nano Nvidia	4	4	microSD	Ethernet Gigabit	\$157.990

Se le asigna un 100 % en caso de que el dispositivo incluya conexión inalámbrica además se establece un porcentaje en base al dispositivo más caro de manera de poder ponderar el valor de los 3 y el resultado de estas asignaciones se muestran en el cuadro 3.15. Luego,

Cuadro 3.15: Comparación porcentual de los dispositivos de computo propuestos para este proyecto

Dispositivo	Conectividad	Precio
Raspberry Pi 4B	100 %	39,87 %
Odroid-xu4	0 %	67,09 %
Jetson Nano Nvidia	0 %	100 %

considerando el mayor porcentaje en el ítem de conectividad y el mayor porcentaje relativo al dispositivo más caro, se elige la **Raspberry Pi 4B** como dispositivo de cómputo local, dicho dispositivo se muestra en la figura 3.18a.

3.4.5. Capa de aplicación

En esta capa se plantea el objetivo a perseguir con el uso de los datos recibidos de la capa de adquisición y para este caso se realizará el monitoreo de los datos obtenidos y se almacenará y dispondrá los datos para futuras aplicaciones de control.

Servicio de computación en la nube (*middleware*)

Para esta sección se elige entre los principales proveedores de estos servicios de los cuales se observa que sus servicios son similares en casi todos los servicios que proveen, siendo líderes en el mercado los 3 primeros proveedores pero por la dificultad para calcular los costos asociados al servicio (el cobro se realiza por hora de uso y depende de las capacidades contratadas), en particular cuando se inicia el proyecto y los requerimientos pueden ir cambiando es que se decide utilizar el servicio de maquinas virtuales de **Digital Ocean**.

Capítulo 4

Implementación

La implementación se lleva a cabo considerando las capas planteadas anteriormente que, para el caso de la presente sección serán descritas desde la capa inferior a la superior.

Los diferentes equipos y tecnologías usadas en cada capa se describen de la siguiente forma

- Capa de Adquisición
 - *Smart Plug*.
 - Sensor de calidad de aire.
 - Medidor de energía total.
 - Modulo de obtención de datos de sistema. fotovoltaico.
 - Módulo de obtención de datos de estación meteorológica.
- Capa Niebla
 - Raspberry Pi
 - Node Red
- Capa de Comunicación
 - MQTT
 - LORA

- Capa de Aplicación:
 - Servidor en Digital Ocean
 - Node-red
 - MySQL

En particular, se puede observar una visión general del trabajo en desarrollo en la figura 4.1. Por otro lado, aunque el sistema se plantea y diseña para los 3 escenarios propuestos, por simplicidad se organiza y se arma en un banco de pruebas el cual se muestra en la figura 4.2 donde se puede observar que se alimenta desde el exterior y se conecta directamente a un pequeño tablero eléctrico, el cual se deja abierto para la facilidad de uso. En el tablero se encuentra una luz piloto para verificar que se tenga energía a la entrada, luego un interruptor termomagnético para protección de un sobreconsumo de corriente y una barra tetrapolar para facilitar la conexión de los equipos en el banco. Se incluyen además 2 circuitos: el de iluminación y el de enchufes. Y se incluyen en el banco un *router* pequeño que provee de una red wifi local para los equipos, la *gateway* LoRa y la Raspberry Pi que implementa la capa niebla propuesta. La configuración de estos equipos se trata en el presente capítulo.

4.1. Capa de Adquisición de Datos

4.1.1. Monitoreo energético de equipos conectados a la red

Este monitoreo se realiza con enchufes inteligentes. En particular, se utiliza el **Sonoff POW R2**.

Este dispositivo viene con un firmware que permite su utilización a través de la aplicación entregada por el fabricante **eWeLink**, pero dado que esta aplicación funciona como un sistema cerrado de información, se decide hacer un cambio de *firmware* de manera de poder acceder a los datos capturados por el equipo.

Se reconoce que este dispositivo utiliza el SoC ESP8266 como corazón del procesamiento por lo que existe varias alternativas de *firmware* a utilizar, pero para el presente proyecto se

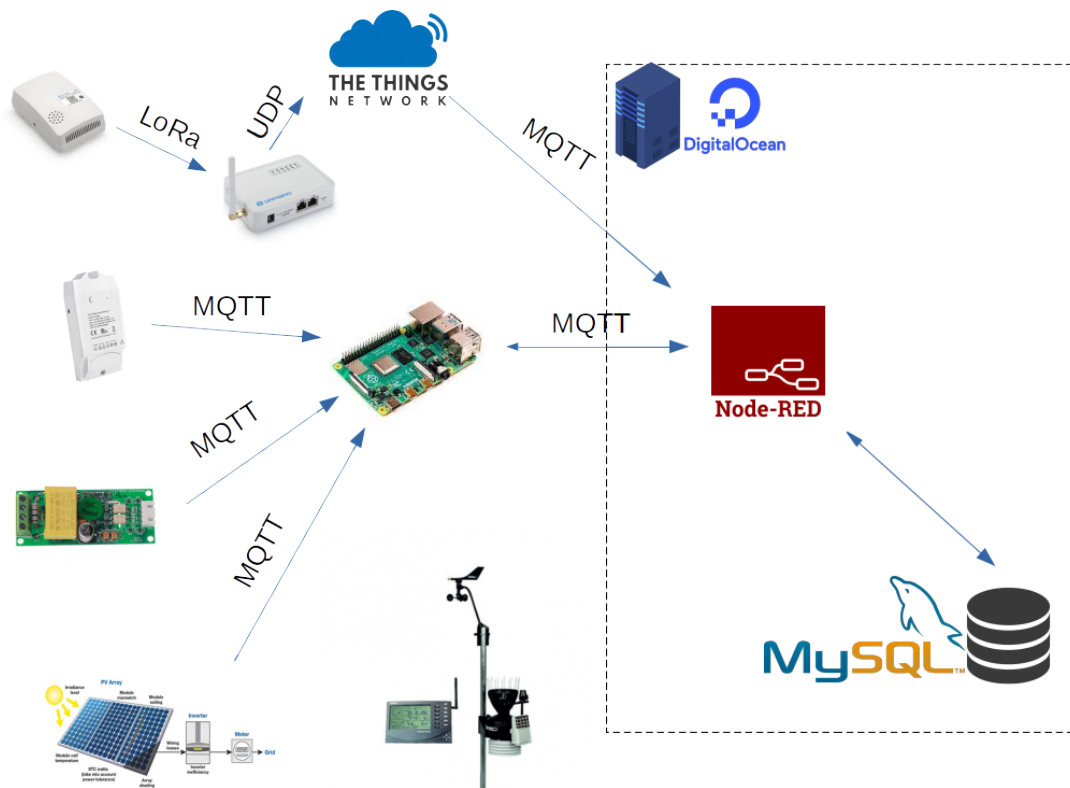


Figura 4.1: Diagrama de implementación de dispositivos de medición conectados al servidor planteado

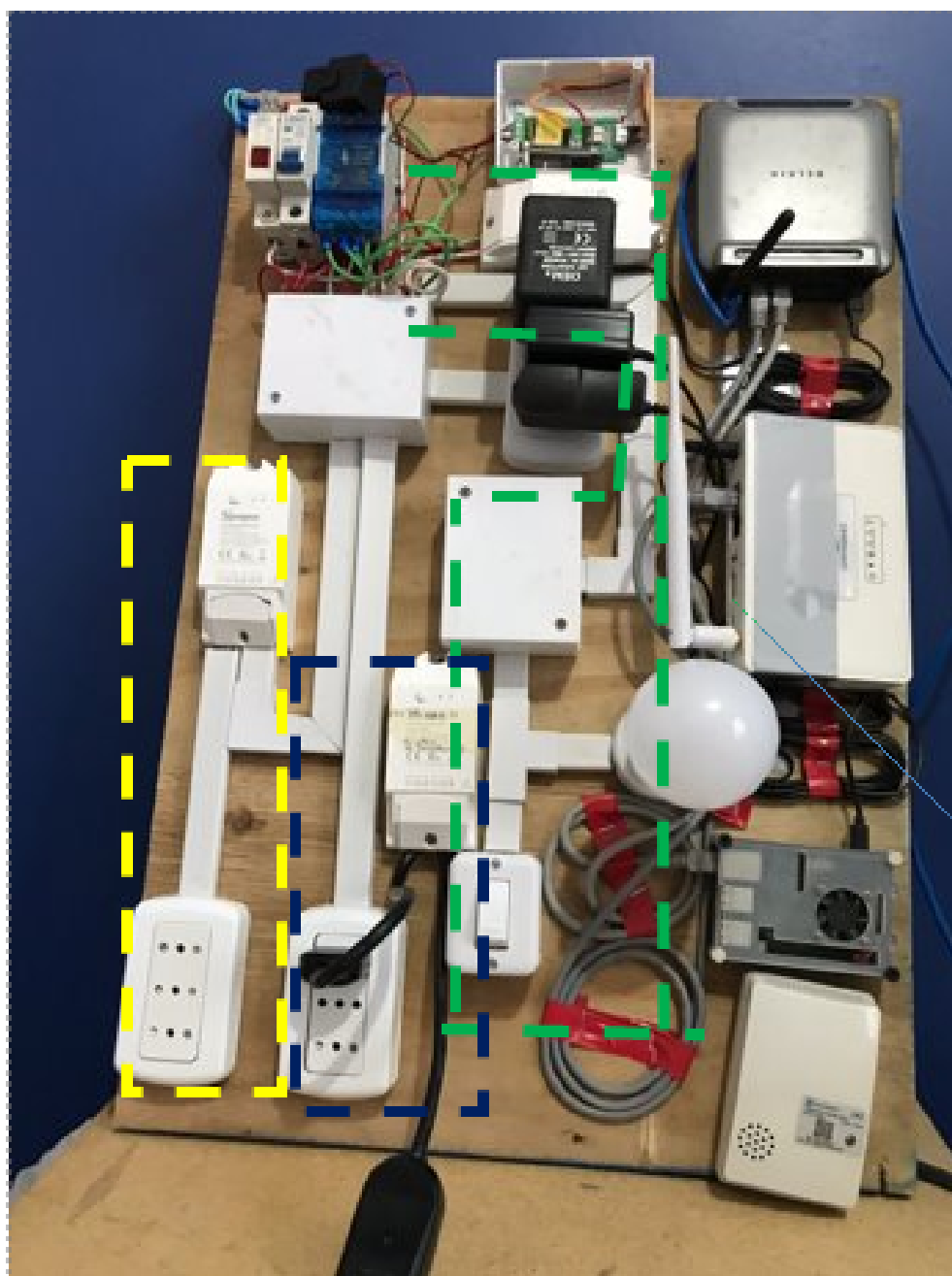


Figura 4.2: Banco de pruebas armado donde se ha segmentado con colores diferentes partes de la implementación.

utiliza el *firmware* de código abierto **ESPurna** (diseñado principalmente por Xose Pérez) debido a las bastas opciones de configuración que posee.

A continuación, se describe el procedimiento realizado para el cambio de firmware.

