

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA



**SISTEMA INTELIGENTE PARA EL CONTROL DE CONSUMO
DE ENERGÍA EN EL HOGAR**

Tesis para obtener el título profesional de Ingeniero mecatrónico

AUTOR:

Armando Christofer Jaimes Chacon

ASESOR:

Diego Martin Arce Cigüeñas

Lima, noviembre, 2021

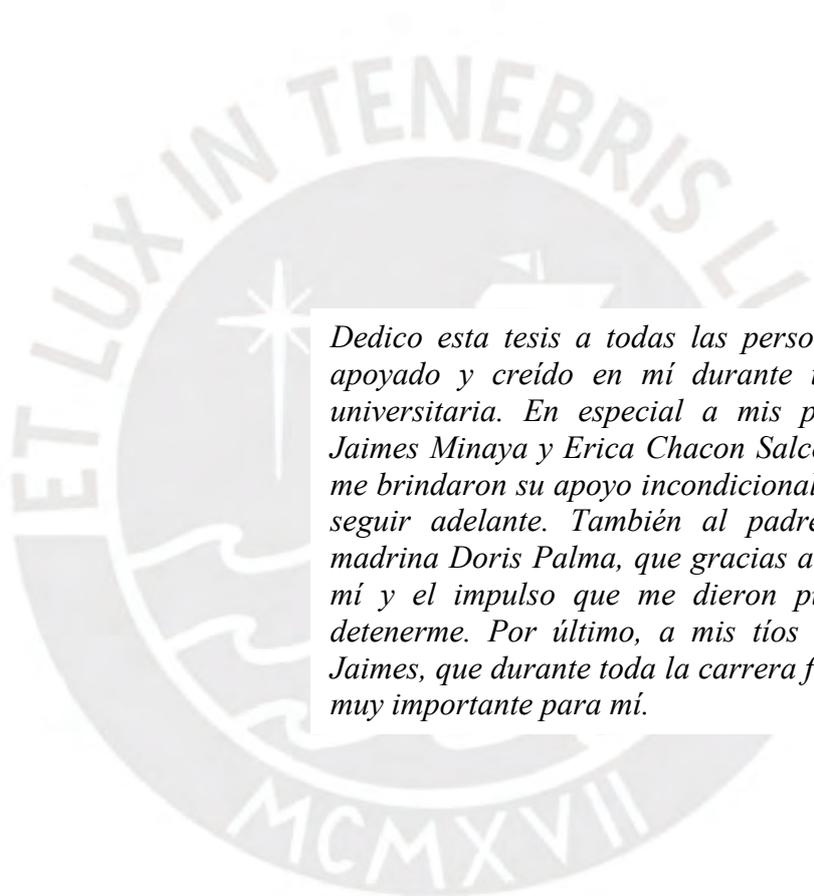
RESUMEN

Esta tesis presentará el desarrollo de un sistema ciberfísico inteligente para el monitoreo y control del consumo de energía en el hogar que abarcará el diseño de los dispositivos físicos y un sistema basado en el internet de las cosas (IoT, por sus siglas en inglés) usando herramientas de computación en la nube (también conocido como Cloud Computing) para el procesamiento y transmisión de la información recolectada y comandos enviados a través de internet.

Para esto se optará por una investigación preliminar y luego el desarrollo de ingeniería pertinente. También se desarrollará la aplicación web y una aplicación móvil. Además, se realizará la fabricación de los prototipos necesarios para la validación del sistema integrado con las aplicaciones mencionadas y ver su correcto funcionamiento.

Luego del trabajo realizado, se puede concluir que es posible el desarrollo de este sistema y que funciona de una manera muy estable. Todo esto se valida gracias a la realización de pruebas por medio de prototipos diseñados e impresos en 3D con filamento de plástico PLA, que da la suficiente resistencia mecánica para el sistema, y su integración con las aplicaciones desarrolladas. Además, el diseño se realizó para que sea un sistema escalable capaz de soportar una gran cantidad de usuarios y que se pueda seguir adicionando diferentes tipos de dispositivos para así aumentar el confort que se tiene.

En vista de que en el Perú el boom de la construcción hace que aumente exponencialmente el consumo energético por habitante, se hace necesario el desarrollo de este sistema que permita monitorear y regular el consumo de energía de una forma práctica y rápida.



Dedico esta tesis a todas las personas que me han apoyado y creído en mí durante toda mi carrera universitaria. En especial a mis padres, Armando Jaimes Minaya y Erica Chacon Salcedo, que siempre me brindaron su apoyo incondicional y la fuerza para seguir adelante. También al padre Moncho y mi madrina Doris Palma, que gracias a que creyeron en mí y el impulso que me dieron pude avanzar sin detenerme. Por último, a mis tíos Sara y Eduardo Jaimes, que durante toda la carrera fueron un soporte muy importante para mí.

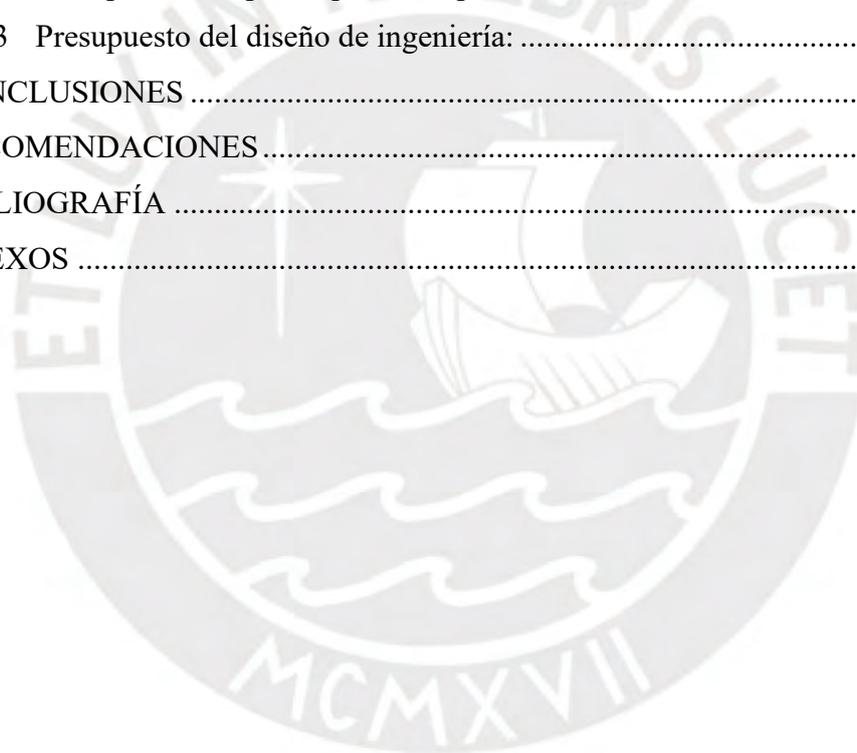
ÍNDICE DE CONTENIDO

RESUMEN.....	i
ÍNDICE DE CONTENIDO.....	iii
ÍNDICE DE FIGURAS.....	vii
ÍNDICE DE TABLAS	x
ÍNDICE DE ECUACIONES	xi
INTRODUCCIÓN	1
1. DEFINICIÓN DEL PROBLEMA	2
1.1 Problemática.....	2
1.2 Objetivos	4
1.2.1 Objetivo General	4
1.2.2 Objetivos específicos	4
1.3 Alcance.....	5
1.4 Estado del arte	5
1.4.1 Propuestas de sistemas de ahorro de energía (tesis).....	5
1.4.1.1 Red de medidores de energía para artefactos domésticos.....	5
1.4.1.2 Sistema inteligente de ahorro de energía eléctrica.....	6
1.4.1.3 Sistema domótico de control centralizado con comunicación por línea de poder	6
1.4.1.4 Sistema de monitoreo y control remoto de viviendas.....	7
1.4.2 Sistemas de ahorro de energía comerciales.....	7
1.4.2.1 Smart House Perú	7
1.4.2.2 Relay switch Simple	8
1.4.2.3 Activa Building control	8
1.4.3 Industria 4.0	9
1.4.4 Tecnologías de comunicación normalmente usadas por los sistemas domóticos.....	9
1.4.5 Protocolos web de envío de datos para IoT	10
1.4.6 Plataformas IoT	11
1.4.7 Plataformas de computación en la nube (cloud computing).....	12
1.4.8 Bases de datos	13
1.4.9 Sensores de consumo eléctrico	13
2. DISEÑO CONCEPTUAL	15
2.1 Caja negra.....	16
2.2 Estructura de funciones	17

2.2.1	Estructura de funciones - Base de datos.....	21
2.2.2	Estructura de funciones - Interfaz	21
2.2.3	Estructura de funciones – Dispositivos	22
2.2.3.1	Dispositivo de iluminación	22
2.2.3.2	Dispositivo de toma de corriente	23
2.3	Matriz Morfológica	23
2.3.1	Dispositivo de toma de corriente.....	23
2.3.2	Dispositivo de iluminación	24
2.3.3	Base de datos e interfaz:.....	24
2.4	Conceptos de solución.....	25
2.4.1	Concepto de solución 1	25
2.4.2	Concepto de solución 2	26
2.4.3	Concepto de solución 3	27
2.5	Evaluación técnico – económica.....	28
2.6	Concepto de solución óptimo.....	30
3.	DISEÑO DE INGENIERÍA	32
3.1	Diagrama de Bloques	33
3.2	Selección de sensores y actuadores.....	33
3.2.1	El sensor de corriente	33
3.2.2	El microcontrolador	35
3.2.3	El relé	36
3.2.4	El dimmer.....	38
3.2.5	El acondicionador de energía.....	39
3.3	Diagrama de flujo.....	42
3.4	Diseño mecánico	44
3.4.1	Materiales:.....	44
3.4.2	Diseño 3D del dispositivo de tomacorriente:	44
3.4.3	Diseño 3D del dispositivo de iluminación:	48
3.4.4	Cálculos mecánicos:.....	50
3.5	Diseño electrónico.....	53
3.5.1	Esquemáticos de dispositivos electrónicos:	54
3.5.2	Cálculos electrónicos:	57
3.6	Diseño de softwares	60
3.6.1	Descripción de las tecnologías a usar.....	60
3.6.1.1	Git y GitHub	60

3.6.1.2	PuTTY	60
3.6.1.3	HeidiSQL	61
3.6.1.4	Figma	61
3.6.1.5	Lenguajes de programación	61
3.6.2	Selección de los entornos de programación	62
3.6.2.1	Editor de código	62
3.6.2.2	Framework para desarrollo web	63
3.6.2.3	Framework para desarrollo móvil	63
3.6.2.4	Framework para desarrollo en microcontroladores	63
3.6.3	Arquitectura del sistema IoT:	64
3.6.4	Selección de los servicios	67
3.6.4.1	Plataforma cloud computing	67
3.6.4.2	Hosting	67
3.6.4.3	Broker	68
3.6.5	Medidas de seguridad	68
3.6.6	Gestión del proyecto – metodología ágil SCRUM	68
4.	IMPLEMENTACIÓN DE LOS APLICATIVOS WEB Y MÓVIL	69
4.1	Preparación para el desarrollo de los aplicativos	69
4.1.1	Creación del logo de SmartHouse	69
4.1.2	Preparación del editor de código	69
4.2	Diseño de la base de datos	70
4.2.1	Descripción de los requerimientos	70
4.2.2	Diagrama entidad – relación	70
4.2.3	Tablas creadas	71
4.3	Broker MQTT	71
4.3.1	Diseño de la estructura de tópicos a usar	71
4.4	Programación del servicio node.js	72
4.4.1	Lógica de programación del servicio node.js	72
4.5	Programación de los dispositivos IoT	73
4.5.1	Lógica de programación de los dispositivos	73
4.6	Programación de la aplicación web	76
4.6.1	Descripción de las pantallas de la aplicación web	77
4.6.2	Lógica de programación de la aplicación web	80
4.7	Programación de la aplicación móvil	82
4.7.1	Descripción de las pantallas de la aplicación móvil	82
4.7.2	Lógica de programación de la aplicación móvil	84

5. PROTOTIPADO Y VALIDACIÓN DEL SISTEMA.....	85
5.1 Prototipos	85
5.1.1 Prototipo del dispositivo de tomacorriente:	87
5.1.2 Prototipo del dispositivo de iluminación:	88
5.2 Pruebas con la aplicación web	89
5.3 Pruebas de dispositivos	91
5.3.1 Pruebas de dispositivo de tomacorriente.....	91
5.3.2 Pruebas del dispositivo de luces.....	92
5.3.3 Pruebas de envío mensajes entre los clientes con el broker.....	93
6. PRESUPUESTOS.....	95
6.1 Presupuesto del prototipo del dispositivo de iluminación:	95
6.2 Presupuesto del prototipo del dispositivo de tomacorriente:	96
6.3 Presupuesto del diseño de ingeniería:	96
CONCLUSIONES	97
RECOMENDACIONES	98
BIBLIOGRAFÍA	99
ANEXOS	102



ÍNDICE DE FIGURAS

Figura 1.1: Proyectos y stock de unidades inmobiliarias por distrito en Lima	2
Figura 1.2: Consumo de energía eléctrica per cápita por habitante 1995 – 2016	3
Figura 1.3: Domótica en el Perú	4
Figura 1.4: Arquitectura final cliente - servidor.....	5
Figura 1.5: Diagrama de bloques del sistema inteligente de ahorro de energía eléctrica	6
Figura 1.6: Diagrama de bloques del sistema domótico	6
Figura 1.7: Arquitectura del sistema de monitoreo y control.....	7
Figura 1.8: Tomacorrientes smart de la compañía Smart House Perú.....	7
Figura 1.9: Relay Switch System de la compañía Domotik	8
Figura 1.10: Sistema domótico de activa building control	8
Figura 1.11: Ranking gráfico de las tecnologías IoT	10
Figura 1.12: Problema de comunicación entre sistemas IoT	11
Figura 1.13: Participación de mercado de las nubes públicas.....	12
Figura 1.14: Comparación de estructura de bases de datos SQL con NoSQL.....	13
Figura 1.15: Esquema de conexión del sensor ACS712	13
Figura 1.16: Esquema de funcionamiento del sensor SCT-013	14
Figura 2.1: Distribución de los dispositivos del sistema.....	16
Figura 2.2: Caja negra	16
Figura 2.3: Base de datos	17
Figura 2.4: Interfaz.....	18
Figura 2.5: Denominación de dispositivos por colores.....	18
Figura 2.6: Dominio mecánico.....	18
Figura 2.7: Dominio de energía	19
Figura 2.8: Dominio de los sensores	19
Figura 2.9: Dominio de los actuadores	20
Figura 2.10: Dominio de control - parte 1.....	20
Figura 2.11: Dominio de control - parte 2.....	20
Figura 2.12: Concepto de solución 1	26
Figura 2.13: Concepto de solución 2	27
Figura 2.14: Concepto de solución 3	28
Figura 2.15: Evaluación técnico – Económica.....	30
Figura 2.16: Concepto de solución óptimo	31
Figura 3.1: Distribución 3D de los dispositivos en el hogar	32
Figura 3.2: Diagrama de bloques del sistema	33
Figura 3.3: Sensor de corriente ACS712.....	34
Figura 3.4: Wemos D1 mini.....	36
Figura 3.5: Relé electromecánico.....	37
Figura 3.6: Dimmer AC 220V	39
Figura 3.7: Acondicionador de energía 220V AC a 5V DC	41
Figura 3.8: Diagramas de flujo general del dispositivo para tomacorrientes.....	42
Figura 3.9: Diagramas de flujo general del dispositivo para luces	43
Figura 3.10: Función de cálculo de variables.....	43
Figura 3.11: Diseño mecánico del dispositivo para tomacorrientes - frontal	45
Figura 3.12: Distribución interna del dispositivo para tomacorrientes 1	45
Figura 3.13: Distribución interna del dispositivo para tomacorrientes 2.....	46
Figura 3.14: Diseño mecánico del dispositivo para tomacorrientes - posterior.....	46
Figura 3.15: Distribución interna de la tapa del dispositivo para tomacorrientes.....	47

Figura 3.16: Medidas internas del dispositivo para tomacorrientes.....	47
Figura 3.17: Dispositivo de iluminación - contenedor.....	48
Figura 3.18: Dispositivo de iluminación - Lugares de los dispositivos electrónicos.	48
Figura 3.19: Dispositivo de iluminación - Tapa.....	49
Figura 3.20: Dispositivo de iluminación - columnas de tapa.....	49
Figura 3.21: Medidas internas del dispositivo para luces	50
Figura 3.22: Diagrama de cuerpo libre (DCL) del dispositivo tomacorriente	53
Figura 3.23: Diseño electrónico inicial del dispositivo para tomacorrientes	54
Figura 3.24: Diagrama de entradas y salidas del Wemos D1 mini	54
Figura 3.25: Esquemático del Wemos D1 mini	55
Figura 3.26: Esquemático del HLK-PM01	55
Figura 3.27: Encapsulado del sensor de corriente ASC712.....	56
Figura 3.28: Esquemático del Relé	56
Figura 3.29: Esquemático del dimmer AC.....	57
Figura 3.30: Relación entre las corrientes en la onda	59
Figura 3.31: Overview del perfil personal en GitHub a utilizar	60
Figura 3.32: Interfaz de Figma para el diseño de aplicaciones	61
Figura 3.33: Cadena de valor del IoT.....	64
Figura 3.34: Esquema básico IoT.....	65
Figura 3.35: Esquema avanzado IoT.....	65
Figura 3.36: Funcionalidades del servidor	66
Figura 3.37: Arquitectura IoT propuesta.....	67
Figura 4.1: Logo de smarthouse.....	69
Figura 4.2: Diagrama entidad - relación	70
Figura 4.3: Tabla devices creada en la base de datos de acuerdo al diseño	71
Figura 4.4: Programación en node.js cuando se recibe el mensaje	73
Figura 4.5: Entorno de platformIO en VSCode	73
Figura 4.6: Declaración de funciones a utilizar	74
Figura 4.7: Cálculo de valores	75
Figura 4.8: Callback	75
Figura 4.9: Repositorio en GitHub para los dispositivos	76
Figura 4.10: Conexión VSCode FTP	77
Figura 4.11: App web – pantalla de ingreso	77
Figura 4.12: App web – pantalla de registro	78
Figura 4.13: App web – pantalla de dashboard de dispositivos.....	78
Figura 4.14: App web – pantalla de registro de casas.....	79
Figura 4.15: App web – pantalla de registro de habitaciones	79
Figura 4.16: App web – pantalla de registro de dispositivos	80
Figura 4.17: App web – pantalla de dashboard de las habitaciones y de las casas....	80
Figura 4.18: App web – codificación de la conexión al broker en VSCode.....	81
Figura 4.19: Repositorio en GitHub para la App web	81
Figura 4.20: App móvil – dashboard dispositivos	82
Figura 4.21: App móvil – dashboard menú lateral.....	83
Figura 4.22: App móvil – seleccionadores superiores	83
Figura 4.23: App móvil – dispositivos.....	84
Figura 5.1: Dispositivos de tomacorrientes e iluminación.....	85
Figura 5.2: Dispositivo de toma de corriente	86
Figura 5.3: Dispositivo de iluminación.....	86
Figura 5.4: Dispositivo para tomacorrientes - vista trasera.....	87
Figura 5.5: Dispositivo para tomacorrientes - vista isométrica	88

Figura 5.6: Interior del prototipo del dispositivo de iluminación	88
Figura 5.7: Tapa y contenedor del dispositivo de iluminación	89
Figura 5.8: Casa principal	90
Figura 5.9: Cuarto de prueba01	90
Figura 5.10: Celular y laptop	90
Figura 5.11: Seleccionando los valores necesarios para probar funcionamiento de aplicación web.....	91
Figura 5.12: Dispositivos desconectados	92
Figura 5.13: Laptop conectada	92
Figura 5.14: Dispositivo de Luces	93
Figura 5.15: Prueba de envío de mensajes del broker.....	93
Figura 5.16: Prueba de envío de mensajes del broker 2.....	94



ÍNDICE DE TABLAS

Tabla 1-1: Comparación de las redes de comunicación más utilizadas	10
Tabla 1-2: Características principales de algunos protocolos	11
Tabla 1-3: Plataformas más comunes y sus propiedades	12
Tabla 2-1: Dispositivo de toma de corriente	24
Tabla 2-2: Dispositivo de Iluminación.....	24
Tabla 2-3: Base de datos e interfaz	25
Tabla 2-4: Evaluación técnica.....	29
Tabla 2-5: Evaluación económica.....	29
Tabla 2-6: Distancia de soluciones	30
Tabla 3-1: Cuadro comparativo de sensores de corriente	34
Tabla 3-2: Cuadro comparativo de microcontroladores.....	35
Tabla 3-3: Cuadro comparativo de relés	37
Tabla 3-4: Cuadro comparativo de Dimmer	38
Tabla 3-5: Consumo de los componentes del dispositivo tomacorriente.....	39
Tabla 3-6: Consumo de los componentes del dispositivo de iluminación	40
Tabla 3-7: Cuadro comparativo de acondicionadores de energía	41
Tabla 3-8: Lista de planos	44
Tabla 3-9: Masa calculada de las impresiones 3D de los dispositivos	51
Tabla 3-10: Peso de los componentes del dispositivo tomacorriente	51
Tabla 3-11: Peso de los componentes del dispositivo de iluminación.....	51
Tabla 3-12: Gestión de proyectos – metodología scrum.....	68
Tabla 5-1: Parámetros de impresión 3D de los prototipos.....	87
Tabla 6-1: Costos del prototipo del dispositivo de iluminación	95
Tabla 6-2: Costos del prototipo del dispositivo de tomacorriente	96

ÍNDICE DE ECUACIONES

Ecuación 3-1: Potencia requerida.....	40
Ecuación 3-2: Masa de filamento.....	50
Ecuación 3-3: Peso del dispositivo	52
Ecuación 3-4: Momentos	52
Ecuación 3-5: Fuerza resultante.....	52
Ecuación 3-6: Prueba de retención o sujeción de enchufes.	53
Ecuación 3-7: Voltaje escalado.....	57
Ecuación 3-8: Corriente calculada en cada lectura de sensor	57
Ecuación 3-9: Filtro pasabajos de la corriente calculada en cada lectura del sensor .	58
Ecuación 3-10: Corriente pico - pico	58
Ecuación 3-11: Corriente pico	58
Ecuación 3-12: Corriente Irms	58
Ecuación 3-13: Potencia.....	59
Ecuación 3-14: Consumo de energía	59
Ecuación 3-15: Consumo de energía cada 5s.....	59



INTRODUCCIÓN

En esta tesis, se presenta el diseño de un sistema ciberfísico inteligente para el monitoreo y control del consumo de energía eléctrica en el hogar. Este sistema debe ser escalable y estar preparado para su uso por parte de una cantidad grande de usuarios y su conexión y comunicación en tiempo real con muchos dispositivos por usuario a la vez.

Para ello, en el capítulo 1, se trata la definición del problema, que es esencial en las metodologías de diseño, ya que con esto se puede encaminar correctamente hacia lo que se quiere lograr.

Luego de esto, en el capítulo 2, se realiza el diseño conceptual, que es el paso en el cual ya se empieza a buscar soluciones al problema planteado. Para esto, se tiene en cuenta el uso de herramientas muy útiles como la matriz morfológica y la estructura de funciones, que son utilizadas para poder seleccionar una solución óptima que pueda dar solución al problema.

Después se pasará al capítulo 3, el cual se enfoca en el diseño de ingeniería. Se realizan planos, cálculos y diagramas, que ayudan a validar la solución óptima que se ha seleccionado. En este capítulo, se utilizarán todos los conceptos de ingeniería relevantes para que se pueda comprobar la funcionalidad y eficiencia de la solución que buscamos.

Posteriormente, en el capítulo 4, se implementan las aplicaciones web y móvil. Se realiza la preparación para el desarrollo de los aplicativos, el diseño de la base de datos y la estructura de tópicos para el bróker MQTT y las programaciones necesarias.

Continuando, en el capítulo 5, se abarcan las pruebas para la validación del sistema. Esto lo realizaremos con pruebas de funcionamiento por medio de prototipos y las aplicaciones desarrolladas. En este paso, se probará el sistema a través de al menos 2 semanas para ver cómo reacciona en la red Wifi.

Por último, en el capítulo 6, se detalla el presupuesto necesario para los prototipos diseñados, tomando en cuenta que toda la implementación del desarrollo móvil y web no tienen costo al estar en etapa de prototipaje.

CAPÍTULO 1

1. DEFINICIÓN DEL PROBLEMA

En este capítulo, se presenta la problemática que se está abordando, así como el alcance y objetivos del presente trabajo. Adicional a ello, se realiza una investigación del estado del arte para poder saber qué tecnologías hay actualmente y cuáles podrían ser útiles para cumplir los objetivos planteados.

1.1 Problemática

Con el boom de la construcción en el Perú, los pobladores están adquiriendo nuevos inmuebles de manera masiva y con ello aumenta la energía eléctrica consumida. Los proyectos inmobiliarios están aumentando y más aún en la capital, siendo que, hasta Julio del 2020, existían 137 proyectos inmobiliarios tan solo en el distrito de Miraflores y más de 2767 unidades inmobiliarias (U.I. por sus siglas) en stock en el distrito de Comas según el informe estadístico del mercado inmobiliario de la asociación de desarrolladores inmobiliarios del Perú (ADIPERÚ por sus siglas) [1]. (Ver figura 1.1)

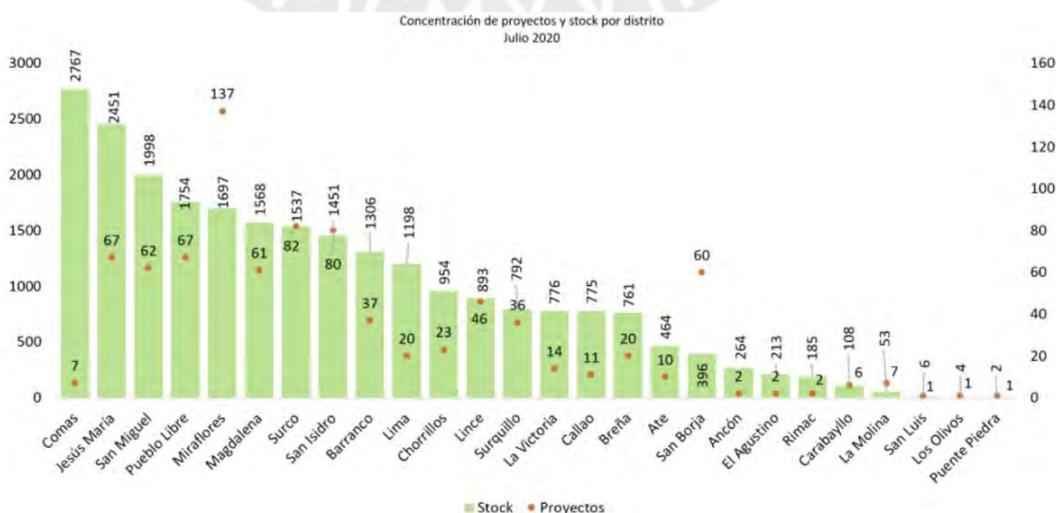


Figura 1.1: Proyectos y stock de unidades inmobiliarias por distrito en Lima. Fuente: [1]

El consumo eléctrico de cada casa o departamento de los peruanos es alto, ya que no hay una cultura de ahorro energético presente en los pobladores. Si bien se puede conocer el consumo mensual, no se conoce el consumo real de cada electrodoméstico o toma de corriente, lo cual no permite realizar un mejor control de consumo de energía total. Cada año, el consumo de energía eléctrica por persona aumenta, siendo para el 2016 la cantidad de 1446 kW.h por cada habitante según el Ministerio de Energía y Minas en su informe de la evolución de indicadores del mercado eléctrico en el Perú [2]. (Ver figura 1.2)

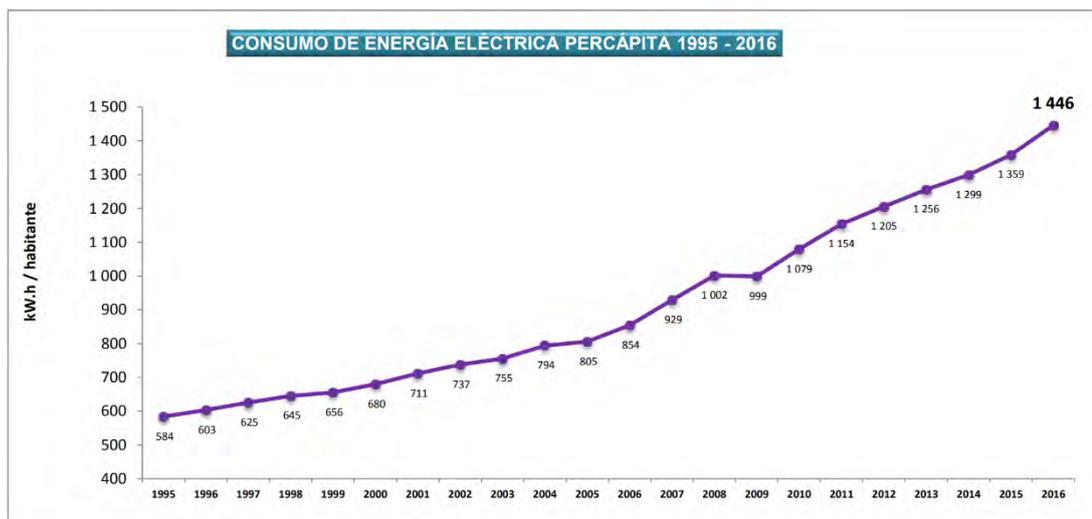


Figura 1.2: Consumo de energía eléctrica per cápita por habitante 1995 – 2016. Fuente: [2]

Con lo anteriormente mencionado, hallar nuevas maneras de controlar el consumo de energía en el hogar se ha convertido en una tarea primordial en las épocas actuales. Por ello, a lo largo de estos años, se han venido desarrollando diversas maneras de realizar esto mismo, ya que se considera una necesidad actual.

La domótica, que son sistemas capaces de automatizar una vivienda de cualquier tipo [3], comenzó en el Perú en los inicios del nuevo milenio, pero no está muy difundida actualmente. Hay una idea errónea con respecto a la domótica, ya que muchas familias la consideran un gasto que no está justificado, pero los dispositivos conectados a los tomacorrientes en las casas, por ejemplo, están en un modo de congelamiento electrónico, el cual consume energía y las personas no lo notan.

En el Perú, hay varias empresas que están en este rubro, tales como LCN (Local Control Network), Activa, Bticino, mtechperu, Domotik, Tec-house, Conauti, Smart House Perú y siguen en aumento. Los productos que ofrecen ahora son tanto cableados, como inalámbricos y los protocolos usados son diversos, pero son muy caros y por ello no son accesibles a todos los sectores de la población. Se centran en entornos corporativos e industriales y en nuevos edificios que se encuentran en proyecto. Además de no gestionar la información y solo basarse en el control de los dispositivos, lo cual no permite tener datos históricos que puedan ser analizados posteriormente. (Ver figura 1.3)



Figura 1.3: Domótica en el Perú. Fuente: [3]

Lo que se busca lograr con el diseño de un sistema inteligente es presentar una opción accesible, estable y remota para el monitoreo y control de energía eléctrica. Además de esto, se pretende solucionar el problema del alto consumo eléctrico por habitante que se tiene actualmente en las casas y departamentos ya terminados de una manera ecológica y aumentando el confort en el hogar.

1.2 Objetivos

1.2.1 Objetivo General

El objetivo es el de diseñar un sistema inteligente para el monitoreo y control de consumo de energía eléctrica en el hogar que permita a una persona reducirlo y optimizarlo.

1.2.2 Objetivos específicos

Los objetivos específicos son los siguientes:

- Realizar la selección de los componentes mecánicos y electrónicos que se van a necesitar para el sistema ciberfísico diseñado.
- Realizar el diseño de ingeniería de los dispositivos, la aplicación web y aplicación móvil y conectarlos como parte de un sistema IoT global interconectado.
- Realizar la lógica de control de cada uno de los subsistemas.
- Realizar el diseño electrónico total del sistema.
- Realizar un diseño mecánico 3D de los dispositivos para implementar prototipos usando impresión 3D.
- Analizar los costos de los dispositivos, procesos de fabricación y servicios en la nube.
- Desarrollar la aplicación web escalable de cara a muchos usuarios alojada en una plataforma en la nube para su uso con los prototipos visualizando los datos en tiempo real, así como el control de los mismos.

1.3 Alcance

El sistema inteligente para el monitoreo y control de consumo de energía en el hogar que será diseñado está enfocado para el ahorro energético controlando inicialmente el paso de energía eléctrica en las tomas de corriente y lámparas del hogar. Con esto, el proyecto se podría aplicar a todas las familias peruanas. El control se realizaría de manera inalámbrica a través de dispositivos móviles o aplicativos web, con los cuales se podría ver el consumo de corriente que se tiene y poder controlar todos los dispositivos de manera remota, incrementando el confort que se tienen en el hogar.

Esta tesis abarcará tanto el diseño mecánico y electrónico de los dispositivos de control en las tomas de corriente y también de la creación de la página web o de una aplicación para dispositivos móviles escalable en el tiempo y proyectada para trabajar con una cantidad alta de usuarios y dispositivos conectados, con las cuales se podrá acceder de manera remota, monitorear el consumo de energía y también conmutar el estado de las tomas de corriente seleccionando si permite el paso de corriente o no.

1.4 Estado del arte

Se presenta la investigación realizada sobre sistemas propuestos, sistemas comerciales y temas de interés para el presente trabajo.

1.4.1 Propuestas de sistemas de ahorro de energía (tesis)

1.4.1.1 Red de medidores de energía para artefactos domésticos

Este sistema, diseñado por el ingeniero Nelson Eduardo Huiman Tocto, propone adaptar las tecnologías ya existentes dentro de las áreas de medición de energía eléctrica para el desarrollo de una red con la capacidad de enviar información de cada dispositivo conectado a la red a un servidor que centralice todo. El sistema tendría información detallada del consumo de energía eléctrica de los artefactos domésticos [4]. (Ver figura 1.4)

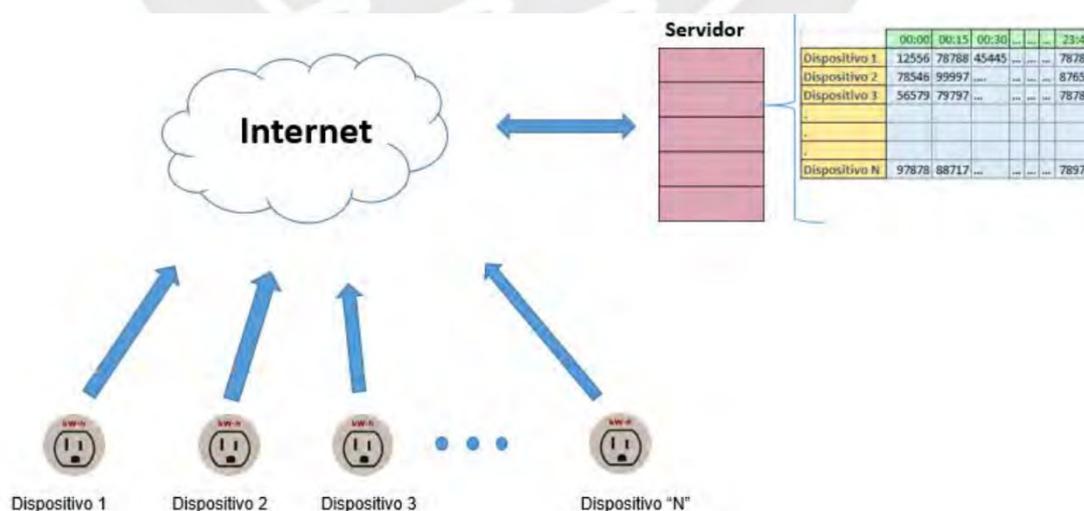


Figura 1.4: Arquitectura final cliente - servidor. Fuente: [4]

1.4.1.2 Sistema inteligente de ahorro de energía eléctrica

Este sistema, diseñado por el ingeniero Luis Felipe Poma Aliaga, es un sistema inteligente por medio de algoritmos de aprendizaje por redes neuronales, las cuales racionalizan el uso de la energía eléctrica. Para medir la potencia eléctrica consumida se usan sensores de potencia y también usan un sensor de presencia. El sistema aprende la rutina del usuario para poder realizar el control inteligente y eficiente de la energía de las casas domésticas. Todo el proceso se da con un uso de una semana como entrenamiento, se registrarán los datos y se procesará todo en un microcontrolador. Luego de esta semana de entrenamiento, la red neuronal podrá predecir y controlar los equipos eléctricos para activarlos o desactivarlos de manera automática [5]. (Ver figura 1.5)

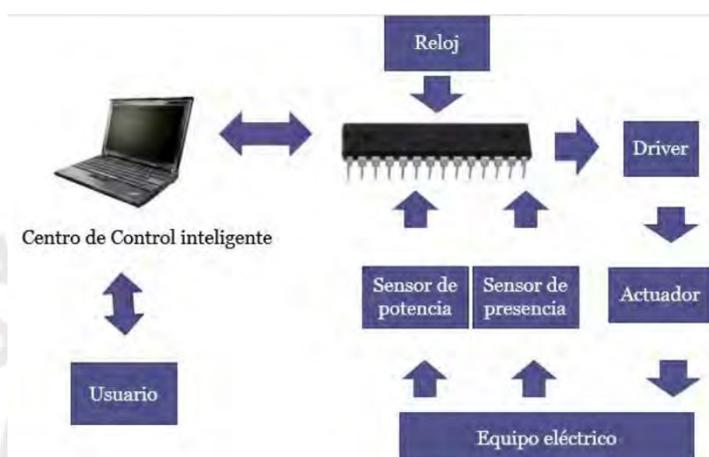


Figura 1.5: Diagrama de bloques del sistema inteligente de ahorro de energía eléctrica. Fuente:[5]

1.4.1.3 Sistema domótico de control centralizado con comunicación por línea de poder

Este sistema, que ha sido diseñado en conjunto por los ingenieros Miguel Guzmán y Renzo Burga, se enfoca en la comunicación de varios dispositivos con un dispositivo central y el control del suministro de energía, la iluminación y la apertura y cierre de las cortinas que pueda haber en la vivienda. Todo esto usando las líneas de poder con el protocolo X10. Se cuenta con una interfaz para un dispositivo móvil que se comunica vía bluetooth a un módulo de control [6]. (Ver figura 1.6)

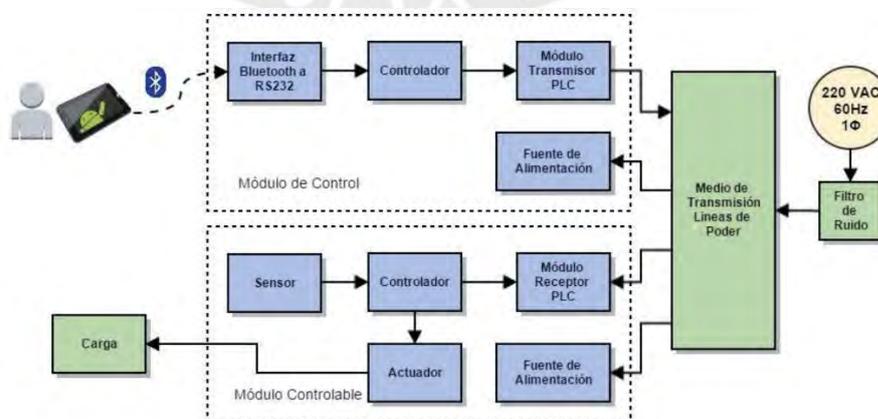


Figura 1.6: Diagrama de bloques del sistema domótico. Fuente: [6]

1.4.1.4 Sistema de monitoreo y control remoto de viviendas

Este sistema, que ha sido diseñado por el ingeniero Juan Culquichicón, se basa en la implementación de un sistema para la interacción de dispositivos electrónicos instalados en cualquier vivienda para monitorear y controlar varios parámetros de esta. El protocolo de comunicación usado es a través de redes Wifi y se conecta a dispositivos móviles por medio de aplicaciones [7]. (Ver figura 1.7)

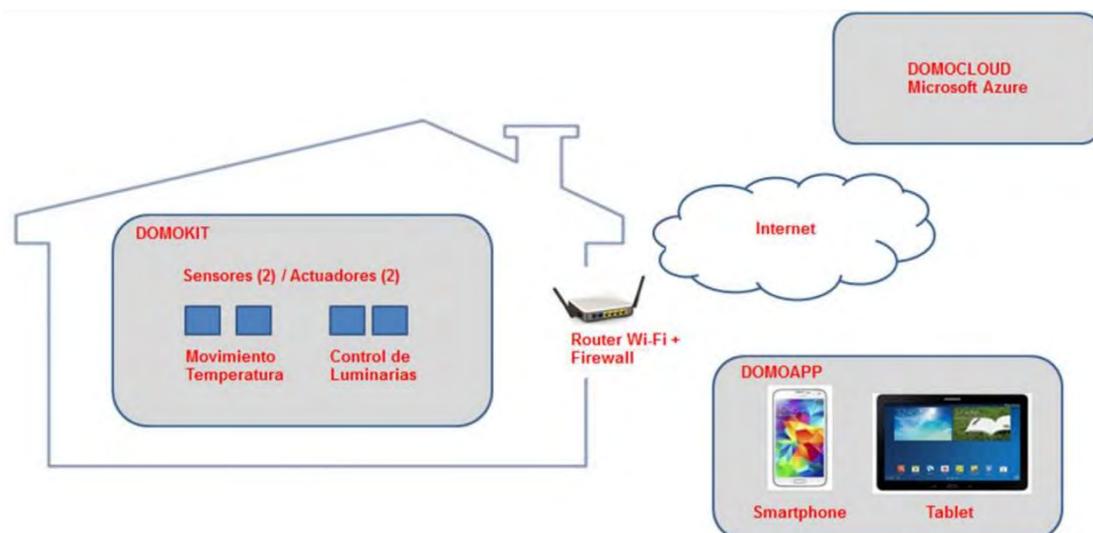


Figura 1.7: Arquitectura del sistema de monitoreo y control. Fuente: [7]

1.4.2 Sistemas de ahorro de energía comerciales

1.4.2.1 Smart House Perú

Es una empresa que se encarga de crear soluciones tecnológicas y de confort para el sector residencial y corporativo. Tienen diferentes productos ya creados y comercializados en diferentes tiendas peruanas. El precio de este dispositivo está actualmente S/. 199 [8]. (Ver figura 1.8)



Figura 1.8: Tomacorrientes smart de la compañía Smart House Perú. Fuente: [8]

1.4.2.2 Relay switch Simple

Es un dispositivo de encendido y apagado de dispositivos por medio de ondas de radio de la compañía Domotik. Este dispositivo funciona con dispositivos de hasta 3 kW. El precio de este dispositivo está actualmente \$121.81 [9]. (Ver figura 1.9)



Figura 1.9: Relay Switch System de la compañía Domotik. Fuente: [9]

1.4.2.3 Activa Building control

Es una compañía que se encarga de crear sistemas muy completos de automatización del hogar. La domótica es uno de sus principales negocios y, junto a otras empresas dedicadas a la domótica, han desarrollado sistemas con interfaz en aplicaciones para el control del hogar en edificios [10]. (Ver figura 1.10)

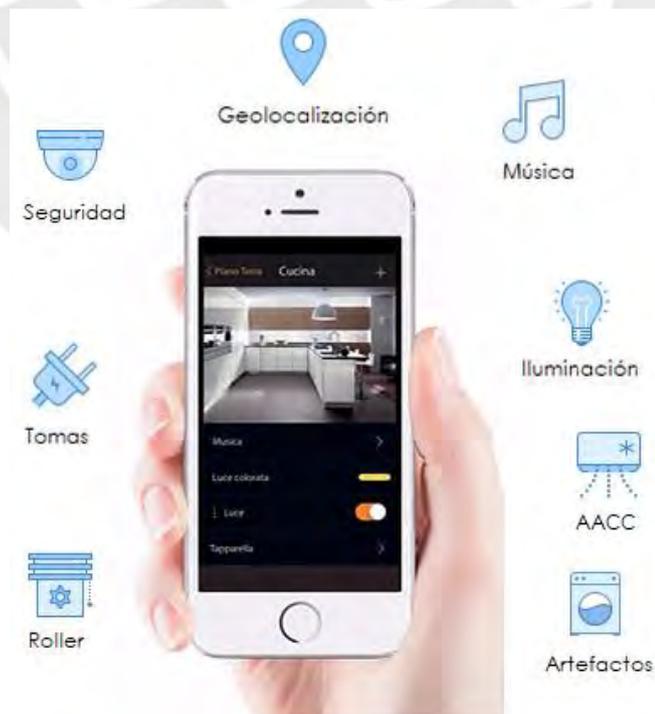


Figura 1.10: Sistema domótico de activa building control. Fuente: [10]

1.4.3 Industria 4.0

Es un término que se refiere a un nuevo modelo de organización para la producción en la industria por medio de nuevas tecnologías interconectadas y la transformación digital [11].