En primer lugar, se debe instalar **esptool**, herramienta que permite la programación de los dispositivos diseñados por **Espressif** y en particular el ESP8266, esto se hace con el comando:

```
$> sudo apt-get update
$> sudo apt-get install esptool
```

Luego, se ingresa al repositorio de github del proyecto **ESPurna** <https://github.com/xoseperez/espurna> donde se accede a la sección de *releases* y se descarga el archivo **.bin** para el equipo buscado, que, para este caso es **espurna-1.14.1-itead-sonoff-pow-r2.bin** de la *release* 1.14.1.

Luego, para continuar, se debe preparar la conexión entre el computador y el dispositivo, la cual se debe realizar a través de una conexión serial. Para ello se abre el dispositivo, se ubican los pines que permiten la conexión UART con un dispositivo externo y se sueldan pines sobre éstos de manera de permitir una conexión estable para realizar la carga del firmware.

Posteriormente, se conecta un dispositivo USB-TTL a los pines TX, RX, GND y VCC (3,3 V), esta conexión se realiza con ayuda de “jumpers de conexión” y luego se conecta al computador con la ayuda de una interfaz USB-TTL (Esta conexión se muestra en la figura 4.3).

Es importante recalcar que el dispositivo debe estar desconectado de la energía eléctrica en el momento de la programación, es decir, solo debe estar conectado a través de la interfaz TTL al computador.

Una vez realizada la conexión para la carga del *firmware* del dispositivo, se debe poner este en modo programación lo que se hace de la siguiente forma:

1. Se desconecta el *jumper* de alimentación (VCC, 3,3[V]).
2. Se mantiene presionado el pulsador del dispositivo.
3. Se conecta nuevamente a la alimentación.

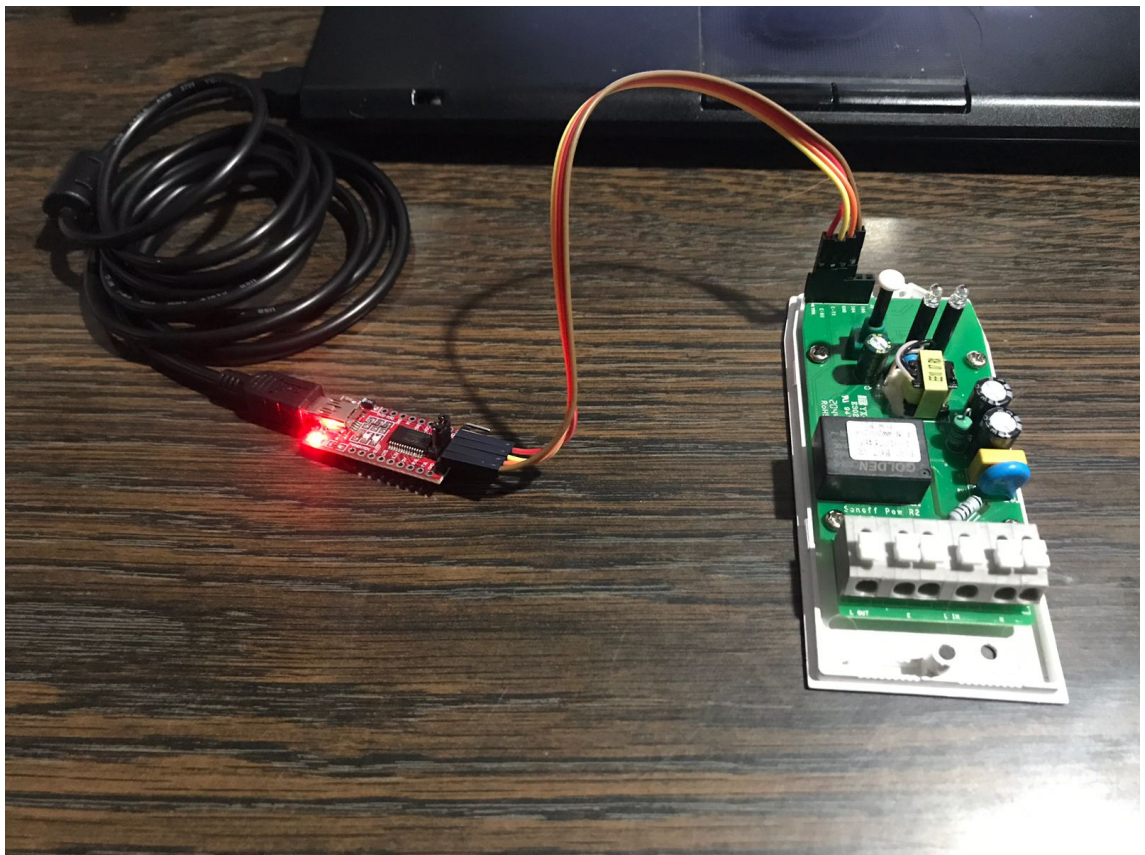


Figura 4.3: Imagen de la conexión para configuración de dispositivo Sonoff Pow R2.

4. Se suelta el pulsador.

Finalmente, se completa el proceso de cambio de firmware, abriendo la consola, ubicándose en el directorio donde se guardó el *firmware* en formato **.bin** e ingresando el siguiente comando:

```
$> esptool.py --port "/dev/ttyUSB0"
--baud 115200 read_flash 0x000000 0x100000
sonoff-backup.bin
```

Que permite realizar una copia del firmware anterior, en este caso, el de fábrica. Luego, se debe borrar la memoria flash del dispositivo con:

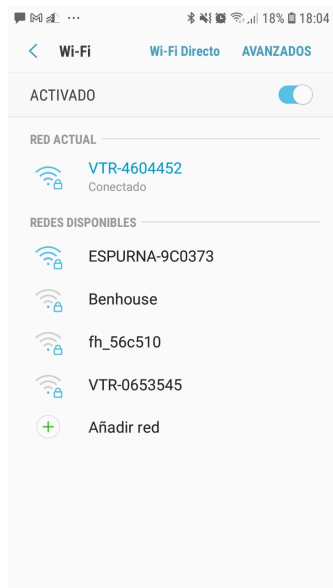
```
$> esptool.py --port "/dev/ttyUSB0"
--baud 115200 erase_flash
```

y por último, se carga el *firmware* con:

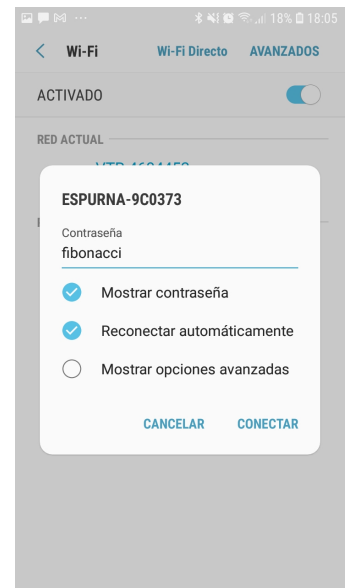
```
$> esptool.py --port "/dev/ttyUSB0"
--baud 115200 write_flash -fs detect -fm dout 0x000000
espurna-1.14.1-itead-sonoff-pow-r2.bin
```

Donde, `/dev/ttyUSB0` es el puerto en que se encuentra conectado el dispositivo USB-TTL. Con todos estos pasos realizados se da por concluida la carga del *firmware* **ESPurna** en el **Sonoff Pow R2**, lo que se puede visualizar verificando las redes **WIFI** disponibles en el entorno cercano al equipo, puesto que al partir genera su red **WIFI** llamada **ITEADXXXXXX** (donde las equis representan el id del dispositivo) y al conectarse a esta red se pueden realizar la configuración del dispositivo para realizar el envío de datos.

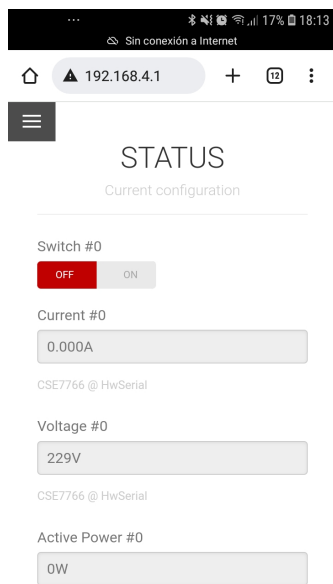
Para conectarse a la red del dispositivo, se debe ingresar la clave **fibonacci** y una vez adentro en la red se debe ingresar a la dirección `http://192.168.4.1` donde el usuario y la contraseña son **admin** y **fibonacci** pero la interfaz obliga a cambiar dicha clave de forma inmediata. Con la nueva clave ingresada, se puede ingresar a la configuración del dispositivo.



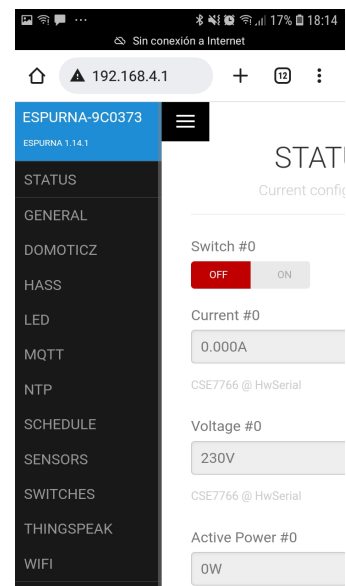
(a) Red WIFI generada por el enchufe inteligente



(b) Clave de acceso a Enchufe inteligente



(c) vista del *status* del enchufe inteligente, visto en `http://192.168.4.1`



(d) Menús de configuraciones del enchufe inteligente

Figura 4.4: Vistas de las configuraciones y vistas disponibles luego de la instalación del *firmware* ESPurna

Cuadro 4.1: Parámetros de conexión WIFI de SonOff Pow R2 con ESPurna.

Parámetro	Valor
Network SSID	smartBuildingNetwork
Password	SmartBuilding01
Static IP	192.168.2.30
Gateway IP	192.168.2.1
Network Mask	255.255.255.0
DNS IP	200.83.1.4

Como se observa en la figura 3.18c, este dispositivo requiere ser conectado con cables tanto al suministro eléctrico como a la carga eléctrica a medir, es por ello que de manera de ejemplo, se arma el banco de pruebas mostrado en la figura 4.2 donde se instalan estos equipos de 3 formas diferentes, las cuales se describen a continuación:

- **Comandando un circuito completo:** Para este caso se desconectó el circuito completo de iluminación y se hizo pasar por el enchufe inteligente permitiendo tener la medición real de todo el circuito, independiente de los equipos de iluminación usados en la instalación (Mostrado en verde en la figura 4.2).
- **Comandando un enchufe:** En donde se desconectó un enchufe y se realizó la conexión a través del equipo lo que permite obtener datos de energía consumida de todos los equipos conectados a dicho enchufe (Mostrado en amarillo en la figura 4.2).
- **Comandando un solo dispositivo:** Donde solo se conectaron enchufes de tipo volante (macho y hembra) al enchufe inteligente de manera de conectar de la manera menos invasiva posible los equipos para la correcta medición de la energía utilizada (Mostrado en azul en la figura 4.2).

Luego, se ingresa a la sección **WIFI** del menú lateral y se ingresan los datos de la red local, que para el caso del banco de pruebas planteado son los mostrados en el cuadro 4.1.