- Impresión 3D: Producción con diversos materiales de objetos mecánicos diseñados en 3D. Permite la creación rápida de prototipos y aumentará la eficiencia de la producción [11].
- Internet de las cosas (IoT): Es un ecosistema en el cual la comunicación, almacenamiento y conectividad se da por medio del internet sin la dependencia del ser humano. En este ecosistema, se encuentran diversos dispositivos con diversas tareas. Pueden ser dispositivos muy simples o también sistemas muy complejos [11].
- Sistemas ciberfísicos: Son los habitantes del ecosistema del internet de las cosas (IoT). Son sistemas que disponen de electrónica, software embebido, y conectividad lo que proporciona la capacidad de comunicación máquina a máquina (M2M), interacción con los humanos. Poseen softwares que los vuelven autónomos y toman decisiones por ellos mismos [11].
- Cloud computing: O computación en la nube, es un término que se refiere al acceso remoto a diferentes servicios en la web, tales como softwares, almacenamiento, procesamiento de archivos, etc. por medio de internet [11].

1.4.4 Tecnologías de comunicación normalmente usadas por los sistemas domóticos

- X10: Es un protocolo de comunicación para dispositivos electrónicos por medio del cableado eléctrico para enviar y recibir señales [12].
- ZigBee: Es un protocolo de comunicación inalámbrica que se caracteriza por el bajo consumo de energía y también por su bajo costo. Este protocolo utiliza la radio frecuencia para comunicarse con los elementos de red [13].
- Wifi: es un protocolo de comunicación inalámbrica para el envío y recepción de datos y está basado en IPs. Este protocolo actualmente es el usado en dispositivos móviles y se aplica a muchos elementos en el hogar [13].
- GSM/GPRS: Estas redes de comunicación están vinculadas siempre a la tarjeta SIM. Las tecnologías que tenemos son 3G, 4G y pronto el 5G [14].
- Sigfox: es una red LPWAN (Low-power Wide-area network) específica para IoT. Tiene una cobertura del 98% del territorio europeo y americano. Su comunicación es unidireccional [14].
- LoRa: Es una red LPWAN preparada para comunicaciones bidireccionales en tiempo real para IoT [14].

- **NarrowBand IoT:** Es una red LPWAN de las operadoras de telecomunicaciones a nivel global. Funciona dentro del rango del LTE o 4G [14].
- **Bluetooth de baja energía (BLE):** Es una tecnología inalámbrica de comunicaciones de bajo consumo energético para interoperar pequeños dispositivos mandando pequeños paquetes de datos [14]. (Ver figura 1.11) (Ver tabla 1-1)

Tabla 1-1: Comparación de las redes de comunicación más utilizadas. Fuente: [14]

TECNOLOGÍA	CONSUMO	ALCANCE	MADUREZ	DISPONIBILIDAD	SEGURIDAD	USABILIDAD	TASA DE DATOS
GSM/GPRS	Muy alto	Alto	Muy Alto	Muy alto	Alta	Alta	Alta
SigFox	Bajo	Medio	Alto	Medio	Media	Alta	Muy baja
LoRa	Bajo	Medio	Bajo	Muy bajo (ad hoc)	N A	Baja	Muy baja
NB IoT							
WiFi	Alto	Bajo	Muy alto	Alto	Baja	Alta	Muy alta
BLE	Muy bajo	Muy bajo	Alto	Bajo	Baja	Media	Baja
ZigBee	Medio	Bajo	Medio	Muy bajo	Alta	Baja	Baja



Figura 1.11: Ranking gráfico de las tecnologías IoT. Fuente: [14]

1.4.5 Protocolos web de envío de datos para IoT

- **HTTP:** (Hypertext Transfer Protocol) Es un protocolo que permite realizar una petición de datos y recursos, como pueden ser documentos HTML. Es la base de cualquier intercambio de datos en la web. Tiene una estructura cliente – servidor [15].
- **MQTT:** (Message Queue Telemetry Transport) Es un protocolo de publicación suscripción que actúa sobre TCP. Es muy ligero y sencillo de implementar. Apropiado para dispositivos de baja potencia [15].
- **XMPP:** (Extensible Messaging and Presence Protocol) Es un protocolo abierto basado en XML diseñado para aplicaciones de mensajería instantánea [15].

- CoAP: (Constrained Application Protocol) Es un protocolo pensado para emplearse en dispositivos de IoT de baja capacidad [15].
- AMQP: (Advanced Message Queuing Protocol) Es un protocolo de publicación suscripción diseñado para asegurar la confiabilidad e interoperabilidad. Diseñado para aplicaciones corporativas, con mayor rendimiento y redes de baja latencia [15]. (Ver figura 1.12) (Ver tabla 1-2)

Tabla 1-2: Características principales de algunos protocolos. Fuente: [15]

	Transporte	Modelo	Ámbito de aplicación	Conocimiento del contenido	Datos principales	Seguridad	Prioridad de los datos	Tolerancia a fallos
AMQP	TCP/IP	Intercambio de mensajes punto a punto	D2D D2C C2C	Ninguno	Codificados	TLS	Ninguno	Específica de la implementación
CoAP	UDP/IP	Petición/Respuesta (REST)	D2D	Ninguno	Codificados	DTLS	Ninguno	Descentralizado
DDS	UDP/IP (unicast + mcast) TCP/IP	Publicación/Suscripción Petición/Respuesta	D2D D2C C2C	Enrutamiento basado en el contenido, consultas	Declarados codificados	TLS, DTLS, DDS	Prioridades de transporte	Descentralizado
MQTT	TCP/IP	Publicación/Suscripción	D2C	Ninguno	No definidos	TLS	Ninguno	El nodo central (broker) es el punto único de fallo (SPoF)



Figura 1.12: Problema de comunicación entre sistemas IoT. Fuente: [15]

1.4.6 Plataformas IoT

- Ubidots: Es una de las herramientas más famosas para recopilación, análisis y visualización de información generada por dispositivos IoT en el mercado [16].
- ThingSpeak: Es una plataforma de código abierto en la que se puede hacer recopilación de información de dispositivos IoT y analizar y visualizar con las herramientas de MATLAB [16].
- Arduino cloud: Plataforma que permite escribir código, compilarlo y subirlo directamente desde el buscador web, conectar los dispositivos IoT y visualizarlos en la plataforma [16]. (Ver tabla 1.3)

Tabla 1-3: Plataformas más comunes y sus propiedades. Fuente: [16]

PLATAFORMAS IOT	Open Source	Visualización rápida de datos	Escalabilidad	Análisis	Administración de datos	Almacenamiento de bases de datos	Seguridad
IBM Watson IoT		✓	✓	✓	✓	✓	✓
Microsoft Azure IoT Suite	✓	✓	✓	✓	✓	✓	✓
AWS IoT		✓	✓	✓	✓	✓	✓
Google Cloud IoT		✓	✓	✓	✓	✓	✓
ThingWorx (ptc)		✓		✓	✓		✓
GE Predix			✓	✓			✓
Kaa	✓	✓	✓	✓	✓	✓	✓
Macchina.io	✓	✓	✓	✓	✓	✓	✓
Thingspeak	✓	✓		✓	✓	✓	
Ubidots		✓	✓	✓	✓	✓	
Carriots		✓	✓	✓		✓	
MyDevices		✓	✓	✓	✓	✓	✓
Temboo		✓	✓	✓	✓	✓	✓
Initial State	✓	✓	✓	✓	✓		
Thinger.io	✓	✓	✓	✓	✓	✓	

1.4.7 Plataformas de computación en la nube (cloud computing)

Las plataformas de computación en la nube son plataformas que prestan su poder de cómputo a los usuarios a través de diferentes servicios especializados para que realicen diversas tareas digitales.

- Amazon web services: Tiene una cantidad muy grande de servicios disponibles y continúa creciendo a día de hoy [17].
- Microsoft Azure: Tiene mucha fuerza en todo el tema del software como servicio (SaaS por sus siglas en inglés) [17].
- Google Cloud Platform: Tiene menos servicios que las anteriores, pero presenta su estrategia multinube que facilitará la combinación de distintos proveedores [17]. (Ver figura 1.13)

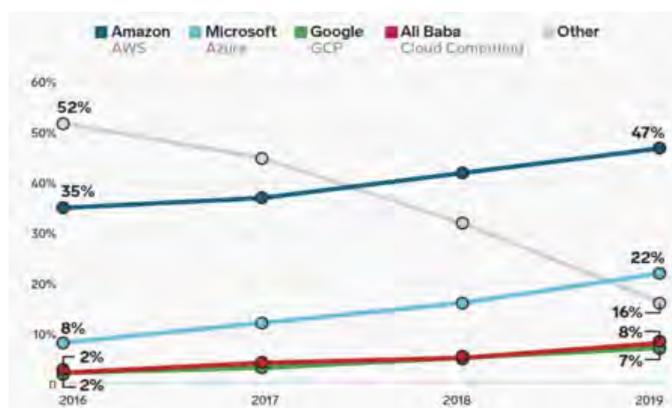


Figura 1.13: Participación de mercado de las nubes públicas. Fuente: [17]

1.4.8 Bases de datos

- SQL: También se le llama bases de datos relacionales. Es un tipo de base de datos estructurada y conformada por tablas, filas, columnas y registros [18].
- NoSQL: También se le llaman bases de datos no relacionales. Poseen un esquema dinámico lo que permite que no se requiera la estructura de los datos para su manipulación [18]. (Ver figura 1.14)

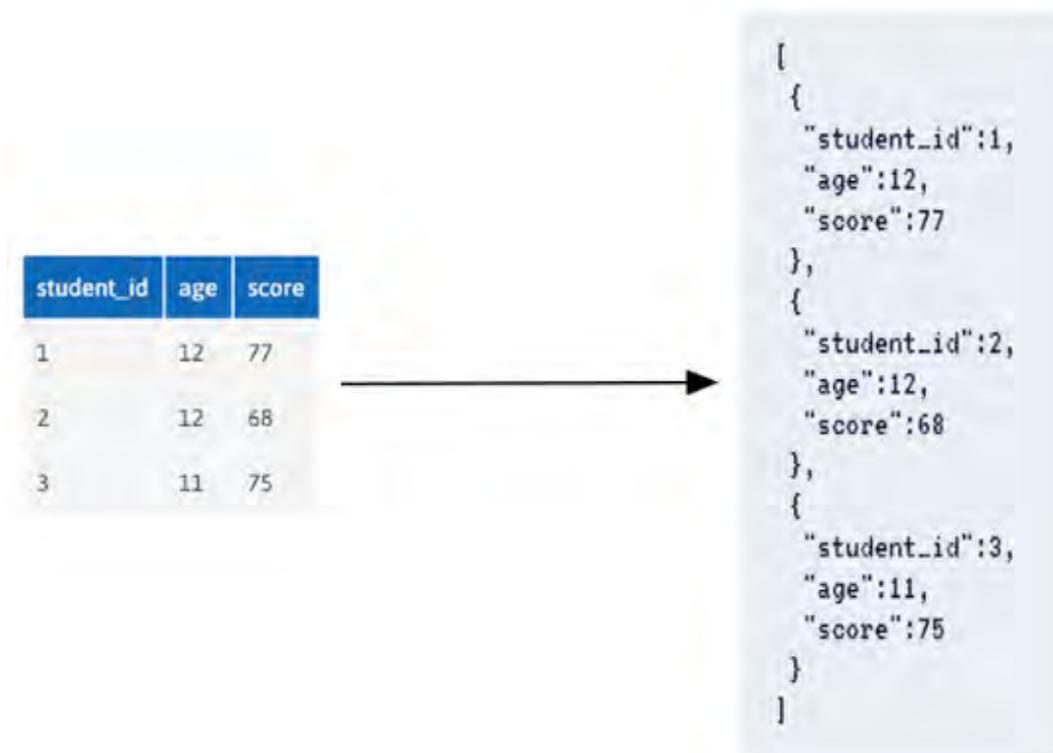


Figura 1.14: Comparación de estructura de bases de datos SQL con NoSQL. Fuente: [18]

1.4.9 Sensores de consumo eléctrico

- ACS712: Es un sensor invasivo de corriente alterna y continua, que mide la intensidad de corriente que pasa por un conductor. Es usado comúnmente con microcontroladores Arduino por su precio económico [19]. (Ver figura 1.15)



Figura 1.15: Esquema de conexión del sensor ACS712. Fuente: [19]

- SCT-013: Es un sensor no invasivo de corriente que trabaja como transformador, ya que la corriente que circula por el cable que se desea medir actúa como un devanado primario e internamente tiene un devanado secundario, que puede tener hasta 2000 espiras [20]. (Ver figura 1.16)

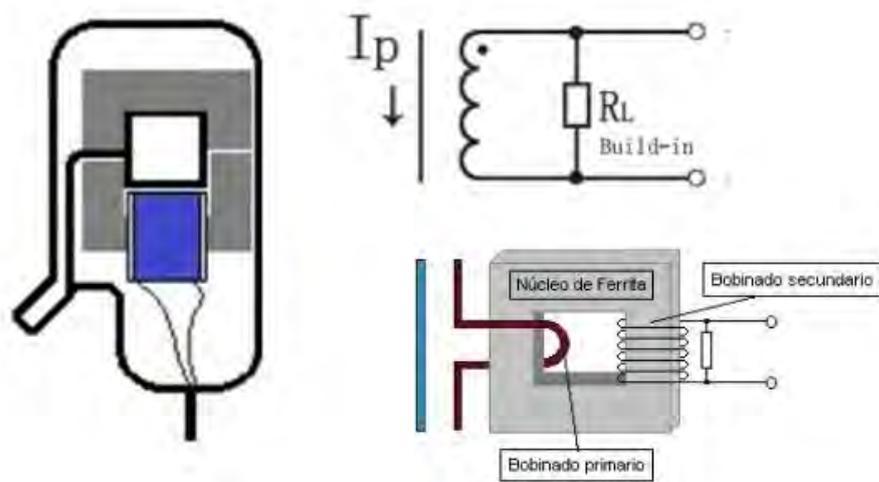
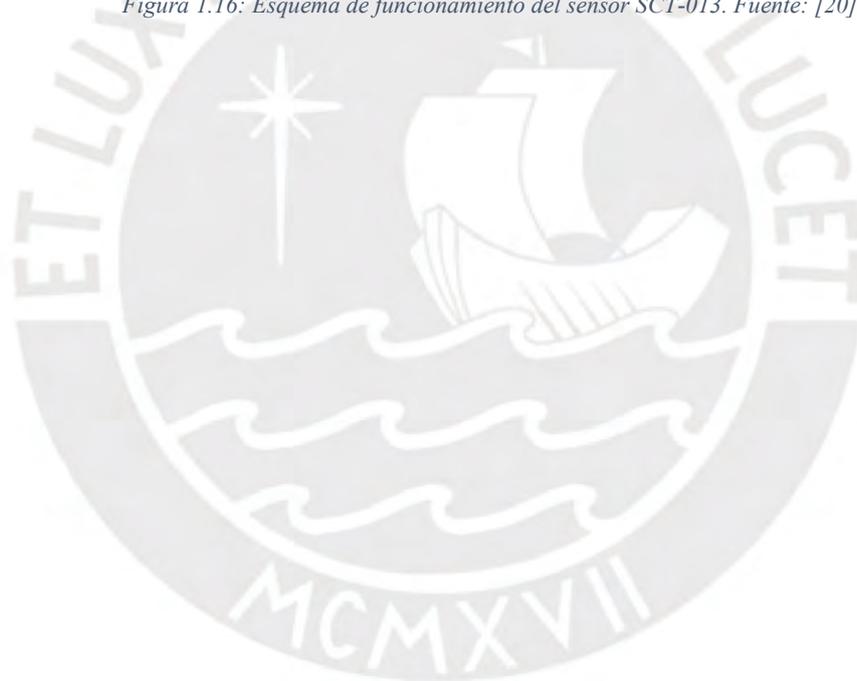


Figura 1.16: Esquema de funcionamiento del sensor SCT-013. Fuente: [20]



CAPÍTULO 2

2. DISEÑO CONCEPTUAL

En este capítulo, se presenta la caja negra, que muestra las entradas y salidas que tendrá el sistema. También se explica la estructura de funciones, así como también la matriz morfológica que se tiene. Luego se muestra los conceptos de solución que se han desarrollado para después realizar una evaluación técnico - económica de los mismos y así elegir el más conveniente. Por último, se presenta el concepto de solución óptimo del sistema.

Además, se detalla el desarrollo de los 2 dispositivos con los que cuenta el sistema. El primero es el dispositivo de control de tomacorrientes, el cual estará ubicado en todos los lugares donde haya tomacorrientes en la casa. El segundo, será el dispositivo de control de luces, el cual se encontrará en los lugares en los que estén los interruptores de luz. El objetivo es que estos sean reemplazados o usados en conjunto.

En la figura 2.1, se tiene un plano de una casa a modo de ejemplo en el que se ubican todos los dispositivos proyectados separándolos en cuadrados de colores, siendo el azul el que representaría a los tomacorrientes y los verdes que representarían a los interruptores de luz.

Se observa que los dispositivos deberán poder adaptarse a cualquier casa o departamento sin importar la distribución eléctrica que tenga. El único requisito que se necesita es que el lugar cuente con cualquier conexión a internet.

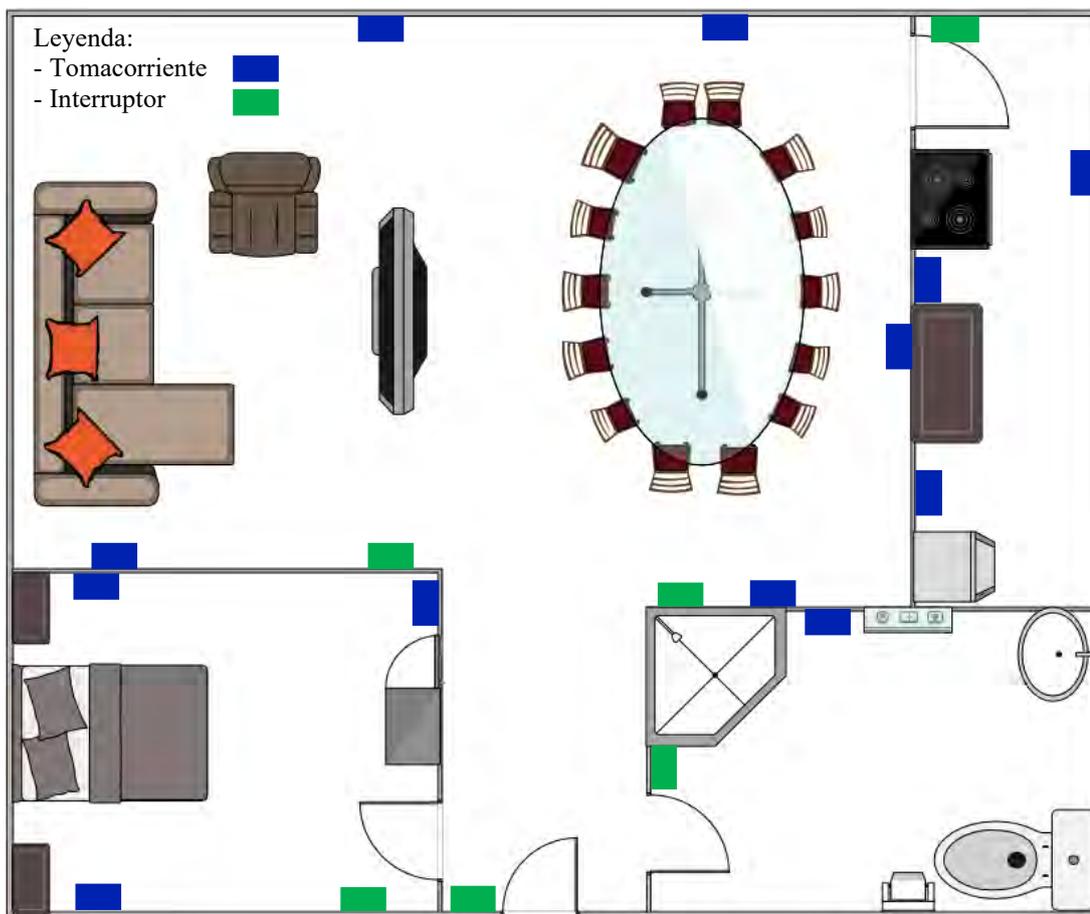


Figura 2.1: Distribución de los dispositivos del sistema. Fuente: [21]

2.1 Caja negra

Las entradas que tendrá el sistema, así como las salidas que producirá, son observadas en la caja negra (black box), en la que se observa 2 subcajas, las cuales representan a cada uno de los dispositivos. Además, en el sistema habrá muchos dispositivos de toma de corriente y de luces, esto se indica con los puntos suspensivos que se encuentran debajo de las subcajas. (Ver figura 2.2)

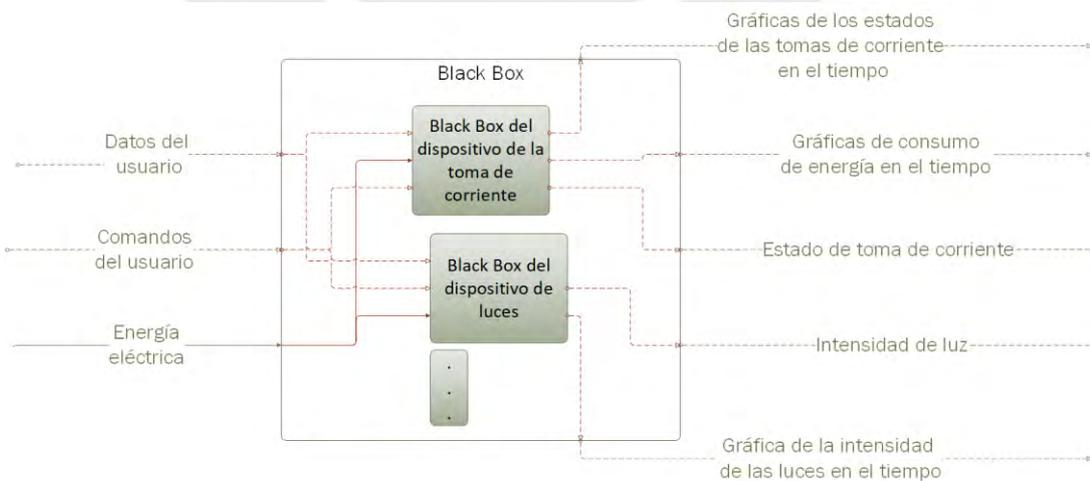


Figura 2.2: Caja negra. Fuente: [21]

El Sistema tendrá como entradas la energía eléctrica, la cual está suministrada en la misma casa y será distribuida a todos los dispositivos. También se realizará el ingreso de los datos del usuario en el interfaz, así como también los comandos que el usuario ejecutará. Las salidas que el sistema va a tener son básicamente las gráficas en la interfaz, tanto de los estados de las tomas de corriente y la intensidad de las luces, como del consumo total de energía. Además, se tendrán la intensidad de la luminosidad de las luces, que será cambiada, y también el estado de las tomas de corriente, que será conmutado por el usuario.

2.2 Estructura de funciones

En la estructura de funciones, se han desarrollado todas las funciones que se necesitarán al interior del sistema. Para ver la estructura de funciones completa, ver Anexo B.

Lo primero que se debe tomar en cuenta es que en la estructura de funciones se han colocado 3 subestructuras que contienen funciones que se realizan en una entidad determinada. La primera entidad identificada es la base de datos, la cual se encarga de almacenar los datos que se transmiten de las otras dos entidades y también de enviar los mismos. (Ver figura 2.3)

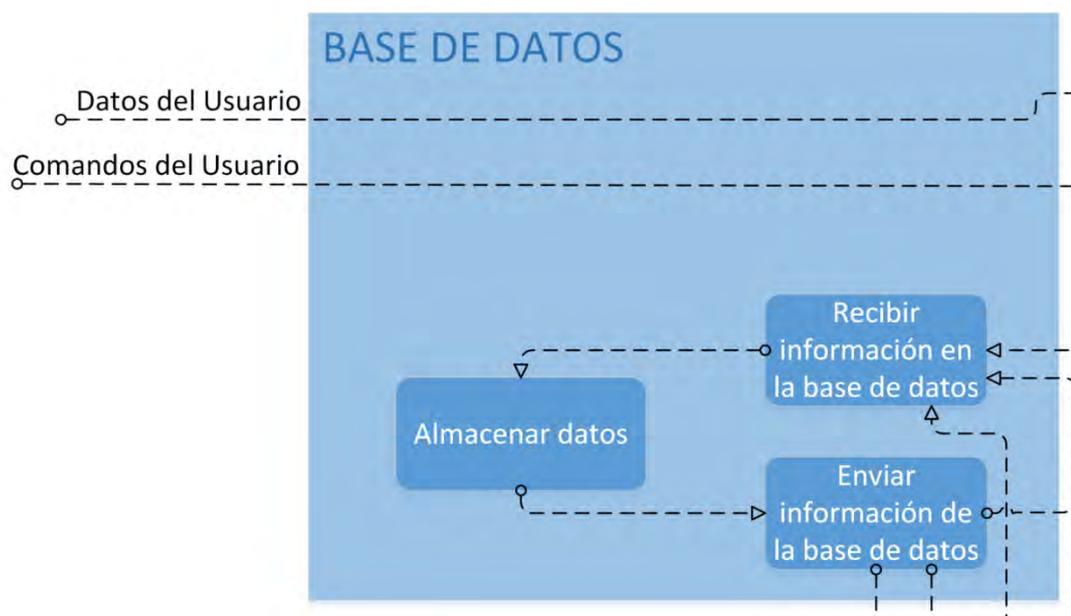


Figura 2.3: Base de datos. Fuente: [21]

La segunda entidad es la interfaz, la cual es la que tendrá contacto directo con el usuario y con ella se podrán realizar diferentes acciones. En esta también se procesarán los datos que se reciben de las otras dos entidades y, una vez procesados, también se enviarán los mismos. (Ver figura 2.4)

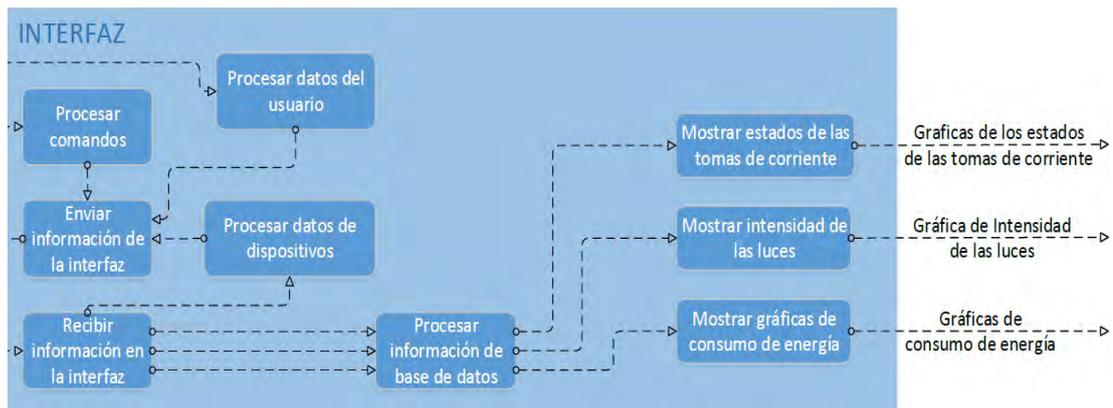


Figura 2.4: Interfaz. Fuente: [21]

Por último, se tiene la última entidad, la cual es la entidad de los dispositivos. Esta entidad se subdivide además en 2 dispositivos que, aunque son similares, tienen algunas funciones que los diferencian (Ver figura 2.5). El primer dispositivo es el dispositivo de iluminación, el cual se va a encargar de controlar la intensidad luminosa del lugar, así como también de conmutar el estado del paso de corriente. El segundo dispositivo se va a encargar de controlar el estado de las tomas de corriente, permitiéndoles el pase de energía eléctrica. Cada dispositivo se comunicará directamente con la base de datos enviando y recibiendo información por medio de funciones especiales.

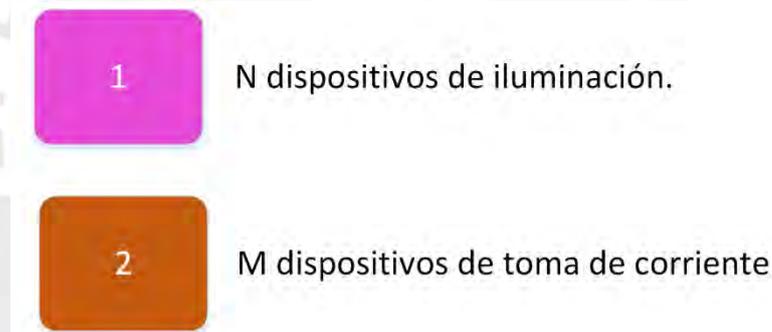


Figura 2.5: Denominación de dispositivos por colores. Fuente: [21]

Estos dispositivos, además, están agrupados en dominios. El dominio mecánico, el cual se representa por el color verde, no tiene conexión directa con el resto de dominios, pero sigue siendo igual de importante. (Ver figura 2.6)

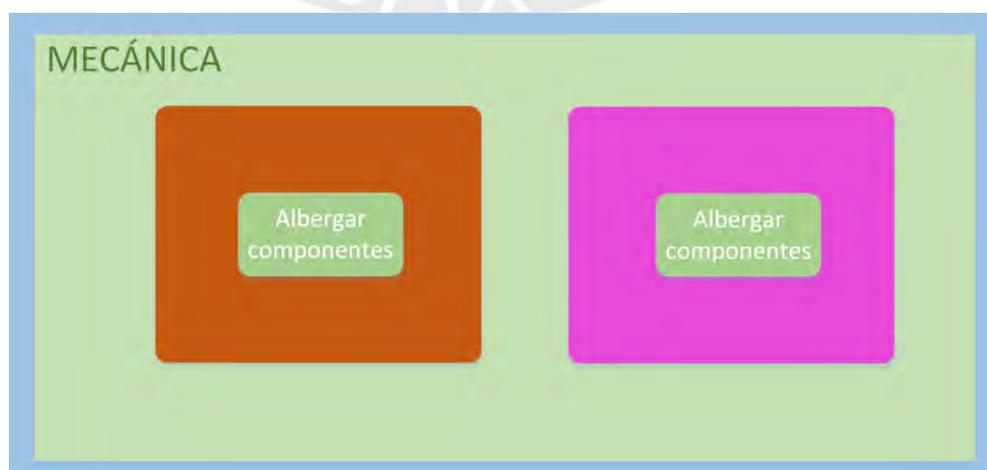


Figura 2.6: Dominio mecánico. Fuente: [21]

El dominio de energía está representado por el color amarillo y además de tener a los 2 dispositivos, también tiene una función que afecta a todos. (Ver figura 2.7)

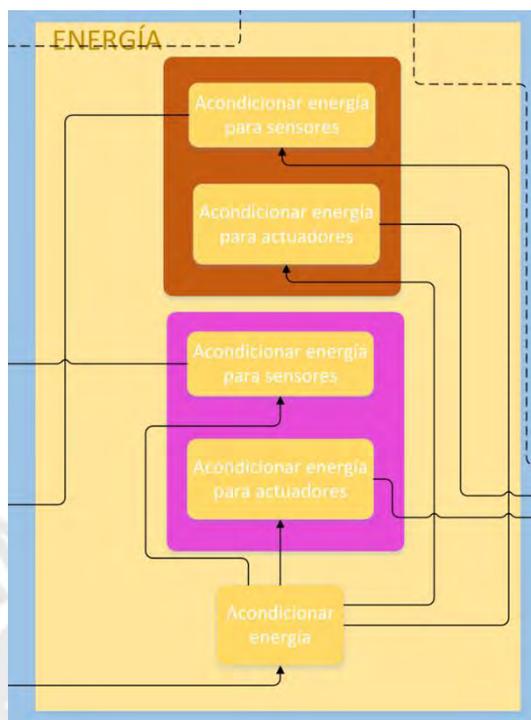


Figura 2.7: Dominio de energía. Fuente: [21]

El dominio de los sensores está representado por el color morado pastel, el cual también tiene a los dos dispositivos. Se puede ver que recibe energía y también envía información. (Ver figura 2.8)

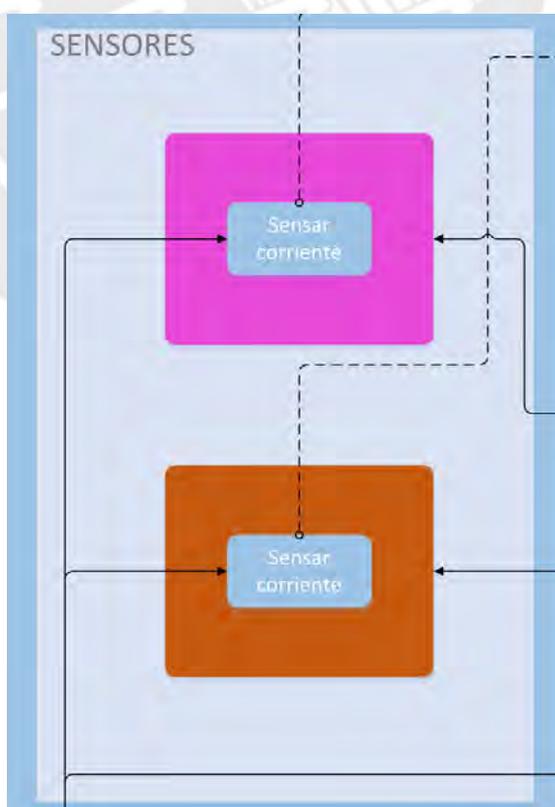


Figura 2.8: Dominio de los sensores. Fuente: [21]

El dominio de los actuadores está representado por el color gris y tiene a los dos dispositivos. También se puede ver el ingreso de energía y la salida de información. (Ver figura 2.9)

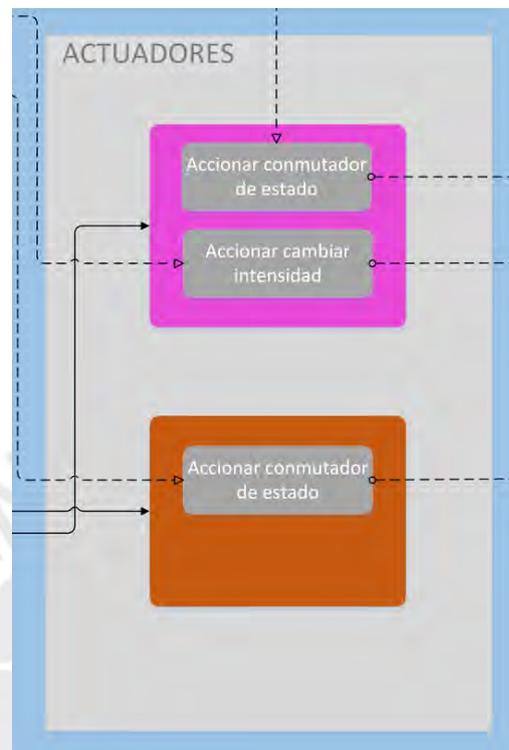


Figura 2.9: Dominio de los actuadores. Fuente: [21]

Por último, se tiene al dominio de control el cual tiene a todos los dispositivos y es el dominio que se comunica con la entidad de la base de datos. (Ver figura 2.10 y 2.11)



Figura 2.10: Dominio de control - parte 1. Fuente: [21]



Figura 2.11: Dominio de control - parte 2. Fuente: [21]

Ahora, se pasará a explicar individualmente las funciones que se tienen en esta estructura.

2.2.1 Estructura de funciones - Base de datos

Solo se tiene 3 funciones:

Almacenar datos:

En esta función, solo almacenarán los datos y también se pueden realizar consultas para poder extraer información.

Recibir información en base de datos:

Se reciben datos desde la interfaz y también desde los dispositivos. Estos datos se almacenarán en la base de datos.

Enviar información de la base de datos:

Se enviará la información que se le solicite a la base de datos a la interfaz y también a los dispositivos.

2.2.2 Estructura de funciones - Interfaz

Se tiene 9 funciones:

Procesar comandos:

Se reciben los comandos del usuario, se procesarán para tomar alguna acción y esa información se transmitirá a la función de enviar.

Procesar los datos del usuario:

Se reciben los datos personales del usuario, se procesarán para poder tener acceso a los dispositivos y se transmitirán a la función de enviar.

Procesar los datos de dispositivos:

Se reciben los datos que envían los dispositivos, se procesarán y luego serán transmitidos por la función de enviar.

Procesar la información de base de datos:

Se reciben los datos enviados por la base de datos, se procesarán para poder ser transmitidos a las funciones de mostrar.

Enviar información de la interfaz:

Se reciben los datos de las funciones procesar comandos, procesar datos del usuario y de procesar datos de dispositivos para poder ser enviados a la base de datos.

Recibir información en la interfaz:

Se reciben datos de la base de datos para poder ser enviados a las funciones de procesar datos de dispositivos y de procesar información de base de datos.

Mostrar estados de las tomas de corriente:

Se reciben los datos procesados de los estados de las tomas de corriente en la función anterior para poder ser mostrados en la interfaz.

Mostrar intensidad de las luces:

Se reciben los datos procesados de la intensidad de las luces en la función anterior para poder ser mostrados en la interfaz.

Mostrar gráficas de consumo de energía:

Se reciben los datos procesados del consumo de energía en la función anterior para poder ser mostrados en la interfaz.

2.2.3 Estructura de funciones – Dispositivos

Ahora pasaremos a presentar las funciones de la entidad de dispositivos separadas por cada uno de los dispositivos que contiene.

2.2.3.1 Dispositivo de iluminación

Se tiene 9 funciones:

Albergar componentes:

Esta función sirve para poder almacenar todos los componentes físicos del dispositivo de iluminación.

Acondicionar energía para sensores:

Se recibiría la energía con un acondicionamiento previo para poder ser acondicionada especialmente para los sensores.

Acondicionar energía para actuadores:

Se recibiría la energía con un acondicionamiento previo para poder ser acondicionada especialmente para los actuadores.

Sensar corriente:

Se recibe directamente la energía eléctrica y se obtienen datos del consumo para poder ser transmitidos.

Procesar datos de dispositivo 1:

Se reciben los datos de la función sensar corriente y de la función de recibir información en dispositivo 1, para que puedan ser procesados. Además, se transmiten los datos para que puedan ser enviados por la siguiente función, así como transmitir órdenes a los actuadores.

Enviar información de dispositivo 1:

Se envían datos procesados previamente a la función de recibir información en la base de datos para que realice su retransmisión.

Recibir información en dispositivo 1:

Se reciben datos desde la función de enviar información de la base de datos directamente y se transmiten para ser procesados.

Accionar conmutador de estado:

Se reciben las órdenes del dominio de control para que se pueda conmutar el estado de las luces.

Accionar cambiar intensidad:

Se reciben las órdenes del dominio de control para que se pueda cambiar la intensidad de las bombillas eléctricas.

2.2.3.2 Dispositivo de toma de corriente

Se tiene 8 funciones:

Albergar componentes:

Esta función sirve para poder almacenar todos los componentes físicos del dispositivo de toma de corriente.

Acondicionar energía para sensores:

Se recibiría la energía con un acondicionamiento previo para poder ser acondicionada especialmente para los sensores.

Acondicionar energía para actuadores:

Se recibiría la energía con un acondicionamiento previo para poder ser acondicionada especialmente para los actuadores.

Sensar corriente:

Se recibe directamente la energía eléctrica y se obtienen datos del consumo para poder ser transmitidos.

Procesamiento de datos de dispositivo 2:

Se reciben los datos de la función sensar corriente y de la función de recibir información en dispositivo 2, para que puedan ser procesados. Además, se transmiten los datos a la siguiente función, así como transmitir órdenes a los actuadores.

Enviar información de dispositivo 2:

Se envían datos procesados previamente a la función de recibir información en la base de datos para que realice su retransmisión.

Recibir información en dispositivo 2:

Se reciben datos desde la función de enviar información de la base de datos directamente y se transmiten para ser procesados.

Accionar conmutador de estado:

Se reciben las órdenes del dominio de control para que se pueda conmutar el estado de los tomacorrientes.

2.3 Matriz Morfológica**2.3.1 Dispositivo de toma de corriente**

Se presentará la matriz morfológica con 3 conceptos de solución diferentes para cada dispositivo o entidad. Para obtener las matrices morfológicas iniciales, ver Anexo C. La primera parte agrupa a las funciones que se aplican al dispositivo de toma de corriente. En este se tienen solo 8 funciones. (Ver tabla 2-1)

Tabla 2-1: Dispositivo de toma de corriente. Fuente: [21]

Funciones	Solución 1	Solución 2	Solución 3
Albergar componentes	Carcasa de plástico	Carcasa de plástico	Carcasa de plástico
Acondicionar energía para sensores	Regulador conmutador	Regulador de tensión lineal	Regulador de tensión lineal integrado
Acondicionar energía para actuadores	Regulador conmutador	Regulador de tensión lineal	Regulador de tensión lineal integrado
Sensar corriente	Sensor de corriente invasivo	Sensor de corriente invasivo	Sensor de corriente no invasivo
Procesamiento de datos de dispositivo 3	Wemos D1 mini	Wemos D1 mini	Wemos D1 mini
Enviar información de dispositivo 3	Wifi	Wifi	Wifi
Recibir información en dispositivo 3	Wifi	Wifi	Wifi
Accionar conmutador de estado	Relé electromecánico	Relé electromecánico	Relé de estado sólido

2.3.2 Dispositivo de iluminación

La segunda parte agrupa las funciones que se aplican al dispositivo de iluminación, en el cual hay 9 funciones. (Ver tabla 2-2)

Tabla 2-2: Dispositivo de Iluminación. Fuente: [21]

Funciones	Solución 1	Solución 2	Solución 3
Albergar componentes	Carcasa de plástico	Carcasa de plástico	Carcasa de plástico
Acondicionar energía para sensores	Regulador conmutador	Regulador de tensión lineal	Regulador de tensión lineal integrado
Acondicionar energía para actuadores	Regulador conmutador	Regulador de tensión lineal	Regulador de tensión lineal integrado
Sensar corriente	Sensor de corriente invasivo	Sensor de corriente invasivo	Sensor de corriente no invasivo
Procesamiento de datos de dispositivo 2	Wemos D1 mini	Wemos D1 mini	Wemos D1 mini
Enviar información de dispositivo 2	Wifi	Wifi	Wifi
Recibir información en dispositivo 2	Wifi	Wifi	Wifi
Accionar conmutador de estado	Relé electromecánico	Relé electromecánico	Relé de estado sólido
Accionar cambiar intensidad	Dimmer AC	Dimmer AC	Dimmer AC

2.3.3 Base de datos e interfaz:

La tercera parte agrupa a las funciones que se aplican en la base de datos y en la interfaz. En esta agrupación se pueden observar 12 funciones, de las cuales 3 están aplicadas a la base de datos, y 9 están aplicadas a la interfaz. (Ver tabla 2-3)

Tabla 2-3: Base de datos e interfaz. Fuente [21]

Funciones	Solución 1	Solución 2	Solución 3
Almacenar datos	Servidor web AWS	Google Cloud	Servidor web propio
Recibir información en la base de datos	Wifi	Wifi	Redes Alámbricas
Enviar información de la base de datos	Wifi	Wifi	Redes Alámbricas
Recibir información en la interfaz	Wifi	Wifi	Wifi
Enviar información de la interfaz	Wifi	Wifi	Wifi
Procesar comandos	Servidor web AWS	Google Cloud	Servidor web propio
Procesamiento de datos del usuario	Servidor web AWS	Google Cloud	Servidor web propio
Procesamiento de datos de dispositivos	Servidor web AWS	Google Cloud	Servidor web propio
Procesamiento de información de base de datos	Servidor web AWS	Google Cloud	Servidor web propio
Mostrar estados de las tomas de corriente	Aplicación y web móvil	Aplicación y web móvil	Aplicación y web móvil
Mostrar intensidad de las luces	Aplicación y web móvil	Aplicación y web móvil	Aplicación y web móvil
Mostrar gráficas de consumo de energía	Aplicación y web móvil	Aplicación y web móvil	Aplicación y web móvil

2.4 Conceptos de solución

Se han desarrollado tres conceptos de solución a partir de la matriz morfológica. Estos se van a presentar a continuación.

2.4.1 Concepto de solución 1:

En este concepto de solución se puede observar que se utilizará una carcasa de plástico para poder albergar todos los componentes adecuados. Para el acondicionamiento de energía se usarán un regulador de voltaje y reguladores conmutadores. Además, para poder sensar la corriente se usarán sensores de corriente invasivos en los dos dispositivos. También se usará la tarjeta Wemos D1 mini para procesar los datos. Para poder enviar y recibir datos entre dispositivos se va a usar la señal Wifi. Para la conmutación de estados, se usarán relés electromecánicos y para el cambio de intensidad se planea usar un dimmer AC. También se tiene previsto usar el servidor web Amazon Web Services, en el cual se podrá almacenar todos los datos y además procesar todo lo que se requiera. Para el envío y recepción de información en la interfaz y la base de datos, se usará también la señal Wifi. Por último, para mostrar los datos se usará una plataforma web y una aplicación móvil. (Ver figura 2.12)

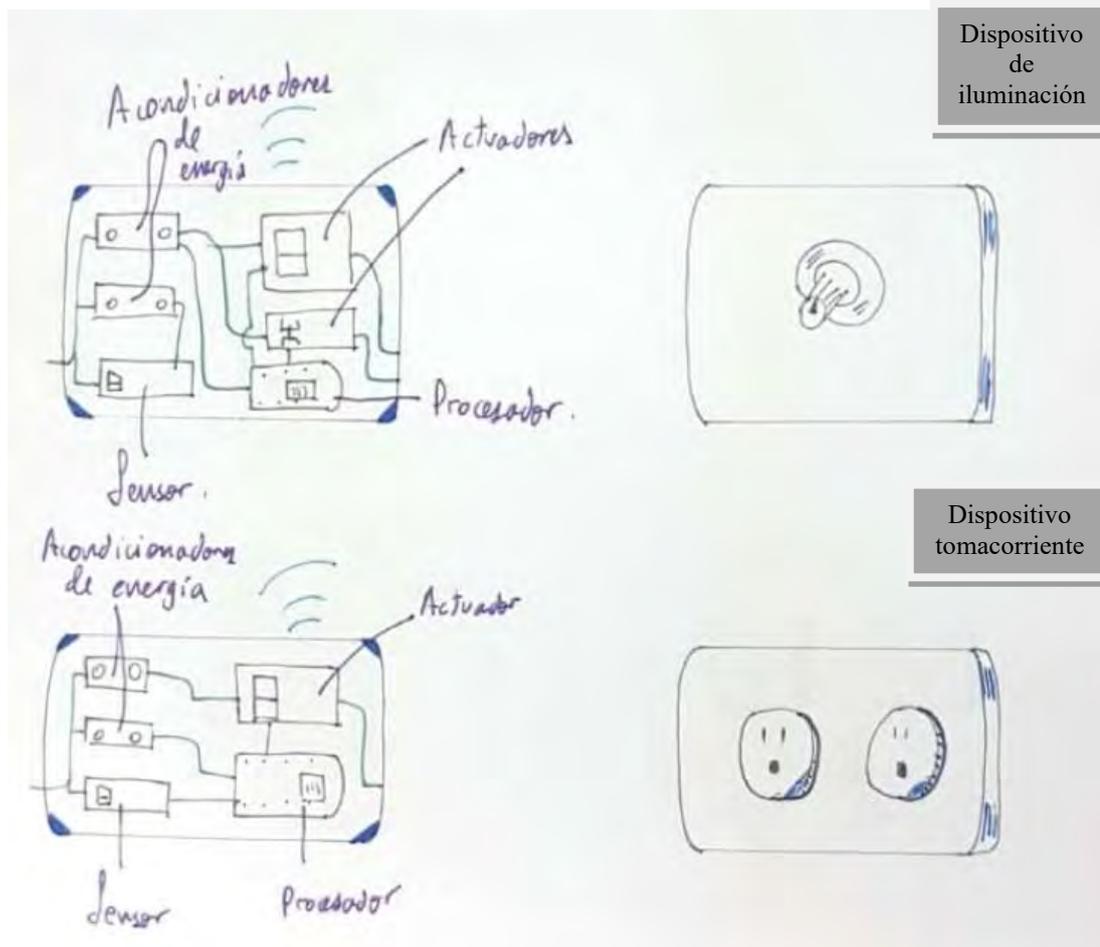


Figura 2.12: Concepto de solución 1. Fuente: [21]

2.4.2 Concepto de solución 2:

En este concepto de solución se usará carcasa de plástico para albergar los componentes. El acondicionamiento de energía se hará con un regulador de voltaje y reguladores de tensión lineal. También se usarán sensores de corriente invasivo. También se usará la tarjeta Wemos D1 mini para procesar los datos. Para poder enviar y recibir datos entre dispositivos se va a usar la señal Wifi. Para la conmutación de estados, se usarán también relés electromecánicos y para el cambio de intensidad se planea usar también un dimmer AC. En esta solución se tiene previsto usar el servidor web Google Cloud para almacenar y procesar la información. Para el envío y recepción de información se usará también la señal Wifi y se mostrarán los datos en una plataforma web y una aplicación móvil. (Ver figura 2.13)

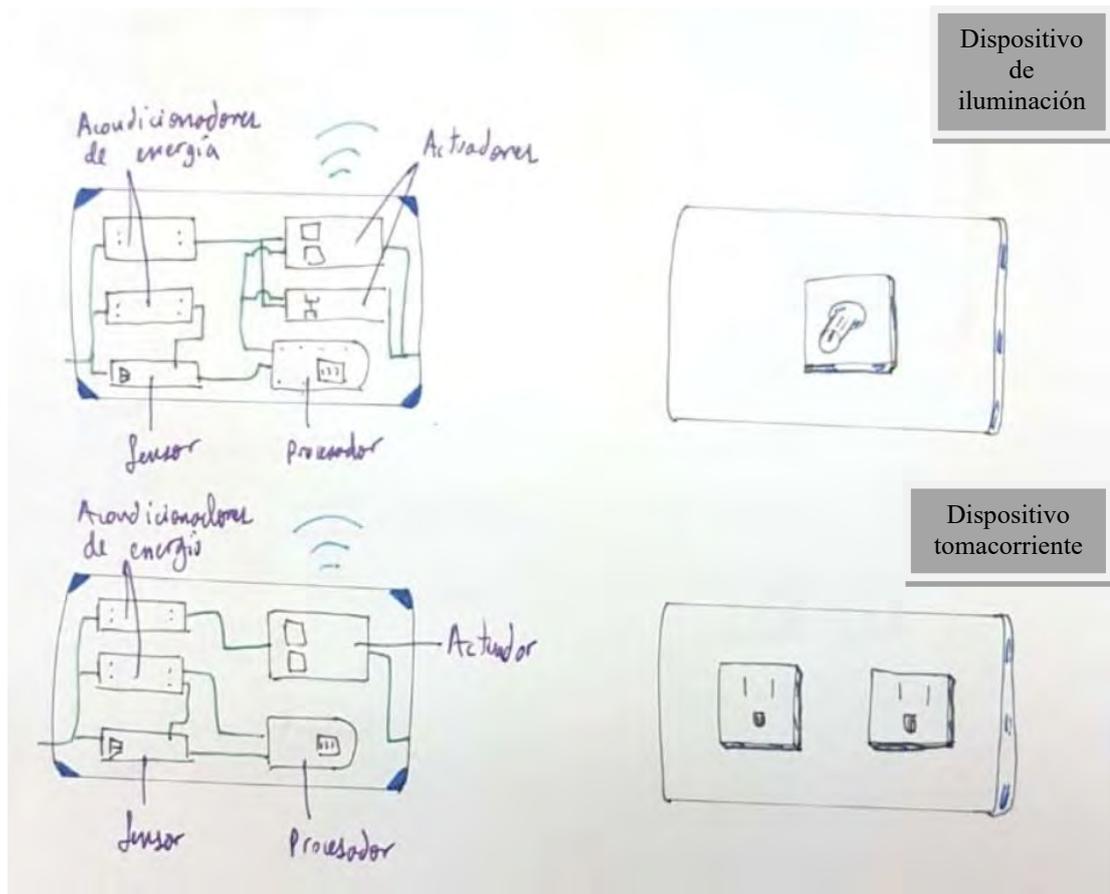


Figura 2.13: Concepto de solución 2. Fuente: [21]

2.4.3 Concepto de solución 3:

En este concepto de solución se usará también una carcasa de plástico para contener todos los componentes necesarios. Para poder acondicionar la energía se usarán reguladores de tensión lineal integrados en todo lo que se requiera. En los dispositivos se usaría un sensor de corriente no invasivo. Para el procesamiento de datos en los dispositivos se usaría un Wemos D1 mini como en todas las demás soluciones. Para enviar y recibir datos entre dispositivos se usará la señal Wifi. Para la conmutación de estados, en esta solución se usará un relé de estado sólido y para el cambio de intensidad se planea usar también un dimmer AC. En esa solución se tiene pensado realizar un servidor web propio y para ello, la recepción y envío de datos en la base de datos se realizaría de manera alámbrica, pero la recepción y envío de datos en la interfaz se realizaría con la señal Wifi. Para mostrar los resultados se desarrollará, como en las soluciones anteriores, una plataforma web y una aplicación móvil. (Ver figura 2.14)

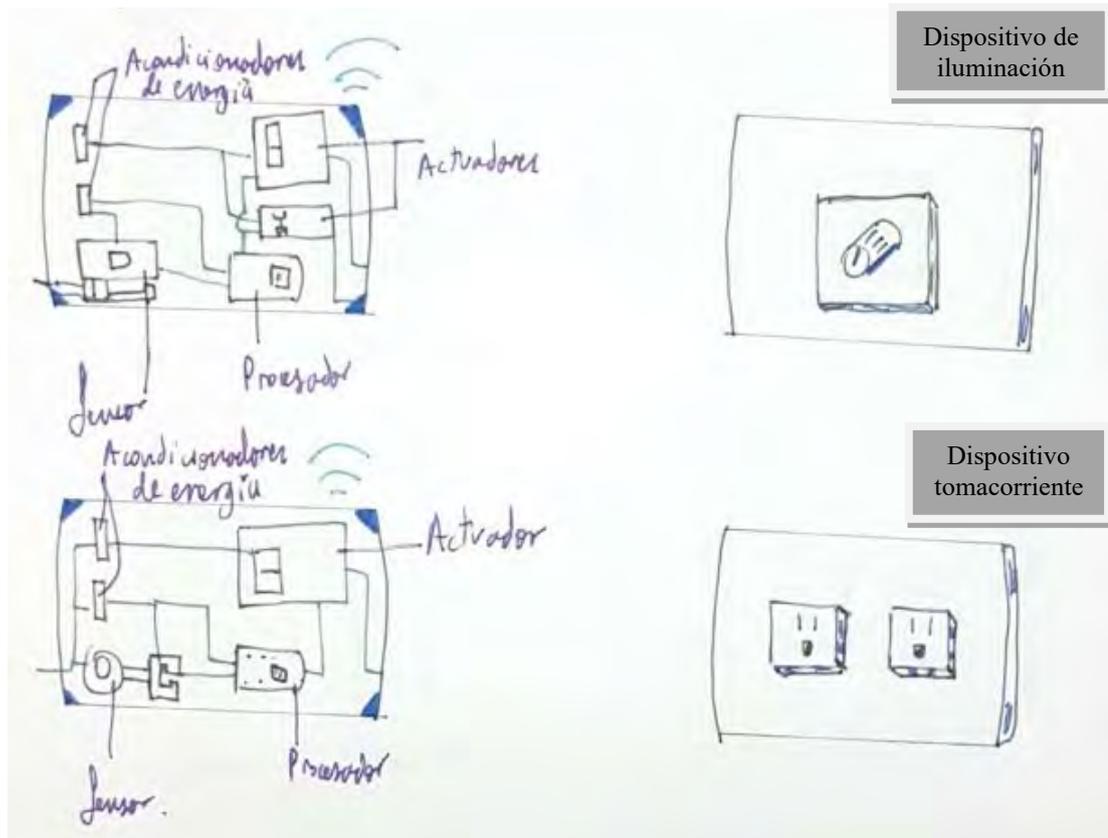


Figura 2.14: Concepto de solución 3. Fuente: [21]

2.5 Evaluación técnico – económica

Para poder elegir entre las soluciones que se han hallado, se va a realizar una evaluación técnico – económica, que es parte de la metodología para el diseño mecatrónico. Para esto, se tendrán que definir los criterios de evaluación tanto para la parte técnica como para la económica. Una vez elegidos los criterios, se les dará un peso a cada uno, que representará el grado de importancia que tienen para la implementación de la solución. Luego, se le dará puntaje a cada uno de los conceptos de solución y se adicionará el concepto de solución ideal, en el que todos los puntajes son perfectos para poder tener una comparación. Se calcula la cantidad alcanzada en base a la ideal y se obtienen los puntajes para cada concepto. Por último, en una gráfica con dos ejes, uno para el puntaje técnico y otro para el puntaje económico, se calcula la distancia a una línea ideal con pendiente de 45 grados, en el que se tiene un equilibrio. Las coordenadas de la solución que esté más cerca a esta línea será la seleccionada.

En el aspecto técnico, se van a tener en cuenta el cumplimiento de la lista de exigencias, la seguridad, la estabilidad, la confiabilidad, el fácil manejo, la transportabilidad y la complejidad del dispositivo. (Ver tabla 2-4)

Tabla 2-4: Evaluación técnica. Fuente: [21]

Evaluación de conceptos de solución											
Valor Técnico (xi)											
Proyecto: Dispositivo inteligente para el control de energía en el hogar											
Variantes de proyectos/conceptos			Solución 1 (S1)			Solución 2 (S2)		Solución 3 (S3)		Solución ideal	
Nr.	Criterios de evaluación	g	p	gp	p	gp	p	gp	p	gp	
1	Lista de exigencias	4	3	12	3	12	3	12	4	16	
2	Seguridad	4	2	8	3	12	3	12	4	16	
3	Estabilidad	4	3	12	3	12	3	12	4	16	
4	Confiabilidad	4	3	12	3	12	3	12	4	16	
5	Facilidad de manejo	3	3	9	3	9	3	9	4	12	
6	Transportabilidad	3	3	9	3	9	3	9	4	12	
7	Complejidad	2	3	6	2	4	2	4	4	8	
Puntaje máximo/total		-	20	68	20	70	20	70	28	96	
Valor técnico xi		-	0.71	0.71	0.71	0.73	0.71	0.73	1.00	1.00	
Orden		-	2		1		1		-	-	

En el aspecto económico, se va a considerar el número de piezas, la fácil adquisición de los materiales, el costo de la tecnología y la facilidad de montaje. (Ver tabla 2-5)

Tabla 2-5: Evaluación económica. Fuente: [21]

Evaluación de conceptos de solución											
Valor Económico (yi)											
Proyecto: Dispositivo inteligente para el control de energía en el hogar											
Variantes de proyectos/conceptos			Solución 1 (S1)			Solución 2 (S2)		Solución 3 (S3)		Solución ideal	
Nr.	Criterios de evaluación	g	p	gp	p	gp	p	gp	p	gp	
1	Número de piezas	2	3	6	3	6	3	6	4	8	
2	Facil adquisición de materiales	3	3	9	3	9	3	9	4	12	
3	Costo de la tecnología	3	3	9	2	6	1	3	4	12	
4	Facilidad de montaje	3	2	6	3	9	2	6	4	12	
Puntaje máximo/total		-	11	30	11	30	9	24	16	44	
Valor económico yi		-	0.69	0.68	0.69	0.68	0.56	0.55	1.00	1.00	
Orden		-	1		1		2		-	-	

Ahora, se va a utilizar el método del VDI 225, el cual indica que la solución más equilibrada será la opción elegida, para ello, se toma como referencia la función identidad en la gráfica. Se deberá hallar la distancia que se tiene desde las soluciones hasta la función identidad. (Ver figura 2.15)

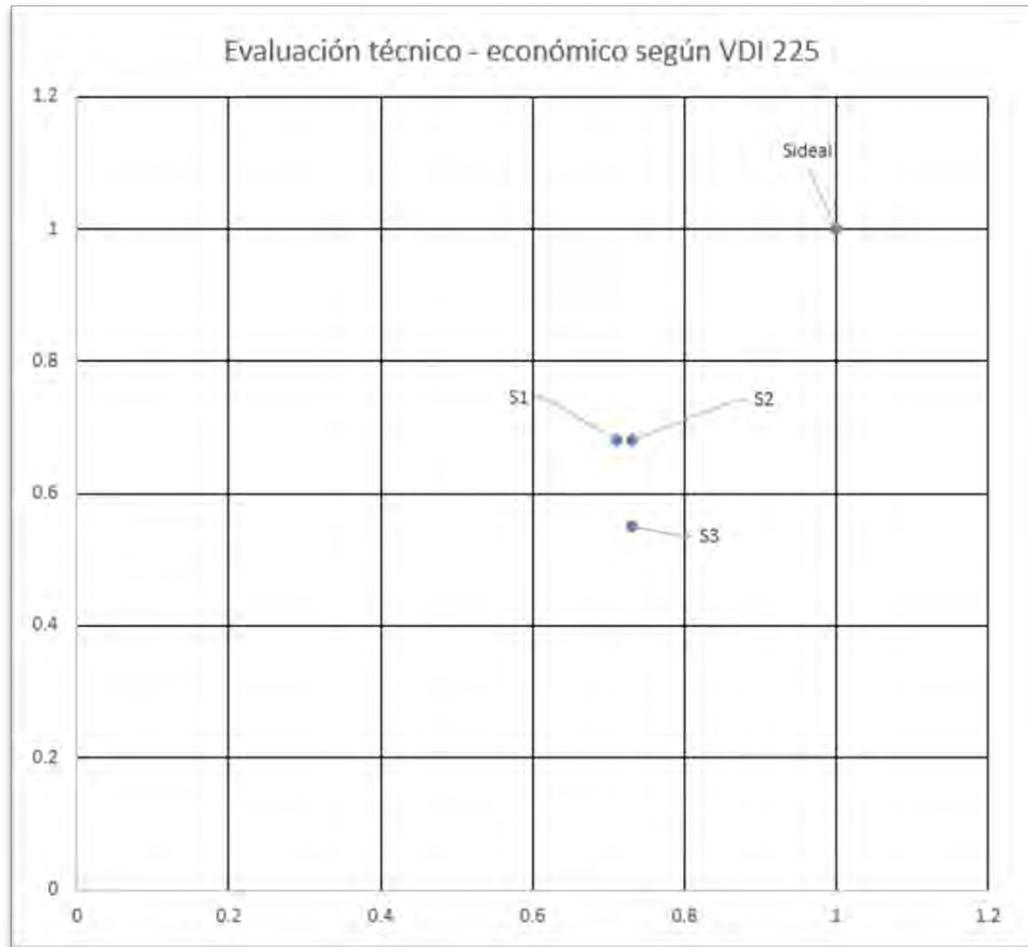


Figura 2.15: Evaluación técnico – Económica. Fuente: [21]

En la gráfica se puede notar que la solución más cercana a la función identidad es la solución 1, lo cual se comprueba realizando los cálculos correspondientes de distancia. (Ver tabla 2-6)

Tabla 2-6: Distancia de soluciones. Fuente: [21]

Solución	1	2	3
Distancia a la línea	0.021213203	0.035355339	0.127279221

Con esto se puede comprobar que la solución 1 es la que está más equilibrada y por lo tanto es la solución elegida.

2.6 Concepto de solución óptimo

Para el concepto de solución óptimo, se ha elegido al concepto de solución 1, ya que cuenta con un mejor equilibrio entre los aspectos técnicos y económicos. (Ver figura 2.16)

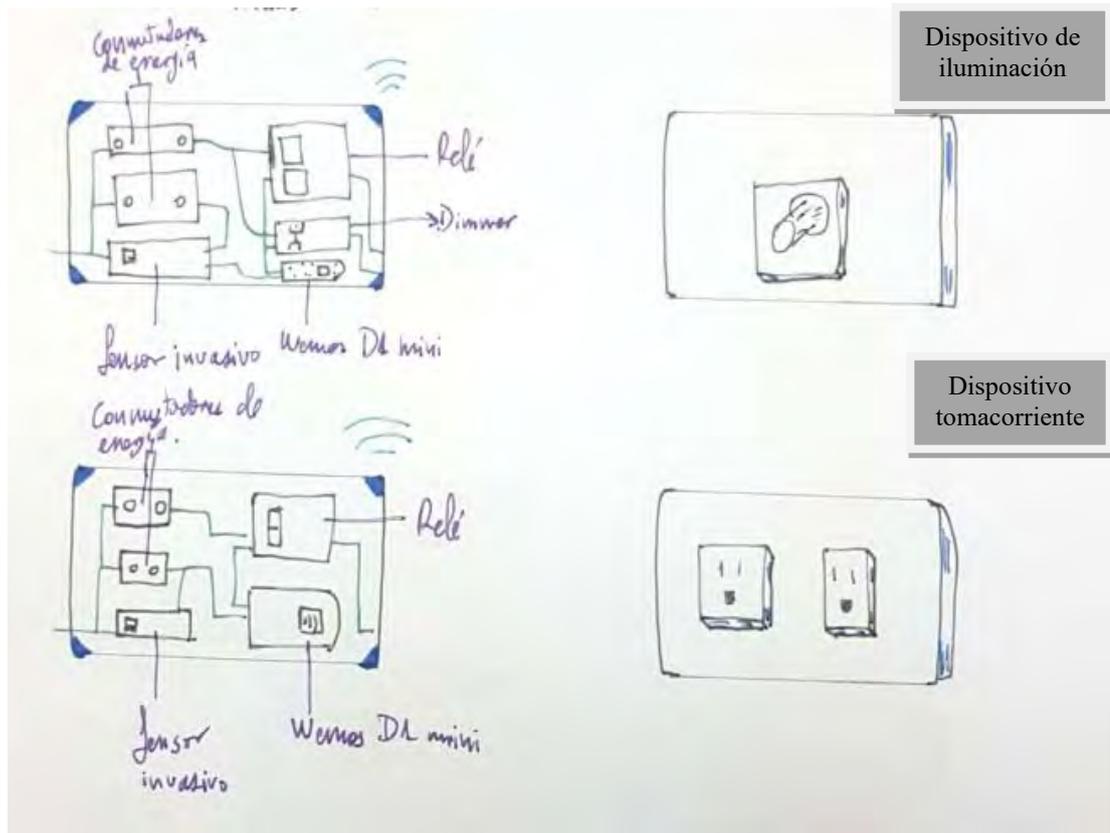
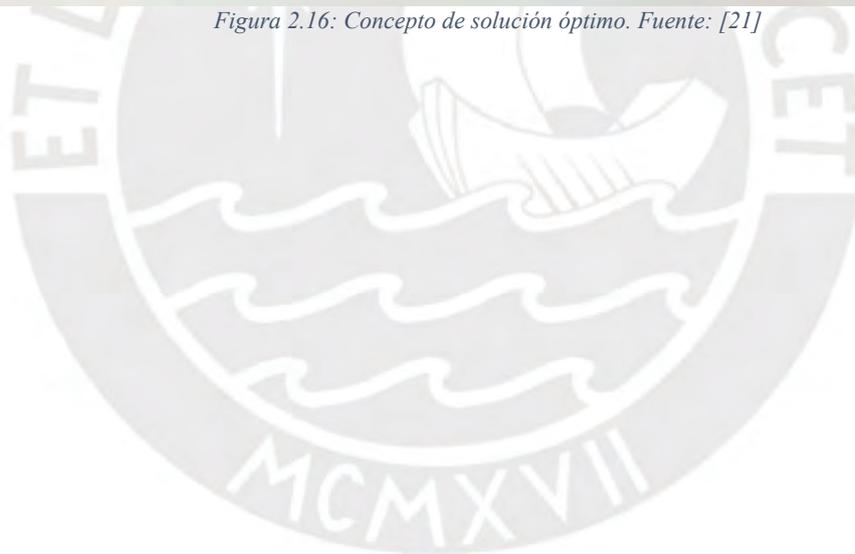


Figura 2.16: Concepto de solución óptimo. Fuente: [21]



CAPÍTULO 3

3. DISEÑO DE INGENIERÍA

En este capítulo, se presentará el diagrama de bloques, que mostrará las conexiones y las relaciones que tienen los dispositivos. También se realizará la selección y descripción de los componentes, así como la explicación del diagrama de flujo del sistema. Luego, se explicarán los diseños mecánico y electrónico iniciales. Adicionalmente, se mostrará la arquitectura de la solución IoT y la selección de las tecnologías que se usarán para el desarrollo de la plataforma web y de la aplicación móvil. Los dispositivos serán distribuidos en toda la casa. El azul representa al dispositivo de toma de corriente, y el verde al de iluminación. (Ver figura 3.1).

Leyenda:

- Tomacorriente
- Interruptor



Figura 3.1: Distribución 3D de los dispositivos en el hogar. Fuente: [22]

Una vez el usuario active el sistema, contará con acceso mediante una página web y mediante una aplicación móvil. Podrá apagar los tomacorrientes que se encuentren distribuidos en toda la casa y disminuir la intensidad de las luces desde cualquier lugar, así como apagarlas. Cada dispositivo, estará encargado de transmitir los datos a la red y se comunicarán con la base de datos para recibir y dar órdenes. El usuario a su vez, tiene que añadir los dispositivos nuevos que entren al sistema, así como eliminar los que ya no formen parte del mismo.

3.1 Diagrama de Bloques

El diagrama de bloques que se muestra se separa en 2 dispositivos diferentes que reciben y envían información. Los dispositivos accederán a la web y a la base de datos para poder almacenar valores o para recibir órdenes. Cada uno de los dispositivos usa un actuador diferente. En el caso del dispositivo de control de toma de corriente, se conmutará el estado y en el caso del dispositivo de control de luces, se variará la intensidad. Los demás bloques son comunes a los 2 dispositivos, como es el caso del microcontrolador, los reguladores AC y también los sensores de corriente, los cuales también están indicados en el gráfico. (Ver figura 3.2)

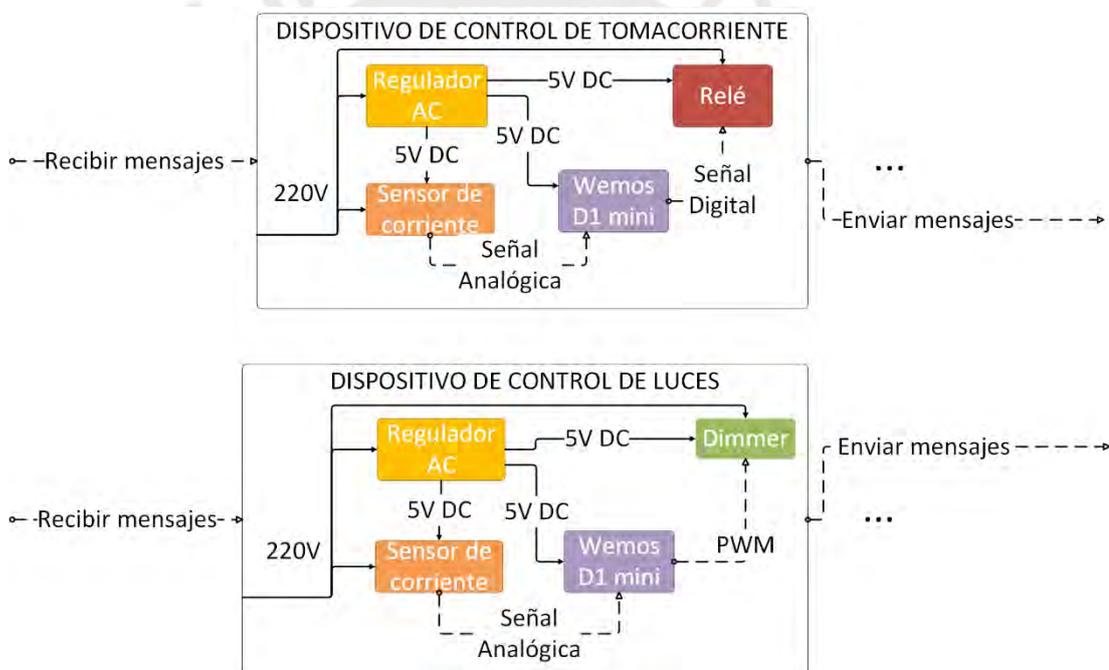


Figura 3.2: Diagrama de bloques del sistema. Fuente: [21]

3.2 Selección de sensores y actuadores

Los componentes que se han seleccionado han sido el acondicionador de energía, el sensor de corriente, el microcontrolador, el relé y el dimmer. La selección de cada uno de estos componentes se detallará a continuación.

3.2.1 El sensor de corriente

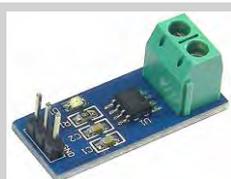
Para la medición de la corriente que pasa en el circuito, se deberá elegir un sensor que cumpla con los siguientes requerimientos:

- Rango de medición de corriente alto (Arriba de los 5 A).
- Tamaño pequeño.

- Voltaje de energización entre 3.3V y 5V.
- Bajo consumo de energía.
- Bajo peso.
- Precio bajo por ser uno de los objetivos del sistema.

Se evalúan 3 alternativas de sensores para este fin. (Ver tabla 3-1)

Tabla 3-1: Cuadro comparativo de sensores de corriente. Fuente: [21]

Dispositivo	ACS712 30A	SCT-013	WCS1600
Imagen referencial			
Dimensiones (mm)	32x14x14	57x32x22	36x31x24
Voltaje de entrada (V)	4.5 ~ 5.5	-	3 ~ 12
Voltaje de Salida (mV/A)	66	33.33	20
Corriente de entrada (mA)	10	0 – 30 A	3.5
Rango de medición. (A)	+/- 30	0 – 30	+/- 70
Temperatura de trabajo (°C)	-40 ~ 85	-25 ~ 70	-20 ~ 125
Peso (g)	5	50	12.4
Precio (S/.)	14	45	95

Luego de la evaluación se concluyó que se usará el modelo ACS712 porque cumple con las especificaciones, el tamaño es el más pequeño, tiene menos peso y menor costo que los otros modelos. (Ver figura 3.3)

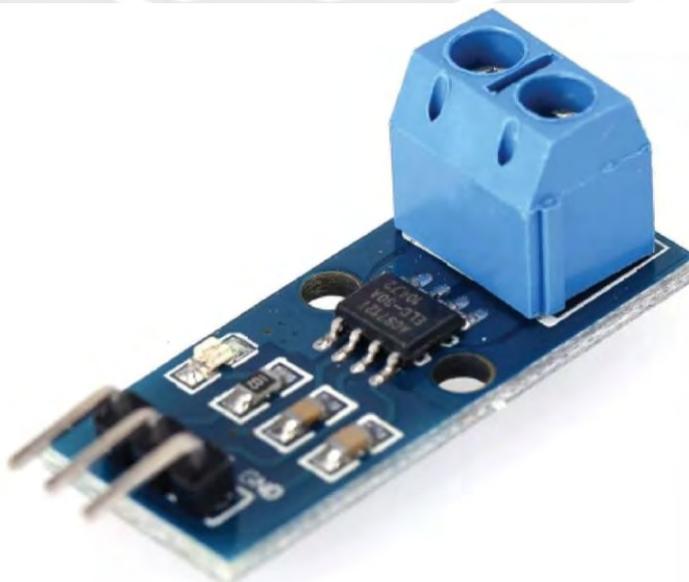


Figura 3.3: Sensor de corriente ACS712. Fuente: [23]

3.2.2 El microcontrolador

Para poder realizar el control, recepción de datos y el envío de las órdenes, se deberá elegir un microcontrolador que cumpla con los siguientes requerimientos:

- Capacidad de tener un módulo wifi.
- 1 entrada analógica como mínimo.
- 2 entradas digitales como mínimo.
- Frecuencia mayor o igual a 16 MHz.
- Tamaño pequeño.
- Bajo peso.
- Precio bajo por ser uno de los objetivos del sistema.

Se evalúan 2 alternativas de microcontroladores. (Ver tabla 3-2)

Tabla 3-2: Cuadro comparativo de microcontroladores. Fuente: [21]

Dispositivo	ARDUINO NANO	WEMOS D1 MINI
Imagen referencial		
Dimensiones (mm)	43.2x18.5	35x26x12
Microprocesador	ATmega328P – 8 bits	ESP8266 – 32 bits
Voltaje de alimentación (V)	5 - 12	5
Corriente consumida (mA)	19	70
Pines digitales	14	11
Pines digitales con PWM	6	0
Pines analógicos	8	1
Corriente máxima por pin (mA)	40	12
Frecuencia de reloj (MHz)	16	80/160
Temperatura de trabajo (°C)	-40 a +85	-20 a +100
Peso (g)	7	6
Precio	S/. 29	S/. 30
Conectividad Wifi	Externo	Interno

Luego de la evaluación, se concluyó que el dispositivo Wemos D1 mini era el adecuado, ya que tiene un precio bastante bajo, es pequeño y no necesita un módulo adicional, ya que tiene un módulo wifi ya instalado. (Ver figura 3.4)

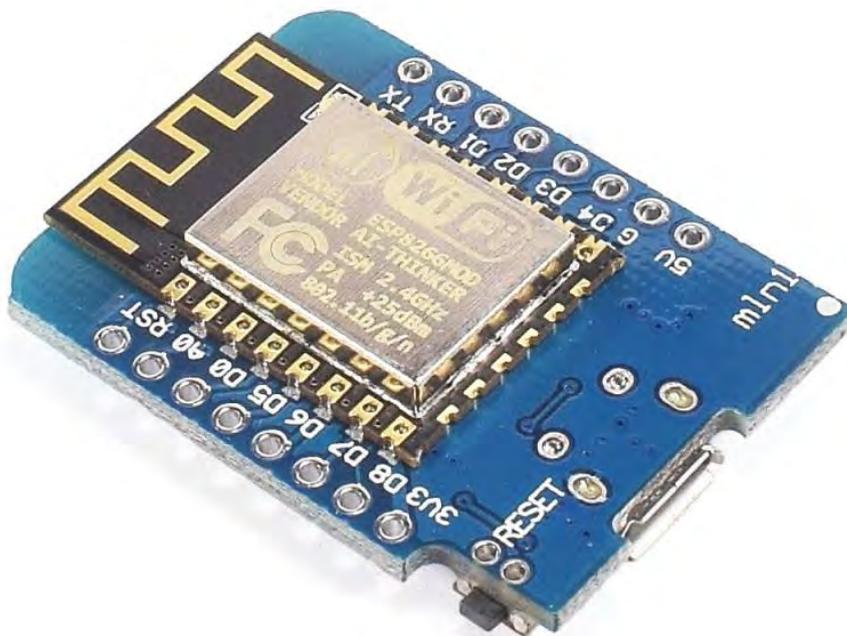


Figura 3.4: Wemos D1 mini. Fuente: [20]

3.2.3 El relé

Para poder conmutar el estado de los tomacorrientes, y permitir o bloquear el paso de la corriente eléctrica, se necesita un dispositivo que cumpla con los siguientes requerimientos:

- Control de al menos 220 VAC.
- Voltaje de energización entre 3.3V y 5V.
- Tamaño pequeño.
- Bajo peso.
- Consumo bajo de corriente.
- Precio bajo por ser uno de los objetivos del sistema.

Se evalúan 2 alternativas de actuadores para este fin. Un relé de estado sólido y un relé electromecánico. (Ver tabla 3-3)

Tabla 3-3: Cuadro comparativo de relés. Fuente: [21]

Dispositivo	G3MB-202P-DC5	SRD-05VDC-SL-C
Imagen referencial		
Dimensiones (mm)	34x25x25	40x20x15
Voltaje de entrada (V)	3.3 ~ 5	5
Voltaje de carga	240VAC	10A/250VAC, 10A/30VDC
Corriente de entrada (mA)	12.5	20
Corriente de control (A)	2 (NO)	10 (NO), 5 (NC)
Temperatura de trabajo (°C)	-30 ~ +80	-25 ~ +70
Peso (g)	13	14
Precio (S/.)	12	8

Luego de la evaluación se concluyó que el relé electromecánico era el adecuado debido a su precio y mayor rango de funcionamiento, tanto en corriente como en voltaje. (Ver figura 3.5)

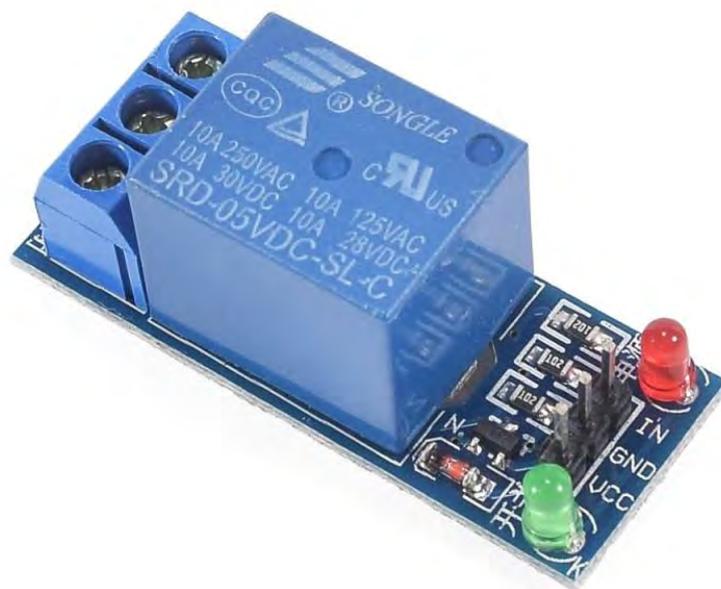


Figura 3.5: Relé electromecánico. Fuente: [20]

3.2.4 El dimmer

Para poder controlar la intensidad de las luces, se necesita un dispositivo que cumpla con los siguientes requerimientos:

- Control de 220 VAC.
- Activación por señal de microcontrolador.
- Voltaje de energización entre 3.3V y 5V.
- Tamaño pequeño.
- Bajo peso.
- Consumo bajo de corriente bajo.
- Precio bajo por ser uno de los objetivos del sistema.

En este caso, solo se ha evaluado un dispositivo que se adecúa para este fin, ya que los demás suelen ser mecánicos y no pueden ser controlados por una señal del microcontrolador. (Ver tabla 3-4)

Tabla 3-4: Cuadro comparativo de Dimmer. Fuente: [21]

Dispositivo	DIMMER AC 1CH BTA16600B
Imagen referencial	
Dimensiones (mm)	63x32x30
Voltaje de entrada (V)	3.3 ~ 5
Voltaje de carga (VAC)	110/220V
Corriente de entrada (mA)	20
Corriente de control (A)	5
Temperatura de trabajo (°C)	-20 ~+80
Peso (g)	24
Precio (S/.)	28

Este dispositivo es el adecuado, ya que se puede controlar por onda PWM y cumple con todos los requisitos mencionados. (Ver figura 3.6)

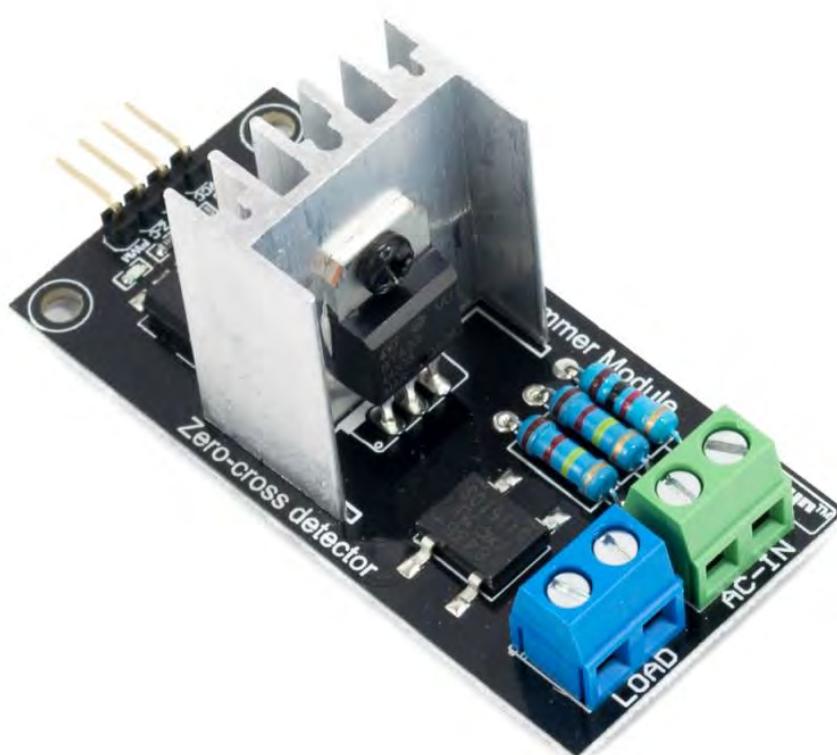


Figura 3.6: Dimmer AC 220V. Fuente: [20]

3.2.5 El acondicionador de energía

Antes de elegir el dispositivo que va a acondicionar la energía, se debe realizar el cálculo del consumo de potencia de todos los componentes electrónicos seleccionados para cada uno de los dispositivos.

Para el dispositivo tomacorriente, se tienen los siguientes cálculos de consumo energético: (Ver tabla 3-5)

Tabla 3-5: Consumo de los componentes del dispositivo tomacorriente. Fuente: [21]

Componente	Código	Cantidad	Voltaje (V)	Corriente (A)	Potencia (W)
Microcontrolador	WEMOS D1 MINI	1	5	0.070	0.350
Sensor	ACS712 30A	1	5	0.010	0.050
Relay	SRD-05VDC-SL-C	1	5	0.020	0.100
Totales	-	3	-	0.100	0.5

Se puede observar en la tabla, que la potencia total requerida para este dispositivo es de 0.5W y que la corriente que consume es de 0.1A.