Luego, al presionar el botón **save** del menú lateral, se apagará la señal **WIFI** generada por el dispositivo y éste se conectará a la red local desde donde se puede acceder estando conectado a dicha red local e ingresando a la **IP** estática definida en su configuración. Con esto

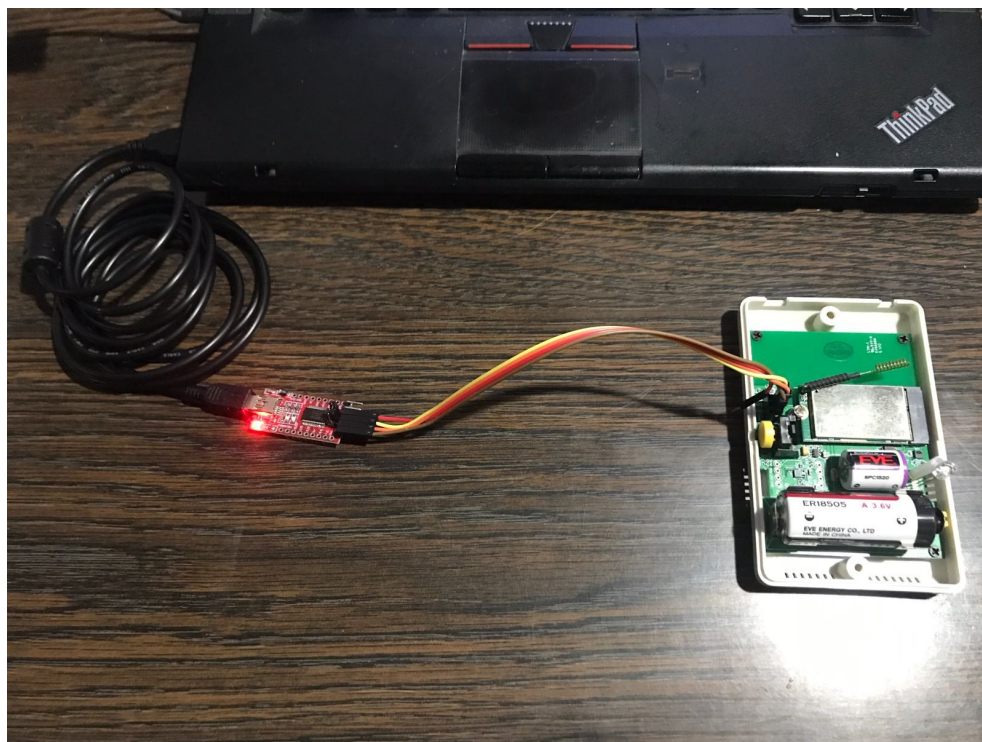


Figura 4.5: Diagrama de conexión para configuración de dispositivo LAQ4

quedan conectados y configurados los enchufes inteligentes y los datos obtenidos pueden ser visualizados accediendo a través del navegador a sus **IPs**.

4.1.2. Monitoreo de las condiciones ambientales en ambientes interiores

Para monitorear las principales condiciones ambientales se utiliza el sensor de calidad del aire Dragino LAQ4 el cual mide parámetros como: Compuesto orgánico volátil total, CO2 equivalente, Temperatura y Humedad Relativa del Aire.

Para la obtención de datos se debe contar con una *Gateway* LORA y para esto se utiliza la *Gateway* **LG308** de Dragino.

Al preparar la conexión óptima del sensor, se abre la carcasa del sensor y se conecta, con ayuda de un adaptador USB a TTL, el sensor al computador (como se muestra en la figura 4.5) y se ingresan los datos descritos en la caja con el título ***Registration Key***.

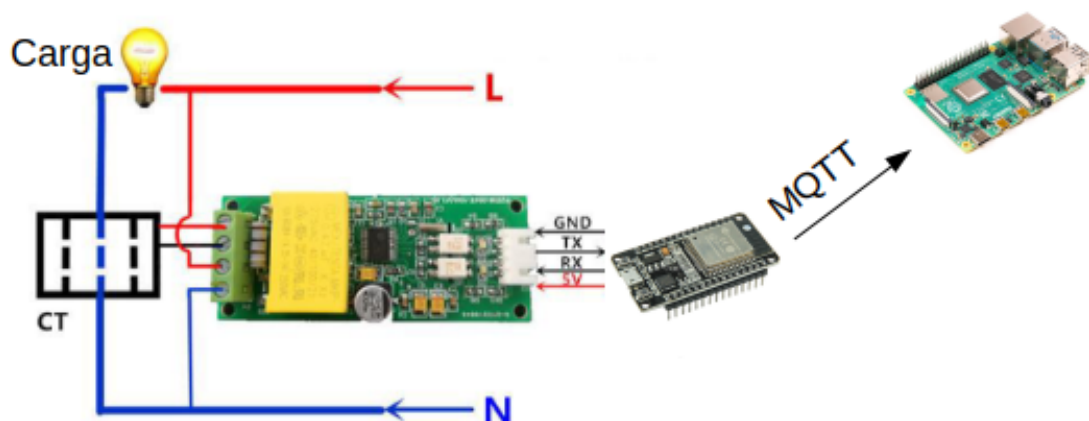


Figura 4.6: Diagrama de conexión del dispositivo de medición energética pzem004t-100A.

Luego, se configura el modo ABP para la conexión directa entre el dispositivo y la *Gateway* LORA.

4.1.3. Monitoreo de la energía total

La medición se realiza tomando el dato de voltaje a la entrada del tablero eléctrico del escenario en estudio y con la ayuda de un sensor transformador de corriente se lee la corriente total de entrada, para ello se utiliza el módulo **pzem-004t-100a** el cual se conecta a la plataforma de desarrollo ESP32 para el posterior envío de datos a la capa niebla, como se muestra en la figura 4.6. Entonces la conexión de adquisición de datos se realiza según la figura 4.6. Luego, para la implementación se considera que el módulo se comunica con comunicación serial, por lo que se conecta a los pines destinados a la comunicación serial del ESP32. Para su programación se utiliza la biblioteca **pzem** construida para el IDE Arduino.

Al concluir con la implementación del módulo descrito se realiza la instalación ubicando el sensor de corriente envolviendo el cable de fase del circuito a medir y sacando una conexión en paralelo para la medición de voltaje, esta implementación se puede observar en la figura 4.7.

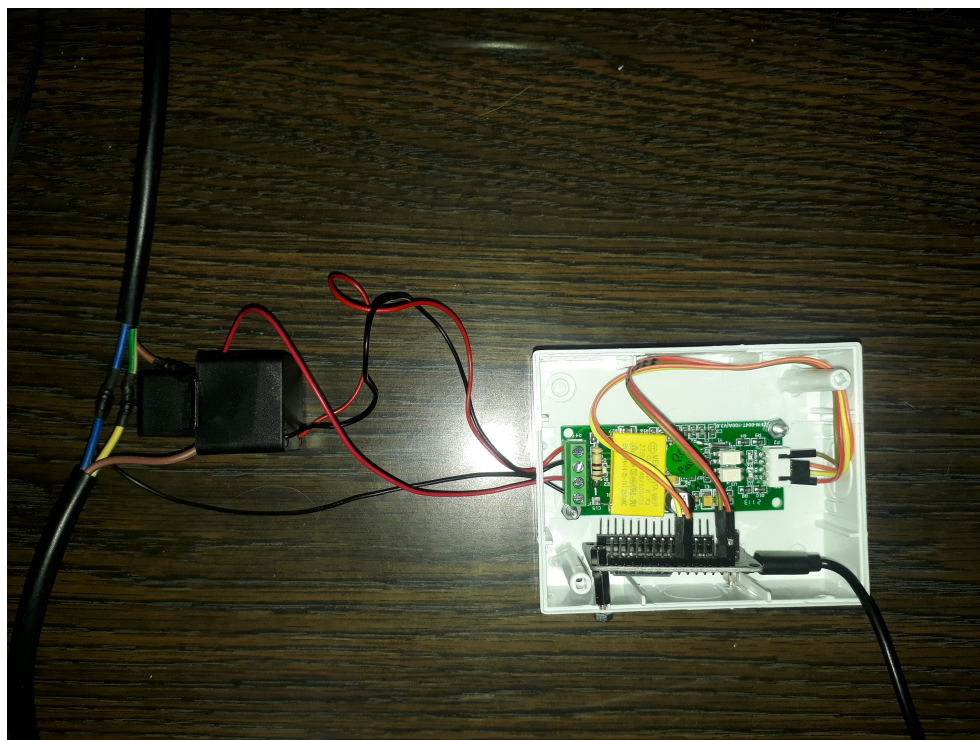


Figura 4.7: Conexión del módulo pzem004t-100A para la lectura general de energía.

4.1.4. Monitoreo del sistema de paneles fotovoltaicos

La medición de los principales parámetros de un panel fotovoltaico se realiza midiendo la corriente con un *CT sensor* (por sus siglas en inglés, *Current Transform Sensor*) que entrega un corriente proporcional a la corriente de medición, pero mucho más baja y un divisor de tensión para la medición de voltaje, ambas mediciones se capturan con pines **ADC** de la tarjeta de desarrollo **Arduino nano**, datos que se envía la tarjeta de desarrollo **NodeMCU Amica** que posee comunicación **WIFI** para luego enviar los datos por **MQTT** a la capa *Niebla* junto con la fecha y hora en que fueron capturados (Dato que se obtiene gracias a que se agregó un módulo **RTC** (*Real Time Clock*) que entrega el valor del tiempo real).

El prototipo del dispositivo de medición de paneles solares se muestra en la figura 4.8 donde el *CT sensor* debe ser conectado envolviendo el cable positivo del panel y los terminales de prueba de voltaje (café - tierra y rojo - VCC) van conectados en paralelo al bus de corriente continua del sistema de paneles fotovoltaicos.

Los datos obtenidos con este dispositivo se envían directamente a la capa **niebla**.

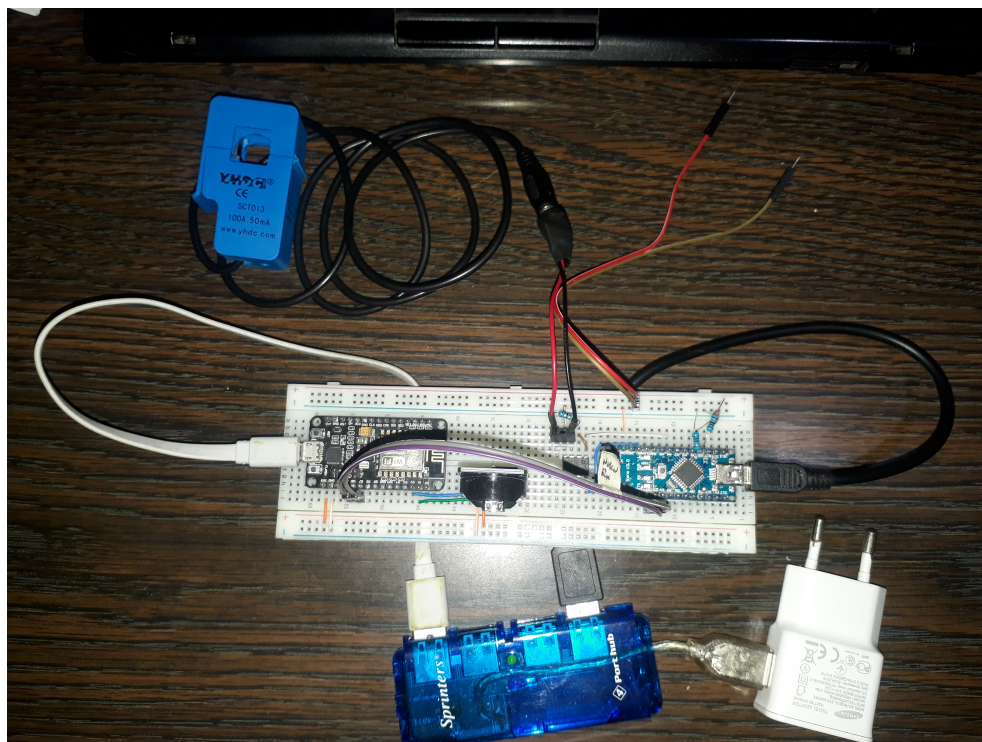


Figura 4.8: Prototipo de equipo de medición de paneles solares

4.1.5. Manejo de datos de estación meteorológica

En el caso de la estación meteorológica *Davis Advantage Pro 2 Plus* se constata que se trata de un sistema cerrado y que no permite la manipulación externa de los datos obtenidos.