Para el dispositivo de iluminación, se tienen los siguientes cálculos de consumo energético: (Ver tabla 3-6)

Tabla 3-6: Consumo de los componentes del dispositivo de iluminación. Fuente: [21]

Componente	Código	Cantidad	Voltaje (V)	Corriente (A)	Potencia (W)
Microcontrolador	WEMOS D1 MINI	1	5	0.070	0.350
Sensor	ACS712 30A	1	5	0.010	0.050
Dimmer AC	1CH BTA16600B	1	5	0.020	0.100
Totales	-	3	-	0.100	0.5

Se puede observar en la tabla, que la potencia total requerida para este dispositivo también es de 0.5W y que la corriente que consume también es de 0.1A.

Con los cálculos realizados, se obtiene la misma potencia requerida para cada uno de los dispositivos, lo cual facilita la selección del acondicionador de energía que será el mismo para ambos. A esta potencia calculada, se le añadirá un factor de seguridad de 1.5 para cualquier imprevisto e incrementar el tiempo de vida útil de los dispositivos.

Se usará la siguiente fórmula:

$$P_r = F.S. * P_c$$

Ecuación 3-1: Potencia requerida

donde:

- P_r es la potencia requerida.
- F.S. es el factor de seguridad.
- P_c es la potencia calculada.

Con los datos determinados previamente, se obtiene una potencia requerida de 0.75W.

Ahora que ya tenemos el dato de la potencia requerida, se va a evaluar la selección del componente de acondicionamiento de energía con los siguientes requerimientos:

- Transformación de 220V AC a 5V DC.
- Potencia mínima de 0.75W.
- Estabilidad en el voltaje de salida.
- Tamaño pequeño.
- Bajo peso.
- Precio bajo por ser uno de los objetivos del sistema.

Se evalúan 2 alternativas de acondicionadores de energía. (Ver tabla 3-7)

Tabla 3-7: Cuadro comparativo de acondicionadores de energía. Fuente: [21]

Dispositivo	HLK-PM01	RAC04-05SC/W
Imagen referencial		
Dimensiones (mm)	34x20.2x20	37.8x23.9x16.4
Voltaje de entrada (VAC)	90 ~ 264	80 ~ 264
Voltaje de Salida (V)	5 ± 4%	5 ± 5%
Potencia (W)	3	4
Corriente de salida máx. (mA)	1000	800
Temperatura de trabajo (°C)	-20 ~ +60	-25 ~ +85
Eficiencia (%)	70	72
Peso (g)	20g	32g
Precio	S/. 20	\$17.59

Luego de la evaluación se concluyó que el dispositivo HLK-PM01 era el adecuado, ya que tiene un precio bastante bajo, es muy compacto y la estabilidad que tiene es alta con solo 4% de error en la salida. (Ver figura 3.7)



Figura 3.7: Acondicionador de energía 220V AC a 5V DC. Fuente: [21]

3.3 Diagrama de flujo

Se ha realizado un diagrama de flujo general para cada uno de los dispositivos. En estos diagramas de flujo se pueden notar las funciones de “Conectarse al servidor”, “Reconectar”, “leer datos de web” y “Enviar datos a la nube”, las cuales dan a entender la comunicación que se tiene con el servidor web usado. (Ver figura 3.8 y 3.9)

Primero, se presenta el diagrama de flujo general del dispositivo para tomacorrientes. Este define las variables para inmediatamente conectarse al wifi y al servidor. Luego de esto, se genera de manera iterativa la verificación de la conexión, lectura de datos y sensores, conmutación del estado del tomacorriente, cálculos de las variables y envío de datos a la nube. (Ver figura 3.8)

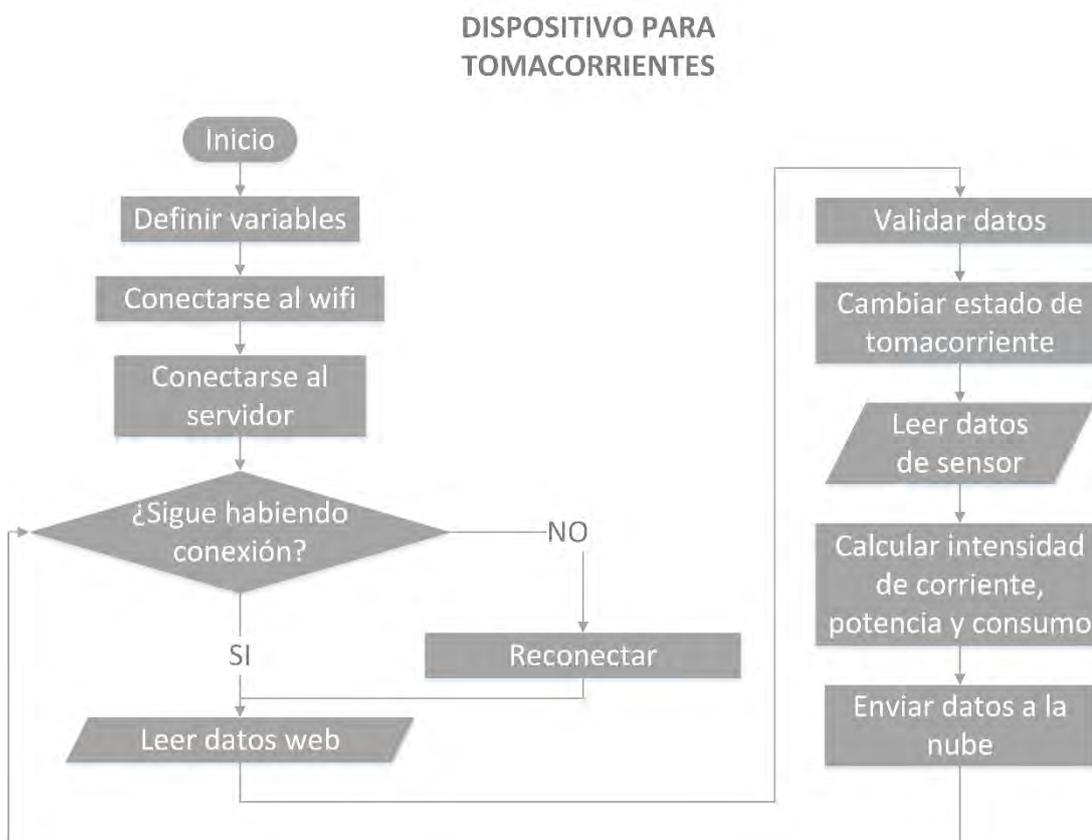


Figura 3.8: Diagramas de flujo general del dispositivo para tomacorrientes. Fuente: [21]

Segundo, se presenta el diagrama de flujo general del dispositivo para luces. Este realiza las mismas funciones que el dispositivo anterior, con la diferencia que tiene una función de cambiar la intensidad de las luces, que reemplaza a la función de cambiar el estado. (Ver figura 3.9)

DISPOSITIVO PARA LUCES

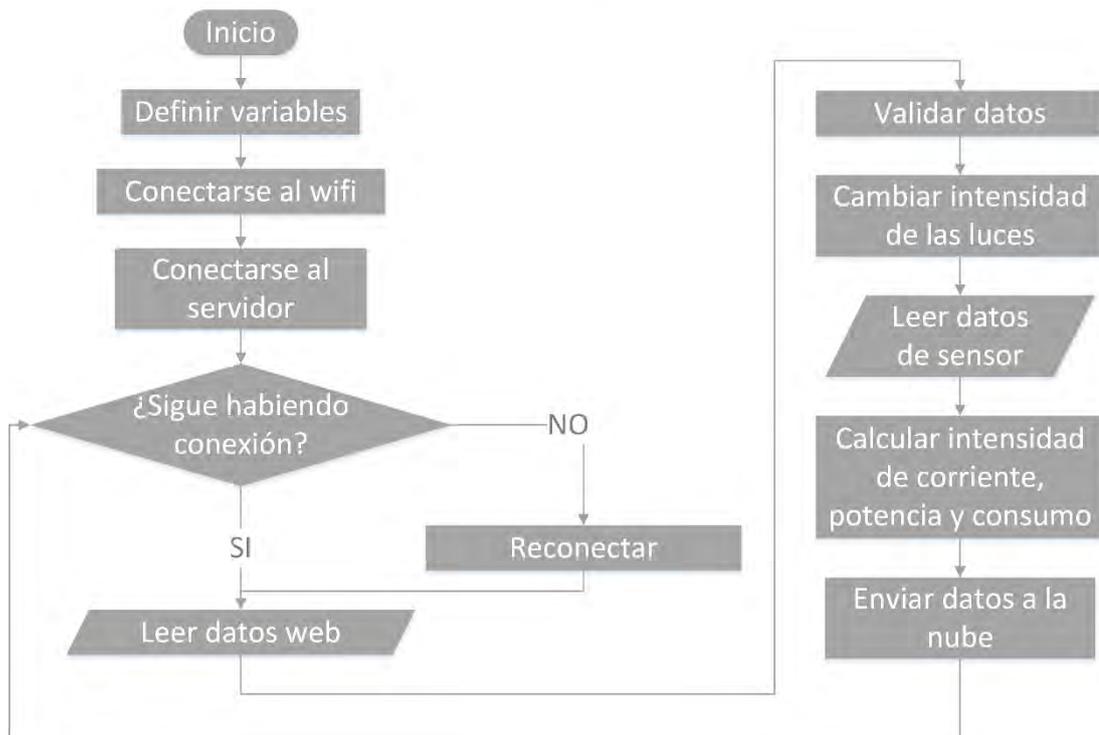


Figura 3.9: Diagramas de flujo general del dispositivo para luces. Fuente: [21]

Ahora, se detallará un poco de la función “calcular intensidad de corriente, potencia y consumo”, la cual es una función que se encarga de obtener la data del sensor por medio segundo; luego, escala el dato obtenido por el sensor para operar la señal y lo opera para después hacer los cálculos de todas las variables necesarias. Se desarrolla fuera de la función principal y sólo es llamada cuando se requiere. (Ver figura 3.10)

Calcular intensidad de corrientes, potencia y consumo

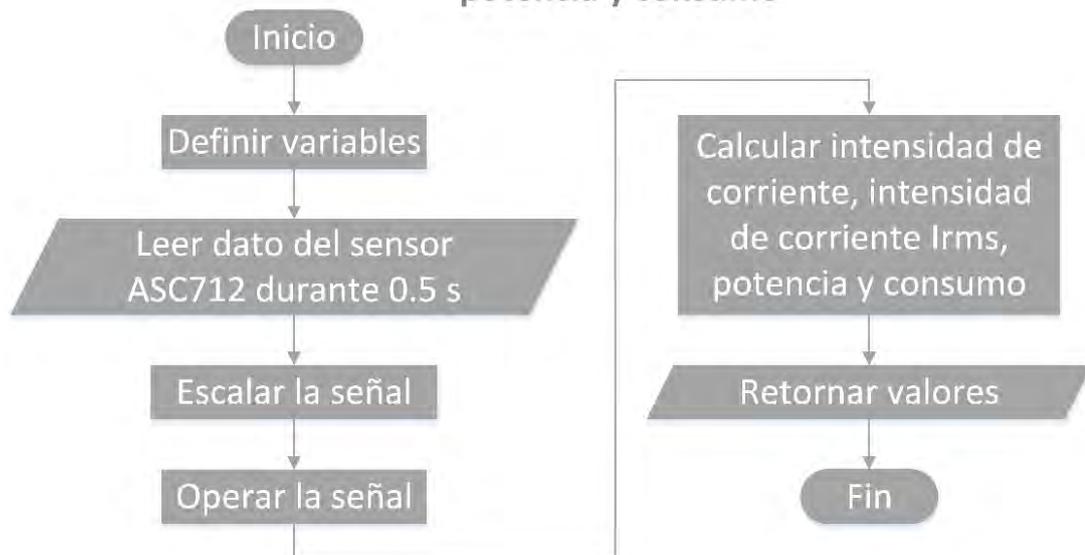


Figura 3.10: Función de cálculo de variables. Fuente: [21]

3.4 Diseño mecánico

Los planos mecánicos del sistema se adjuntarán en el Anexo F, pero se presentará la lista de los mismos y se detallarán a continuación. Se tiene dos planos por cada dispositivo. Un plano es el contenedor y el otro representa la tapa. Cada dispositivo tiene ranuras en el lugar donde se encuentra el módulo Wemos D1 mini para la conexión Wifi se realice sin problemas. Además, dependiendo del calentamiento del dispositivo, se tienen ranuras cerca al causante de este calentamiento. Por ejemplo, en el dispositivo de iluminación se tiene ranuras en la parte posterior, ya que el disipador de calor del dimmer tiende a calentarse y para que ese calor sea expulsado, se diseñaron esas ranuras. Además de esto, se tienen lugares destinados para cada componente electrónico. Por último, las tapas llevan columnas para mantener los dispositivos en su lugar y también llevan una marca de ejemplo. (Ver tabla 3-8)

Tabla 3-8: Lista de planos. Fuente: [21]

Nº	Nombre	Código
1	Contenedor – Tomacorriente	TOM-01
2	Tapa – Tomacorriente	TOM-02
3	Contenedor – Iluminación	ILU-01
4	Tapa – Iluminación	ILU-02

3.4.1 Materiales:

Para los materiales, se toma de referencia la norma NTP-IEC 60884-1, que es la norma para enchufes y tomacorrientes para uso doméstico y propósitos similares. Esta norma define 2 tipos de tomacorrientes, el simple y el empotrable. Además, también se ve el tema de los enchufes y los interruptores. Los materiales permitidos para los dispositivos eléctricos son materiales aislantes los cuales pueden ser PVC y cuerpo moldeado en policarbonato resistente a impactos.

Para fabricarlos, se utilizan pellets de plástico que van a moldearse por medio del método de extrusión para darle la forma de la carcasa del tomacorriente o interruptor. Para esto, se deben calentar los pellets hasta más de 200 °C.

Adicional a esto, para la etapa de pruebas y prototipado, se van a fabricar los productos con una impresora 3D y el material a usar será plástico PLA. Este material tiene las siguientes características:

- Temperatura de fusión entre 190 a 210 °C.
- Resistencia mecánica de tracción – compresión aceptable.
- Dureza para pruebas aceptable.

3.4.2 Diseño 3D del dispositivo de tomacorriente:

Se diseño la carcasa del tomacorriente de forma cuadrada y se puede observar que se tienen ranuras en la parte frontal para que la señal Wifi no tenga obstáculos. (Ver figura 3.11)

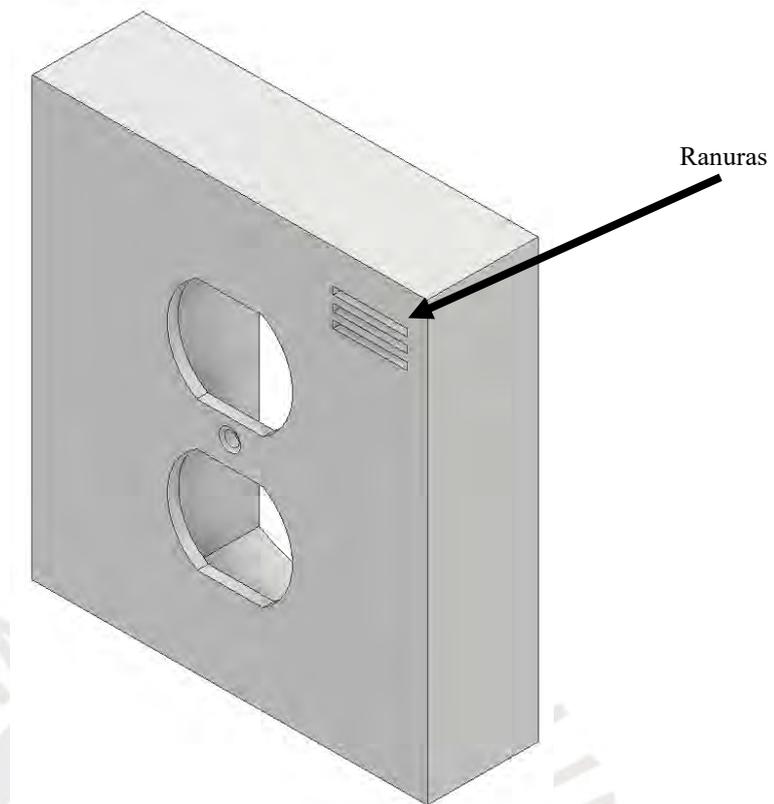


Figura 3.11: Diseño mecánico del dispositivo para tomacorrientes - frontal. Fuente: [21]

El diseño interior con la distribución de los dispositivos electrónicos del tomacorriente se muestra en las siguientes figuras. Se puede notar que hay separaciones para cada uno de los dispositivos para que encajen por apriete y por atornillado. (Ver figuras 3.12 y 3.13)

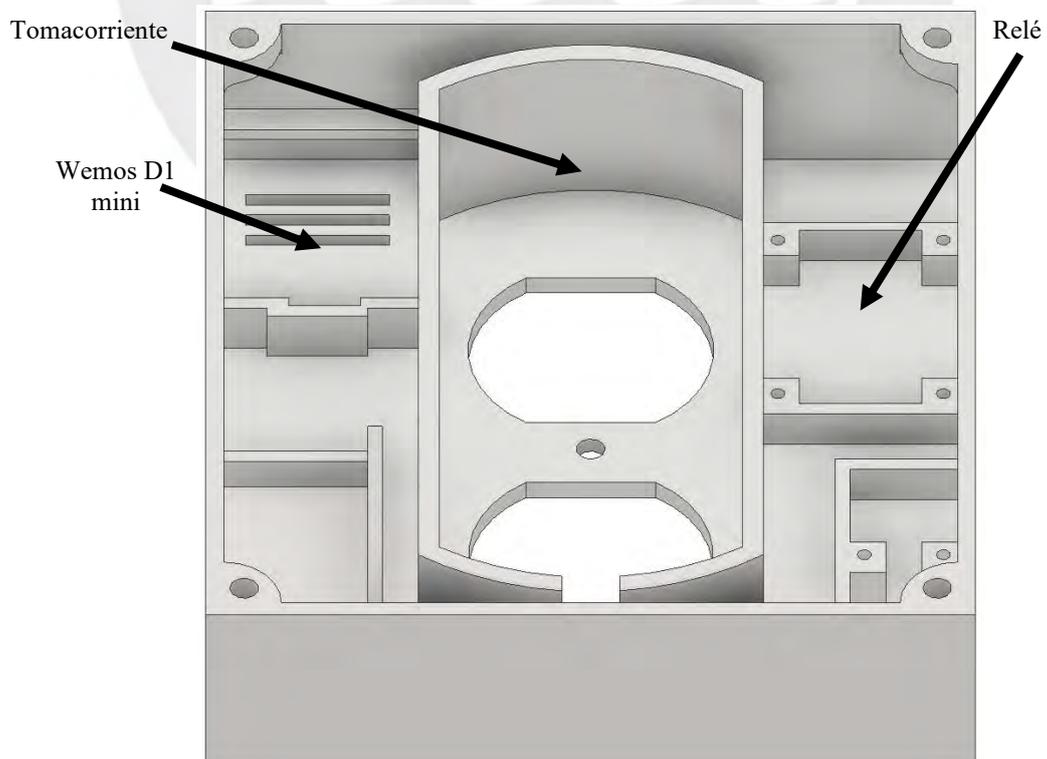


Figura 3.12: Distribución interna del dispositivo para tomacorrientes 1. Fuente: [21]

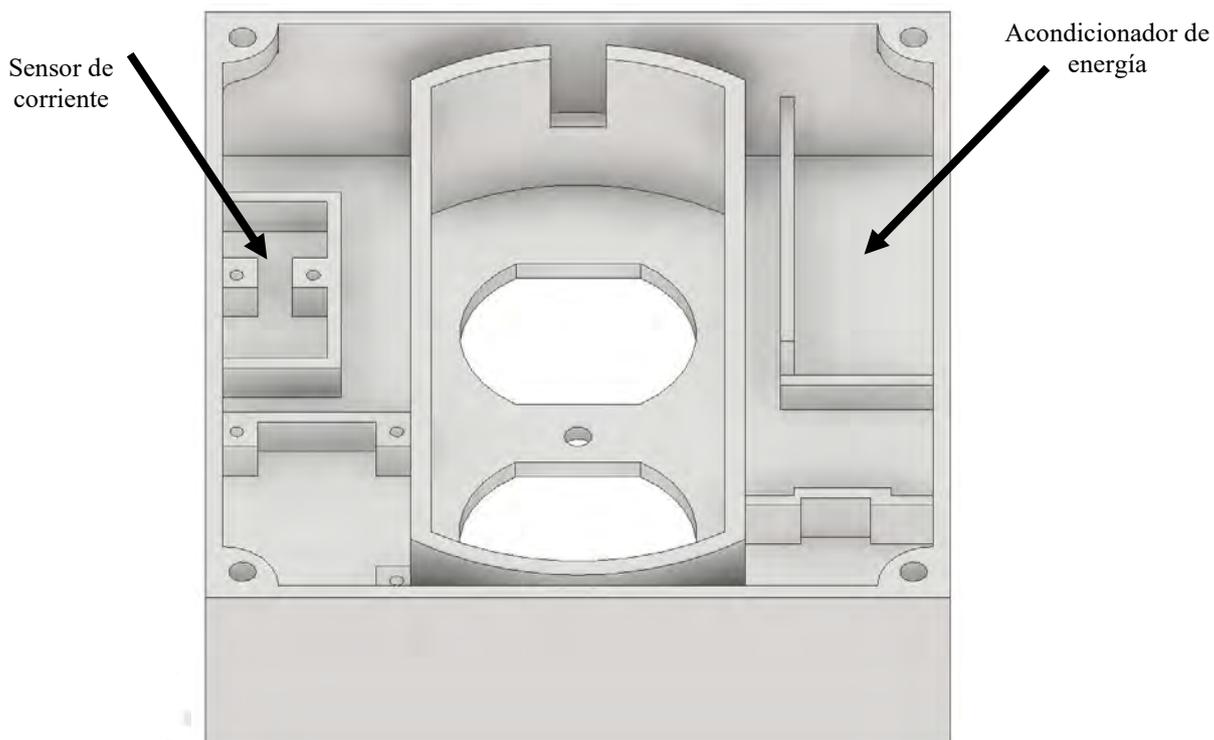


Figura 3.13: Distribución interna del dispositivo para tomacorrientes 2. Fuente: [21]

En el diseño de la tapa, se han colocado una ranura para el enchufe y una marca de ejemplo. También tiene columnas para que los dispositivos electrónicos permanezcan en su lugar. (Ver figura 3.14 y 3.15)

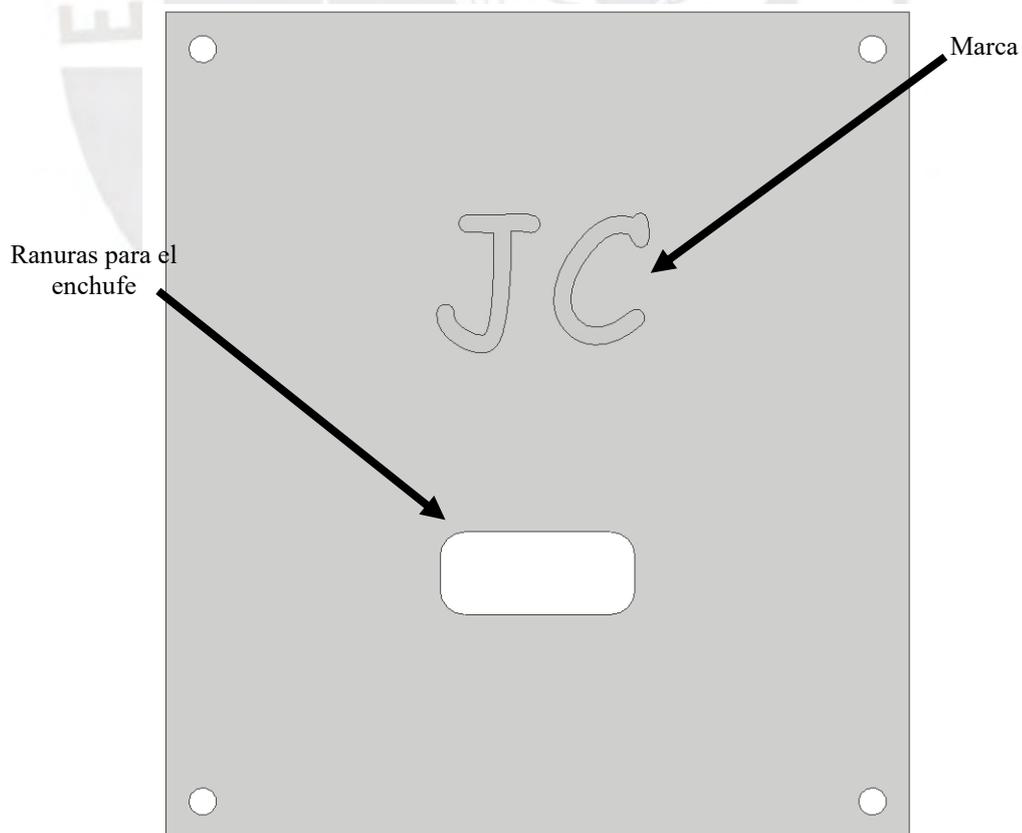


Figura 3.14: Diseño mecánico del dispositivo para tomacorrientes - posterior. Fuente: [21]

3.4.3 Diseño 3D del dispositivo de iluminación:

El dispositivo para la iluminación también tiene ranuras para que la señal Wifi que se transmite no tenga obstáculos. (Ver figura 3.17)



Figura 3.17: Dispositivo de iluminación - contenedor. Fuente: [21]

El dispositivo de iluminación también tiene una distribución adecuada para los dispositivos electrónicos. Se puede notar las separaciones encajan por apriete y por atornillado. (Ver figura 3.18)

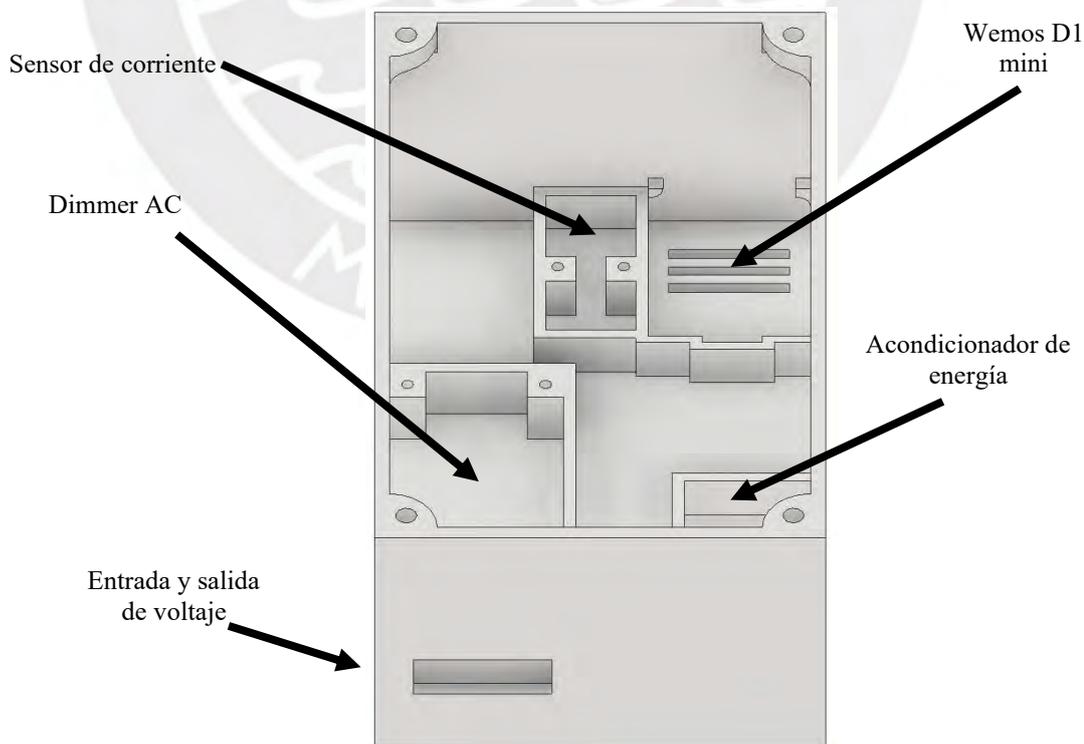


Figura 3.18: Dispositivo de iluminación - Lugares de los dispositivos electrónicos. Fuente: [21]

La tapa del dispositivo de iluminación tiene ranuras para que el calor provocado por el dimmer pueda ser evacuado. Además, cuenta con una marca para hacer notar que es un producto comercial. (Ver figura 3.19) También, la tapa tiene columnas para que los dispositivos electrónicos permanezcan en su lugar. (Ver figura 3.20)

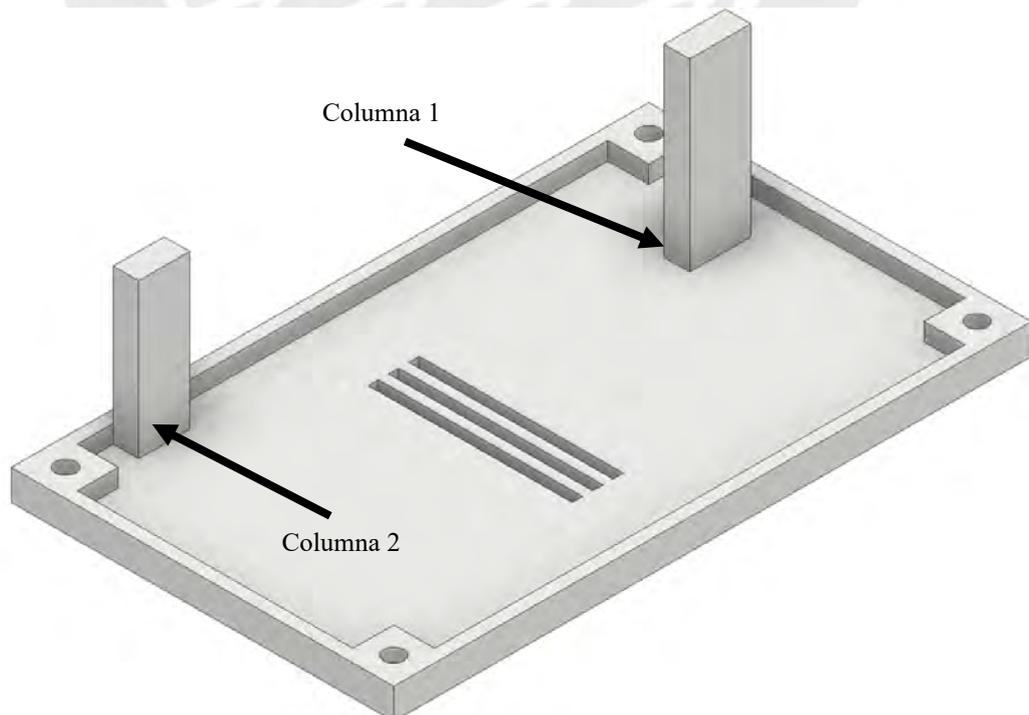
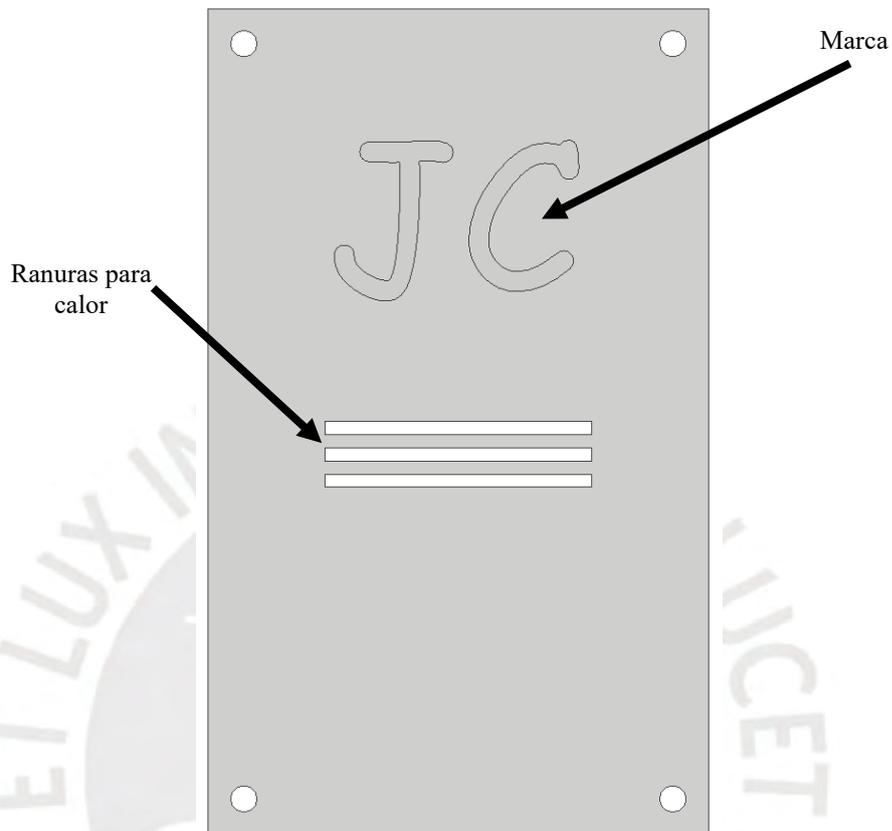


Figura 3.20: Dispositivo de iluminación - columnas de tapa. Fuente: [21]

Por último, se muestran las medidas internas de los lugares para los componentes del dispositivo para las luces. (Ver figura 3.21)

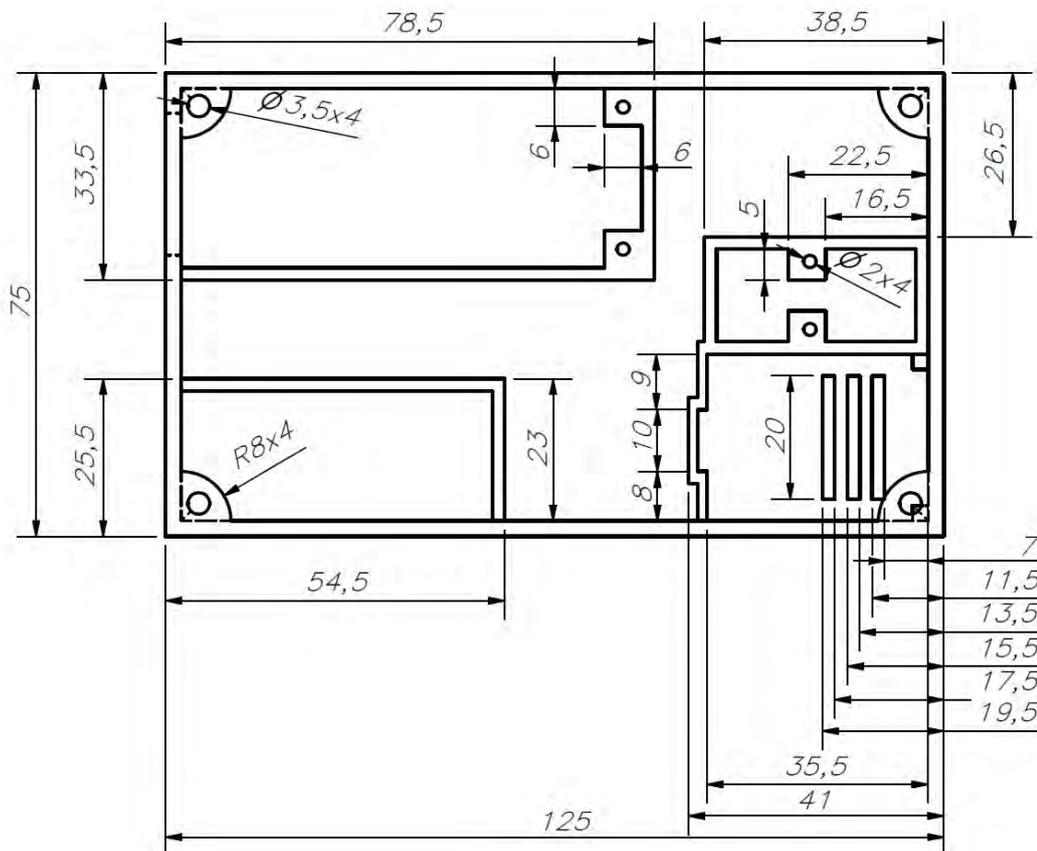


Figura 3.21: Medidas internas del dispositivo para luces. Fuente: [21]

3.4.4 Cálculos mecánicos:

Ahora que se tienen los diseños y las medidas realizadas, se puede obtener un aproximado de la cantidad de material que se va a necesitar para imprimir en 3D los dispositivos. Como dato, vamos a tener la longitud del filamento a usar. La densidad del filamento PLA es de 1.24 g/cm^3 o de $1.24 * 10^{-3} \text{ g/mm}^3$. El diámetro estándar del filamento que se usa para las impresiones es de 1.75 mm.

Usaremos la siguiente fórmula:

$$W_{\text{fil}} = L_{\text{fil}} * \rho_{\text{fil}} * A_{\text{fil}}$$

Ecuación 3-2: Masa de filamento

siendo:

- L_{fil} es la longitud del filamento en mm.
- ρ_{fil} es la densidad del PLA en g/mm^3 .
- A_{fil} es el área circular transversal del filamento en mm^2 .
- W_{fil} es la masa del filamento en g.

Con los datos y fórmula determinados previamente, se obtendrá un peso por cada impresión 3D que se realice y se presentará resumido. (Ver tabla 3.9)

Tabla 3-9: Masa calculada de las impresiones 3D de los dispositivos. Fuente: [21]

Componente	Longitud de filamento usado (mm)	Densidad de material (g/mm ³)	Área transversal (mm ²)	Masa calculada (g)
Tomacorriente – frontal	25273	1.24 * 10 ⁻³	$\pi * (1.75/2)^2$	75.38
Tomacorriente – posterior	26666	1.24 * 10 ⁻³	$\pi * (1.75/2)^2$	79.53
Luces - frontal	22792	1.24 * 10 ⁻³	$\pi * (1.75/2)^2$	67.98
Luces - posterior	12142	1.24 * 10 ⁻³	$\pi * (1.75/2)^2$	36.21
Totales	86873	-	-	259.10

Ahora, según la tabla 3-9, la masa de la carcasa del dispositivo tomacorriente sería 154.91 g y la masa de la carcasa del dispositivo de luces sería 104.19 g. Cabe mencionar que la densidad del PVC y otros materiales para las carcasas del plástico es de 1.3 a 1.5 g/cm³. Estos pesos hacen que la diferencia en peso que se tiene con los prototipos en PLA es de 4.8 % a 20.1 %.

Ahora, se calcula la masa total del dispositivo de tomacorriente. (Ver tabla 3.10)

Tabla 3-10: Peso de los componentes del dispositivo tomacorriente. Fuente: [21]

Componente	Código	Cantidad	Peso (g)
Acondicionador de energía	HLK-PM01	1	20
Microcontrolador	WEMOS D1 MINI	1	6
Sensor	ACS712 30A	1	5
Relay	WCS1600	1	14
Carcasa	-	1	154.91
Totales	-	3	199.91

Y también, se calcula la masa total del dispositivo de Luces. (Ver tabla 3.11)

Tabla 3-11: Peso de los componentes del dispositivo de iluminación. Fuente: [21]

Componente	Código	Cantidad	Peso (g)
Acondicionador de energía	HLK-PM01	1	20
Microcontrolador	WEMOS D1 MINI	1	6
Sensor	ACS712 30A	1	5
Dimmer AC	1CH BTA16600B	1	24
Carcasa	-	1	104.19
Totales	-	3	159.19

Según las tablas 3-10 y 3-11, la masa del dispositivo tomacorriente sería de 199.91 g y del dispositivo de luces 159.19 g, pero se considerará alrededor de 10 g extra por los cables y tornillos de unión. Con esto, la masa aproximada sería 210 g y 170 g respectivamente.

Con este dato, podemos obtener la fuerza de sujeción en los tomacorrientes que tiene el dispositivo tomacorriente y ver si está dentro de lo permitido. Para esto se usarán las fórmulas:

$$F_1 = m_t * g$$

Ecuación 3-3: Peso del dispositivo

Donde:

- F_1 es el peso del dispositivo.
- m_t masa total del dispositivo.
- g es la aceleración de la gravedad.

Con los datos previos, tenemos que F_1 es 2.1 N.

También se calculará la fuerza de apoyo considerando que el punto de aplicación se encuentra en la esquina inferior. Por los planos, se sabe que la distancia desde el enchufe hasta la esquina inferior izquierda es de 81.5 mm. Se considera también el punto de aplicación del peso en el centro geométrico por comodidad, lo cual hace que esté a 25 mm del enchufe. Usando la ley de momento de resultante considerando el punto cero al punto de conexión del enchufe se tiene:

$$d_1 * F_1 = d_2 * F_2$$

Ecuación 3-4: Momentos

donde:

- d_1 es la distancia de la línea vertical que pasa por el centro geométrico.
- F_1 es el peso del dispositivo.
- d_2 es la distancia desde el enchufe hasta la esquina inferior.
- F_2 es la fuerza de reacción de la pared.

Con los datos previos, tenemos que F_2 es 0.64 N.

Ahora, la F_3 es la fuerza resultante de las F_1 y F_2 .

$$F_3 = (F_1^2 + F_2^2)^{(1/2)}$$

Ecuación 3-5: Fuerza resultante

Con esta fórmula, obtenemos que F_3 es 2.20 N.

Se presentará el diagrama de cuerpo libre del dispositivo tomacorriente con los resultados encontrados. (Ver figura 3.22)

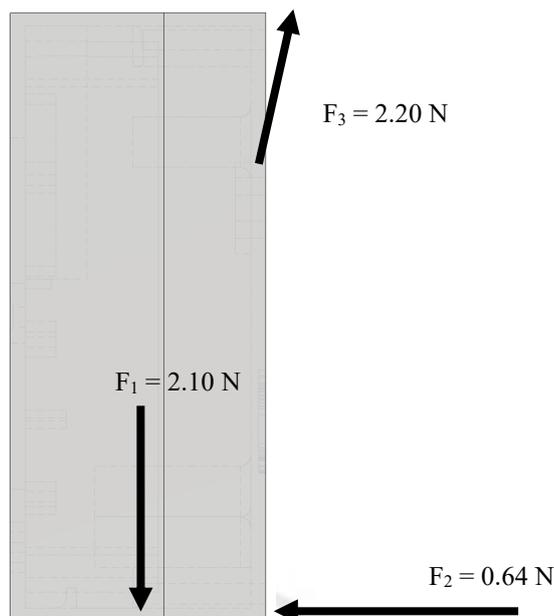


Figura 3.22: Diagrama de cuerpo libre (DCL) del dispositivo tomacorriente. Fuente: [21]

Como las componentes de la F_3 son F_1 y F_2 , la componente que se usará para la comparación será F_2 , ya que es la que se encuentra en la horizontal.

En las pruebas de calidad que se realizan a los tomacorrientes de manera horizontal apuntando hacia el piso, según las normas IEC / UL (ensayo de Nema), se debe poder retener un enchufe de prueba con una carga de 1.4 kg por un minuto. Esto asegura que la retención de los tomacorrientes sea la óptima. Esto equivale a 14 N de fuerza de retención o sujeción para los tomacorrientes. Comparando con la fuerza F_2 que se usará para la comparación tenemos:

$$F_2 \leq 14 \text{ N}$$

Ecuación 3-6: Prueba de retención o sujeción de enchufes.

Como sabemos, F_2 es 0.64 N, lo que hace que sea una fuerza muy pequeña y se concluye que teóricamente no debería tener ningún problema de sujeción en ningún tomacorriente estándar. Lo mismo pasaría si el material fuera PVC u otro plástico, ya que no hay mucha diferencia de peso.

3.5 Diseño electrónico

En el siguiente diseño electrónico, se ven las conexiones que se tendrán en el tomacorriente y las ubicaciones de cada dispositivo. Este diseño es referencial, ya que en el dispositivo irán separados dependiendo donde este el conector principal. El otro dispositivo también tiene un arreglo similar. Por ello, solo se tomará de ejemplo el dispositivo tomacorriente.

El microcontrolador está conectado al relé y al sensor de corriente para mandar y recibir datos. Todos los dispositivos se alimentan del HLK-PM01, ya que este arroja 5V y estos se distribuyen en todo el sistema. (Ver figura 3.23)

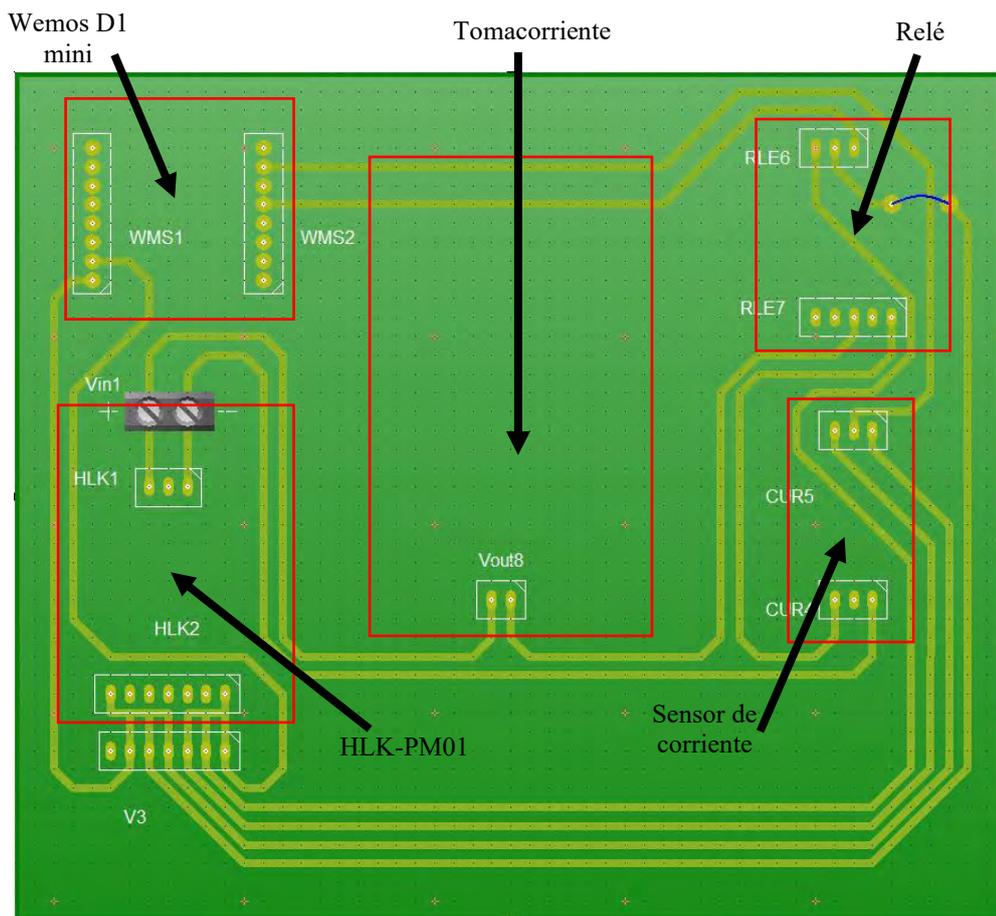


Figura 3.23: Diseño electrónico inicial del dispositivo para tomacorrientes. Fuente: [21]

3.5.1 Esquemáticos de dispositivos electrónicos:

Ahora se pasarán a mostrar los esquemáticos de los dispositivos electrónicos seleccionados. Primero se presenta el diagrama de entradas y salidas, y el esquemático del Wemos D1 mini. (Ver figuras 3.24 y 3.25)

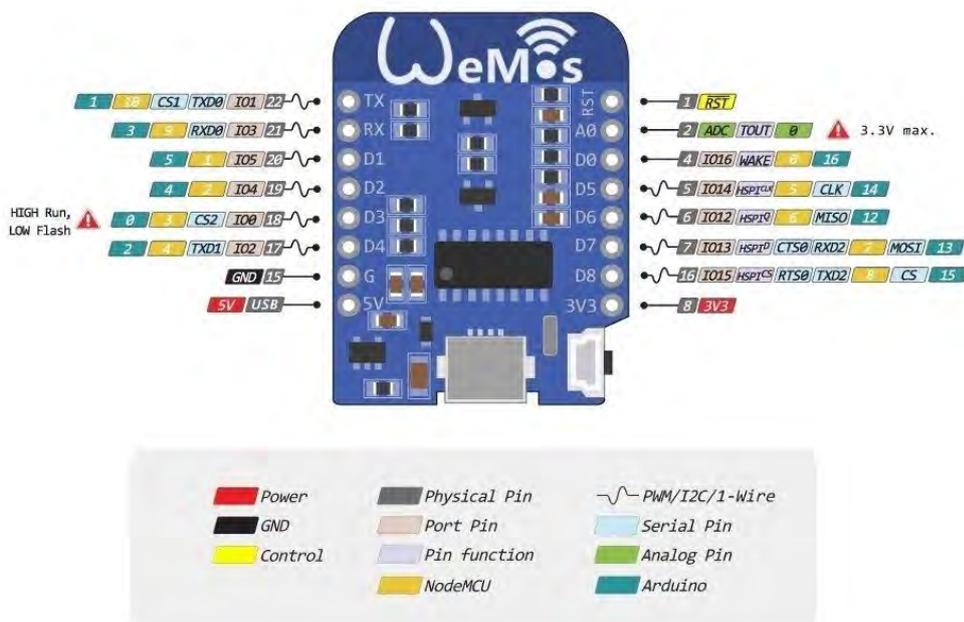


Figura 3.24: Diagrama de entradas y salidas del Wemos D1 mini. Fuente: [20]

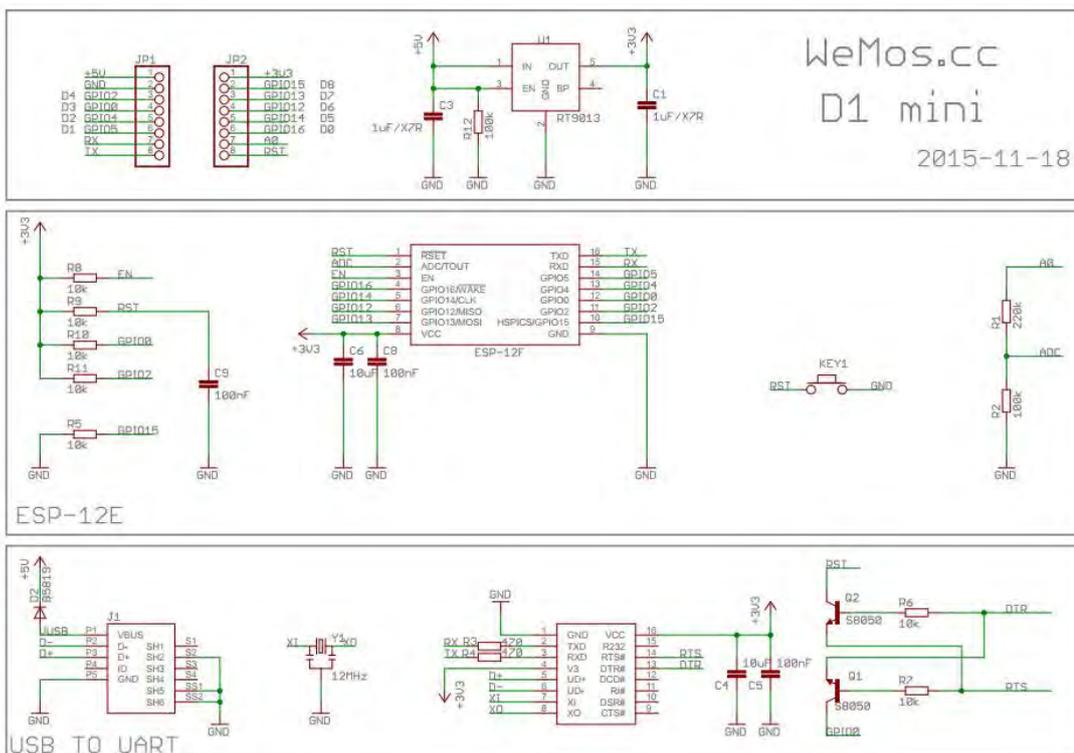


Figura 3.25: Esquemático del Wemos D1 mini. Fuente: [20]

El esquemático del HLK-PM01 que da el fabricante es bastante simple, y solo se preocupa de las dimensiones y del voltaje de salida y entrada. (Ver figura 3.26)

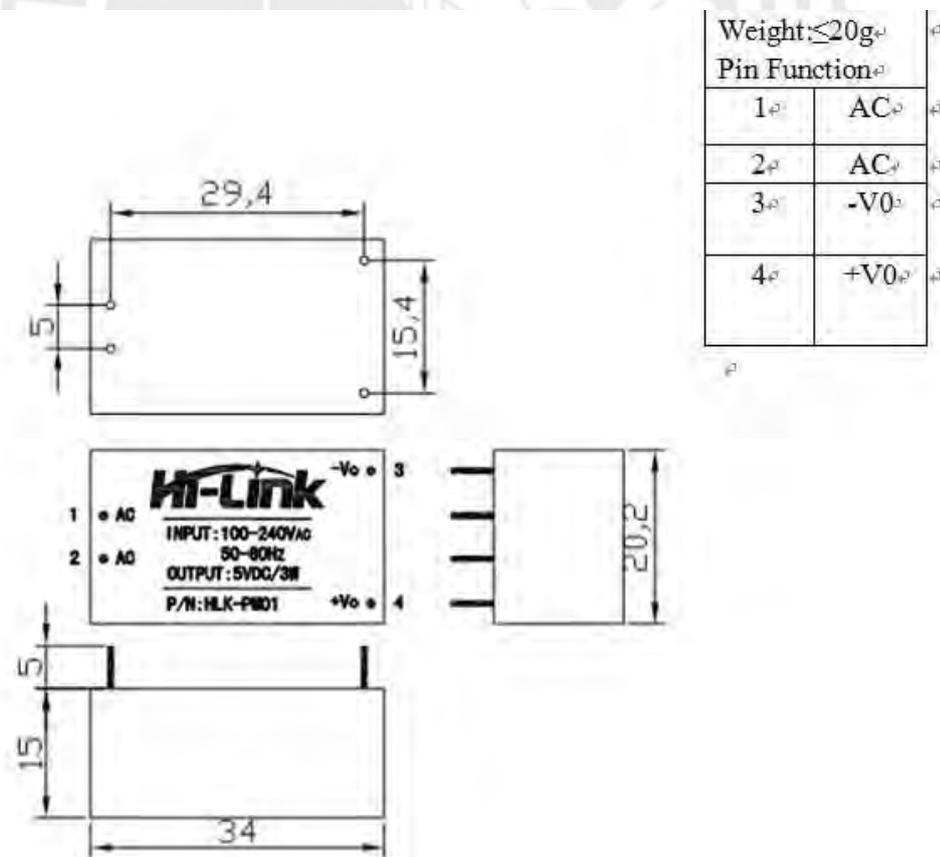


Figura 3.26: Esquemático del HLK-PM01. Fuente: [20]

El módulo ASC712 tiene un esquemático que parte de la aplicación de poder medir la corriente AC y DC por efecto Hall. Internamente, los pines IP+ e IP- se unen en uno solo. (Ver figura 3.27)

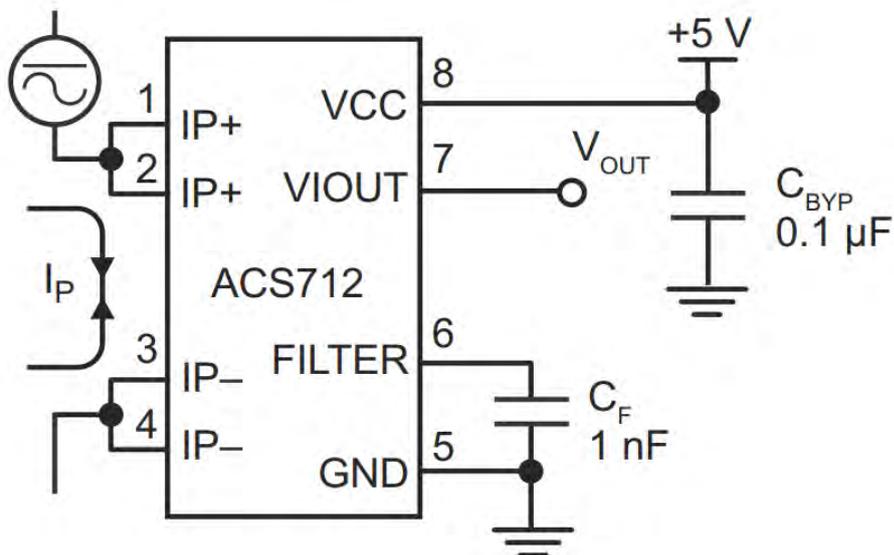


Figura 3.27: Encapsulado del sensor de corriente ASC712. Fuente: [20]

El relé electromecánico que usaremos, tiene la siguiente distribución en la que se nota los dos tipos de funcionamiento, el normalmente abierto y el normalmente cerrado. (Ver figura 3.28)

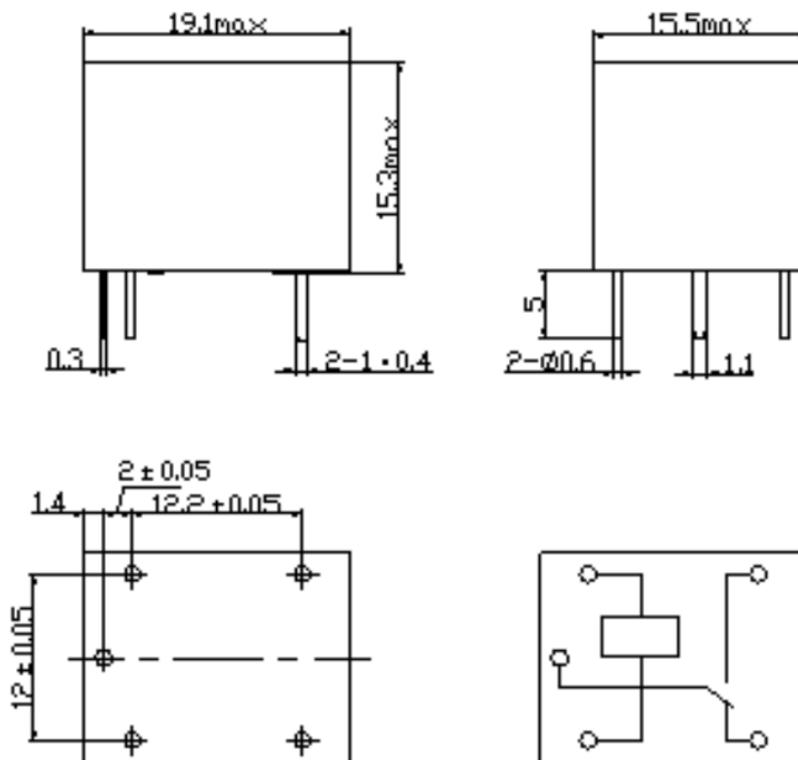


Figura 3.28: Esquemático del Relé. Fuente: [20]

Por último, se presenta el esquemático del dimmer AC. Este último funciona con un triac BTA16-600B, un optoacoplador y un optoaislador. (Ver figura 3.29)

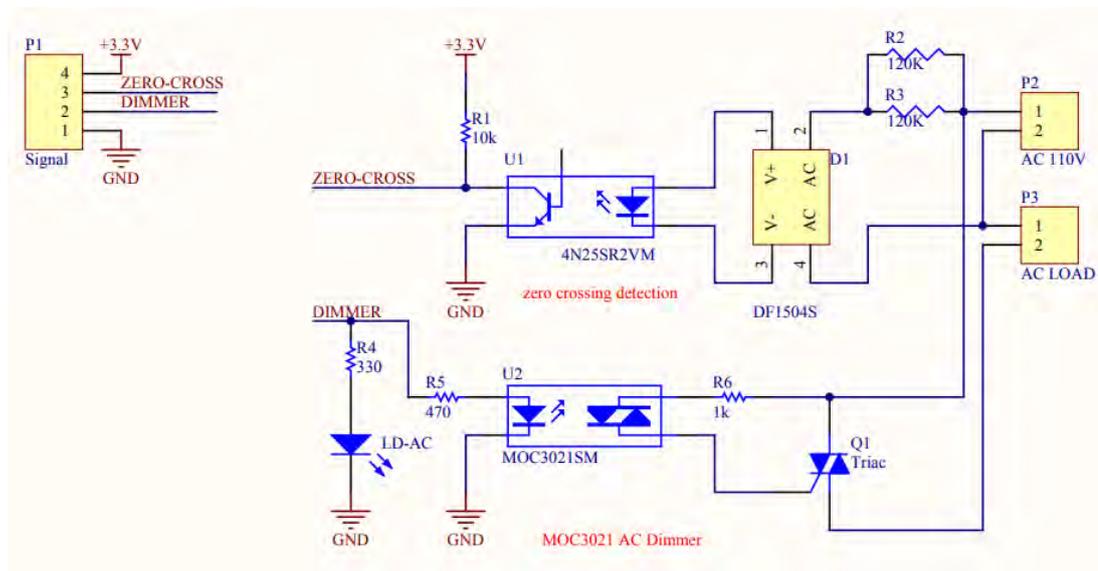


Figura 3.29: Esquemático del dimmer AC. Fuente: [20]

3.5.2 Cálculos electrónicos:

Ahora que se tienen los esquemáticos y los dispositivos conectados, se planteará el método de cálculo de las variables que se tienen. Se sabe que el sensor ASC712 entrega un voltaje proporcional a la corriente, y la resolución que tiene es de alrededor de 66 mV/A. Al momento de leer el sensor, se lo tiene que escalar. El Wemos D1 mini tiene convertidor análogo digital con una resolución de 10 bits, por ende, sus valores van desde 0 a 2^{10} o 1023 y estos a su vez, se debe pasar a una escala de 0 a 5V. Para esto solo basta con una regla de 3 simple para obtener:

$$V_s = L_s * (5/1023)$$

Ecuación 3-7: Voltaje escalado

Donde:

- V_s es el voltaje calculado en el rango de 0 a 5V.
- L_s es la lectura del sensor que va de 0 a 1023.

Una vez obtenido el voltaje del sensor, se debe transformarlo a corriente. Para esto, el voltaje debe estar en un rango negativo con centro en la mitad del valor medido, también llamado valor de reposo (valor en el que el voltaje valdría 0). Esto se hace porque el sensor mide entre -30A ~ 30A. Luego de esto se lo divide por la resolución que se tiene.

$$I_o = (V_s - 2.5 V) / (0.066 V/A)$$

Ecuación 3-8: Corriente calculada en cada lectura de sensor

Donde:

- I_o es la corriente calculada inicial.

Luego de esto, se le aplicará un filtro pasabajos para tener la mejor precisión posible. Esto se logra obteniendo el 90% del resultado anterior y sumarle el 10% de la medición actual. Con esto la corriente sería la siguiente:

$$I = 0.9 * I_0 + 0.1 * (V_s - 2.5 V) / (0.066 V/A)$$

Ecuación 3-9: Filtro pasabajos de la corriente calculada en cada lectura del sensor

Donde:

- I es la corriente calculada actual.

Después, se comparará el resultado con valores anteriores para hallar la mayor medición de corriente con la menor medición de corriente, ya que al estar midiendo corriente alterna, hay partes negativas y partes positivas. Este proceso se hace de manera continua e iterativa por 0.5 s.

A continuación, se resta esas dos corrientes para hallar la I_{pp} o corriente punto a punto.

$$I_{pp} = I_{m\acute{a}x} + I_{m\acute{i}n}$$

Ecuación 3-10: Corriente pico - pico

Donde:

- I_{pp} es la corriente punto a punto.
- $I_{m\acute{a}x}$ es la corriente máxima medida.
- $I_{m\acute{i}n}$ es la corriente mínima medida.

Ahora, se divide esa corriente entre 2 para poder hallar la corriente pico.

$$I_p = I_{pp} / 2$$

Ecuación 3-11: Corriente pico

Donde:

- I_p es la corriente pico.

Para poder hallar la corriente I_{rms} , se debe dividir la corriente pico entre raíz de dos.

$$I_{rms} = I_p / 2^{(1/2)}$$

Ecuación 3-12: Corriente Irms

Donde:

- I_p es la corriente pico.
- I_{rms} es la corriente rms.

Se tiene la manera de calcular las variables de corriente, las cuales están relacionadas también por el gráfico de onda. (Ver figura 3.30)

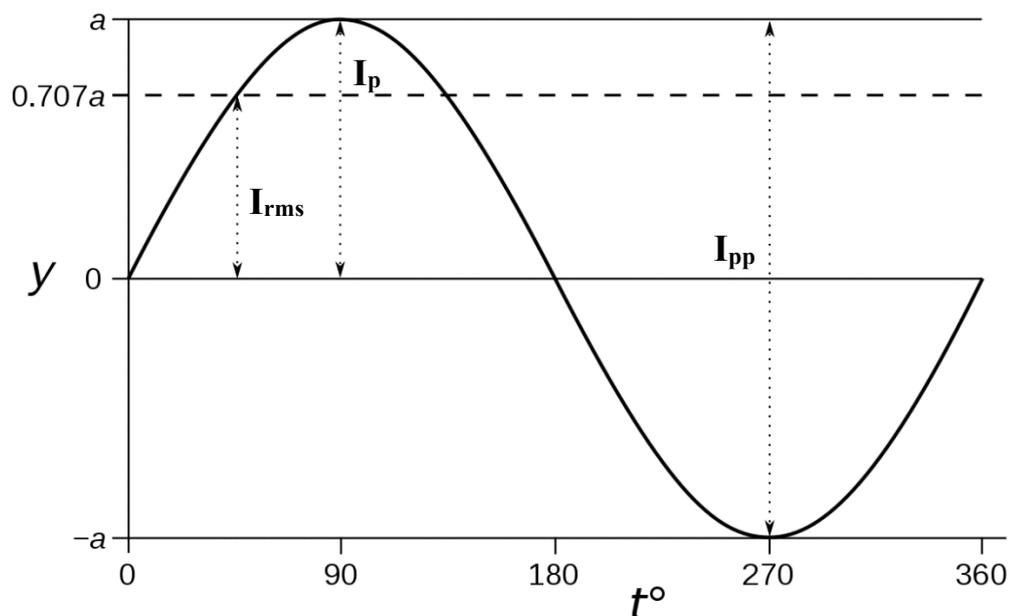


Figura 3.30: Relación entre las corrientes en la onda. Fuente. [21]

Ahora, se calcula la potencia multiplicando la tensión de la red, que es en promedio 230 V, por la corriente rms.

$$P = I_{rms} * 230$$

Ecuación 3-13: Potencia

Donde:

- P es la potencia en W.

Por último, se calcula los kilowatts hora.

$$E = P * t / (1000 * 3600)$$

Ecuación 3-14: Consumo de energía

Donde:

- E es el consumo en kWh.
- P es la potencia en W.
- t es el periodo de tiempo de muestreo en s.

En este caso, el periodo de muestreo que se va a utilizar es 5s. Por ello, la fórmula a usar sería:

$$E = 5 * P / (1000 * 3600)$$

Ecuación 3-15: Consumo de energía cada 5s

Con esto, ya tenemos la manera de obtener las 4 variables de interés:

- Corriente pico en amperios (\$I_p\$).
- Corriente rms en amperios (\$I_{rms}\$).
- Potencia en watts (P).
- Consumo kilowatt por hora (E).

3.6 Diseño de softwares

En este apartado del diseño de los aplicativos, se van a definir las tecnologías que se van a utilizar, los entornos en los que se va a trabajar, la arquitectura que se va a utilizar en el sistema IoT y la selección de los servicios adicionales. Se estará diseñando para 3 entornos diferentes, el primero es el del dispositivo IoT, el segundo es la aplicación web y el tercero será la aplicación móvil.

3.6.1 Descripción de las tecnologías a usar

Las tecnologías que se van a utilizar van a ser bastante variadas, ya que al ser un proyecto que abarca un sistema ciberfísico completo, se necesitan múltiples lenguajes de programación, herramientas de diseño, herramientas de versionamiento, herramientas de conexión, etc. Se van a presentar una por una y se describirá el uso que se le dará.

3.6.1.1 Git y GitHub

Este proyecto es bastante grande y es muy propenso a tener fallas y que se cometan errores. Para esto, se usará la plataforma GitHub, la cual está basada en el gestor de versiones Git. En ella, se almacena la programación de los dispositivos, de la aplicación web y también de la aplicación móvil.

Se crearán 3 repositorios privados en GitHub y cada uno almacenará una aplicación o codificación diferente. Todo esto se hará bajo una cuenta personal en la plataforma. (Ver imagen 3.31)

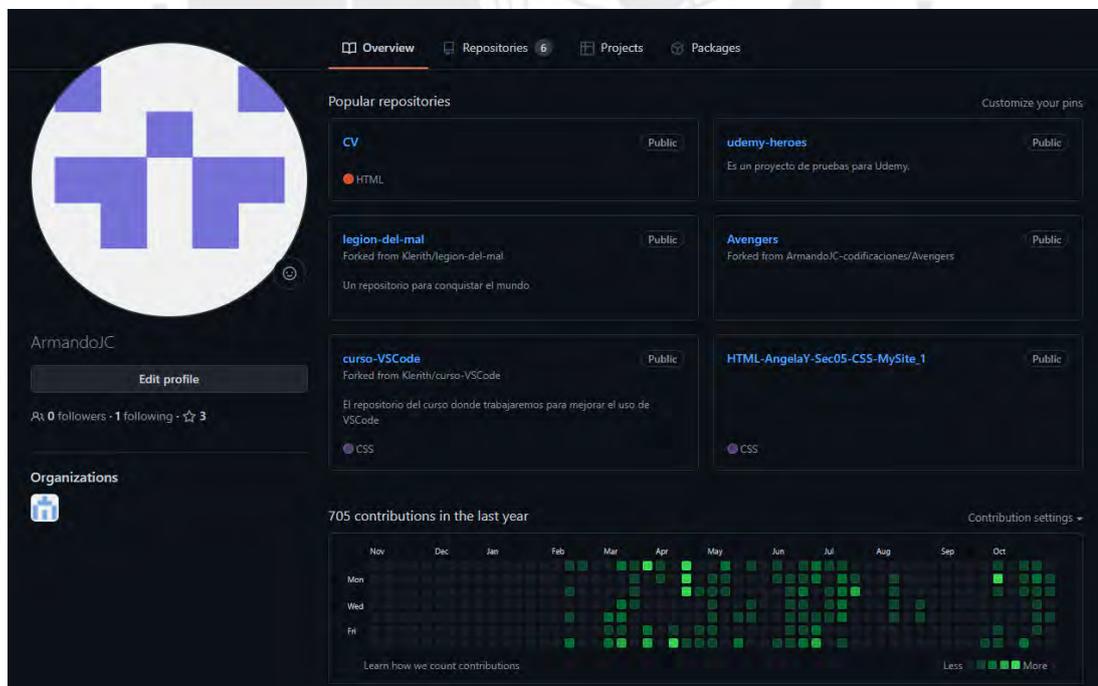


Figura 3.31: Overview del perfil personal en GitHub a utilizar. Fuente. [21]

3.6.1.2 PuTTY

Si el equipo de cómputo que se usará para poder conectarse al servidor es Windows, entonces se hace necesario este programa. En otros equipos con otros sistemas

operativos, no es necesario el uso de PuTTY y se puede conectar directamente por medio de SSH.

3.6.1.3 HeidiSQL

En caso de usar una base de datos SQL, se necesita un gestor bastante cómodo y versátil. Uno de estos gestores bastante recomendables es HeidiSQL, que puede conectarse a las bases de datos y gestionarlas desde su interfaz.

3.6.1.4 Figma

Para poder hacer los diseños de la aplicación web y de la aplicación móvil se requiere una herramienta de diseño para poder hacer los maquetados. Figma es una herramienta bastante versátil y útil. Una de las principales ventajas es que es una aplicación web que funciona en la nube y no un programa instalado localmente. Se puede compartir proyectos para trabajo colaborativo y también se tiene un gestor de versiones en la capa de pago. (Ver imagen 3.32)

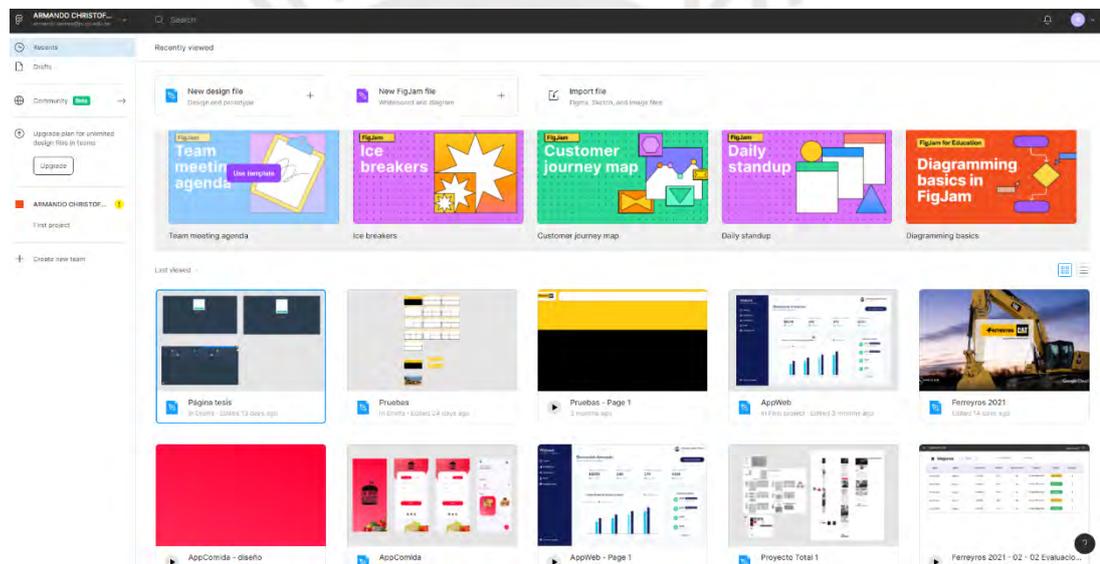


Figura 3.32: Interfaz de Figma para el diseño de aplicaciones. Fuente. [21]

3.6.1.5 Lenguajes de programación

Para poder programar la aplicación web serán necesarios varios lenguajes. En primer lugar, para la parte de la aplicación que ve el usuario y se ejecuta en los navegadores web (también llamado Frontend) se usarán los siguientes lenguajes:

- HTML5: Es un lenguaje de marcado y sirve para poder estructurar el documento web.
- CSS3: Es un lenguaje de diseño para darle estilos a la aplicación web. Se encarga del apartado visual.
- Bootstrap4: Es un framework con plantillas preparadas para facilitar el diseño web.
- JavaScript: Es un lenguaje de programación para darle interacción a la aplicación web con el usuario.

En segundo lugar, para la programación que se ejecuta en el servidor web (también llamado Backend) se usarán los siguientes lenguajes:

- PHP: Es un lenguaje de programación orientado a objetos para la parte del servidor y que está muy difundido en el mundo.
- NodeJS: Es el lenguaje de programación JavaScript, pero usado para el lado del servidor. Es especialmente útil para aplicaciones en tiempo real.

En tercer lugar, para el tratamiento de la base de datos y de la conexión con los servidores donde se alojará la página web se usarán los siguientes lenguajes:

- SQL: Es un lenguaje de consultas hacia bases de datos relacionales. Se elije este lenguaje en lugar de NoSQL, ya que se requieren datos que sean consistentes y que no den pie a errores y las bases de datos relacionales cumplen con 4 propiedades que son bien conocidas que son la atomicidad, consistencia, aislamiento y durabilidad (ACID por sus siglas en inglés) [26]. El lenguaje NoSQL es altamente escalable y da mayor facilidad para trabajar con volúmenes grandes de información, pero para este proyecto en específico se eligió lenguaje SQL también por motivos académicos.
- Terminal: Es un lenguaje de comandos para poder moverse y utilizar las diferentes terminales que existen.

Para programar la aplicación móvil, se podría utilizar también JavaScript, pero también se considerará un lenguaje de programación relativamente joven que sirve para desarrollo móvil:

- Dart: Es un lenguaje de programación de Google que sirve para poder programar aplicaciones móviles y es usado en algunos frameworks de Google.

Para programar los dispositivos IoT, el lenguaje a usar sería el siguiente:

- C: Es un lenguaje de programación que tiene muchas aplicaciones y que se usar también para la programación de microcontroladores.

Estas son algunas tecnologías que se planean usar para el desarrollo de las aplicaciones y programación en general.

3.6.2 Selección de los entornos de programación

3.6.2.1 Editor de código

Los entornos de programación que se van a utilizar tienen que ser cómodos y ayudar a que el desarrollo sea mucho más sencillo y rápido. Para esto, se tendrá que elegir un editor de código bastante flexible y potente. Actualmente se tienen varios disponibles como:

- Sublime Text 3
- Atom
- Visual Studio Code
- Brackets

Para este proyecto se ha seleccionado a Visual Studio Code como el editor de código a utilizar por las ventajas que trae y la cantidad inmensa de plugins que posee y facilitan mucho el trabajo.

3.6.2.2 Framework para desarrollo web

Para la aplicación web, se requiere un framework que ayude a que el proceso de programación sea bastante rápido. Para este existen varios como:

- Angular
- React
- Vue
- Code Igniter
- Laravel

Las 3 tecnologías más famosas son Angular, React y Vue. React es una librería y tiene un rango de posibilidades mucho más pequeño. Un framework es más robusto que las librerías y eso es lo que se necesita. Por lo tanto, las opciones que se tendrían serían Vue y Angular. El primero tiene una curva de aprendizaje bastante corta en comparación al segundo, pero Angular tiene una curva de aprendizaje alta, pero luego de dominarlo, se podrá crear aplicaciones web en el menor tiempo posible. [24]

En este caso, se seleccionará Angular debido a su robustez y la velocidad de desarrollo que se tendría al dominarlo. Además, tiene plantillas bastante buenas para iniciar la aplicación y es bastante versátil.

3.6.2.3 Framework para desarrollo móvil

Para la aplicación móvil, se requiere otro framework de trabajo. Para este caso, existen de varios tipos. Algunos son para desarrollar aplicaciones nativas, otros, para aplicaciones que se ejecutan en el navegador y también otros frameworks para aplicaciones multiplataforma. Algunos de los mencionados son los siguientes:

- React native
- Ionic
- Flutter

Para el rápido desarrollo de la aplicación, se ha escogido el framework Flutter, ya que es multiplataforma y solo tendría que hacer un solo desarrollo para sistemas operativos Android y iOS. Es un framework relativamente nuevo, pero que tiene mucha aceptación y una comunidad bastante grande que lo respalda. El rendimiento que alcanzan las aplicaciones desarrolladas en este framework son bastante es bastante bueno.

3.6.2.4 Framework para desarrollo en microcontroladores

Para la programación de los dispositivos, se necesita un entorno de desarrollo adecuado. Se tiene las siguientes opciones:

- Arduino

- PlatformIO
- VisualGDB
- Espressif ESP

Como la tarjeta seleccionada es un Wemos D1 mini, esta tarjeta puede programarse en el entorno de Arduino, pero este no es el único framework con el que se puede programar estas tarjetas y se lo considera uno de los más básicos. Por eso, se usará PlatformIO, ya que este entorno de desarrollo puede ser usado en el editor Visual Studio Code que ha sido seleccionado líneas arriba.

Estos son los entornos de programación que se usarán para el desarrollo de las aplicaciones.

3.6.3 Arquitectura del sistema IoT:

Para poder crear el sistema IoT, se debe conocer cuál es la cadena de valor del mismo. Esta cadena de valor está compuesta por 4 entidades, la primera es el dispositivo físico que es la que tiene los sensores y se encargará de recopilar la información del exterior. La segunda entidad es la conectividad, en esta se define la manera de envío de datos dependiendo de las condiciones y las distancias. Existen muchas maneras de transmitir los datos de manera inalámbrica, pero la elección de una u otra, va a depender del caso de uso. La tercera entidad es la plataforma en la nube, la cual se encarga de almacenar, proteger y gestionar los datos. La cuarta entidad es la aplicación en sí misma, esta toma los datos y los utiliza para generar conocimiento y valor. (Ver figura 3.33)



Figura 3.33: Cadena de valor del IoT. Fuente. [21]

Hasta ahora, se tienen diseñados los dispositivos y seleccionada la conectividad, que en este caso sería Wifi. Ahora mismo, falta programar la lógica en los dispositivos, preparar la plataforma para que reciba los datos y también crear las aplicaciones. Se puede ver que cada una de estas etapas es importante y dependiente de la anterior.

Los dispositivos IoT normalmente tienen un esquema en el que se tiene al servidor, los dispositivos con sus sensores y actuadores y por detrás una base de datos con una aplicación. Este esquema es algo básico y se usa para desarrollar proyectos pequeños y prototipos, pero que no son escalables a largo plazo. (Ver figura 3.34)



Figura 3.34: Esquema básico IoT. Fuente: [21]

Para que el sistema sea escalable en el tiempo y robusto, se debe añadir funcionalidades extras que ayuden con este fin. En el lado del servidor, es necesario añadir medidas de seguridad para que el sistema sea más confiable. Adicional a esto, se debe tener un motor de reglas que reaccione en base a los datos que envían los sensores. En la parte de los dispositivos, si es muy crítica la tarea, se debe crear un motor de reglas offline, este se encargará de operar los dispositivos si es que ocurre alguna caída de red. Con estas funcionalidades añadidas, el sistema es más robusto y listo para trabajar en la industria. Claro que hay más funcionalidades por detrás, pero lo más general ya está considerado. (Ver figura 3.35)

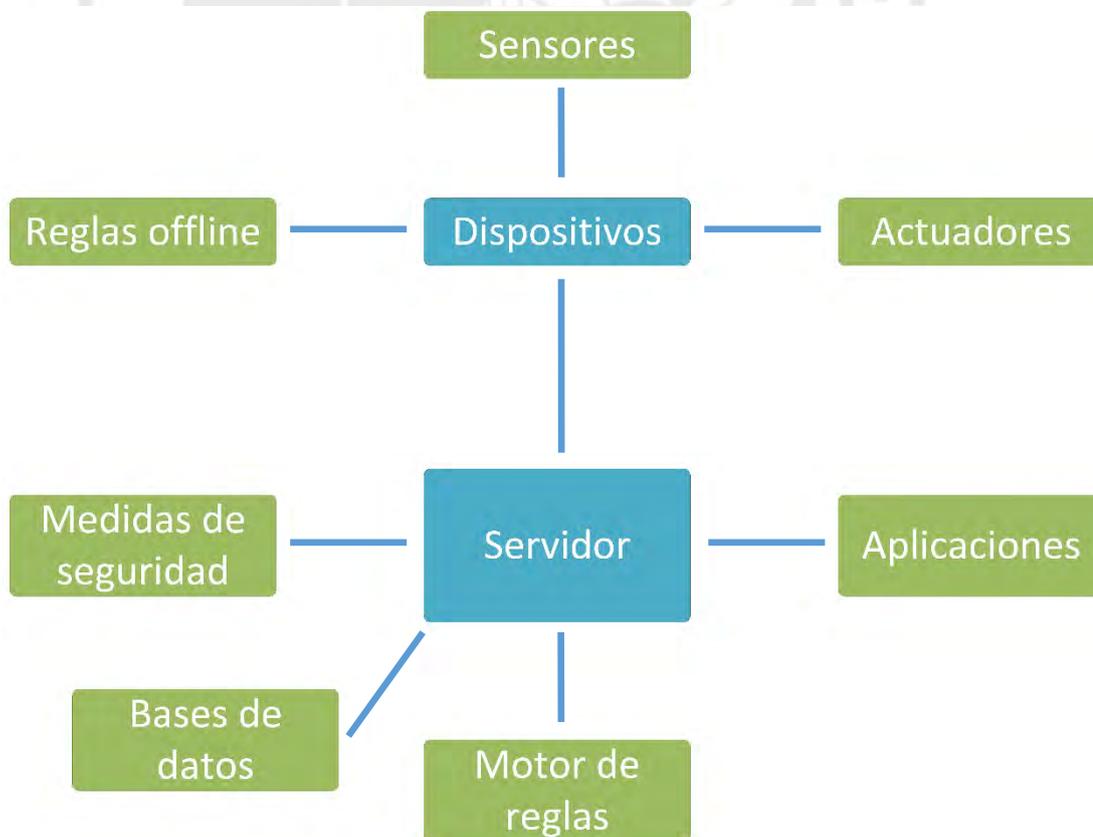


Figura 3.35: Esquema avanzado IoT. Fuente: [21]

Con este esquema, podemos ver que el servidor es bastante importante y se debe seleccionar de manera adecuada, ya que este tiene varias funcionalidades que cubrir y albergar.

El hosting es una de las funcionalidades que estarán dentro del servidor y será el encargado de alojar todas las aplicaciones web, páginas institucionales, etc. A parte del hosting, también el broker estará alojado en el servidor y este es el encargado de las comunicaciones. Las medidas de seguridad y el motor de reglas también se alojarán en el servidor, así como las bases de datos que se puedan crear o necesitar. (Ver figura 3.36)

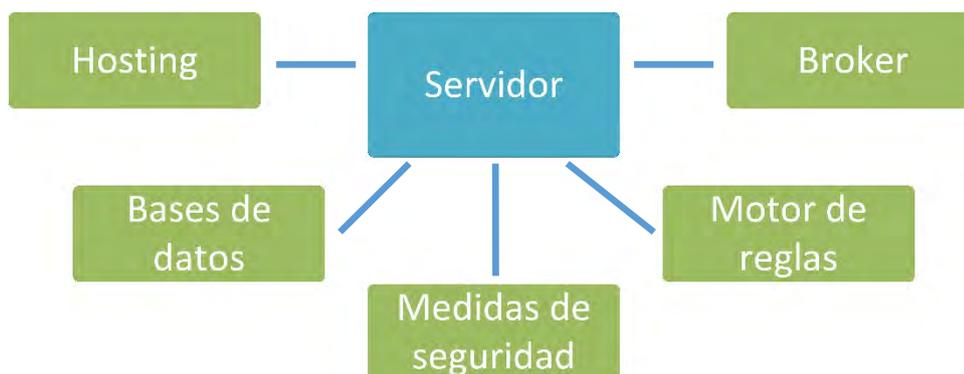


Figura 3.36: Funcionalidades del servidor. Fuente:[21]

Ahora que tenemos claras las actividades más importantes que va a realizar el servidor, se tiene que decidir si esas funcionalidades las van a realizar el mismo o se va a requerir servicios de terceros para que realicen esas actividades en sus propios servidores.

Para este proyecto, se prescindirán de servicios de terceros que se realicen fuera de nuestro servidor. Todos los servicios que tengamos lo haremos dentro del mismo. Para esto, la arquitectura de la solución no dependerá de servicios especializados de las plataformas, si no solo del espacio para alojar el servidor.