En particular el sistema consiste de la siguiente forma:

La estación misma obtiene los datos de los sensores conectados físicamente a ésta, luego, se envía de forma inalámbrica a la consola **Vantage Pro 2** y en caso de tener conecta el *datalogger* vendido por la marca, se puede conectar dicha consola con un cable USB al computador. Se debe tener el programa propietario del sistema llamado *WhetherLink Computer Software* donde se pueden visualizar los datos medidos en tiempo real como se observa en la figura 4.9.

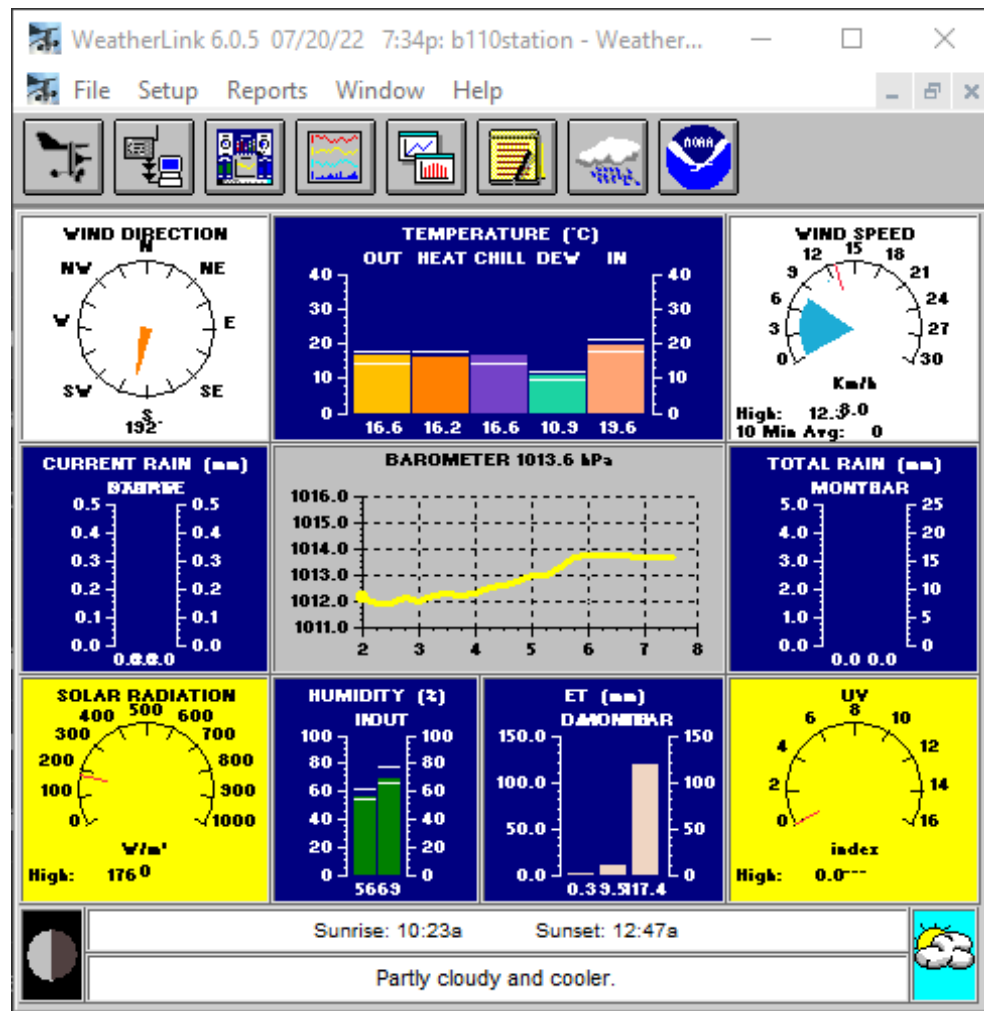


Figura 4.9: Vista general de los datos de la estación meteorológica obtenida desde el *WeatherLink Computer Software*.

4.2. Capa de comunicación

Para la implementación de esta capa se utilizan los protocolos de comunicación MQTT y LORA.

En particular para el caso de los enchufes inteligentes la comunicación por **MQTT** se activa y configura eligiendo la sección **MQTT** del menú lateral mostrado en la interfaz web de los enchufes inteligentes con el *firmware* **ESPurna** instalado como se observa en la figura 4.10.

Para el caso de los dispositivos de medición general y de paneles fotovoltaicos la integración del protocolo **MQTT** se realiza dentro del código con que fueron programados.

La configuración para todos los dispositivos que se conectan por **MQTT** es similar, es decir, se conectan al *broker* que está corriendo en la **Raspberry Pi** en la **IP** 192.168.2.2 por el puerto 1883 y se diferencia por el tópico en el cual publican. Tópicos que fueron definidos de manera descriptiva de la medición y la locación en que se toman los datos y que se muestran en el cuadro 4.2.

Para el caso del dispositivo de medición de calidad de aire (**LAQ4**) se debe realizar un procedimiento un poco más largo. Se crea una cuenta en el servidor **LORAWAN the things Network** en <https://www.the-thingsnetwork.org/>, luego, al registrarse se debe entrar en el apartado *start building* (que lleva a la dirección <https://console.cloud.thethings-network/>) donde se elige el *Cluster Nam1* ubicado en el estado de carolina en Estados Unidos (que es el *Cluster* más cercano a nuestro país) y una vez seleccionado se entra a la pestaña de *Gateways* en donde se presiona el botón **+ add gateway** rellenando todos los datos pedidos, en particular, el **Gateway ID** que es el número único asociado a cada *gateway* **LoRa**. Para el caso del presente trabajo se utiliza el nombre de **GatewayOfBuilding** y el *gateway ID* de **gatewaylorabuilding01**.

Una vez concluida la inscripción de la *gateway* **LoRa** se puede volver a la pestaña de **gateways** y seleccionar la *gateway* inscrita de manera de ver información de la conexión de ésta con el servidor *the things Network* como se muestra en la figura 4.11.

Ya verificada la conexión de la *gateway* se debe entrar a la pestaña *applications* y presionar el botón **+ add application** donde solo se deben llenar 3 parámetros, a saber, *Application ID*,

MQTT

Configure an **MQTT broker** in your network and you will be able to change the switch status via an MQTT message.

Enable MQTT

MQTT Broker

MQTT Port

MQTT User

You can use the following placeholders:
{hostname}, {mac}

MQTT Password

Figura 4.10: Ventana de activación y configuración de la comunicación de los enchufes inteligentes usando MQTT

Cuadro 4.2: Descripción de tópicos de los dispositivos conectados por MQTT.

Dispositivo	Carga	Locación	Tópico
Sonoff Pow R2	enchufes	oficina B-349	officeB349/ enchufes01/ ESPURNA- A9F0E4
Sonoff Pow R2	iluminación	oficina B-349	officeB349/ ilu- minacionGene- ral/ ESPURNA- 9CFBF8
Sonoff Pow R2	Computador	oficina B-349	officeB349/ connectedDevi- ce/ Computer/ ESPURNA- 9CFBF8
Dispositivo de medición DC	Panel fotovol- taico	Exterior	photovoltaicSystem/ panel01
Dispositivo de medición AC	Panel General	sala B-213	classroomB213/ generalPanel

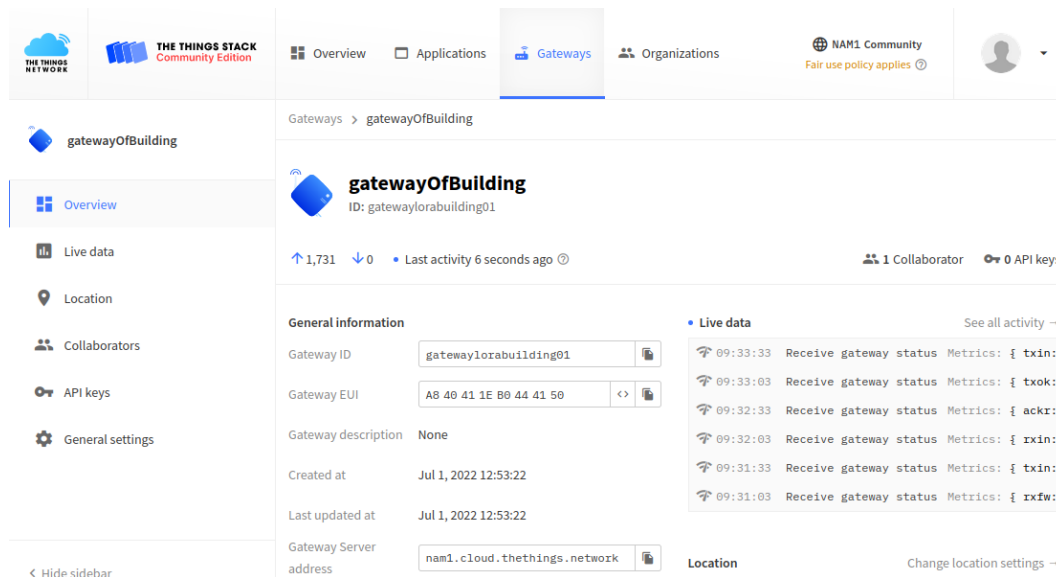


Figura 4.11: Ventana se estado de la conexión entre la *gateway* LoRa y el servidor en *The Things Network*

Application Name y *Description* que para este caso se llena con los datos de *smart-buildings-lora-sensors-usmcc* en *Application ID* y *lora sensors in usmcc* en *Application Name*.

Una vez creada la aplicación y al entrar en ella se presiona el botón **+ add end device** e ingresando los datos para el sensor deseado, que en este caso es el **LAQ4** por lo que se ingresan los datos correspondientes al fabricante y modelo del dispositivo.

A continuación, se ingresan los parámetros propios del sensor, llamados *Registration Key*, además, en este caso, Se agregó el valor de *airqualitysensorusmcc* al parámetro **End Device ID** y se presiona *Register End Device*.

Una vez concluido el registro, se vuelve a entrar a la pestaña *applications* y en el apartado de *end devices* se entra al recién creado, como se observa en la figura 4.12.

Finalmente, estando dentro del dispositivo agregado se entra al apartado del menú lateral *Integrations* y luego a **MQTT** donde se obtienen los datos de subscripción al **broker** de *The things network* como se muestra en la figura 4.13, datos que serán usados en la capa de aplicación para visualizar de mejor manera los datos.

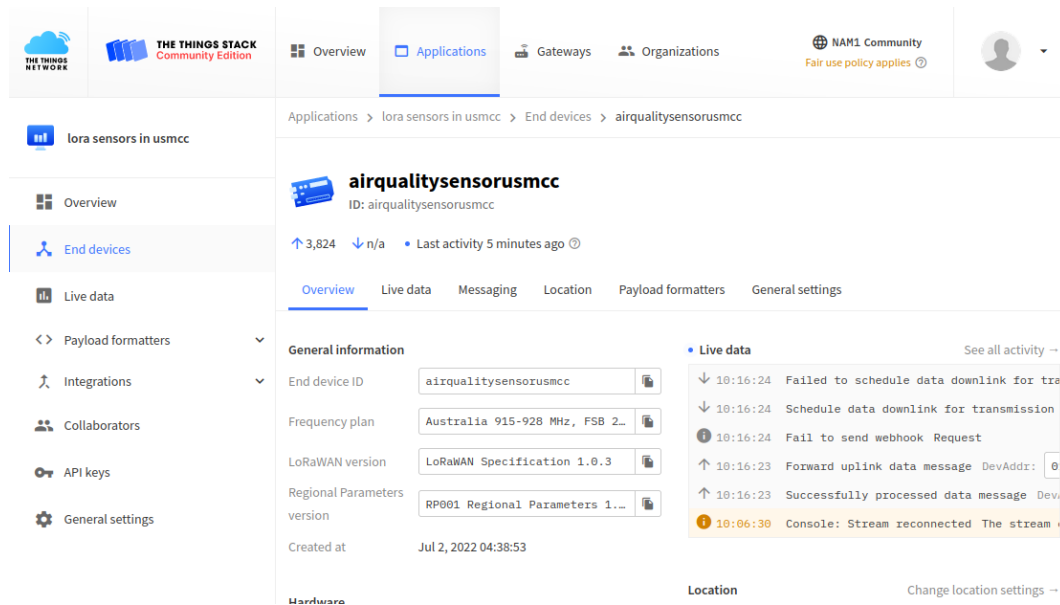


Figura 4.12: Ventana se estado de la conexión entre el dispositivo final LAQ4 y el servidor en *The Things Network*

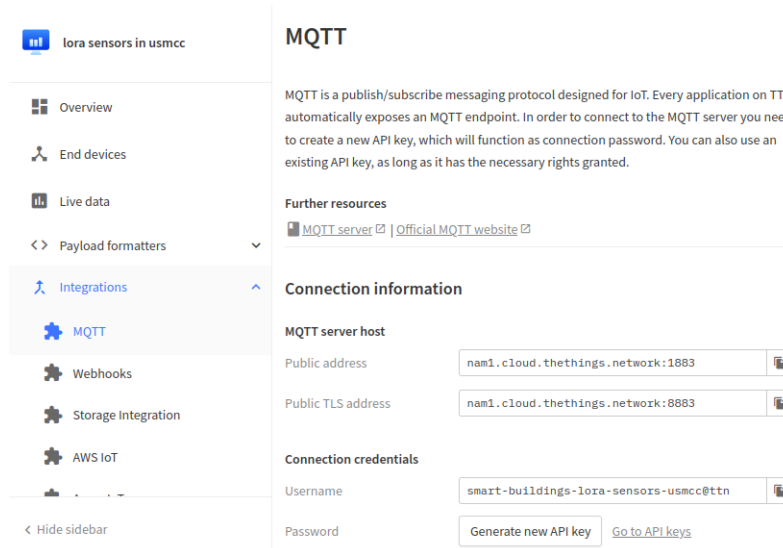


Figura 4.13: Ventana de configuración de envío de datos por MQTT en *The Things Network*

4.3. Capa Niebla

Para esta capa se utiliza una **Raspberry Pi** con el sistema operativo **Raspberry Pi OS** el cual se carga en una memoria micro SD descargando el software de instalación de <https://www.raspberrypi.com/software/>, se elige el sistema a instalar y se selecciona la memoria micro SD a utilizar en la Raspberry Pi mencionada.

Para la configuración posterior, se entra al sistema usando una pantalla con entrada HDMI y un *mouse* para activar la opción de permitir la conexión por el protocolo SSH y posterior a ello, simplemente se conecta a la red local por el cable de *Ethernet*.

Luego, en un computador dentro de la misma red se realiza la conexión por SSH (considerando que la IP de la Raspberry Pi es 192.168.2.2) usando el siguiente comando.

```
$> ssh buildingadmin@192.168.2.2
```

Y se ingresa la clave.

Una vez conectado el computador a la **Raspberry Pi** por SSH, se instala Mosquitto como *broker MQTT* seleccionado, para ello se usan los comandos:

```
$> sudo apt update
$> sudo apt upgrade
$> sudo apt-get install mosquitto mosquitto-clients
```

Luego, se instala Node-RED con que se manejan los datos de manera local, para ello, siguiendo la recomendación de la pagina oficial (**nodered.org**) se utiliza el siguiente comando.

```
$> bash <(curl -sL
https://raw.githubusercontent.com/node-red/
linux-installers/master/deb/update-nodejs-and-nodered)
```

Para que el programa se ejecute desde el arranque del sistema operativo en la Raspberry, se hace

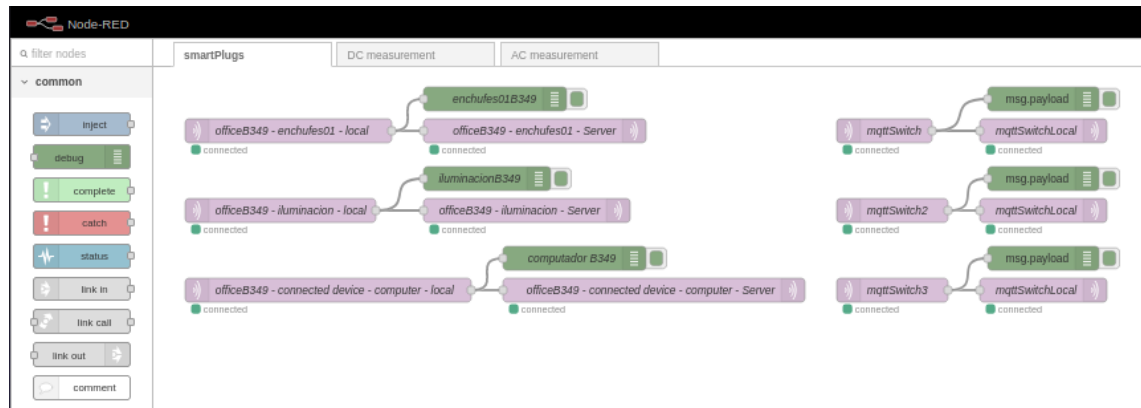


Figura 4.14: Esquema implementado para el reenvío de los datos recibidos de los enchufes inteligentes conectados

```
$> sudo systemctl enable nodered.service
```

Aunque para ejecutarlo directa mente basta con escribir el comando

```
$> node-red
```

Para ingresa a la paleta de programación de node-red basta con estar dentro de la misma red local de la Raspberry Pi e ingresar a <http://192.168.2.2:1880>

Luego, una vez dentro de la paleta de programación se realiza la división en 3 flujos que corresponden a: los enchufes inteligentes (ver figura 4.14), el dispositivo de medición de corriente alterna (ver figura 4.15) y el dispositivo de medición de corriente continua (ver figura 4.16). En los 3 flujos se reciben datos **MQTT** de cada equipo y se reenvía usando el mismo protocolo (**MQTT**) al servidor alojado en **Digital Ocean**.

4.4. Capa de Aplicación

La capa de aplicación está enteramente soportada en una maquina virtual contratada con los servicios de Digital Ocean, dicha maquina tiene 2 Gb de RAM y 50 Gb de disco duro.

Para la contratación de estos servicios se debe entrar a la pagina del proveedor <https://www.digitalocean.com/go/developer-brand> Y se debe registrar (considerando que

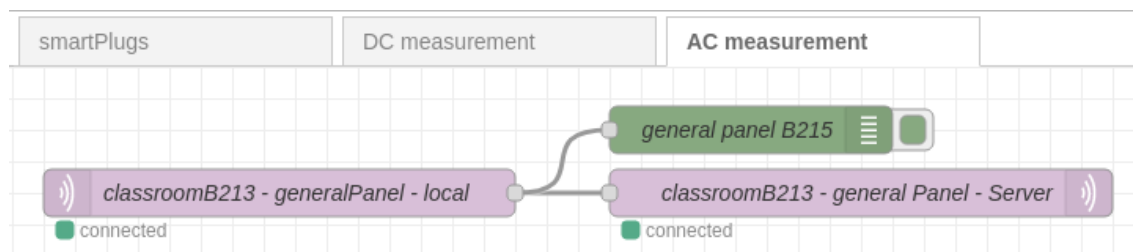


Figura 4.15: Esquema implementado para el reenvío de los datos recibidos del dispositivo de medición de energía general

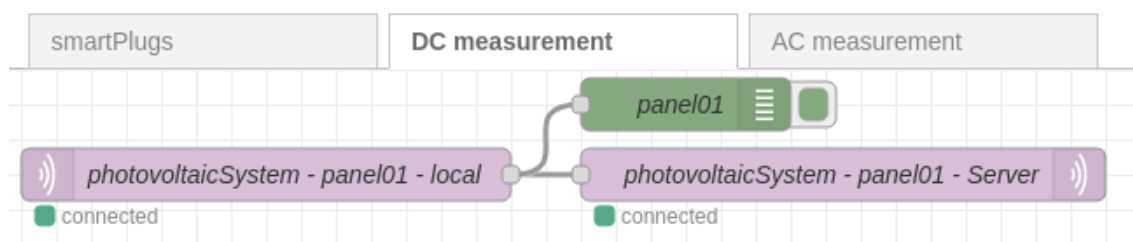


Figura 4.16: Esquema implementado para el reenvío de los datos recibidos del dispositivo de medición de corriente continua

se debe incluir una tarjeta de crédito válida durante el proceso de registro).

Luego de iniciar sesión en la página, se genera un nuevo proyecto que, en este caso, se llama **IoTPlatform4ManageEnergy** y luego, se presiona el botón **create** para crear un nuevo *Droplet* que es la forma en que **Digital Ocean** le llama a las máquinas virtuales y esta se crea con 2 *Gb* de RAM, 50 *Gb* de disco duro y se instala el sistema operativo Ubuntu 20.04 (LTS) x64.

Al terminar la configuración del *Droplet* se obtiene la vista general que se muestra en la figura 4.17.

Una vez creado el **Droplet** se puede acceder a esta con la IP fija entregada por el proveedor (165.232.139.50) a través de SSH.