Por esto, se tendrán una gran cantidad de dispositivos, los cuáles se conectarán a internet por medio de la señal wifi y transmitirán sus mensajes al router o fuente del internet para que este los redirija hacia el servidor, específicamente a la plataforma MQTT alojada en el servidor. Esta plataforma se encargará de redirigir todos los mensajes tanto a las aplicaciones web y móvil como a los dispositivos y a un servicio en nodeJS que estará encargado de la inserción de la información enviada por los dispositivos a la base de datos, también a la autenticación de los dispositivos y los usuarios. Las aplicaciones web y móvil, también deberán conectarse directamente con la base de datos para obtener la información necesaria para poder funcionar correctamente y de poder insertar datos que no serán enviados por medio del protocolo MQTT. Esta arquitectura, tendrá las 4 partes de la cadena de valor del dispositivo IoT planteado al principio. (Ver imagen 3.37)

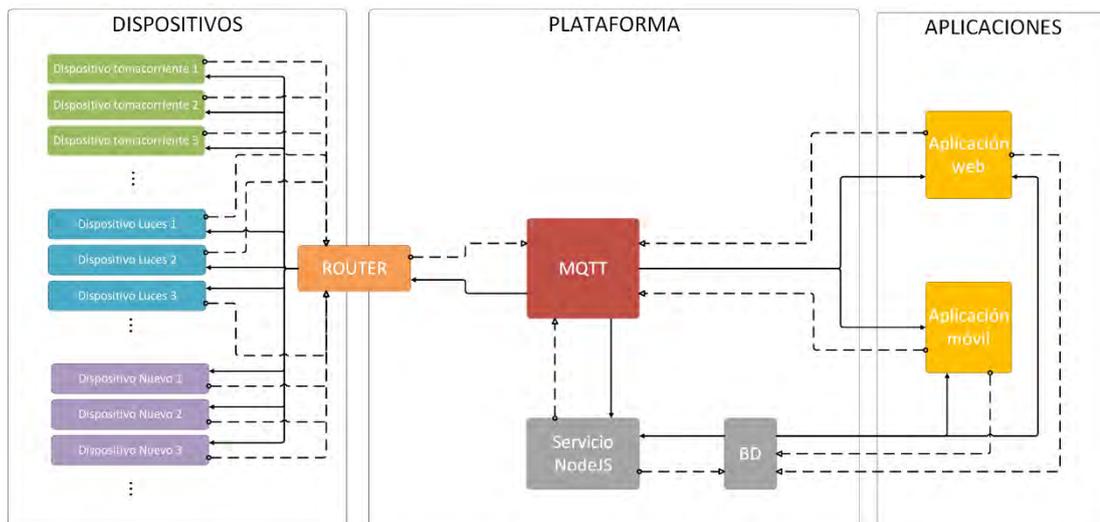


Figura 3.37: Arquitectura IoT propuesta Fuente: [21]

3.6.4 Selección de los servicios

Se van a seleccionar todos los servicios que se van a usar por parte de los softwares. Para el protocolo de comunicación, la mejora alternativa que se tiene es usar MQTT, ya que este protocolo está muy preparado para sistemas IoT y, además, hay mucha información disponible para poder utilizarlo. Ahora solo corresponde seleccionar la plataforma de cloud computing que alojará al servidor, el hosting y el broker.

3.6.4.1 Plataforma cloud computing

En el caso del servidor, se tiene 3 opciones para poder elegir:

- Tener una computadora física que se encargue de ser el servidor.
- Contratar o instalar racks físicos en donde podamos alojar nuestro ordenador.
- Contratar un servidor privado virtual (VPS por sus siglas en inglés).

Debido a que las dos opciones iniciales suponen un gasto alto y un rendimiento que depende del mantenimiento que le pueda dar, se decide por contratar un VPS. Estos VPS los ofrecen diversas empresas por medio de servicios.

- Amazon Web Services
- Google Cloud
- Microsoft Azure
- Hostinger

En este caso, se va a trabajar con *Amazon Web Services* y se utilizará el servicio EC2 en el que se creará una instancia en la que se concentrarán todos los servicios; por lo tanto, se usará una arquitectura *Stand-Alone* que no garantiza alta disponibilidad.

3.6.4.2 Hosting

Para el web hosting, se va a usar Vesta Control Panel, que es un panel muy conocido y que tiene prestaciones bastante aceptables y unos requisitos mínimos para funcionar muy bajos, lo que lo hace perfecto para que se pueda instalar fácilmente en cualquier servidor.

3.6.4.3 Broker

Para el broker MQTT, se tiene varias opciones bastante conocidas como:

- Mosquito
- Mosca
- EMQ

Para este caso, se va a seleccionar EMQ, aunque sea el menos conocido, ya que, a diferencia de los otros dos, es bastante escalable y su rendimiento es muy bueno. Esto queda demostrado en las pruebas de rendimiento y escalabilidad que realizan con 10 millones de conexiones en simultáneo. [25] Mosquito y Mosca son los más conocidos, pero usados mayormente en ambientes de desarrollo.

3.6.5 Medidas de seguridad

Las medidas de seguridad que se toman de cara al usuario son inicialmente por medio de la creación de una cuenta con un correo de preferencia y una contraseña. Estos datos serán guardados en la base de datos y el password se encriptará. Toda esta operación se realizará en el backend para que no sea de fácil acceso.

Adicionalmente, *Amazon Web Services* provee de cortafuegos (firewall) que proporciona grupos de seguridad en el que podemos abrir puertos específicos para los programas o servicios que se utilizan. También, *Vesta Control Panel* provee del mismo servicio de cortafuegos, lo cual hace que se tenga 2 líneas de defensa contra programas maliciosos o atacantes que quieran tener acceso a la red o a la información.

3.6.6 Gestión del proyecto – metodología ágil SCRUM

Para la parte de la gestión del proyecto, se toma como base la metodología ágil SCRUM que es muy usada para el desarrollo de software. En esta se definen sprints para entregar y se pueda tener un producto mínimo viable (MVP por sus siglas en inglés) en el menor tiempo posible.

Solo se lo tomará como referencia para mostrar el camino a seguir para el desarrollo de estos softwares, ya que solo hay un programador encargado del proyecto. En cada término de release, se tiene un MVP. Para la presente tesis, solo se llegará hasta el release 1, que es la etapa inicial y en donde solo se muestra funcionalidad, quedando abierto mejoras y usos de otros servicios como los que tiene *Amazon Web Services*. (Ver tabla 3-12)

Tabla 3-12: Gestión de proyectos – metodología scrum. Fuente: [21]

Aplicaciones		Q3-2021			Q4-2021			Q1-2022		
		Agosto	Setiembre	Octubre	Noviembre	Diciembre	Enero	Febrero	Marzo	
Dispositivo	Release 1	User Stories	P. Backlog	Sprint1	Sprint2					
	Release 2				Sprint3	Sprint4				
	Release 3						Sprint5	Sprint6		
Aplicación web	Release 1	User Stories	P. Backlog	Sprint1	Sprint2	Sprint3				
	Release 2					Sprint4	Sprint5	Sprint6		
	Release 3							Sprint7	Sprint8	Sprint9
Aplicación móvil	Release 1		User Stories	P. Backlog	Sprint1	Sprint2				
	Release 2						Sprint3	Sprint4		
	Release 3							Sprint5	Sprint6	

CAPÍTULO 4

4. IMPLEMENTACIÓN DE LOS APLICATIVOS WEB Y MÓVIL

En este capítulo, se detallará el proceso de implementación de los aplicativos y el desarrollo de la programación de los dispositivos.

4.1 Preparación para el desarrollo de los aplicativos

Se despliega todas las herramientas y programas seleccionados en el capítulo pasado. En el Anexo D, se podrán ver la creación de la instancia y la instalación del panel Vesta.

4.1.1 Creación del logo de SmartHouse

Ahora que se va a diseñar la aplicación web y móvil, es necesario tener una identidad y para esto se ha diseñado un logo con la herramienta Figma, que representa el proyecto de tesis. (Ver figura 4.1)



Figura 4.1: Logo de smarthouse. Fuente: [21]

4.1.2 Preparación del editor de código

Usaremos VSCode y para poder editar los archivos directamente al servidor y para esto se deberá crear una conexión FTP por medio de los plugins que tiene el editor. Una vez creada la conexión, se podrán editar los documentos localmente y se actualizarán en el servidor, lo que hará que se pueda ver el cambio actualizando la página web en el navegador.

4.2 Diseño de la base de datos

Se procederá a ver el diseño de la base de datos que se usará para el proyecto. En este caso, se usará una base de datos relacional en MySQL, pero para administrarla se usará la herramienta presentada anteriormente, HeidiSQL.

4.2.1 Descripción de los requerimientos

Se necesita administrar usuarios para que puedan ingresar al sistema y usarlo. Estos usuarios tienen una fecha de registro a la aplicación, sus nombres, apellidos, un alias o apodo, un email, un password, así como también tiene un tipo de usuario. Este usuario puede tener varias casas, las cuales, tienen un alias, la cantidad de personas que residen en ella, una descripción y un tipo de casa para poder diferenciarlas. Cada una de las casas tendrán varios cuartos, estos tienen una fecha de registro, un alias, una descripción y un tipo de cuarto. Luego se definen los dispositivos, que están registrados a nombre de un cuarto. Estos dispositivos deberán tener un alias, un número de serie, el tipo del dispositivo y una pequeña descripción. También se debe tener una tabla de valores, en donde se va a ir registrando todos los datos que los sensores envían. Los datos de interés son la corriente Irms, la corriente pico, la energía gastada, la energía consumida por hora, un valor para colocar si el dispositivo está o no encendido y la medida de la intensidad luminosa que se necesita. Adicional a esto, se necesita los dispositivos producidos. Esto debe tener el tipo del dispositivo, la serie, el propietario, el costo y si ya fue entregado. Este apartado solo es para emular que se tiene un lugar en la nube para consultar con respecto a la autenticación. Con esta información, ya se podría realizar el diseño de la base de datos. (Ver imagen 4.2)

4.2.2 Diagrama entidad – relación

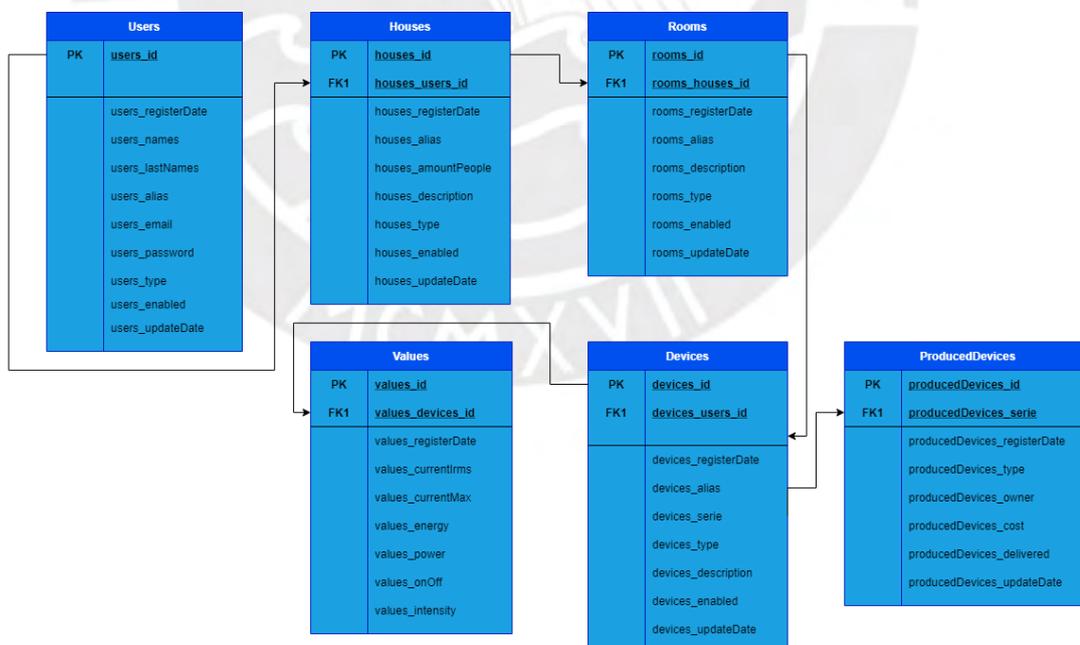


Figura 4.2: Diagrama entidad - relación. Fuente: [21]

4.2.3 Tablas creadas

Utilizando la herramienta HeidiSQL, podemos observar las tablas creadas. Se agregó la columna de registro de la fecha de actualización como buena práctica para hacer seguimiento a los registros. Además de esto, se le añadió la columna de habilitación, que controlará si el dispositivo se muestra o no. Esto se hace para que no se elimine la data ingresada, sino solo se deje de visualizar para un posterior seguimiento. (Ver figura 4.3)

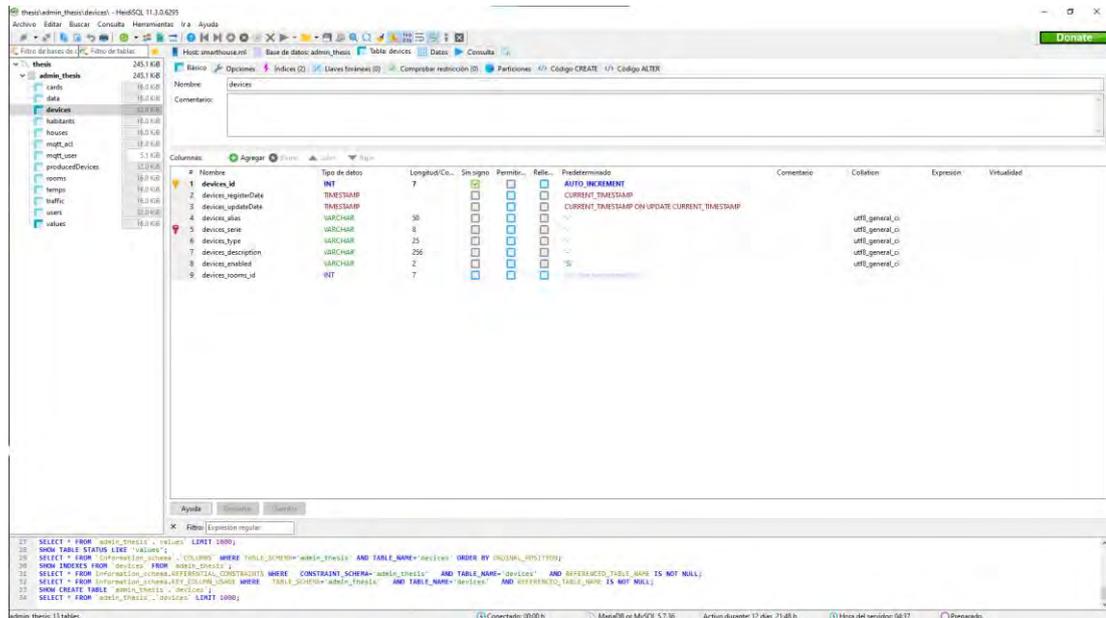


Figura 4.3: Tabla devices creada en la base de datos de acuerdo al diseño. Fuente [21]

4.3 Broker MQTT

El broker que se va a usar es el EMQX. Para no tener problemas en su uso, deben permitir el acceso de los puertos en los grupos de seguridad de AWS y en el cortafuegos de Vesta Panel. La instalación del broker se encuentra en el Anexo D.

4.3.1 Diseño de la estructura de tópicos a usar

Ahora que se tiene el broker instalado, se deberá diseñar la estructura de tópicos a utilizar. En el diseño de la base de datos, tenemos varias entidades y para poder separarlas e identificarlas sería recomendable tenerlas en cuenta. Las entidades que son de interés para separar son las casas, las habitaciones y el dispositivo. Como existen varios tipos de dispositivos, se tendría que tomar en cuenta para el diseño de tópicos. Como propuesta se tiene esta estructura de tópico:

{Negocio}/{Casa}/{Habitación}/{TipoDispositivo}/{SerieDispositivo}/{Variable}

Con este tipo de tópico abarcamos diferentes negocios que se puedan realizar, se separa por casa, habitación y tipo de dispositivo. Con esto, se está bastante segmentado, pero para poder identificar el dispositivo individualmente se necesitaba poner el número de serie. Luego de eso, se coloca la variable general que se usará.

Como ejemplo de tópicos formados se tienen:

- SmartHouse/1/2/Tomacorriente/44447777/valores
- SmartHouse/1/5/Luces/55679877/Comandos

Adicionalmente, el dispositivo para operar deberá solicitar credenciales a la base de datos y esto se hará por medio del servicio node.js. Para esto, se usará este tópico especial:

{Negocio}/Credentials/{SerieDispositivo}

Como ejemplo de tópicos formados se tienen:

- SmartHouse/Credentials/44447777
- SmartHouse/Credentials/12345678

4.4 Programación del servicio node.js

Este servicio es el que se encargará de la inserción de los datos recopilados por los sensores a la base de datos, de la autenticación de dispositivos y de la autenticación de usuarios. Se creará en una carpeta en el servidor para que no sea vista por el usuario, ya que tiene información sensible como la conexión a la base de datos.

4.4.1 Lógica de programación del servicio node.js

El dispositivo, para poder autenticarse, mandará los tópicos adecuados sin el número de serie en el tópico, sino en el mensaje. Esto ayuda a que el servicio pueda identificar a todos los dispositivos porque usan un único tópico y lo que los diferencia es el mensaje. Una vez recibido el mensaje, se consultará en la base de datos para poder obtener los valores de la casa, la habitación, el tipo de dispositivo y el id del dispositivo y eso se mandará bajo el tópico de credenciales con el número de serie, porque a ese tópico es al único al que inicialmente se puede suscribir.

El otro trabajo del servicio es la inserción a la base de datos y eso se da cada vez que llega algún dato bajo el tópico adecuado con la variable “values”. Se separa la cadena de texto que se recibe en un formato conocido y se extrae la información que se va a insertar en la base de datos. Luego solo se hace la consulta de inserción y se termina el proceso. (Ver figura 4.4)

```

41 //CUANDO SE RECIBE MENSAJE:
42 client.on('message', function(topic, message){
43     console.log("Mensaje recibido desde -> " + topic + " | Mensaje -> "
44
45     if(topic == "SmartHouse/Credentials"){
46         let newTopic = topic + "/" + message.toString();
47
48         // let query = "SELECT * FROM `admin_thesis`.`devices` WHERE de
49         let query = "SELECT r.rooms_houses_id, d.devices_rooms_id, d.de
50         con.query(query, function(err, result, fields){
51             if(err) throw err;
52             console.log(result);
53             if(result.length > 0){
54
55                 let casa = result[0]['rooms_houses_id'];
56                 let habitacion = result[0]['devices_rooms_id'];
57                 let tipoDispositivo = result[0]['devices_type'];
58                 let dispositivoId = result[0]['devices_id'];
59                 Slater [10 days ago] • Conexión con el dispositivo
60                 let mensaje = casa + "/" + habitacion + "/" + tipoDispo

```

Figura 4.4: Programación en node.js cuando se recibe el mensaje. Fuente: [21]

4.5 Programación de los dispositivos IoT

La programación de los dos dispositivos se hará por medio del entorno escogido que es PlatformIO. En el editor de código VSCode, existe una extensión que permite su uso. Se crea un proyecto y se empezará a realizar la programación. (Ver figura 4.5)

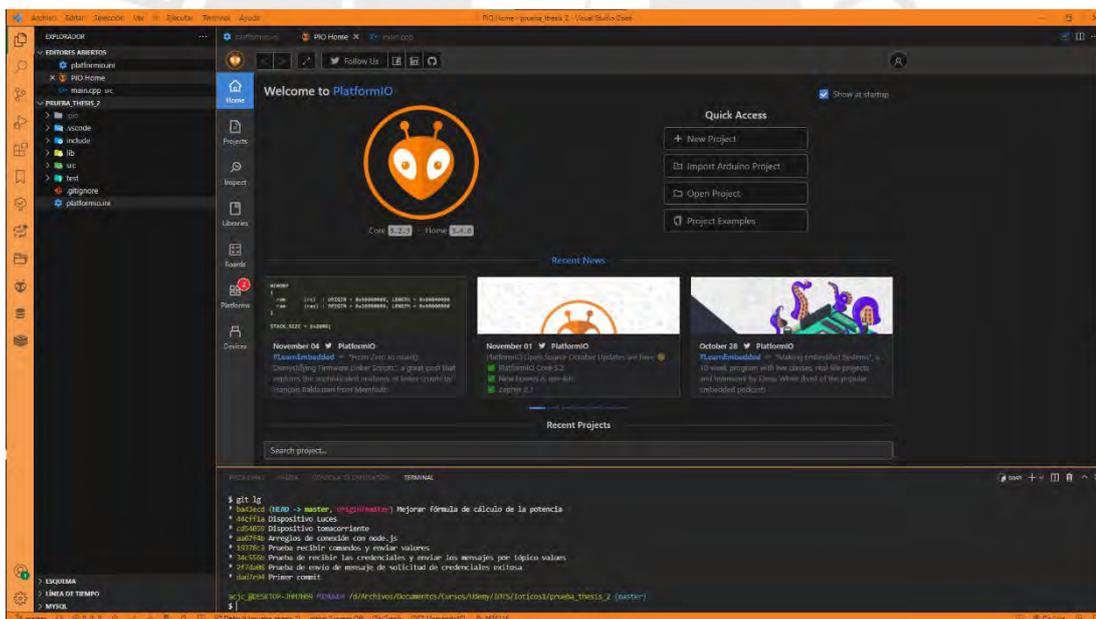


Figura 4.5: Entorno de platformIO en VSCode. Fuente: [21]

4.5.1 Lógica de programación de los dispositivos

Con el proyecto creado, se realiza la programación y es muy similar para los dos dispositivos. Se podría usar la misma, ya que no habría ningún tipo de contradicción.

Lo primero que se realiza es que se añade las librerías que se van a usar. Estas serían la librería “PubSubClient” y la librería “ESP8266WiFi”. Luego se definen todas las constantes necesarias, incluidas las credenciales del WiFi y del servidor. La variable más importante para el dispositivo es el número de serie, ya que este debe ser único

para cada dispositivo y con este se identifica para que pueda conectarse correctamente a la base de datos y autenticarse.

Luego se crean todas las variables que van a ser necesarias para el programa y se las inicializa. Después, se hace la declaración de las funciones que se crean como buena práctica de programación. (Ver figura 4.6)

```

76  //*****
77  //***** DECLARACIÓN DE FUNCIONES *****
78  //*****
79
80  void setup_wifi();
81  void callback(char *topic, byte *payload, unsigned int length);
82  void reconnect();
83
84  String formarTopico(String comando);
85  String obtenerSubStrings(String data, char separator, int index);
86
87  //Funciones prueba:
88  float calcularIntensidadPico();
89  float calcularCorrienteIrms();
90  float calcularPotenciaRMS();
91  double calcularKilowatts();
92
93  void correctValue();
94
95  //*****
96  //***** SETUP *****
97  //*****

```

Figura 4.6: Declaración de funciones a utilizar. Fuente: [21]

Ahora se empieza con el setup, en este se configura el puerto serial, se conecta al WiFi, se le da las credenciales para que se conecte al servidor y se asigna la función que será llamada cuando haya algún mensaje (callback()). Se configuran todos los pines como entradas o salidas. Por último, se inicializa al dimmer.

Por último, se entra al loop y se realiza una comprobación en cada ciclo para ver si se sigue conectado, y si no, la función “reconnect ()” se encargará de la reconexión. Luego de eso, se genera un condicional que solo se ejecutará después de 5s usando el tiempo transcurrido y no delays que hacen que el flujo pare y genere errores. En esta condicional, se calculan todas las variables que se necesitan y si el dispositivo está autenticado, se procederá a mandar la información por medio de una publicación MQTT y el tópico único del dispositivo. (Ver figura 4.7)

```

154   if (now - lastMsg > periodoMilisegundos)
155   {
156
157       // Serial.println("Pasaron 5s");
158       // Serial.println("*****");
159       currentMaxPico = calcularIntensidadPico();
160       currentIrms = calcularCorrienteIrms();
161       energy = calcularPotenciaRMS();
162       power = calcularKilowatts();
163       Serial.println("=====");
164       Serial.println(currentMaxPico);
165       Serial.println(currentIrms);
166       Serial.println(energy);
167       Serial.println(power);
168       Serial.println("=====");
169
170       lastMsg = now;
171
172       if (topicoRecuperado)
173       {
174           String to_send = String(currentIrms) + "," + String(currentMaxPico);
175           to_send.toCharArray(msg, MSG_BUFFER_SIZE); //Se transforma en un char
176
177           Serial.print("Publicamos mensaje: ");
178           Serial.println(msg);
179           topicoDispositivo = formarTopico("values");
180           // Serial.println("+++++++");
181           char topicoValores[MSG_BUFFER_SIZE];
182           topicoDispositivo.toCharArray(topicoValores, MSG_BUFFER_SIZE);
183           client.publish(topicoValores, msg);

```

Figura 4.7: Cálculo de valores. Fuente: [21]

Este es el comportamiento cuando se envían mensajes, pero cuando llegan se activa la función “callback ()” la cual se encarga de ejecutar los comandos que se envían. (Ver imagen 4.8)

```

290   if (onOff == 0)
291   {
292       Serial.println("Se apaga el dispositivo");
293       digitalWrite(LED_BUILTIN, HIGH);
294       digitalWrite(14, HIGH);
295   }
296   else if (onOff == 1)
297   {
298       digitalWrite(14, LOW);
299       Serial.println("Se enciende el dispositivo");
300       digitalWrite(LED_BUILTIN, LOW);
301   }
302   else
303   {
304       Serial.println("Dato onOff incorrecto");
305   }
306
307   // TODO: SETEAR INTENSIDAD
308   if (tipoDispositivo == "Luces")
309   {
310       Serial.print("Se seteará la intensidad a: ");
311       Serial.println(intensity);
312       potValue = intensity;
313       correctValue();

```

Figura 4.8: Callback. Fuente: [21]

Por último, indicar que se creó el reposito en GitHub para este proyecto y se realizaron varios commits para poder preservar el avance que se tenía. (Ver figura 4.9)

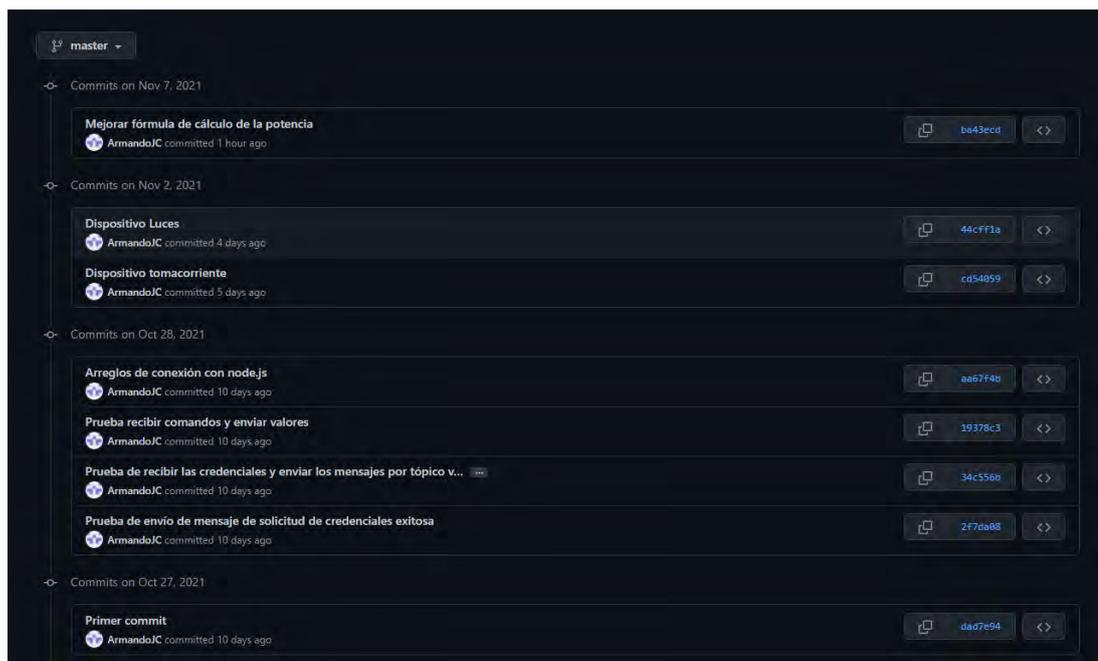


Figura 4.9: Repositorio en GitHub para los dispositivos. Fuente: [21]

Esta sería toda la programación de los 2 dispositivos. Se logró que se enviara y recibiera mensajes por medio de MQTT sin ningún problema y que se pueda identificar correctamente al dispositivo con el servicio node.js

4.6 Programación de la aplicación web

La aplicación web en este sprint ha sido programada para poder soportar registros de nuevas cuentas, autenticaciones para ingresar a la aplicación, ingresar nuevas casas, nuevas habitaciones, nuevos dispositivos, todos relacionados y se implementó el dashboards de los dispositivos, pero no de las habitaciones y las casas.

Se usó el logo diseñado para darle identidad y se realizaron todas las conexiones necesarias. Todo esto se ha realizado en el editor de código VSCode por medio de conexión FTP. (Ver imagen 4.10) También se ha realizado commits para poder ir guardando toda la información en el repositorio privado que se creó en GitHub como medida de seguridad.

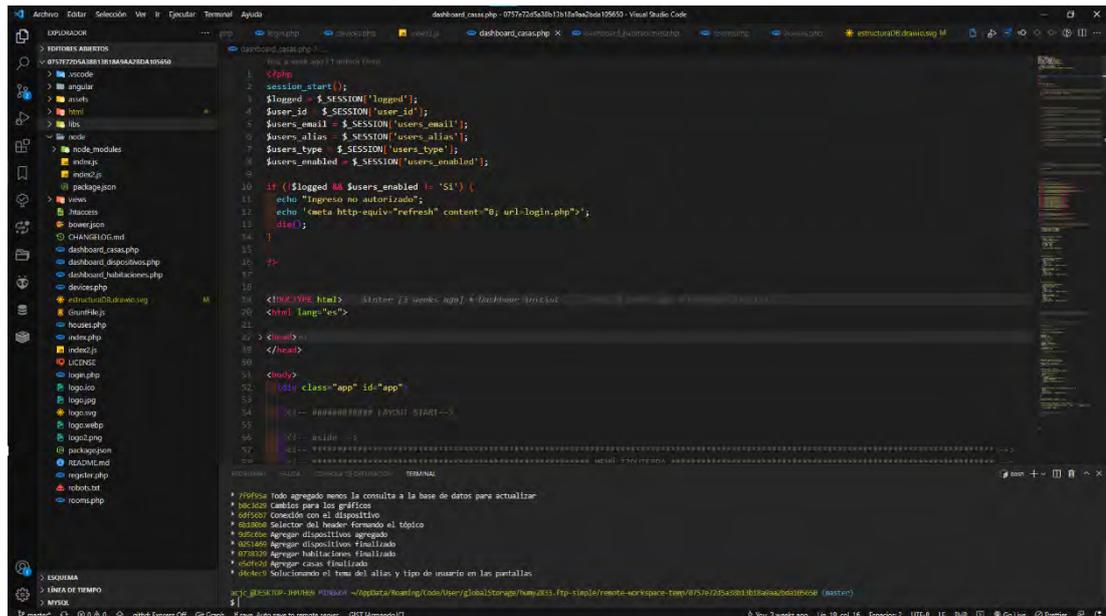


Figura 4.10: Conexión VSCode FTP. Fuente: [21]

4.6.1 Descripción de las pantallas de la aplicación web

La primera pantalla es la de ingreso, en la que, colocando las credenciales, pueden entrar al aplicativo sin ningún inconveniente. (Ver figura 4.11)

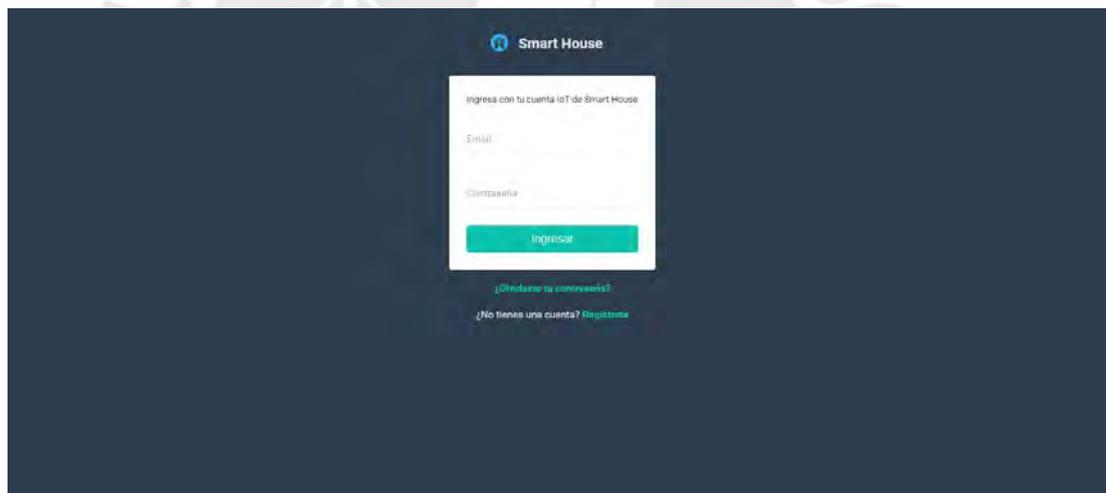


Figura 4.11: App web – pantalla de ingreso. Fuente: [21]

Si no tiene una cuenta, y desea crearse un, está el botón de “Regístrate” en color verde. Al darle clic, se redirige a la pantalla de registro. (Ver imagen 4.12)

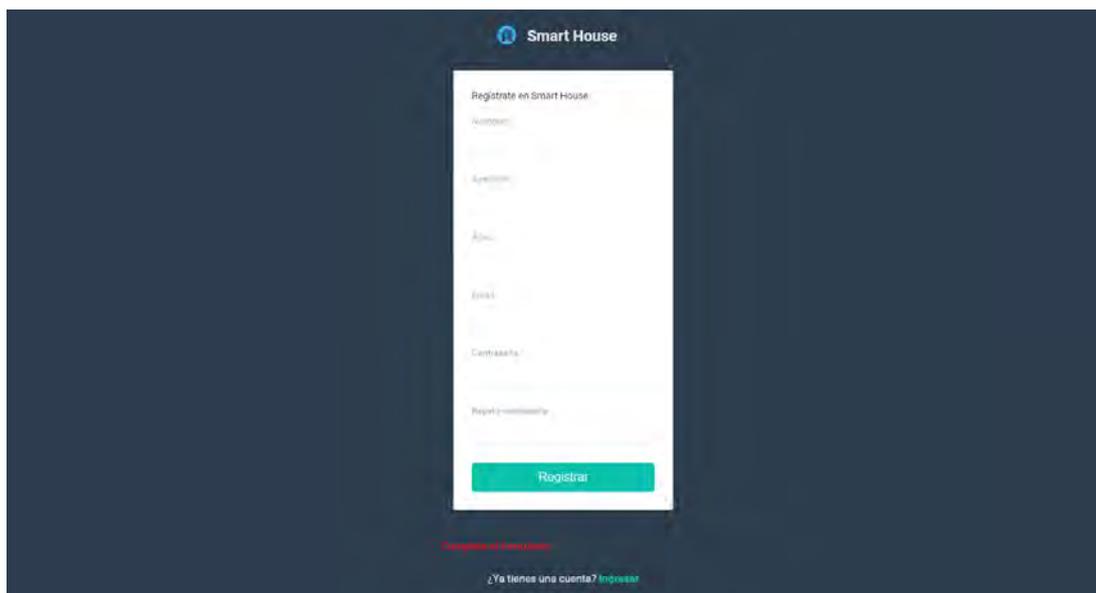


Figura 4.12: App web – pantalla de registro. Fuente: [21]

Una vez se haya ingresado, la primera pantalla es el dashboard de indicadores de los dispositivos. Aparecen los gráficos vacíos y también están los seleccionadores en la parte superior que permitirá filtrar el dispositivo que se quiere revisar. (Ver figura 4.13)

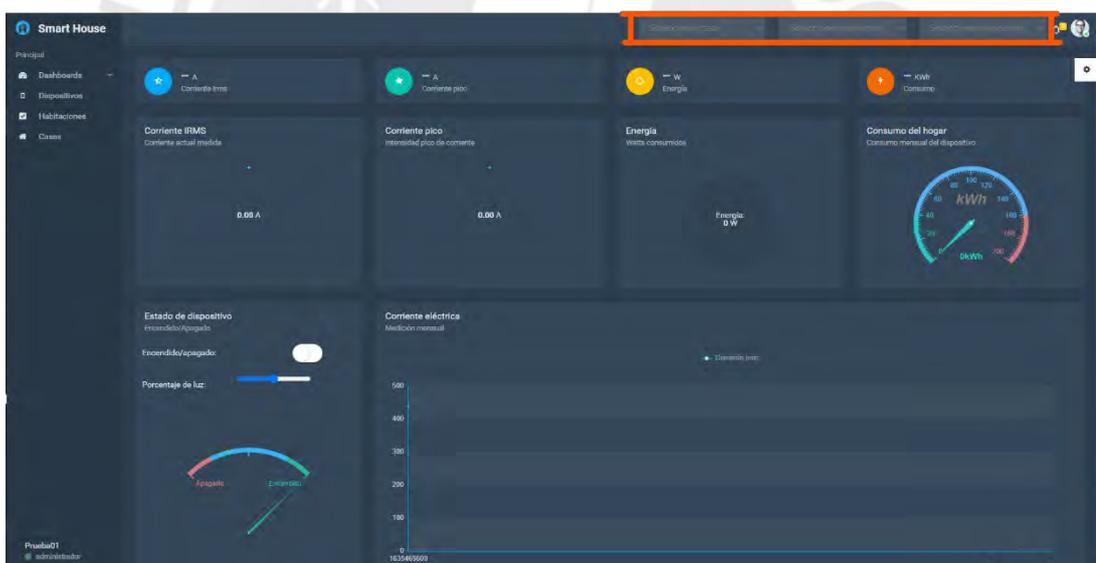


Figura 4.13: App web – pantalla de dashboard de dispositivos. Fuente: [21]

La pestaña para agregar o eliminar una casa te permite llenar un formulario para agregar una casa y también permite eliminar casas anteriores. (Ver figura 4.14)

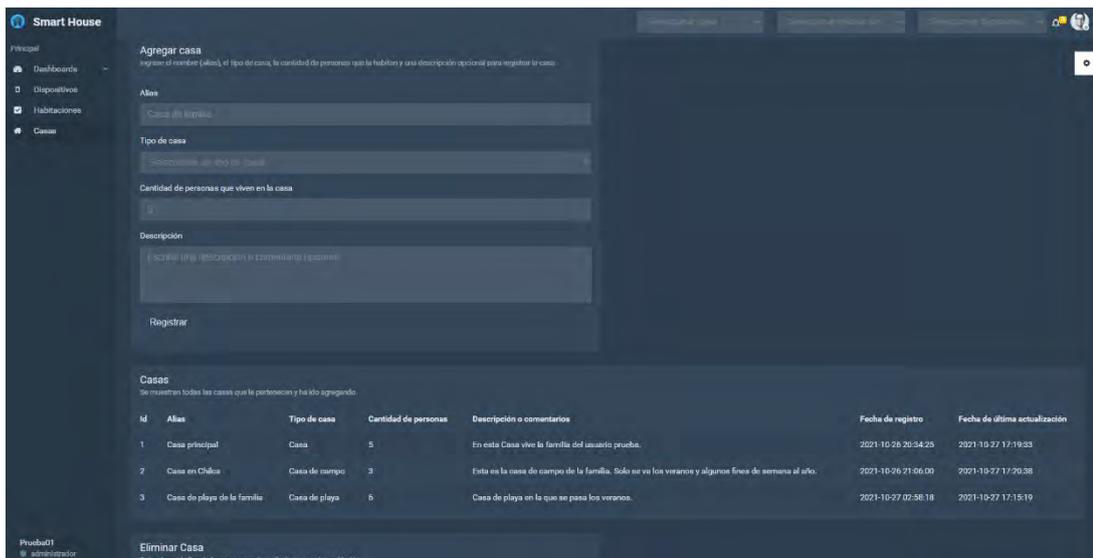


Figura 4.14: App web – pantalla de registro de casas. Fuente: [21]

La pestaña para agregar o eliminar una habitación permite que llenes un formulario para agregar una nueva habitación, pero debes seleccionar una de las casas registradas para enlazarla con esta. (Ver figura 4.15)

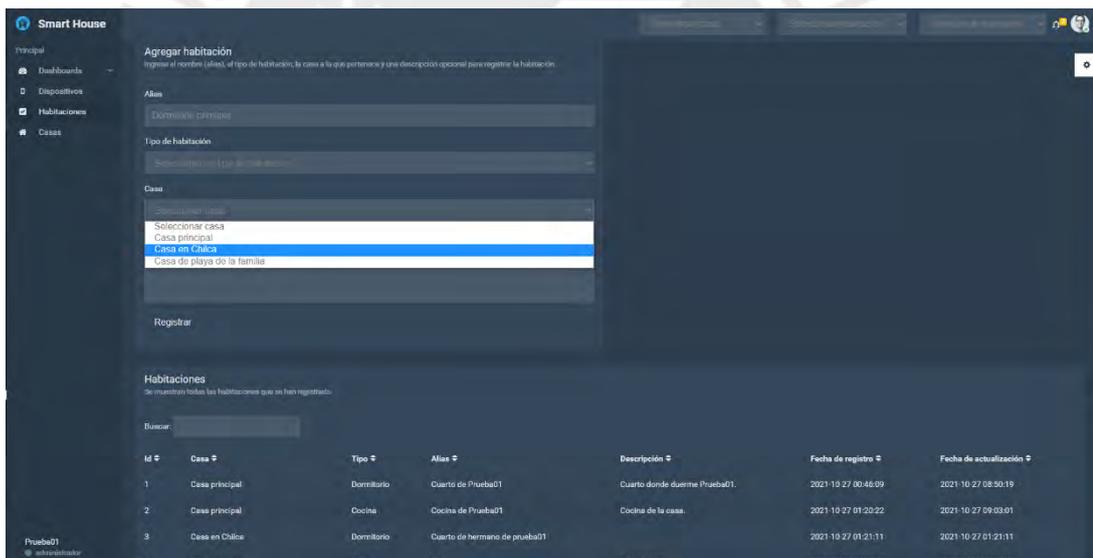


Figura 4.15: App web – pantalla de registro de habitaciones. Fuente: [21]

La pestaña para agregar o eliminar dispositivos, te permite agregar un dispositivo enlazándolo con una habitación y para esto también debes seleccionar una casa para que te aparezcan las opciones. También realiza la consulta para verificar si el número de serie del dispositivo que se está ingresando es el correcto. (Ver figura 4.16)

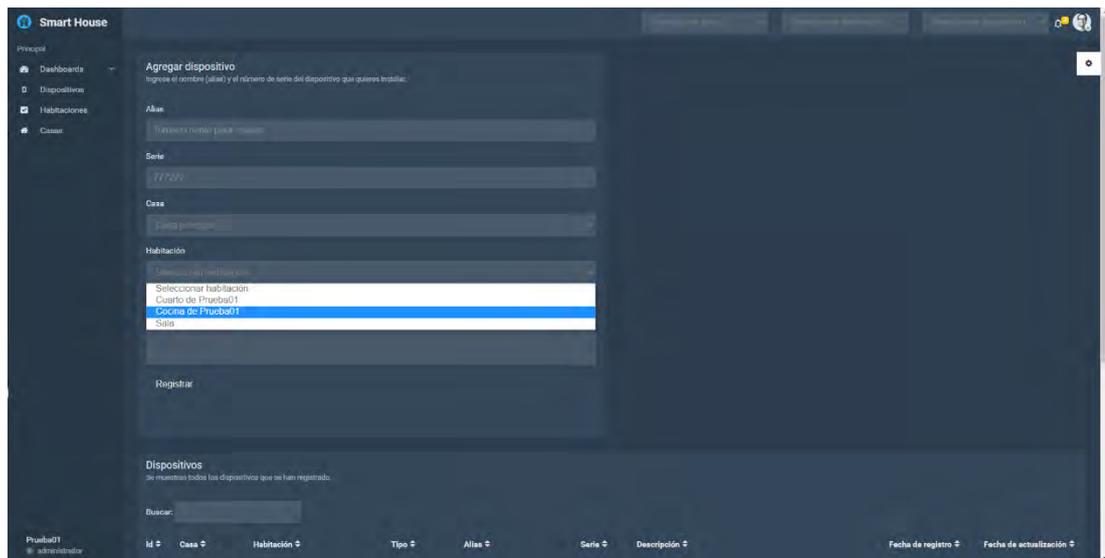


Figura 4.16: App web – pantalla de registro de dispositivos. Fuente: [21]

Las pantallas del dashboard de las casas y habitaciones, son básicamente lo mismo. (Ver figuras 4.17)



Figura 4.17: App web – pantalla de dashboard de las habitaciones y de las casas. Fuente: [21]

4.6.2 Lógica de programación de la aplicación web

La programación para que la aplicación funcione se está realizando con PHP como backend y como frontend se utilizan los lenguajes de programación más comunes que son HTML5, CSS3 y JavaScript. La conexión con el broker es una de las partes más importantes, porque si no funciona eso, no se mostraría ningún dato en ninguno de los dashboards. (Ver figura 4.18)

```

1247
1248 ***** CONEXIÓN *****
1249
1250 */
1251 var host = 'wss://smarthouse.ml[REDACTED]/mqtt'
1252 let client_id = "[REDACTED]";
1253
1254 client_id += Math.random().toString(36).slice(-8);
1255
1256 > var options = {...
1276 }
1277
1278 console.log('connecting mqtt client');
1279 var client = mqtt.connect(host, options);
1280
1281 > client.on('error', function(err) {...
1284 })
1285
1286 > client.on('connect', function() {...
1315 })
1316
1317 > client.on('reconnect', (error) => {...
1319 });
1320
1321 > client.on('message', function(topic, message, packet) {...
1325 })
1326
1327 // client.on('close', function() {
1328 //   console.log(clientId + ' disconnected')
1329 // })

```

Figura 4.18: App web – codificación de la conexión al broker en VSCode. Fuente: [21]

Por último, indicar que se creó el repositorio en GitHub para este proyecto y se realizaron varios commits para poder preservar el avance que se tenía. (Ver figura 4.19)

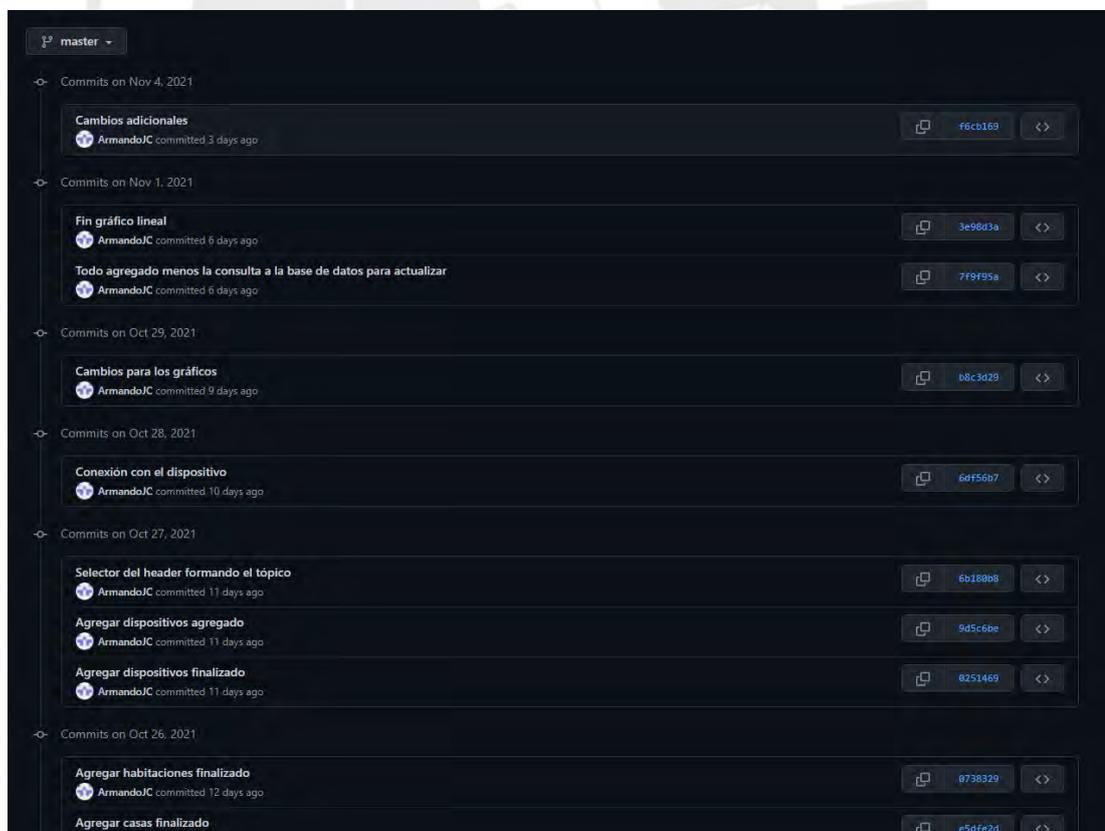


Figura 4.19: Repositorio en GitHub para la App web. Fuente: [21]

4.7 Programación de la aplicación móvil

La aplicación móvil ha llegado al mismo nivel que la aplicación web en este sprint, lo que hace que pueda soportar la gestión de usuarios, los ingresos de nuevas casas, habitaciones y dispositivos y el dashboard de dispositivos.