En este servidor se instaló el *broker Mosquitto* para definir la comunicación por MQTT, con el comando:

```
$> sudo apt-get update
$> sudo apt-get upgrade
$> sudo apt-get install mosquitto
```

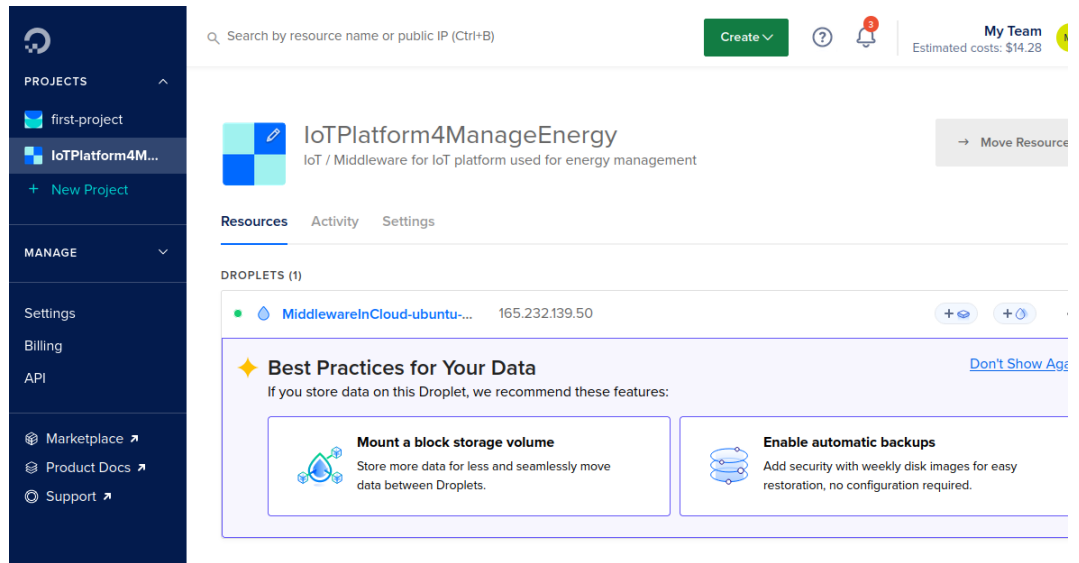


Figura 4.17: Ventana de configuración de *Droplet* en *Digital Ocean*

`mosquitto-clients`

Además, se instala Node-RED para la manipulación de los datos con los comandos que se describen a continuación.

```
$> bash <(curl -sL https://raw.githubusercontent.com/ node-red/
linux-installers/master/deb/
update-nodejs-and-nodered)
```

Se instalan los nodos extras con:

```
$> sudo apt install build-essential
```

Luego, para iniciar el proceso simplemente se escribe en la línea de comando:

```
node-red &
```

Continuando, se instala **MySQL** (para permitir el almacenamiento de datos) con el comando:

```
$> sudo apt install mysql-server
```

Se cambian algunas opciones predeterminadas para permitir su uso más seguro, ejecutando:

```
$> sudo mysql_secure_installation
```

Donde pedirá el cambio de la contraseña del usuario `root`.

A continuación, se inicia el proceso de **MySQL** con el comando:

```
$> mysql -u root -p
```

ingresando la contraseña pedida, y se crea la base de datos a usar, con:

```
mysql> CREATE DATABASE energydatausmcc;
```

Se crea un nuevo usuario con el comando:

```
mysql> CREATE USER 'smartbuildingoperator01'@'localhost'  
IDENTIFIED BY 'password';
```

Donde se agrega una contraseña válida para luego definir los privilegios asociados a dicho usuario con:

```
mysql> GRANT SELECT, INSERT, UPDATE, DELETE ON energydatausmcc *  
TO 'smartbuildingoperator01'@'localhost';  
mysql> FLUSH PRIVILEGES;
```

Se crean las tablas para almacenar los datos de los enchufes inteligentes, con el comando:

```
mysql> CREATE TABLE smartplugsedificiob  
(ID INT AUTO_INCREMENT, internalID INT, host VARCHAR(30),  
timecaptured DATETIME, timereception DATETIME,
```



```
current FLOAT, voltage INT, power INT,  
reactive INT, apparent INT, factor INT,  
energy INT, location VARCHAR(30), PRIMARY KEY (ID));
```

La tabla para almacenar los datos de la medición del tablero eléctrico, con:

```
mysql> CREATE TABLE acmeasurement  
(ID INT AUTO_INCREMENT, internalID INT, host VARCHAR(30),  
timecaptured DATETIME, timereception DATETIME,  
current FLOAT, voltage FLOAT, power FLOAT, frequency FLOAT,  
factor FLOAT, energy FLOAT, location VARCHAR(30),  
PRIMARY KEY (ID));
```

La tabla asociada a la medición del voltaje y corriente de cada panel solar, con:

```
mysql> CREATE TABLE dcmeasurement (ID INT AUTO_INCREMENT,  
panelID INT, host VARCHAR(30), timecaptured DATETIME,  
timereception DATETIME, current FLOAT, voltage FLOAT,  
power FLOAT, location VARCHAR(30), PRIMARY KEY (ID));
```

Y, la tabla asociada a los parámetros ambientales medidos, con:

```
mysql> CREATE TABLE airquality (ID INT AUTO_INCREMENT,  
internalID INT, host VARCHAR(30), timecaptured DATETIME,  
timereception DATETIME, batV FLOAT, co2 INT,  
humidity FLOAT, tvoc INT, temperature FLOAT, location VARCHAR(30),  
PRIMARY KEY (ID));
```

Finalmente, con todo esto instalado se crea los diagramas con **Node Red** como se muestran en la figuras 4.18, 4.19, 4.20, 4.21, 4.28.

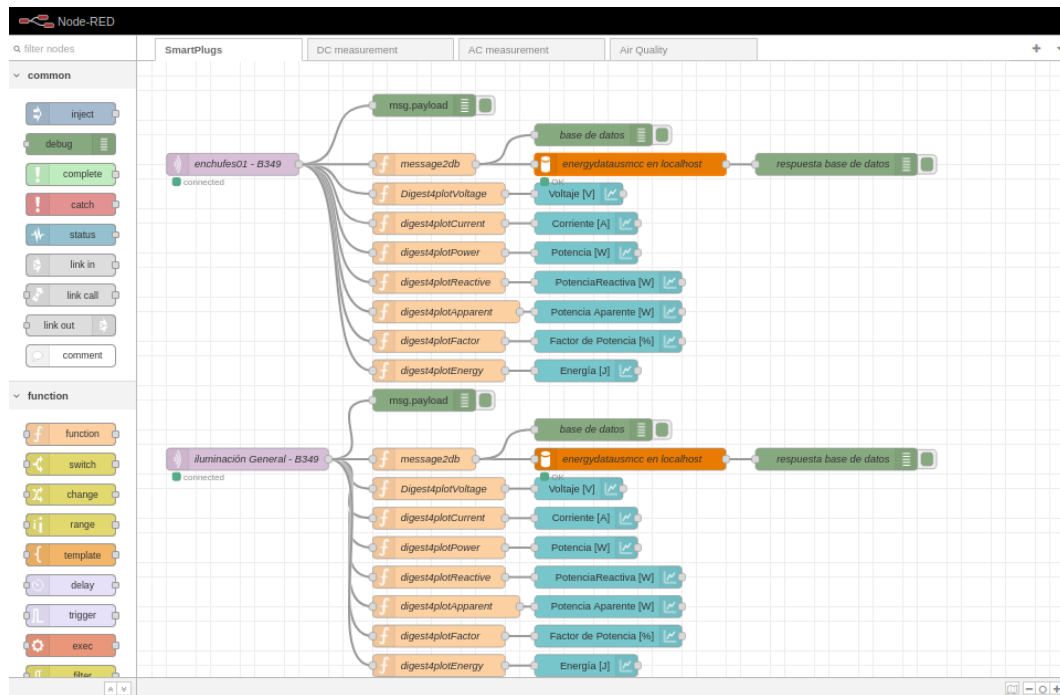


Figura 4.18: Ventana configuración de flujos para la visualización de datos de los enchufes inteligentes.

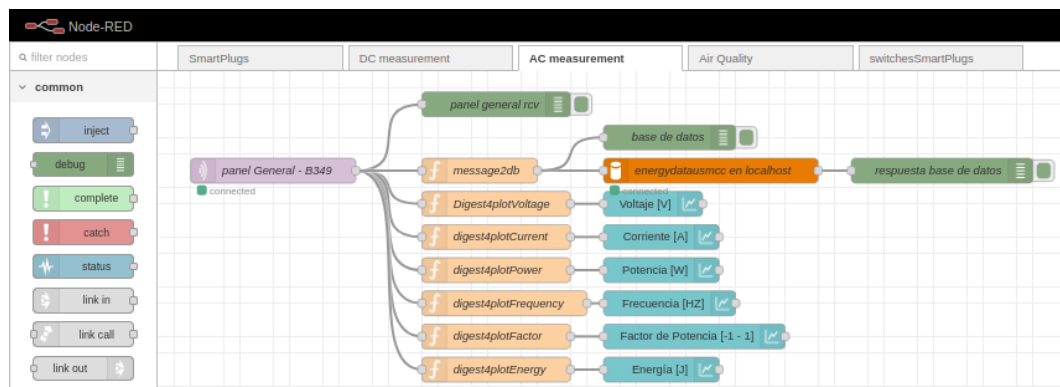


Figura 4.19: Ventana configuración de flujos para la visualización de datos del medidor general de tablero eléctrico.

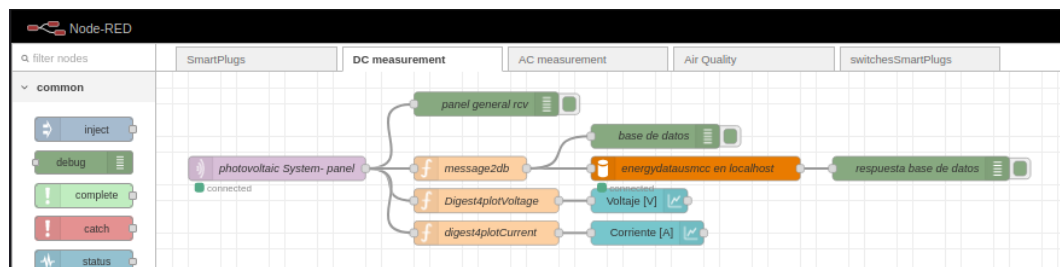


Figura 4.20: Ventana configuración de flujos para la visualización de datos del medidor del panel fotovoltaico.

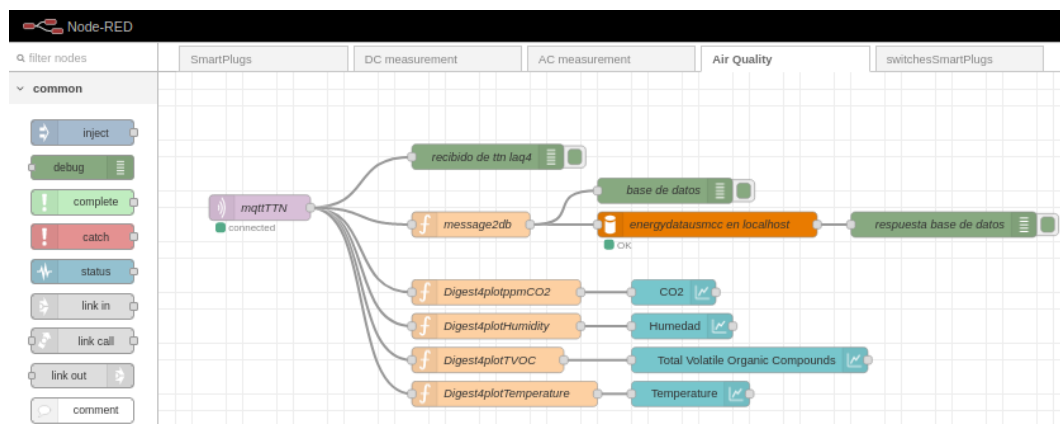


Figura 4.21: Ventana configuración de flujos para la visualización de datos ambientales.