En este caso, no fue necesaria ninguna conexión FTP ni nada parecido, ya que existe algo llamado “Hot reload” en Flutter, que hace que los cambios ocurran casi instantáneamente, que es lo mismo que pasaba en la aplicación web. También se ha realizado commits para poder ir guardando toda la información en el repositorio privado que se creó en GitHub como medida de seguridad.

4.7.1 Descripción de las pantallas de la aplicación móvil

Las pantallas son básicamente las mismas que en la aplicación web, solo que la distribución de elementos cambia. Todo es mucho más compacto y pequeño. (Ver figuras 4.20, 4.21, 4.22 y 4.23)

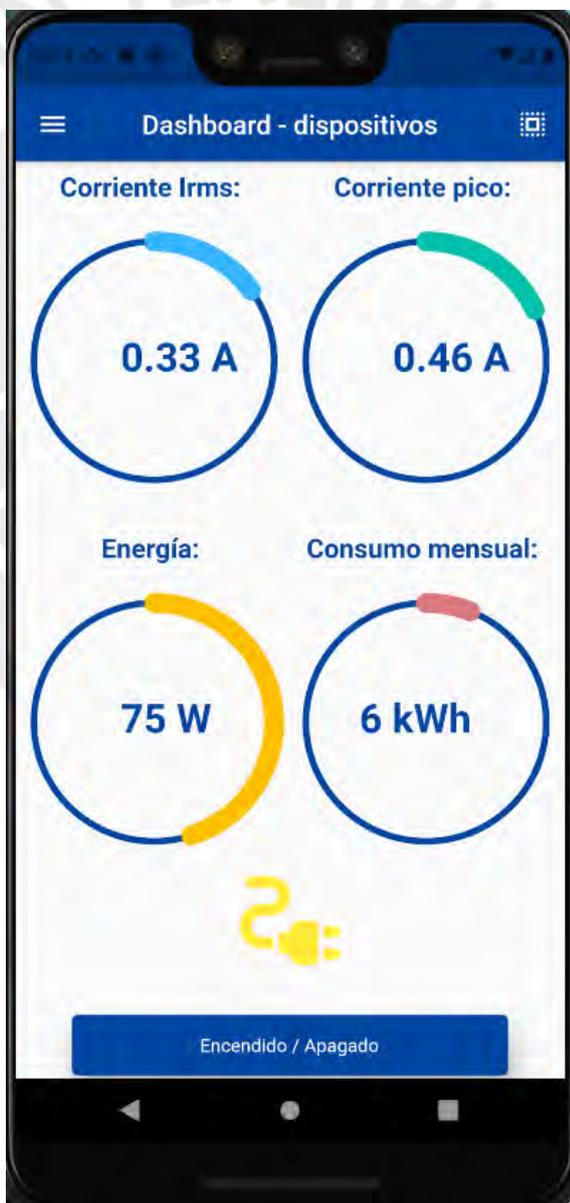


Figura 4.20: App móvil – dashboard dispositivos. Fuente: [21]

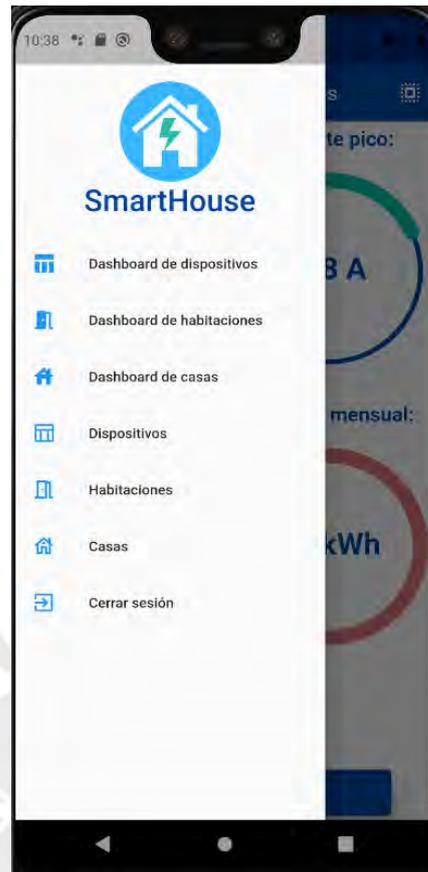


Figura 4.21: App móvil – dashboard menú lateral. Fuente: [21]

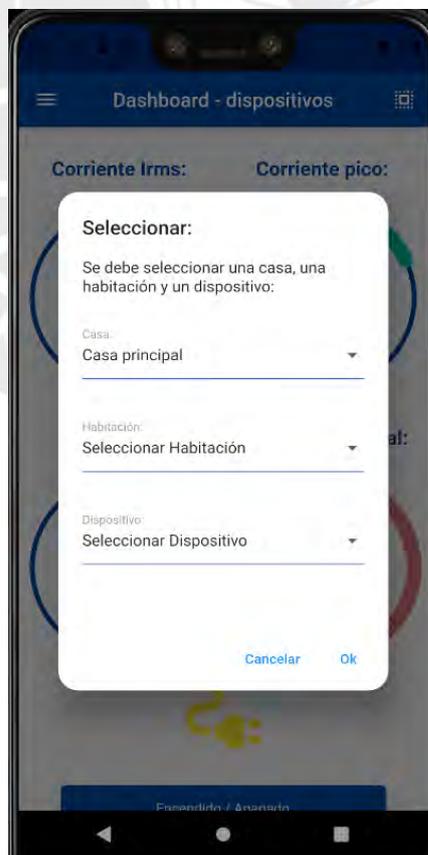


Figura 4.22: App móvil – Seleccionadores superiores. Fuente: [21]



Figura 4.23: App móvil – dispositivos. Fuente: [21]

4.7.2 Lógica de programación de la aplicación móvil

La lógica de programación para la aplicación móvil es la misma que para la aplicación web. Lo único en lo que se ha diferenciado es en las credenciales para la autenticación.

CAPÍTULO 5

5. PROTOTIPADO Y VALIDACIÓN DEL SISTEMA

En este capítulo, se presentarán los diseños de los prototipos del sistema, que serán utilizados en las pruebas de funcionamiento. Por último, se realizarán las últimas pruebas con los dos prototipos desarrollados.

5.1 Prototipos

Ahora se presentarán los prototipos que se desarrollaron para poder realizar las pruebas de funcionamiento y de estabilidad del sistema. Estos prototipos fueron realizados con la ayuda de la impresión 3D utilizando el filamento PLA, que es un filamento bastante utilizado y uno de los más famosos que se tiene ahora en el método de impresión 3D. Se utilizó el color azul en los prototipos por considerarlo un color llamativo. (Ver figura 5.1, 5.2 y 5.3)



Figura 5.1: Dispositivos de tomacorrientes e iluminación. Fuente: [21]



Figura 5.2: Dispositivo de toma de corriente. Fuente: [21]



Figura 5.3: Dispositivo de iluminación. Fuente: [21]

Para poder realizar una impresión 3D de manera correcta y duradera, se debe tener en cuenta varios parámetros de impresión, los mismos que se detallan a continuación. (Ver tabla 5.1)

Tabla 5-1: Parámetros de impresión 3D de los prototipos. Fuente: [21]

Componente	Filamento (mm)	Calidad (mm)	Tiempo (m)	Temperatura (°C)	Relleno (%)	Precio (S/.)
Tomacorriente – frontal	25273	0.2	315	215	10	35
Tomacorriente – posterior	26666	0.2	345	215	10	25
Luces - frontal	22792	0.2	149	210	10	40
Luces - posterior	12142	0.2	85	210	10	20

Ahora se mostrarán los prototipos individualmente, indicando las conexiones que se tiene y la distribución de los dispositivos electrónicos.

5.1.1 Prototipo del dispositivo de tomacorriente:

Este prototipo va a servir para probar si el monitoreo y el cambio de estado se pueden realizar sin problemas. Cuenta con todos los dispositivos electrónicos seleccionados en la sección 3.2, los cuales están indicados en el dibujo. La entrada de la corriente eléctrica se dará por un cable de conexión directa a un tomacorriente del hogar. (Ver figuras 5.4 y 5.5)

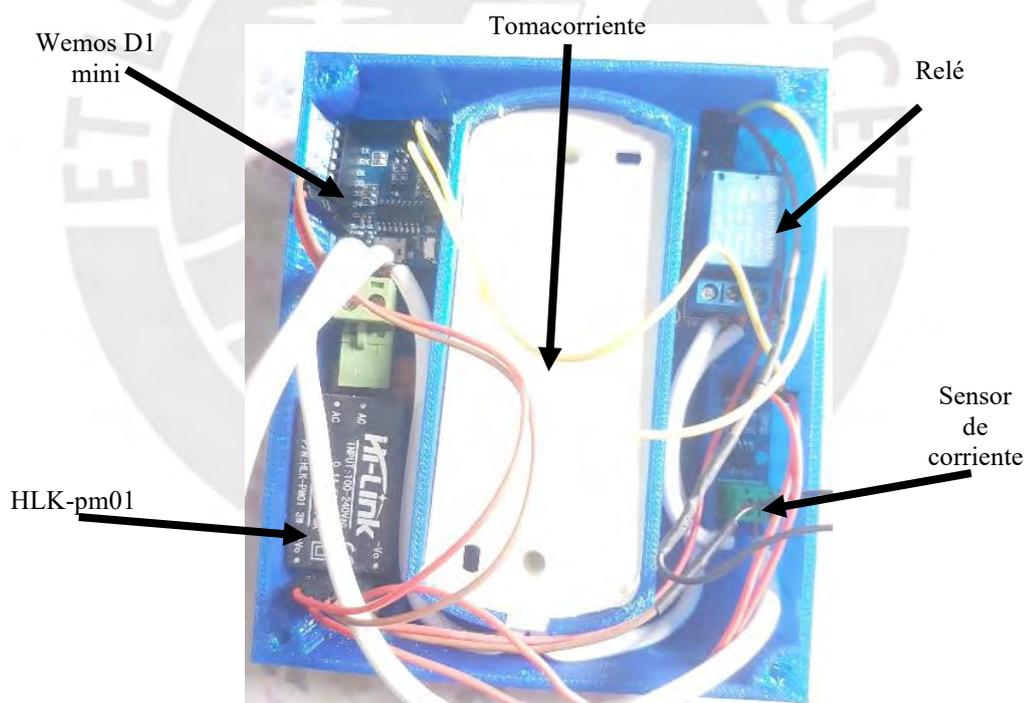


Figura 5.4: Dispositivo para tomacorrientes - vista trasera. Fuente: [21]

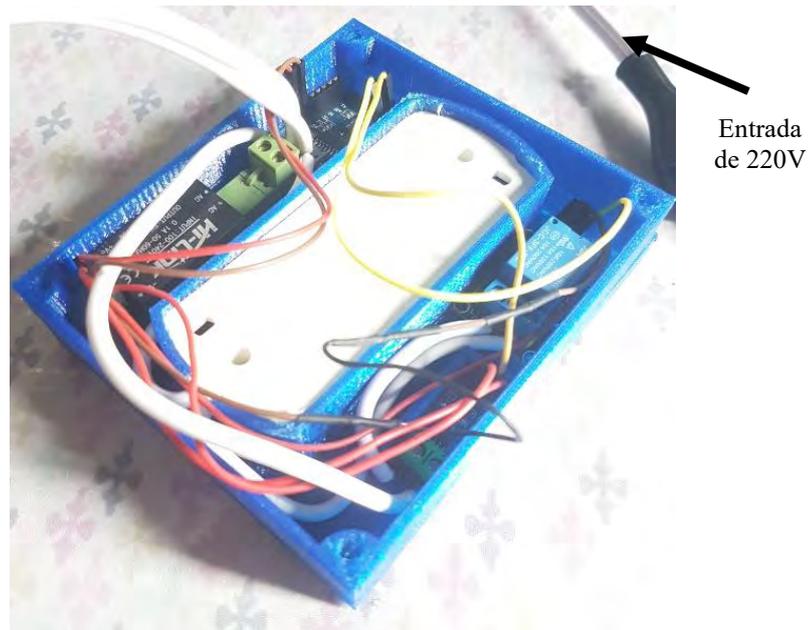


Figura 5.5: Dispositivo para tomacorrientes - vista isométrica. Fuente: [21]

5.1.2 Prototipo del dispositivo de iluminación:

Se presenta el interior del prototipo del dispositivo de iluminación mostrando todas sus conexiones. En este se puede observar el Wemos D1 mini, el sensor de corriente, el HLK-PM01, y el dimmer AC, que son los dispositivos que se seleccionaron previamente. (Ver figura 5.6)

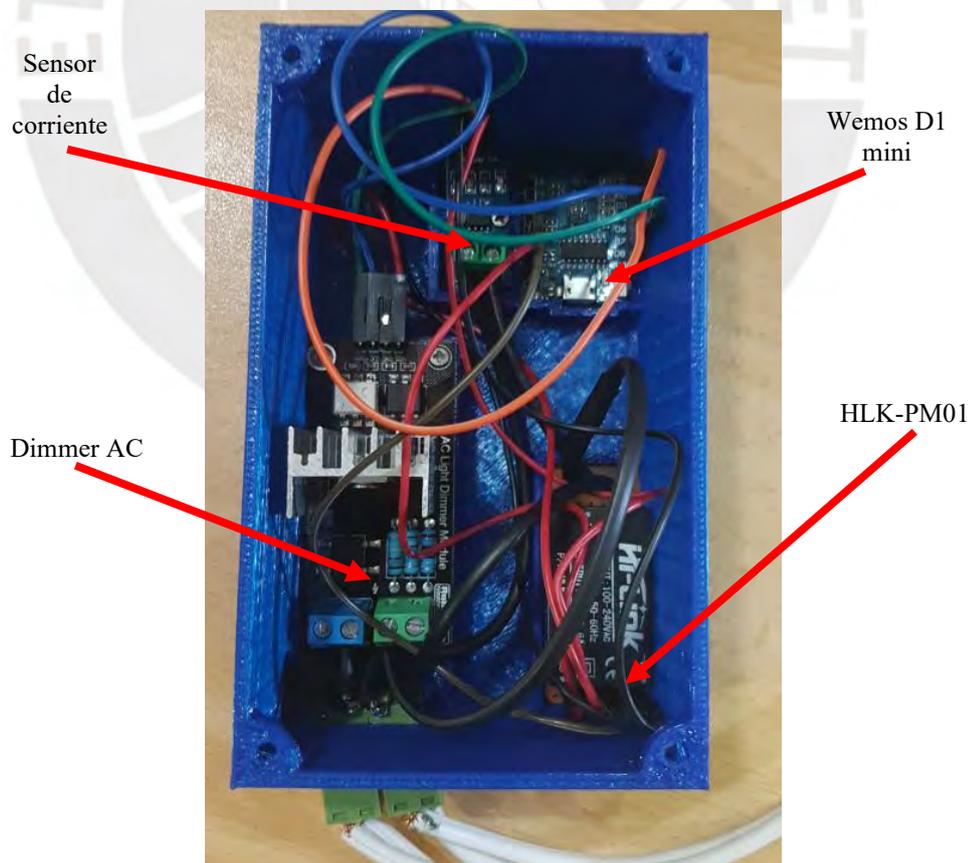


Figura 5.6: Interior del prototipo del dispositivo de iluminación. Fuente: [21]

Ahora se mostrará las dos partes de las que consta el diseño mecánico del prototipo, que son el contenedor de los dispositivos electrónicos, y la tapa con la que se cierra el dispositivo para poder ser atornillado, que además cuenta con dos columnas para mantener los dispositivos electrónicos fijos a sus posiciones. También se puede ver la entrada de la carga de 220V AC y la salida de la carga controlada de voltaje para variar la intensidad de la iluminación. (Ver figura 5.7)

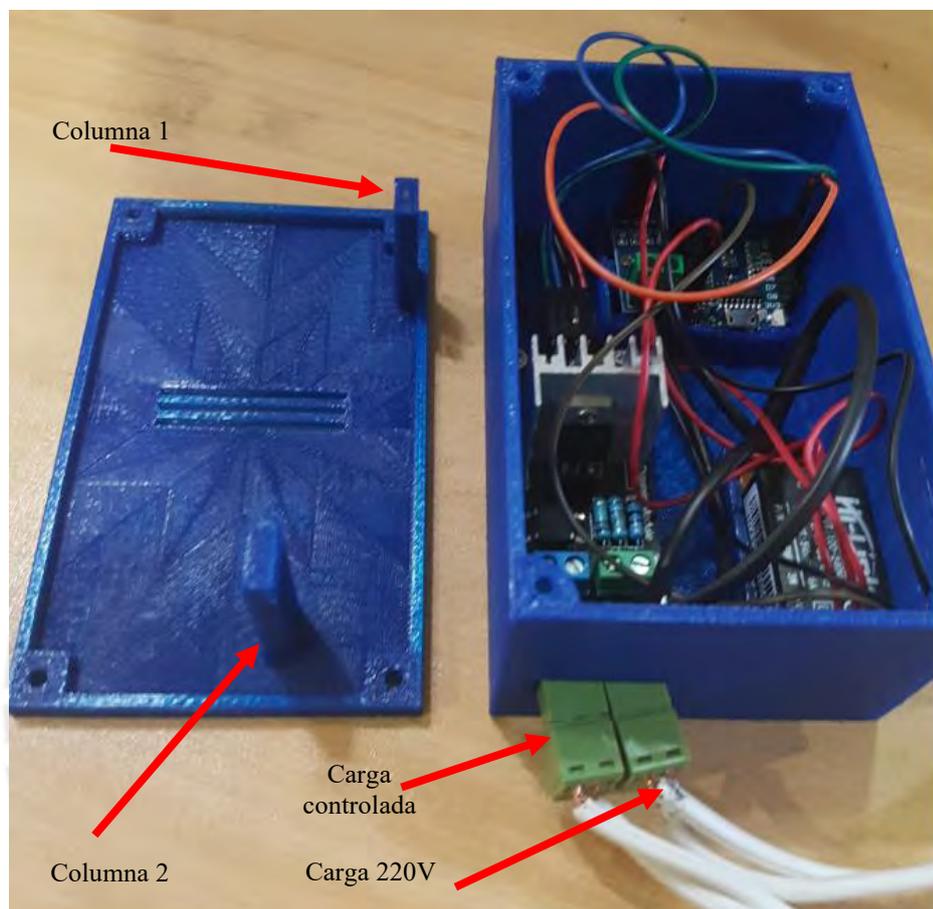


Figura 5.7: Tapa y contenedor del dispositivo de iluminación. Fuente: [21]

5.2 Pruebas con la aplicación web

Para las pruebas se agregaron los siguientes valores:

- Casa: Casa principal
- Habitación: Cuarto de prueba01
- Dispositivo: Celular y Laptop

Estos datos corresponden a un dispositivo de tipo tomacorriente. (Ver figuras 5.8, 5.9, y 5.10)

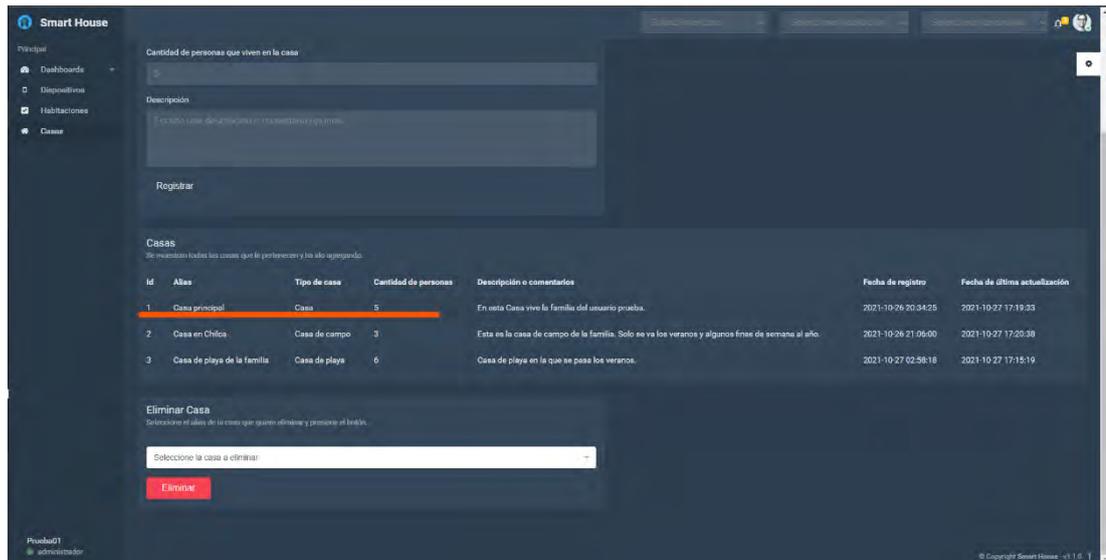


Figura 5.8: Casa principal. Fuente: [21]

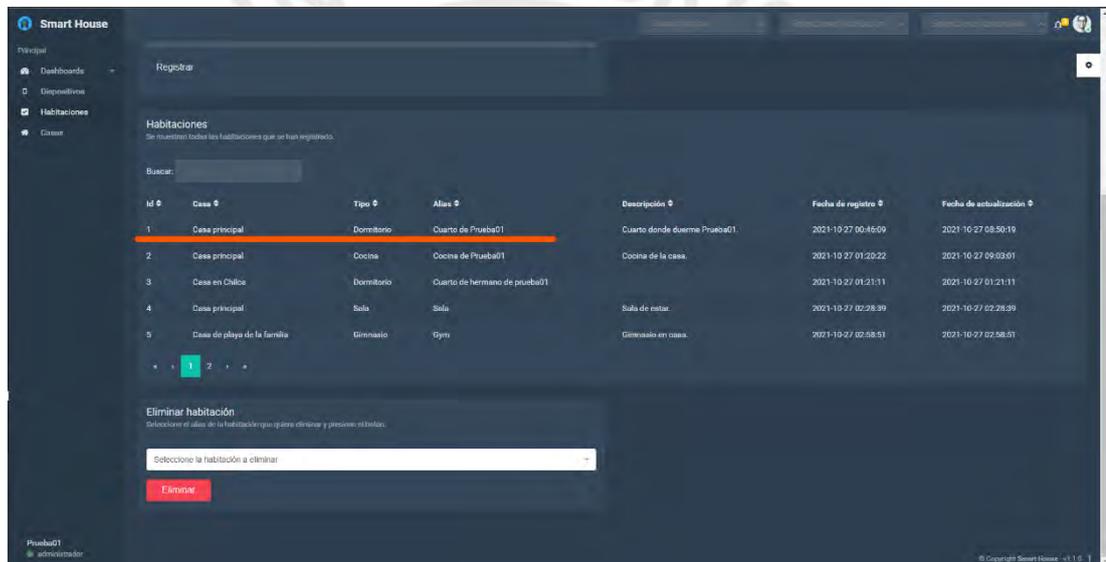


Figura 5.9: Cuarto de prueba01. Fuente: [21]

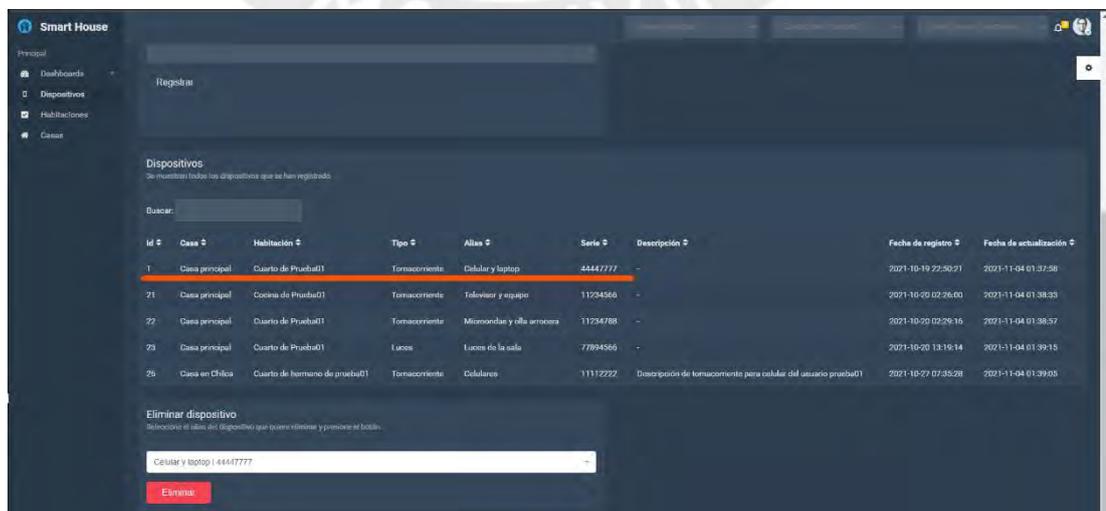


Figura 5.10: Celular y laptop. Fuente: [21]

Luego de esto, en el selector de la esquina superior derecha, se debe seleccionar estos valores. (Ver figura 5.11)

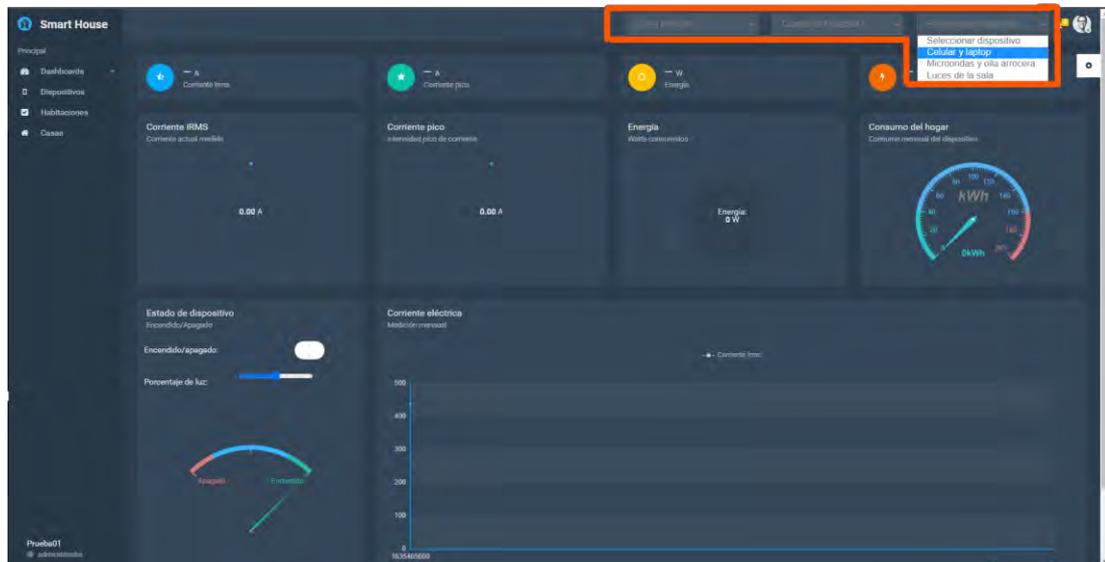


Figura 5.11: Seleccionando los valores necesarios para probar funcionamiento de aplicación web. Fuente: [21]

También se tiene como dispositivo de prueba al que tiene los siguientes valores:

- Casa: Casa principal
- Habitación: Cuarto de prueba01
- Dispositivo: Luces de la sala

Estos datos corresponden a un dispositivo de tipo Luces.

Esta funcionalidad de la página web funciona perfectamente. Ahora que se seleccionó los valores necesarios, es momento de probar los dispositivos para verificar que la información llega y se muestra.

5.3 Pruebas de dispositivos

Las pruebas que se realizarán son para ver si los dispositivos pueden trabajar correctamente por sí mismos y en conjunto.

5.3.1 Pruebas de dispositivo de tomacorriente

En esta sección se realizarán las pruebas considerando solo al dispositivo de tomacorriente. Esto se hace para ver cómo reacciona el dispositivo estando solo en la red. Se quiere observar si no tiene caídas de red, o si las reacciones son muy lentas. Para ello se usará la interfaz antes mencionada. Estas pruebas se realizaron en la casa del tesista por unas dos semanas en periodos de 5 a 24 horas.

Ahora se usa el tomacorriente con la aplicación web desarrollada y se observa las variaciones de corriente consistentes con la teoría. En este escenario, se está cargando una laptop y al desconectarla por medio del switch que tiene la página para cambiar el estado del dispositivo, las medidas de corriente bajan hasta la medida de 0.100 A aproximadamente, que es la corriente que gasta el sistema. (Ver figura 5.12)



Figura 5.12: Dispositivos desconectados. Fuente: [21]

Ahora, se pasará a encender el switch digital y se le hace zoom a la gráfica. Se puede observar que las medidas de la corriente son las esperadas. También se comprueba el funcionamiento del switch, ya que se puede encender y apagar desde la misma plataforma de manera instantánea. (Ver figura 5.13)

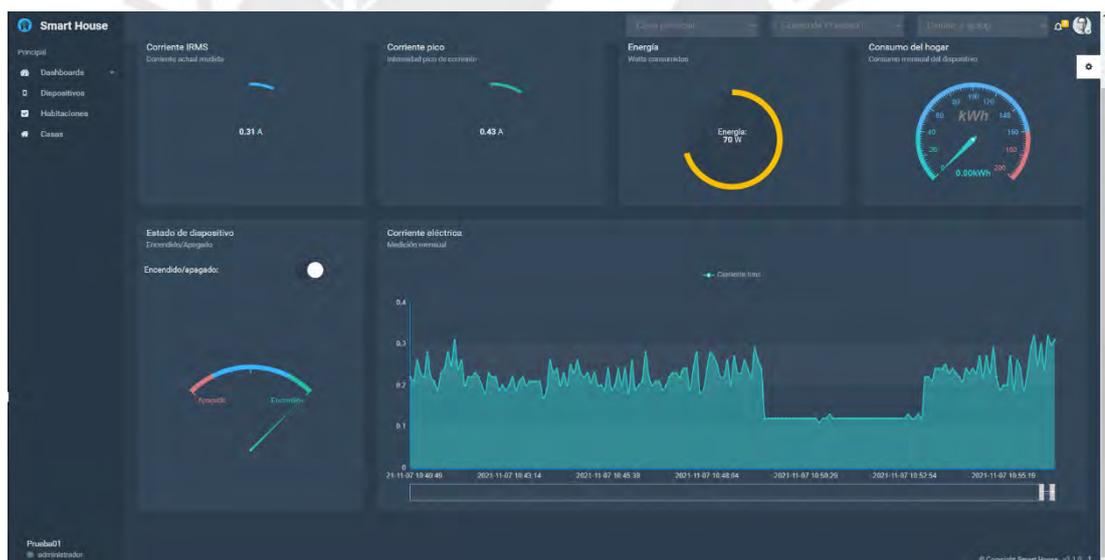


Figura 5.13: Laptop conectada. Fuente: [21]

Se pudo notar el confort por parte del tesista y su familia, al tener un dispositivo que permitió el control de encendido de varios aparatos a distancia.

5.3.2 Pruebas del dispositivo de luces

Ahora se probará el dispositivo de iluminación. En la gráfica se puede observar que cuando el dispositivo que se selecciona es de tipo “Luces”, aparecerá un controlador del porcentaje de luz del ambiente. En la gráfica de corriente, se puede observar la variación que se tuvo al encender el dispositivo y luego bajando la intensidad de la corriente. Se puede ver que la cresta más grande corresponde al 100 %. Luego de eso, se disminuyó la intensidad hasta el 40% y después se lo dejó en cero. Las gráficas muestran el comportamiento esperado. (Ver figura 5.14)



Figura 5.14: Dispositivo de Luces. Fuente: [21]

5.3.3 Pruebas de envío mensajes entre los clientes con el broker

Los mensajes enviados por el broker no han tenido ningún problema, se agregaban a la base de datos como se esperaba, la aplicación web los recibía y también el servicio node.js. Además, el dispositivo de toma de corriente y de luces, también recibía la información, todo con una latencia casi imperceptible. Cabe resaltar que se hizo una prueba con 7 cuentas conectadas a la vez a la misma información y no hubo caída de velocidad en la recepción de la información. (Ver figuras 5.16 y 5.17)

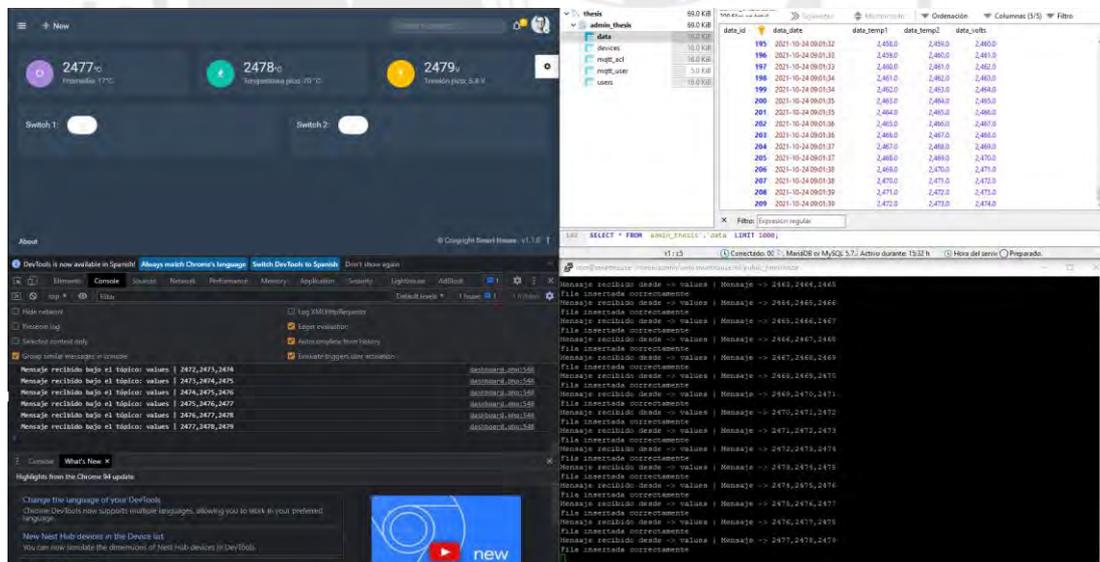


Figura 5.15: Prueba de envío de mensajes del broker. Fuente: [21]

```

40 float power = 3;
41 int onOff = 0;

```

```

Mensaje recibido desde -> SmartHouse/1/1/Tomacorriente/44447777/values
3.00,4.00,5.00,6.00,0,10,1
Mensaje -> 3.00,4.00,5.00,6.00,0,10,1
Pasaron 5s
*****
Publicamos mensaje: 4.00,5.00,6.00,7.00,0,10,1
*****
Mensaje recibido desde -> SmartHouse/1/1/Tomacorriente/44447777/values
4.00,5.00,6.00,7.00,0,10,1
Mensaje -> 4.00,5.00,6.00,7.00,0,10,1

```

Figura 5.16: Prueba de envío de mensajes del broker 2. Fuente: [21]

Como conclusión se pudo observar que el sistema se mantiene estable en el tiempo y la conexión depende del internet que se tenga en el hogar. Los datos se mandan y reciben sin error alguno, siendo las gráficas las esperadas. El broker y todos los sistemas demuestran ser robustos y estar preparados para poder trabajar con muchos usuarios.



CAPÍTULO 6

6. PRESUPUESTOS

Los gastos que se necesitan para poder desarrollar los 3 dispositivos a nivel de prototipo son los siguientes:

6.1 Presupuesto del prototipo del dispositivo de iluminación:

En el dispositivo de iluminación, se tienen los precios del sensor de corriente invasivo, del dimmer AC, del acondicionador de energía, del microcontrolador, de la carcasa impresa en 3D y de los dispositivos electrónicos que se usarán. (Ver tabla 6-1)

Tabla 6-1: Costos del prototipo del dispositivo de iluminación. Fuente: [21]

	Precio (S/.)
Sensor de corriente invasivo (ACS712)	14
HLK-PM01	20
Wemos D1 mini	30
Dimmer AC	28
Carcasa impresión 3D	60
Dispositivos electrónicos varios (cables, pasta de soldar, etc.)	5
TOTAL	157

6.2 Presupuesto del prototipo del dispositivo de tomacorriente:

En el dispositivo de tomacorriente, se tienen los precios del sensor de corriente invasivo, del relé, del acondicionador de energía, del microcontrolador, de la carcasa impresa en 3D y de los dispositivos electrónicos que se usarán. (Ver tabla 6-2)

Tabla 6-2: Costos del prototipo del dispositivo de tomacorriente. Fuente: [21]

	Precio (S/.)
Sensor de corriente invasivo (ACS712)	14
HLK-PM01	20
Wemos D1 mini	30
Relé	8
Carcasa impresión 3D	60
Dispositivos electrónicos varios (cables, pasta de soldar, etc.)	5
TOTAL	137

6.3 Presupuesto del diseño de ingeniería:

El costo del diseño es calculado a partir de las horas de trabajo en el mismo, el cual ha sido 14 semanas de trabajo, a un promedio de 10 horas de trabajo por semana, lo cual harían 140 horas trabajadas en el diseño. A esto se le debe multiplicar el sueldo por hora de un ingeniero promedio, que es 20 soles por hora. Con estos datos se concluye que el costo del diseño serían aproximadamente S/. 2800.

El costo total sería $2800 + 157 + 137 = \text{S/}. 3094$

CONCLUSIONES

Las conclusiones se detallan a continuación:

- El sistema es funcional para la casa de las familias limeñas, ya que se realizaron pruebas de uso doméstico por alrededor de 1 semana y la utilidad y comodidad fueron evidentes.
- Se comprobó que el sistema es bastante estable, porque no hubieron caídas de red, ni pérdida de datos a través del tiempo en la plataforma web.
- La conexión con la página web no mostro dificultades, salvo en el dispositivo de iluminación, ya que había un pequeño retardo a la hora de ejercer el control.
- Se cumplió con la lista de requerimientos presentada en el anexo A, siendo las más resaltantes la función principal de regular el cambio de estado de los dispositivos del sistema y monitorizar la energía eléctrica consumida en el hogar; la geometría, ya que no se excedió las medidas planteadas; la energía, ya que se utiliza la energía de la red eléctrica; y los costos, ya que no se excedió el costo máximo de 200 soles por dispositivo.
- El costo total de los 2 dispositivos a nivel prototipo es de S/. 294.
- Se logró desarrollar todo como un sistema ciberfísico interconectado que es escalable en el tiempo.
- Se implementó los dispositivos usando impresión 3D
- Se cumplió el objetivo de realizar una aplicación web alojada en *Amazon Web Services* para su uso con los prototipos. Esta aplicación es robusta y la arquitectura está preparada para trabajar con múltiples dispositivos al mismo tiempo. Para esto, se realizó una prueba con 7 dispositivos de diferentes partes de Lima conectados a la misma cuenta y no hubo ningún problema ni disminución de velocidad.
- Este proyecto sí se puede usar a futuro en un emprendimiento, ya que está preparado y pensado para eso.