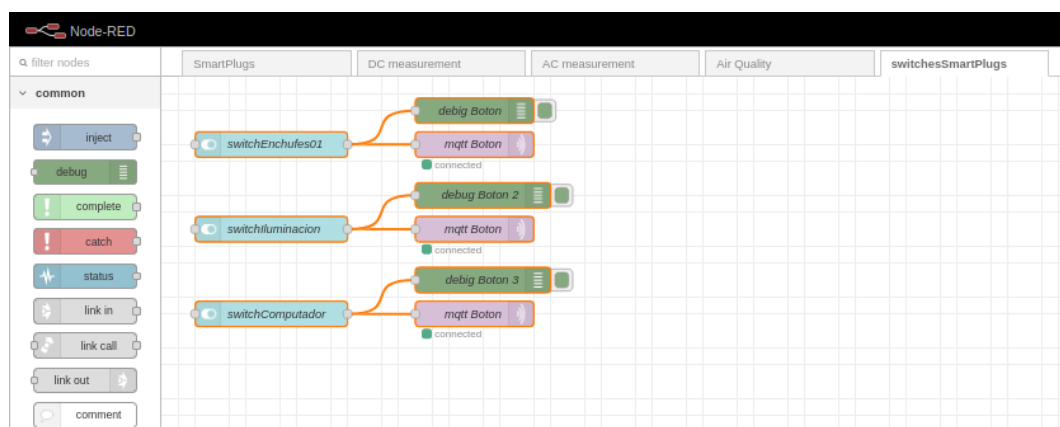
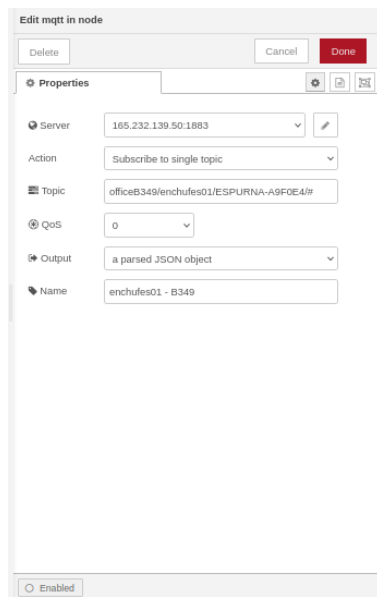
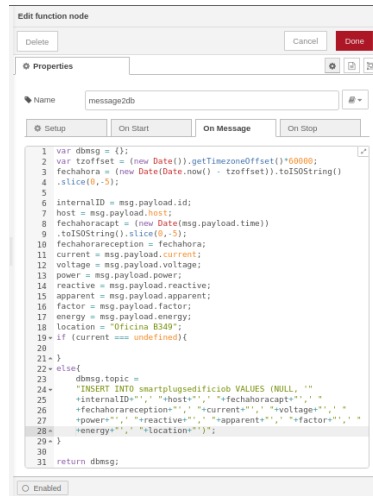


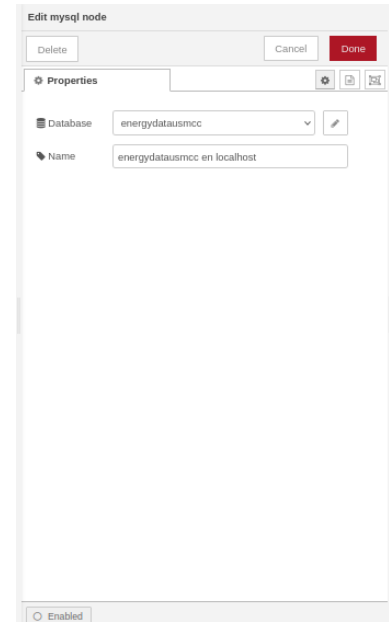
Figura 4.22: Ventana configuración de flujos para la activación de los enchufes inteligentes utilizados.



(a) Configuración nodo MQTT para la recepción de los datos desde la capa niebla.



(b) Configuración del nodo que procesa los datos para insertarlos en la base de datos.



(c) Configuración del nodo encargado de escribir en la base de datos.

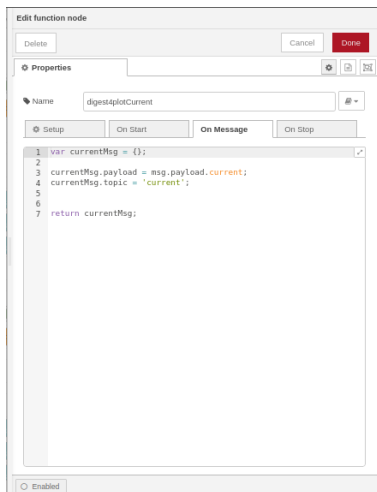
Figura 4.23: Configuraciones de nodos característicos para el almacenamiento de datos en MySQL

En donde la configuración de los principales nodos para la inserción de datos en la base de datos y para su visualización se muestra en las figuras 4.23 y 4.24.

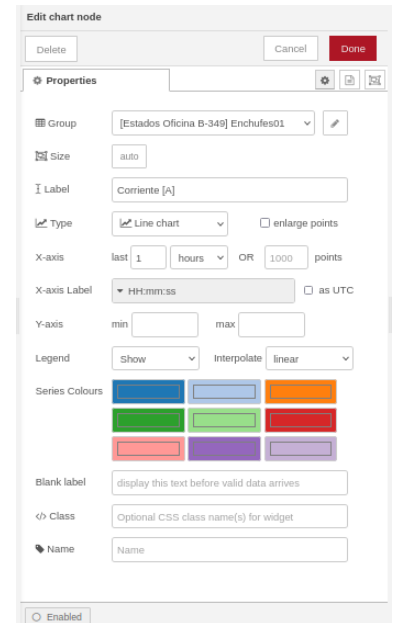
Y con ello se obtiene el *dashboard* mostrado en las figuras 4.25, 4.26, 4.27, 4.29. Estos ejemplos son los observados desde un computador, ya que al ser Node-red responsivo, la visualización se adapta de acuerdo al dispositivo de visualización, como puede ser un *smartphone* o una tablet.

Además, con la implementación del bloque de interacción con la base de datos de MYSQL, los datos quedan guardados en la nube, como se muestra en las figuras 4.30.

Ya con el *Dashboard* finalizado, se puede tener acceso a la visualización de los datos desde la URL <http://165.232.139.50:1880/ui> dando por concluido el desarrollo de la plataforma planteada.



(a) Configuración nodo que procesa uno de los datos para su posterior visualización.



(b) Configuración del nodo que dibuja el punto del valor en función del tiempo en que llega el dato.

Figura 4.24: Configuraciones de nodos característicos para la visualización de datos recibidos por MQTT.

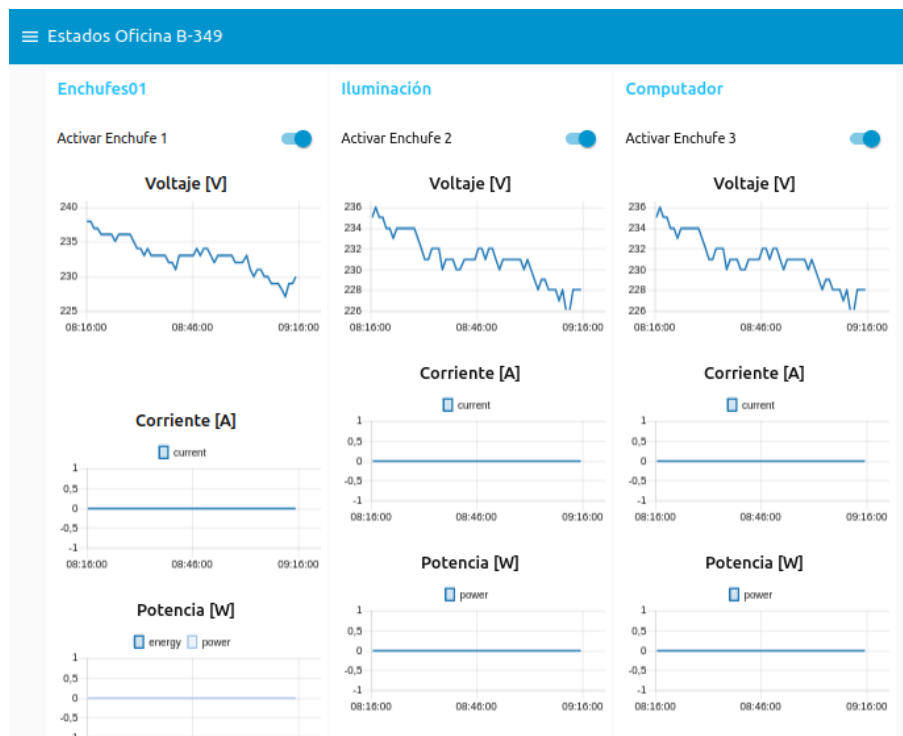


Figura 4.25: Captura de pantalla de la visualización de los datos de los enchufes inteligentes obtenidos.

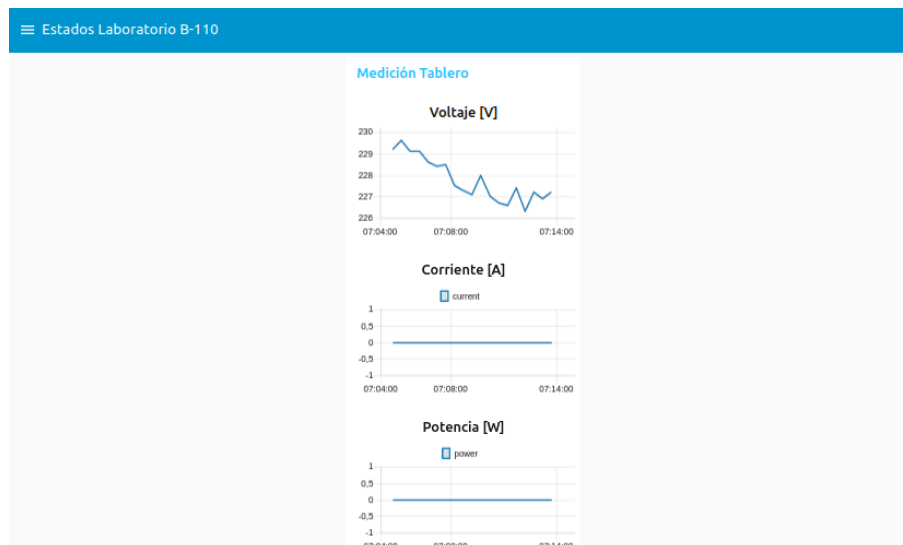


Figura 4.26: Captura de pantalla de la visualización de los datos de medición del tablero eléctrico.



Figura 4.27: Captura de pantalla de la visualización de los datos del sistema de paneles solares.

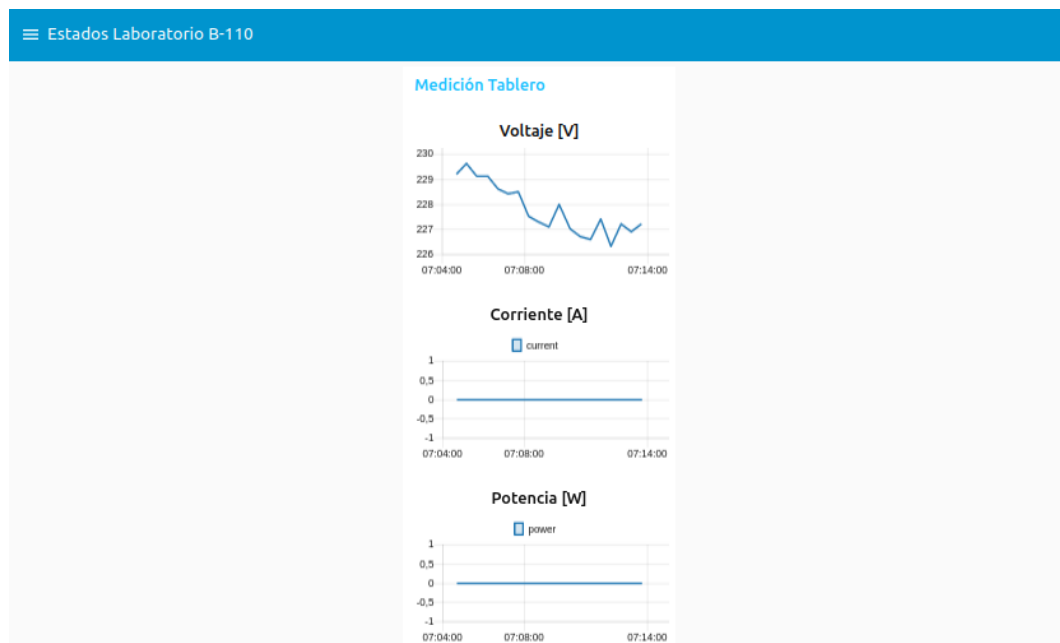


Figura 4.28: Captura de pantalla de la visualización de los datos de energía consumida en el tablero eléctrico.

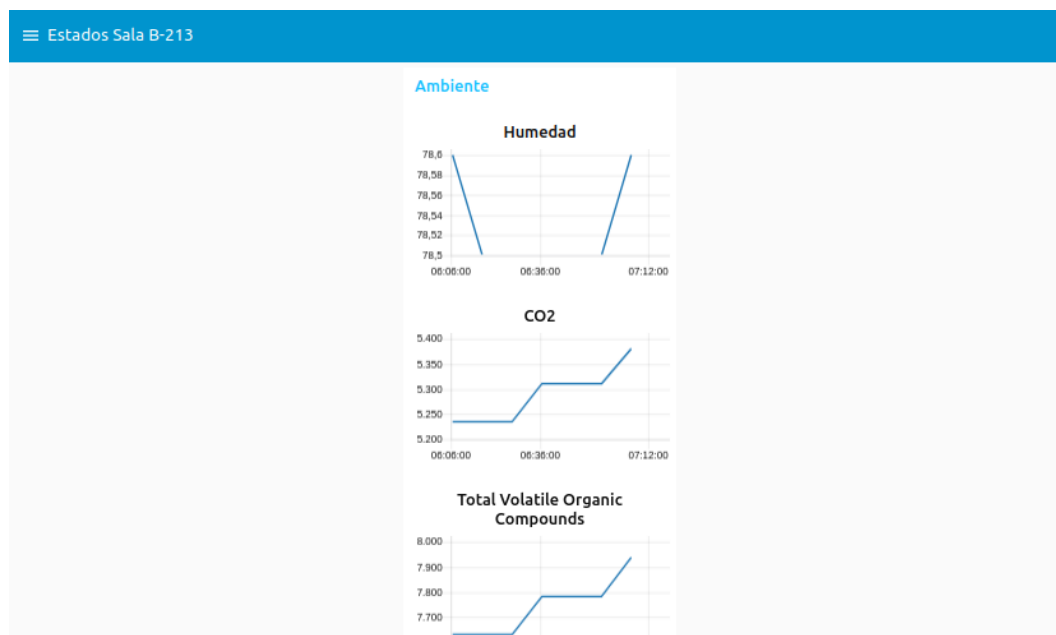


Figura 4.29: Captura de pantalla de la visualización de los datos ambientales obtenidos.

```
mysql> SELECT * FROM smartplugsedificio;
```

ID	internalID	host	timecaptured	timereception	current	voltage	power	reactive	apparent	factor	energy	location
4127	1283	ESPURNA-A9F0E4	2022-07-11 10:21:52	2022-07-11 08:21:52	0	231	0	0	0	100	115120	NULL
4128	1283	ESPURNA-9CFBF8	2022-07-11 10:21:55	2022-07-11 08:21:53	0	230	0	0	0	100	21165	NULL

(a) Captura de pantalla de una línea característica de la tabla **smartplugsedificio**.

```
mysql> SELECT * FROM acmeasurement;
```

ID	internalID	host	timecaptured	timereception	current	voltage	power	frequency	factor	energy	location
953	1	tablero eléctrico 01	2022-07-11 10:22:36	2022-07-11 10:22:36	0	229.7	0	49.9	0	0.074	Oficina B-349

(b) Captura de pantalla de una línea característica de la tabla **acmeasurement**.

```
mysql> SELECT * FROM dcmeasurement;
```

ID	panelID	host	timecaptured	timereception	current	voltage	power	location
143	1	panel fotovoltaico 01	2022-07-11 10:25:24	2022-07-11 10:25:24	0	0	0	Laboratorio B-110

(c) Captura de pantalla de una línea característica de la tabla **dcmeasurement**.

```
mysql> SELECT * FROM airquality;
```

ID	internalID	host	timecaptured	timereception	batv	co2	humidity	tvoc	temperature	location
205	1	laq4-01	2022-07-11 10:27:30	2022-07-11 10:27:30	3.289	6014	76.7	9296	14.4	Sala B-213

(d) Captura de pantalla de una línea característica de la tabla **airquality**.

Figura 4.30: Capturas de pantalla de los parámetros guardados en cada tabla en MySQL.

Conclusiones

Al concluir el desarrollo del presente trabajo se obtiene una plataforma funcional que reúne datos de consumo energético dentro de un edificio inteligente. Para ello se define una arquitectura de 5 capas que se describen a continuación:

En primer lugar, se estudia la capa física presente en el edificio que se caracterizan por tableros eléctricos por dependencia, donde se conectan circuitos de enchufes, iluminación entre otros, enchufes de aparatos específicos, enchufes de uso general y el sistema de paneles fotovoltaicos.

Para la capa de adquisición se utilizan dispositivos de medición para tablero eléctrico, enchufes inteligentes, dispositivos de medición de energía de corriente continua de diseño propio, equipos de monitoreo de calidad de aire y una estación meteorológica para capturar parámetros medioambientales.

Se define una capa niebla capaz de reunir la información captada en la red local y reenviarla al servidor remoto para ser visualizada. Esta capa se implementa en una **Raspberry Pi** y los datos se manejan con **Node Red**.

La capa de comunicación se define de manera de obtener la información de los equipos instalados por comunicación **WIFI** o **LoRa** y luego enviarlas al servidor a través del protocolo de mensajería **MQTT**.

Finalmente, se contratan el servicio de **Dropplet** de **Digital Ocean** para utilizarlo como servidor de propósito general en donde se instala el **Broker MQTT**, para el manejo de los datos enviados por este protocolo de mensajería, se instala **Node Red** para el manejo general de los datos y la visualización de ellos por parte del usuario final y se instala **MySQL** que permite almacenar la información en tablas definidas para cada dispositivo de adquisición de datos.

La implementación de este proyecto se realiza en un banco de pruebas definido para caracterizar los equipos y condiciones en 3 escenarios característicos dentro de la universidad Técnica Federico Santa María, los cuales son la oficina B-349, la sala de clases B-213 y el laboratorio B-110. Dicho banco de pruebas permite probar el sistema completo en un espacio reducido.

Bibliografía

- [1] W. Tushar, N. Wijerathne, W. Li, C. Yuen, H. Poor, T. Saha, and K. Wood. Iot for green building management. *IEEE Signal Processing Magazine*, 05 2018.
- [2] J. Sinopoli. *Smart building systems for architects, owners, and builders*. Elsevier, 2010.
- [3] A. Nugur, M. Pipattanasomporn, M. Kuzlu, and S. Rahman. Design and development of an iot gateway for smart building applications. *IEEE Internet of Things Journal*, PP:1–1, 07 2019.
- [4] A. Eltamaly, M. Alotaibi, a. Alolah, and M. Ahmed. Iot-based hybrid renewable energy system for smart campus. *Sustainability*, 13, 08 2021.
- [5] G. Alsuhli and A. Khattab. A fog-based iot platform for smart buildings. pages 174–179, 02 2019.
- [6] M. Jia, A. Komeily, Y. Wang, and R. Srinivasan. Adopting internet of things for the development of smart buildings: A review of enabling technologies and applications. *Automation in Construction*, 101:111–126, 05 2019.
- [7] D. S. Vijayan, A. Leema Rose, S. Arvindan, J. Revathy, and C. Amuthadevi. Automation systems in smart buildings: a review. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–13, 11 2020.
- [8] X. Zhang, M. Pipattanasomporn, T. Chen, and S. Rahman. An iot-based thermal model learning framework for smart buildings. *IEEE Internet of Things Journal*, PP:1–1, 11 2019.
- [9] A. P. Plageras, K. E. Psannis, C. Stergiou, H. Wang, and B. B. Gupta. Efficient iot-based sensor big data collection–processing and analysis in smart buildings. *Future Generation Computer Systems*, 82:349–357, 2018.
- [10] M. Elsisi, M. Tran, K. Mahmoud, M. Lehtonen, and M. M F. Darwish. Deep learning-based industry 4.0 and internet of things towards effective energy management for smart buildings. *Sensors*, 21:1038, 02 2021.

- [11] A. Pandharipande, M. Zhao, and E. Frimout. Connected indoor lighting based applications in a building iot ecosystem. *IEEE Internet of Things Magazine*, 2:22–26, 03 2019.
- [12] C. Metallidou, K. Psannis, and E. Egyptiadou. Energy efficiency in smart buildings: Iot approaches. *IEEE Access*, PP:1–1, 03 2020.
- [13] D. Siddiqui and V. Srivastava. Internet-of-things-based smart cities: Recent advances and challenges. 05 2020.
- [14] Aditya A. Nugur, M. Pipattanasomporn, M. Kuzlu, and S. Rahman. Design and development of an iot gateway for smart building applications. *IEEE Internet of Things Journal*, PP:1–1, 07 2019.
- [15] C. Stolojescu-Crisan, C. Crisan, and B. Butunoi. An iot-based smart home automation system. *Sensors*, 21:3784, 05 2021.
- [16] P. Pierleoni, R. Concetti, A. Belli, and L. Palma. Amazon, google and microsoft solutions for iot: Architectures and a performance comparison. *IEEE Access*, PP:1–1, 12 2019.
- [17] Amazon. benefits of application-hosting, 2022.
- [18] Azure. What is azure, 2022.
- [19] Google. Why google cloud, 2022.
- [20] Digital Ocean. Developer brand, 2022.