RECOMENDACIONES

Las recomendaciones se detallan a continuación:

- Para proyectos en los que se requiera del uso de la red Wifi, es recomendable que siempre se realicen con microcontroladores que incluyan ya un módulo Wifi integrado, porque esto facilita en gran medida la codificación y las conexiones de los dispositivos electrónicos, además que reduce el espacio que utilizan.
- Utilizar microcontroladores conocidos y que se tenga bibliografía sobre su utilización, ya que un microcontrolador poco conocido implica un problema a la hora de buscar información y esto retrasa el proceso de diseño.
- Realizar prototipos para validar los diseños es de gran ayuda, ya que indican si vamos por buen camino.
- Tener muy en cuenta las medidas de seguridad cuando se trabaje con altos voltajes, ya que, de no hacerlo, se pone en riesgo la vida.
- Verificar maneras de simplificar el diseño en los siguientes releases utilizando servicios de la plataforma para reducir el trabajo, ya que hacer un sistema completo desde cero, requiere mucho conocimiento técnico.
- Se podría usar de referencia arquitecturas proporcionadas por *Amazon Web Services*.
- Para la base de datos relacional, se podría usar el servicio AWS RDS.
- Para la autenticación y seguridad, se podría usar el servicio Amazon Cognito.
- Para realizar microservicios, como el que realiza node.js para el ingreso de data a la base de datos, se podría usar el servicio de funciones de AWS Lambda.
- Para redirigir las peticiones, se podría usar el servicio AWS API Gateway.
- Hacer pruebas periódicas y someter el sistema a una cantidad grande de usuarios y dispositivos.
- Crear otros dispositivos diferentes como un sensor de presencia que encienda y apague las luces, un sensor de temperatura que encienda los ventiladores o aire acondicionado, etc.
- Al realizar un proyecto que abarca muchos campos, pensar en realizarlo en un grupo multidisciplinario.
- Se puede hacer muchas mejoras a las aplicaciones, como incluirle creación de alertas, informes, un apartado para recomendaciones de uso implementado con machine learning, etc.
- Las aplicaciones, de cara al usuario, podrían permitir que se puedan crear dashboards personalizados.
- En los apartados de dashboards para casas y habitaciones, se debe implementar para que los datos y comandos sean para todos los dispositivos de un mismo tipo en la casa o habitación.
- Se recomienda cambiar el lenguaje backend de PHP a Node.js en su totalidad, ya que eso puede permitir que se facilite la actualización en tiempo real porque PHP no está diseñado específicamente para eso.

BIBLIOGRAFÍA

- [1] Asociación de desarrolladores inmobiliarios (ADIPERÚ), “Informe estadístico del mercado inmobiliario”, Perú, 2020. Obtenido de <http://adiperu.pe/wp-content/uploads/Bolet%C3%ADn-Estad%C3%ADstico-Mensual-VF.pdf>
- [2] Ministerio de Energía y Minas, “Estadísticas eléctricas anuales – Evolución de indicadores del mercado eléctrico 1995 - 2016”, Perú, 2016. Obtenido de <http://www.minem.gob.pe/descripcion.php?idSector=6&idTitular=638&idMenu=sub115&idCateg=350>
- [3] Diario El Comercio, “¿Cómo ayuda la domótica a ahorrar energía en el hogar?”, Perú, 2017. Obtenido de <https://elcomercio.pe/tecnologia/actualidad/ayuda-domotica-ahorrar-energia-hogar-421104>
- [4] N. E. Huiman Tocto, “Diseño e implementación de una red de medidores de energía para artefactos domésticos”, tesis de licenciatura, Ingeniería Electrónica, PUCP, Lima, Perú, 2017.
- [5] L. F. Poma Aliaga, “Diseño de un sistema inteligente de ahorro de energía eléctrica”, tesis de licenciatura, Ingeniería Electrónica, PUCP, Lima, Perú, 2017.
- [6] M. R. Guzmán Guerra y R. A. Burga Velarde, “Sistema domótico de control centralizado con comunicación por línea de poder”, tesis de licenciatura, Ingeniería electrónica, PUCP, Lima, Perú, 2014.
- [7] J. C. Culquichicón Valentín, “Domolab: sistema de monitoreo y control remoto de viviendas”, Tesis de maestría, Ingeniería telecomunicaciones, PUCP, Lima, Perú, 2015.
- [8] Smart House Perú. Obtenido de <https://www.falabella.com.pe/falabella-pe/product/17758020/Toma-corriente-smart-X2/17758020>
- [9] Domotik. Obtenido de <https://www.domotik.com.pe/producto/relay-switch-simple/>
- [10] Activa building control. Obtenido de <https://www.e-activa.com/domotical>
- [11] J.L. del Val Román, “Industria 4.0: la transformación digital de la industria”, informe de la conferencia de directores y decanos de ingeniería informática, Ingeniería Informática, Universidad de Deusto, Bilbao, España, 2016.
- [12] S. Khan y J. Lloret Mauri, “Green Networking and Communications: ICT for Sustainability”, 2013.
- [13] G. Reiter, “Wireless connectivity for the Internet of Things – White Paper”, Dallas: Texas Instruments, 2014

- [14] Efor internet y tecnología, “Tecnologías de comunicación para IoT”, España, 2017. Obtenido de <https://www.efor.es/sites/default/files/tecnologias-de-comunicacion-para-iot.pdf>
- [15] L. Llamas, “Protocolos de comunicación para IoT”, España, 2019. Obtenido de <https://www.luisllamas.es/protocolos-de-comunicacion-para-iot/>
- [16] J.J. Manotas Campos y N. Martínez Marín, “Exploración de las plataformas IoT en el mercado para fomentar el conocimiento, buen uso y efectividad de los dispositivos IoT creados en la Facultad de Ingeniería y Ciencias Básicas de la Institución Universitaria Politécnico Grancolombiano.”, tesis de licenciatura, Ingeniería de Telecomunicaciones e Ingeniería de Sistemas, Institución Universitaria Politécnico Grancolombiano, Bogotá, Colombia, 2018.
- [17] T. Rodríguez, “La otra guerra entre Microsoft, Google y Amazon: la batalla por controlar los servicios en la nube para desarrolladores”, México, 2019. Obtenido de <https://www.xataka.com/servicios/otra-guerra-microsoft-google-amazon-batalla-controlar-servicios-nube-para-desarrolladores>
- [18] Y. A. Rendón, “Bases de datos relacionales vs. no relacionales”, Colombia, 2019. Obtenido de <https://www.pragma.com.co/academia/lecciones/bases-de-datos-relacionales-vs.-no-relacionales>
- [19] L. Llamas, “Medir Intensidad y consume eléctrico con Arduino y ACS712”, España, 2017. Obtenido de <https://www.luisllamas.es/arduino-intensidad-consumo-electrico-ac712/>
- [20] Naylamp Mechatronics, “Tutorial sensor de corriente AC no invasivo SCT-013”, Perú, 2016. Obtenido de https://naylampmechatronics.com/blog/51_tutorial-sensor-de-corriente-ac-no-invasivo-s.html
- [21] Elaboración propia.
- [22] 123RF. Obtenido de https://fr.123rf.com/photo_83554614_int%C3%A9rieur-3d-de-la-maison-moderne-avec-cuisine-salon-salle-de-bain-et-chambre-%C3%A0-coucher-jeu-d-illustr.html
- [23] Electronics Universal, “Sensores”, México, 2018. Obtenido de <https://electronicsuniversal.com/categoria-producto/sensores/otros-sensores/>
- [24] Platzi, “React vs. Angular vs. Vue vs. Svelte | ¿Cómo elegir tu próxima herramienta frontend? ¿Cuál es mejor?”, 2021. Obtenido de https://platzi.com/blog/react-angular-vue-svelte/?gclid=Cj0KCQiAkZKNBhDiARIsAPsk0Wh2k9ay9MoVfszXerFsToKc7iRLqFDolVA20zC6nrOte9idYRfEbD8aAiVSEALw_wcB&gclsrc=aw.ds

- [25] EMQ, “EMQ X 3.1.1 Ten Million Connections Performance Test Report”, 2021. Obtenido de <https://www.emqx.com/en/resources/emqx-ten-million-connections-performance-test-report>
- [26] IBM, “Propiedades ACID de las transacciones”, 2017. Obtenido de <https://www.ibm.com/docs/en/cics-ts/5.4?topic=processing-acid-properties-transactions>



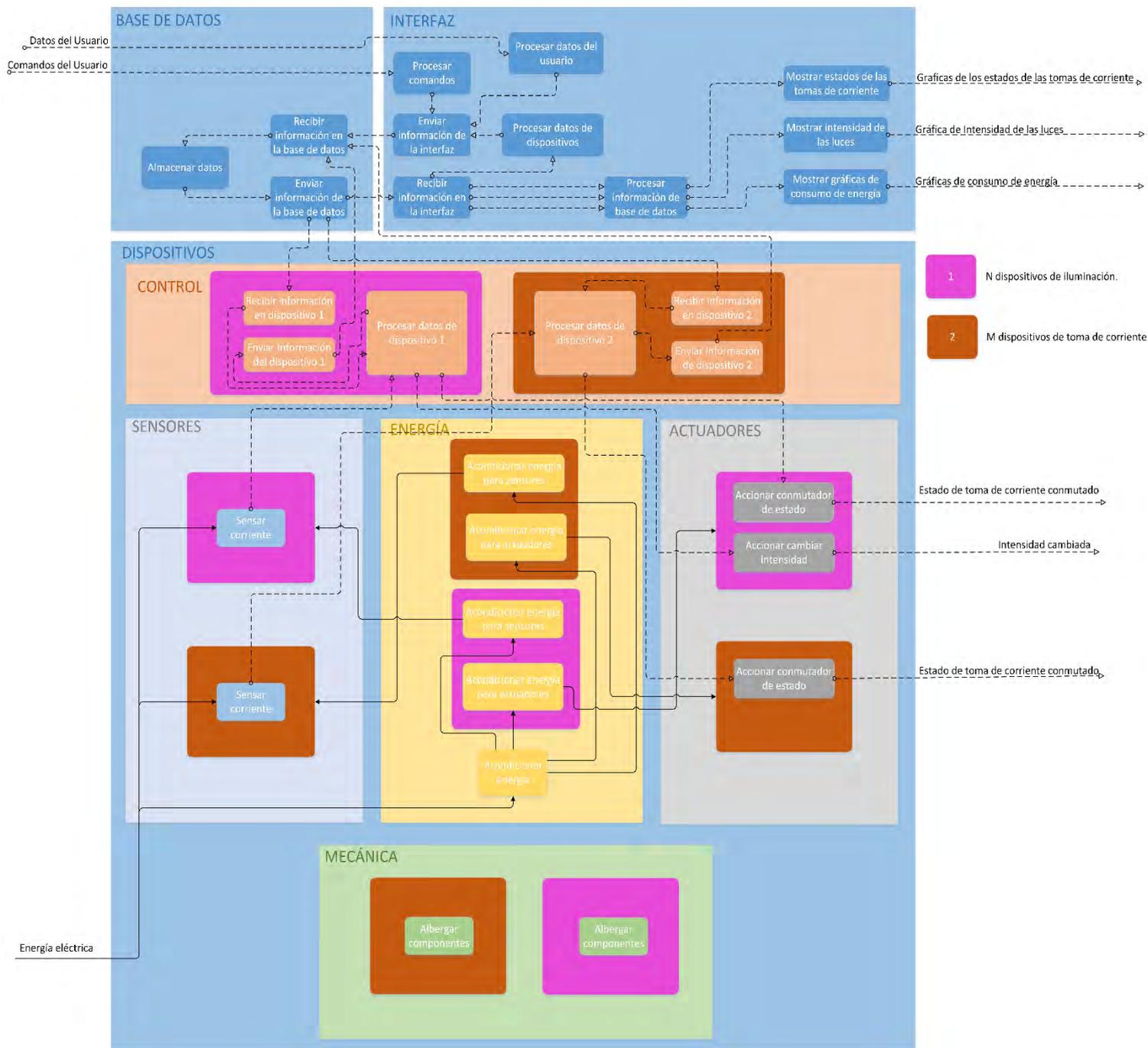
ANEXOS

Anexo A: Lista de requerimientos

LISTA DE REQUERIMIENTOS			Pág. 1 de 2
			Edición: 1
PROYECTO:		SMART HOUSE 1	Fecha: 07/09/18
			Revisado:
CLIENTE:		PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ	Elaborado: Armando J.
Fecha (cambios)	Deseo o Exigencia	Descripción	Responsable
FUNCIÓN PRINCIPAL:			
07/09/18	E	<ul style="list-style-type: none"> Regular el cambio de estado de los dispositivos del sistema y monitorizar la energía eléctrica consumida en el hogar 	Armando J.
GEOMETRÍA:			
07/09/18	E	<ul style="list-style-type: none"> Las dimensiones no deben exceder los 15 cm de largo. Las dimensiones no deben exceder los 12 cm de ancho. Las dimensiones no deben exceder los 8 cm de espesor. 	Armando J.
SOFTWARE:			
07/09/18	E	<ul style="list-style-type: none"> El software a utilizar será software libre 	Armando J.
07/09/18	E	<ul style="list-style-type: none"> Se utilizará una página web para el control vía computadora. 	Armando J.
COMUNICACIONES:			
07/09/18	E	<ul style="list-style-type: none"> La comunicación será inalámbrica y deberá tener un rango de comunicación efectivo no menor a 100 m 	Armando J.
ENERGÍA:			
07/09/18	E	<ul style="list-style-type: none"> Será alimentado con la energía que está en la misma toma de corriente. 	Armando J.
COSTOS:			
07/09/18	E	<ul style="list-style-type: none"> No excederá los 200 soles por dispositivo. 	Armando J.

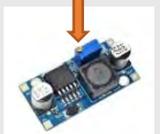
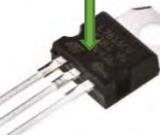
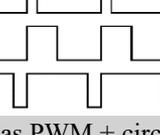
LISTA DE REQUERIMIENTOS			Pág. 2 de 2
			Edición: 1
PROYECTO:		SMART HOUSE 1	Fecha: 07/09/18
			Revisado:
CLIENTE:		PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ	Elaborado: Armando J.
Fecha (cambios)	Deseo o Exigencia	Descripción	Responsable
		SEÑALES:	
07/09/18	E	<ul style="list-style-type: none"> Se tendrá señales de salida de datos de consumo de corriente en tiempo real. 	Armando J.
07/09/18	E	<ul style="list-style-type: none"> Se comunicará vía internet indicando dispositivos conectados 	Armando J.
07/09/18	E	<ul style="list-style-type: none"> Recibirá señales de entrada de encendido y apagado de dispositivos. 	Armando J.
		CONTROL:	
07/09/18	E	<ul style="list-style-type: none"> Se realizará de manera remota. 	Armando J.
		ELECTRÓNICA:	
07/09/18	E	<ul style="list-style-type: none"> Se usarán sensores de corriente. Se utilizarán actuadores para conmutar el paso de corriente o atenuarlo. 	Armando J.
07/09/18	E	<ul style="list-style-type: none"> Se acondicionará la energía eléctrica dentro del dispositivo. 	Armando J.
		SEGURIDAD:	
07/09/18	E	<ul style="list-style-type: none"> La manipulación del dispositivo por parte del usuario no pondrá en riesgo su integridad. 	Armando J.
07/09/18	E	<ul style="list-style-type: none"> Se basará en el código nacional de electricidad del Perú del ministerio de energía y minas. 	Armando J.
		FABRICACIÓN:	
07/09/18	E	<ul style="list-style-type: none"> Los materiales y elementos a fabricar se conseguirán en el mercado local. 	Armando J.
		MANTENIMIENTO:	
07/09/18	E	<ul style="list-style-type: none"> No se realizará mantenimiento. 	Armando J.

Anexo B: Estructura de funciones

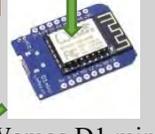


Anexo C: Matriz morfológica

Dispositivo de iluminación:

Funciones	Opción 1	Opción 2	Opción 3
Albergar componentes	 Carcasa de metal	 Carcasa de plástico	 Carcasa de madera
Acondicionar energía para sensores	 Regulador de tensión lineal	 Regulador conmutador	 Regulador de tensión lineal integrado
Acondicionar energía para actuadores	 Regulador de tensión lineal	 Regulador conmutador	 Regulador de tensión lineal integrado
Sensar corriente	 Sensor de corriente invasivo	 Sensor de corriente no invasivo	 Amperímetro - Voltímetro
Procesamiento de datos de dispositivo 2	 Arduino Nano	 Wemos D1 mini	 Arduino UNO
Enviar información de dispositivo 2	 Wifi	 ZigBee	 X10
Recibir información en dispositivo 2	 Wifi	 ZigBee	 X10
Accionar conmutador de estado	 Relé de estado sólido	 Relé electromecánico	-----
Accionar cambiar intensidad	 Dimmer AC	 Ondas PWM + circuito	-----

Dispositivo de toma de corriente:

Funciones	Opción 1	Opción 2	Opción 3
Albergar componentes	 Carcasa de metal	 Carcasa de plástico	 Carcasa de madera
Acondicionar energía para sensores	 Regulador de tensión lineal	 Regulador conmutador	 Regulador de tensión lineal integrado
Acondicionar energía para actuadores	 Regulador de tensión lineal	 Regulador conmutador	 Regulador de tensión lineal integrado
Sensar corriente	 Sensor de corriente invasivo	 Sensor de corriente no invasivo	 Amperímetro - Voltímetro
Procesamiento de datos de dispositivo 3	 Arduino Nano	 Wemos D1 mini	 Arduino UNO
Enviar información de dispositivo 3	 Wifi	 ZigBee	 X10
Recibir información en dispositivo 3	 Wifi	 ZigBee	 X10
Accionar conmutador de estado	 Relé de estado sólido	 Relé electromecánico	-----

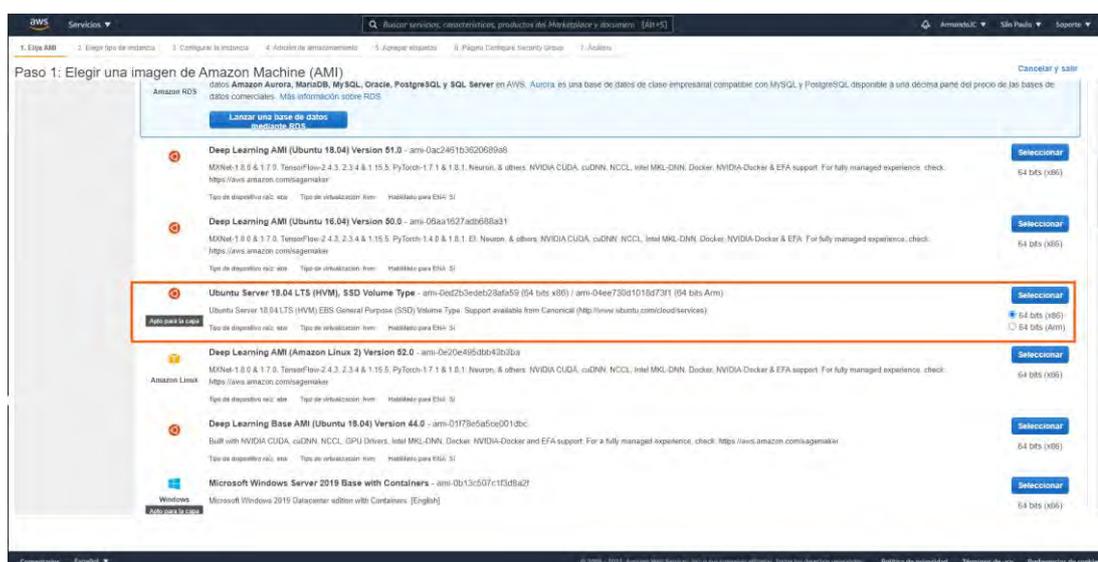
Base de datos e interfaz:

Funciones	Opción 1	Opción 2	Opción 3
Almacenar datos	 Servidor web AWS	 Servidor web propio	 Google Cloud
Recibir información en la base de datos	 Wifi	 Redes Alámbricas	----
Enviar información de la base de datos	 Wifi	 Redes Alámbricas	----
Recibir información en la interfaz	 Wifi	 Redes Alámbricas	----
Enviar información de la interfaz	 Wifi	 Redes Alámbricas	----
Procesar comandos	 Servidor web AWS	 Servidor web propio	 Google Cloud
Procesamiento de datos del usuario	 Servidor web AWS	 Servidor web propio	 Google Cloud
Procesamiento de datos de dispositivos	 Servidor web AWS	 Servidor web propio	 Google Cloud
Procesamiento de información de base de datos	 Servidor web AWS	 Servidor web propio	 Google Cloud
Mostrar estados de las tomas de corriente	 Página Web	 Aplicación móvil	 Aplicación y web móvil
Mostrar intensidad de las luces	 Página Web	 Aplicación móvil	 Aplicación y web móvil
Mostrar gráficas de consumo de energía	 Página Web	 Aplicación móvil	 Aplicación y web móvil

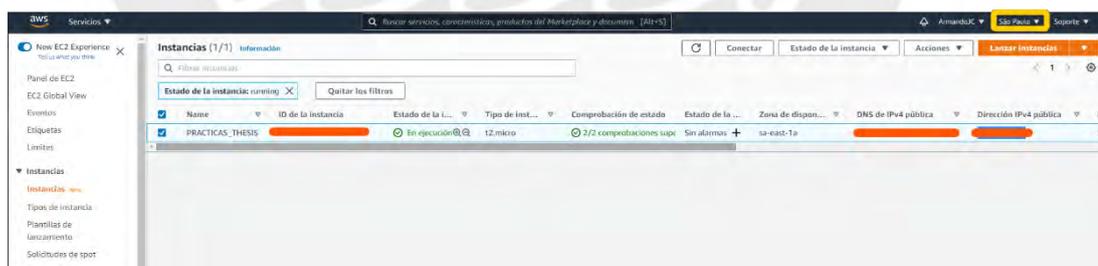
Anexo D: Instalaciones

Creación de instancia en *Amazon Web Services*:

Se tendrá que obtener una cuenta en *Amazon Web Services* (AWS por sus siglas en inglés). Luego de esto, se deberá hacer uso del servicio EC2. En este, se debe crear una instancia con los atributos requeridos. En este caso se elegirá Ubuntu Server 18.04 LTS que es un despliegue de Linux ya bastante conocido.



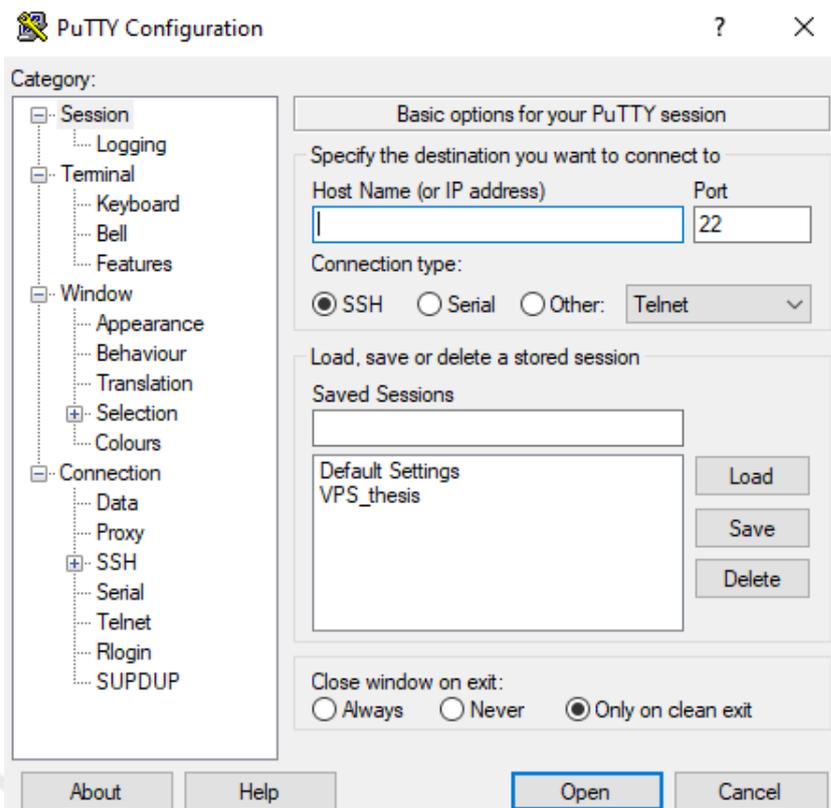
Cabe mencionar que las instancias están alojadas en servidores de todo el mundo, pero es posible seleccionar dónde puede estar el nuestro. De todas las opciones, se encontró que lo más conveniente es seleccionar la opción más cercana, que es Sao Paulo. Con esto, ya se tiene la instancia creada correctamente.



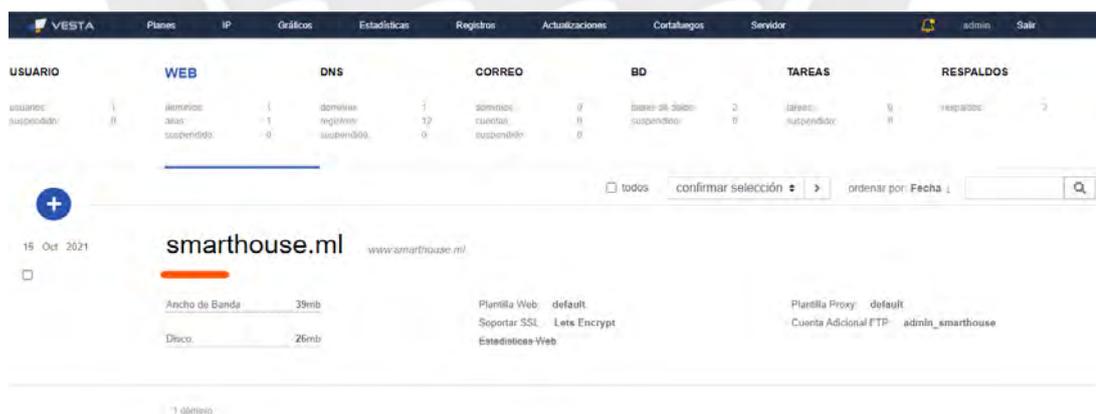
Instalación de *Vesta Panel*:

Previamente a la instalación del panel, se debe conseguir un dominio para la página web y verificar que esté desplegado en todo el mundo para que pueda ser reconocido. El dominio que se consiguió fue "smarthouse.ml". Con el dominio tramitado y desplegado pasamos a la instalación de Vesta Panel en el servidor.

Para poder instalar en el servidor, primero necesitamos conectarnos y con ese fin usaremos nuestra herramienta PuTTY para conectar con el servidor vía SSH.



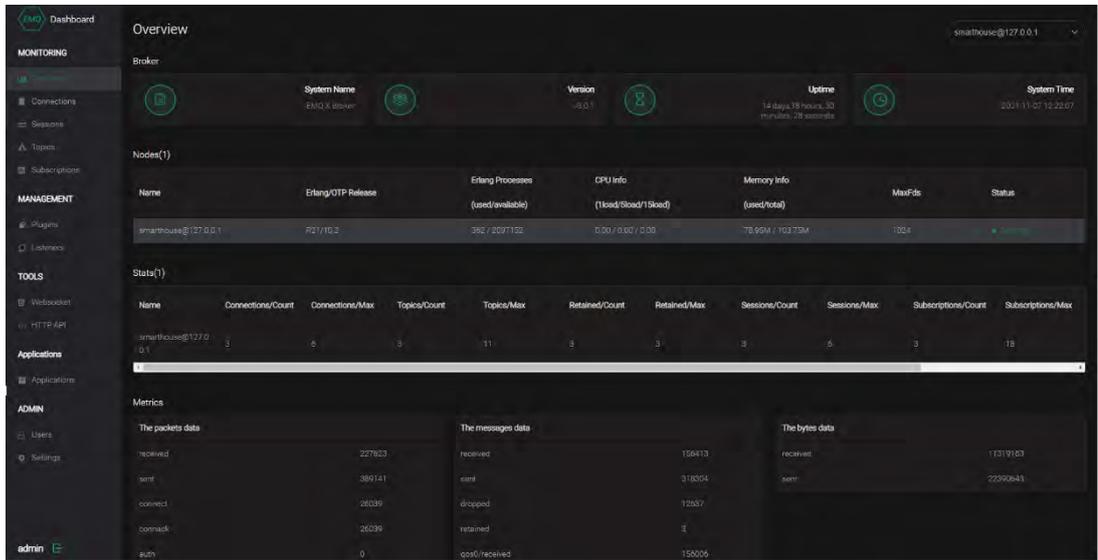
Una vez conectados por terminal, se procederá a hacer la instalación de Vesta Panel indicando el dominio que se tiene y un usuario y contraseña seguros. Luego de esto, se deberá ir al navegador e ingresar a la dirección IP por el puerto 8083. Si ya tenemos el dominio, se puede usar también (www.smarthouse.ml:8083). También se deberá configurar el certificado SSL de la página y por medio de PuTTY, hacer que Vesta Panel también tenga el certificado SSL para poder entrar de manera segura.



Luego de instalar Vesta Panel, se debe configurar su cortafuegos y también los grupos de seguridad de AWS para que los puertos que usemos tengan la conexión permitida y podamos ingresar sin ningún problema a Vesta, conectarnos por FTP, conectarnos con la base de datos, conectarse con el broker EMQ, etc.

Instalación del *broker EMQ*:

Para poder instalar EMQX, se conecta con PuTTY al servidor vía SSH. Una vez conectados, procedemos a realizar la instalación. Con la instalación, se irá al panel por medio del puerto 18083. Si ya tenemos el dominio, se puede usar también (www.smarthouse.ml:18083). También se deberá configurar el certificado SSL por medio de PuTTY.



The screenshot displays the EMQX Dashboard Overview page. The interface is dark-themed and includes a sidebar with navigation options like MONITORING, MANAGEMENT, TOOLS, and ADMIN. The main content area shows the following data:

- Broker Overview:** System Name (EMQX broker), Version (v5.0.1), Uptime (14 days, 19 hours, 53 minutes, 28 seconds), and System Time (2021-11-07 12:22:07).
- Nodes(1):** A table listing the node 'smarthouse@127.0.0.1' with details on Erlang/OTP Release (R21/19.2), Erlang Processes (362 / 2007152), CPU Info (0.00 / 0.00 / 0.00), Memory Info (78.99M / 1027.9M), MaxFds (1024), and Status (Running).
- Stats(1):** A table showing statistics for 'smarthouse@127.0.0.1' including Connections/Count (3), Connections/Max (6), Topics/Count (3), Topics/Max (11), Retained/Count (8), Retained/Max (3), Sessions/Count (3), Sessions/Max (5), Subscriptions/Count (3), and Subscriptions/Max (18).
- Metrics:** Three tables showing data for 'The packets data', 'The messages data', and 'The bytes data'.

The packets data		The messages data		The bytes data	
received	227623	received	158413	received	11319153
sent	389141	sent	318304	sent	22959643
connect	26039	dropped	12657		
connack	26039	retained	8		
auth	0	qos0/received	155000		

Anexo E: Cotizaciones

Hlk-pm01:

mercado libre hlk pm01 Descarga gratis la app d

Categorías Ofertas de la semana Tiendas oficiales Tu historial Tus compras Vender Armando ...

Búsquedas relacionadas: hllink

Hlk pm01
8 resultados

Ordenar publicaciones
Más relevantes

Fuente Aislada Step Down Hlk-pm01 De 220 Vac/vdc A 5vdc/3w
S/ 20
12x S/ 167
3 vendidos - Lima

Relé:

mercado libre modulo rele 1 canal optoacoplador Solo en Arduino

Categorías Ofertas de la semana Tiendas oficiales Tu historial Tus compras Vender

Electrónica, Audio y Video > Componentes
Electrónicos > Arduino

Modulo rele 1 canal optoacoplador
15 resultados

Ordenar publicaciones

Modulo Relé 1 Canal Optoacoplador Arduino
S/ 8
12x S/ 067
Lima

Sensor de corriente invasivo:

mercado libre modulo sensor de corriente acs712 30a Solo en Arduino

Categorías Ofertas de la semana Tiendas oficiales Tu historial Tus compras Vender

Electrónica, Audio y Video > Componentes
Electrónicos > Arduino

Modulo sensor de corriente acs712 30a
9 resultados

Modulo Sensor De Corriente Acs712 30a
S/ 14
12x S/ 117
8 vendidos - Lima

Sensor de corriente no invasivo:

mercado libre

sct 013 030 sensor de corriente ac no invasivo 0 30a ard Solo en Arduino Descarga gratis

Categorías Ofertas de la semana Tiendas oficiales Tu historial Tus compras Vender Armar

Electrónica, Audio y Video > Componentes Electrónicos > Arduino

Sct 013 030 sensor de corriente ac no invasivo 0 30a arduino

7 resultados



Sct-013-030 Sensor De Corriente Ac No Invasivo 0-30a

S/ 45

12x S/ 375

4 vendidos - Lima

Wemos D1 mini:

Llámanos ahora: 997646858 Contacte con nosotros Iniciar sesión

NAYLAMP MECHATRONICS

wemos d1 mini S/ 28,00

INICIO ARDUINO COMUNICACIÓN SENSORES ACTUADORES BLOG TUTORIALES

Comunicación Inalámbrico **Wemos D1 mini ESP8266** Volver a los resultados de búsqueda de "wemos d1 mini" (2 resultados extra)



Ver más grande

Wemos D1 mini ESP8266

S/ 30,00

Modelo ESP8266-WM1

Plataforma de desarrollo para Internet de las cosas (IoT) de fácil integración. Te permite crear rápidamente proyectos que se pueden conectar a Internet por WiFi.

Enviar a un amigo Imprimir

Cantidad: 1 Añadir al carrito Añadir a la lista de deseos

Dimmer AC:

Llámanos ahora: 997646858 Contacte con nosotros Iniciar sesión

NAYLAMP MECHATRONICS

dimmer ac S/ 28,00

INICIO ARDUINO COMUNICACIÓN SENSORES ACTUADORES BLOG TUTORIALES

Drivers **Modulo Dimmer AC** Volver a los resultados de búsqueda de "dimmer ac" (2 resultados extra)



Ver más grande

Modulo Dimmer AC

S/ 28,00

Modelo DRV-DIMMER

Controla cargas de corriente alterna (ventiladores, bombillas, motores) desde tu Arduino.

Producto Agotado

tu@email.com

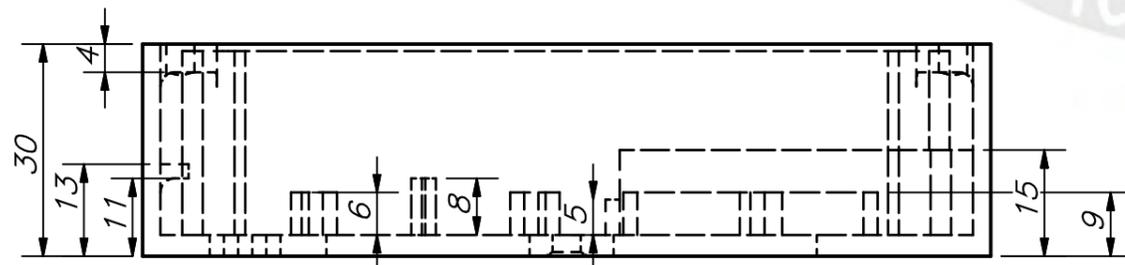
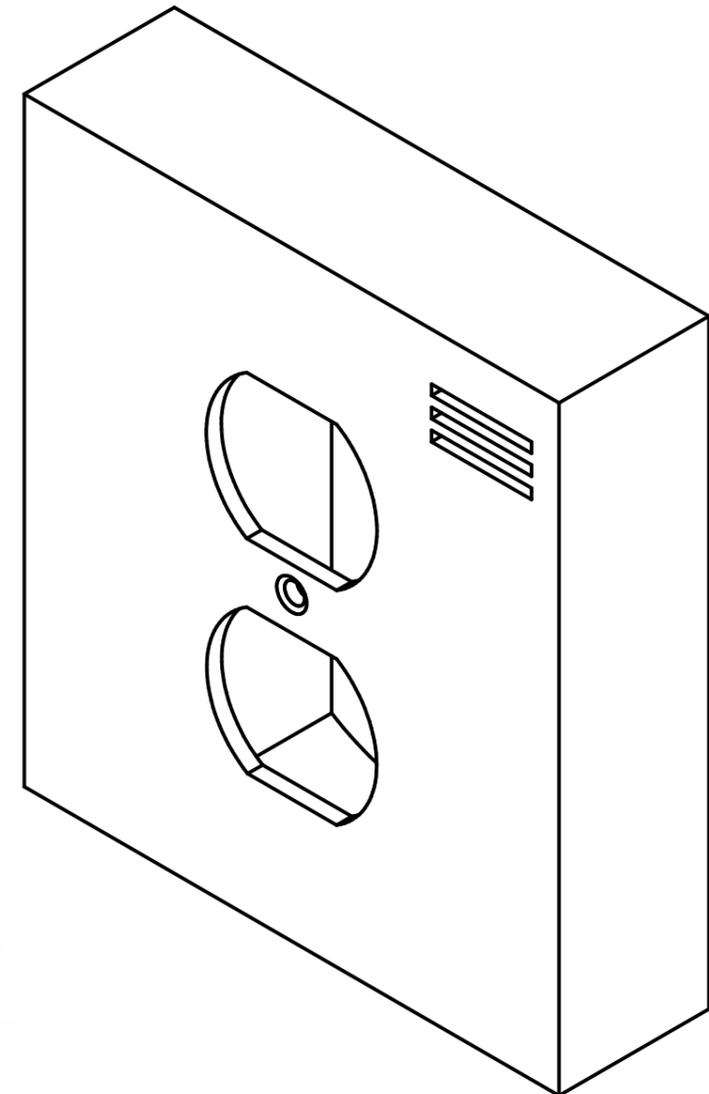
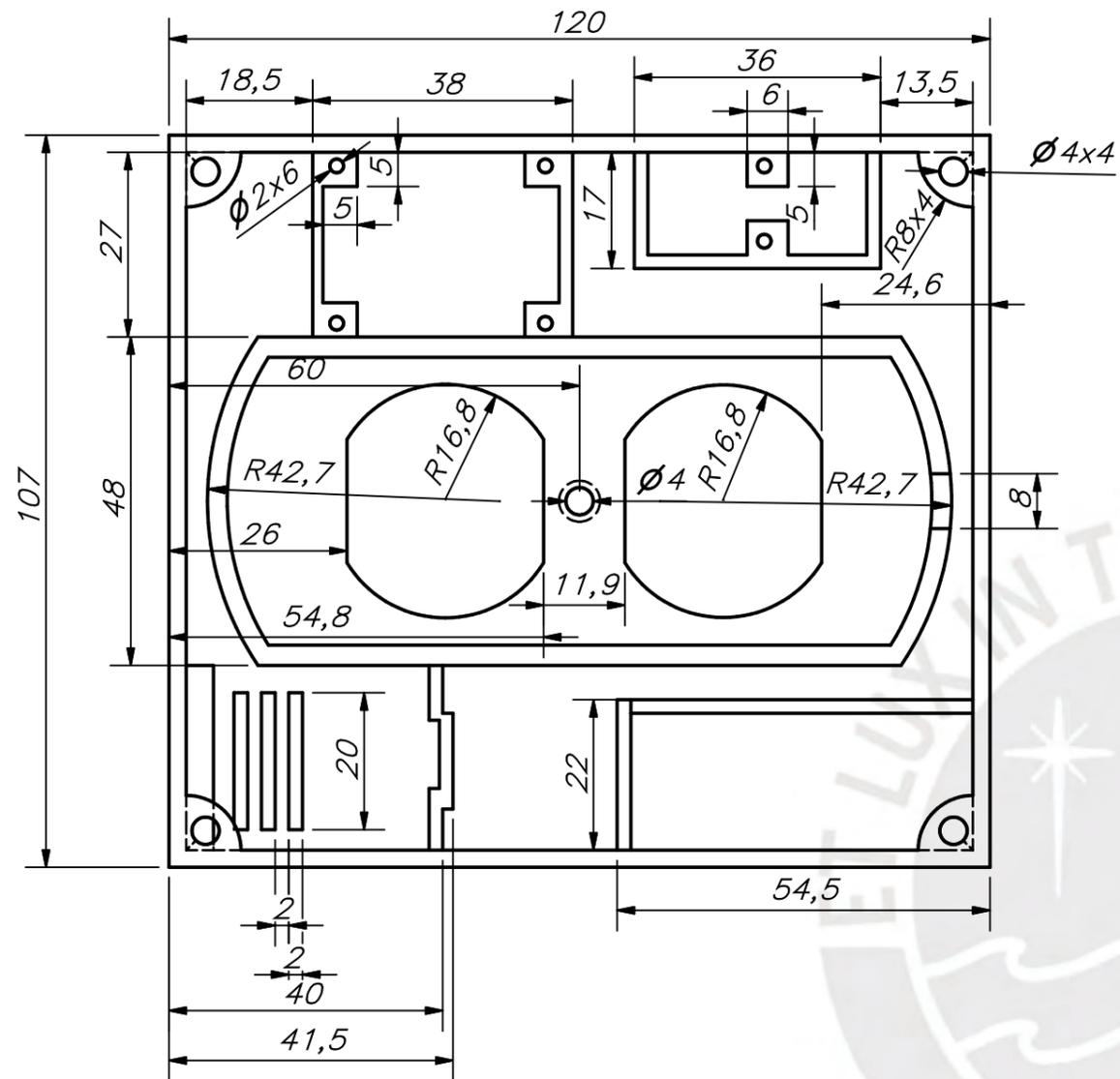
Avísame cuando esté disponible

Enviar a un amigo Imprimir

Añadir a la lista de deseos

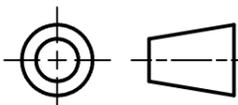
Anexo F: Planos

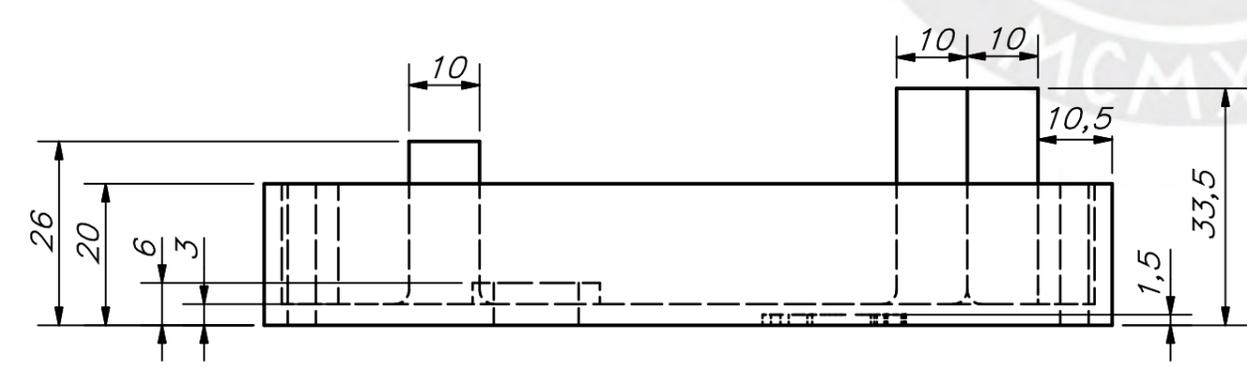
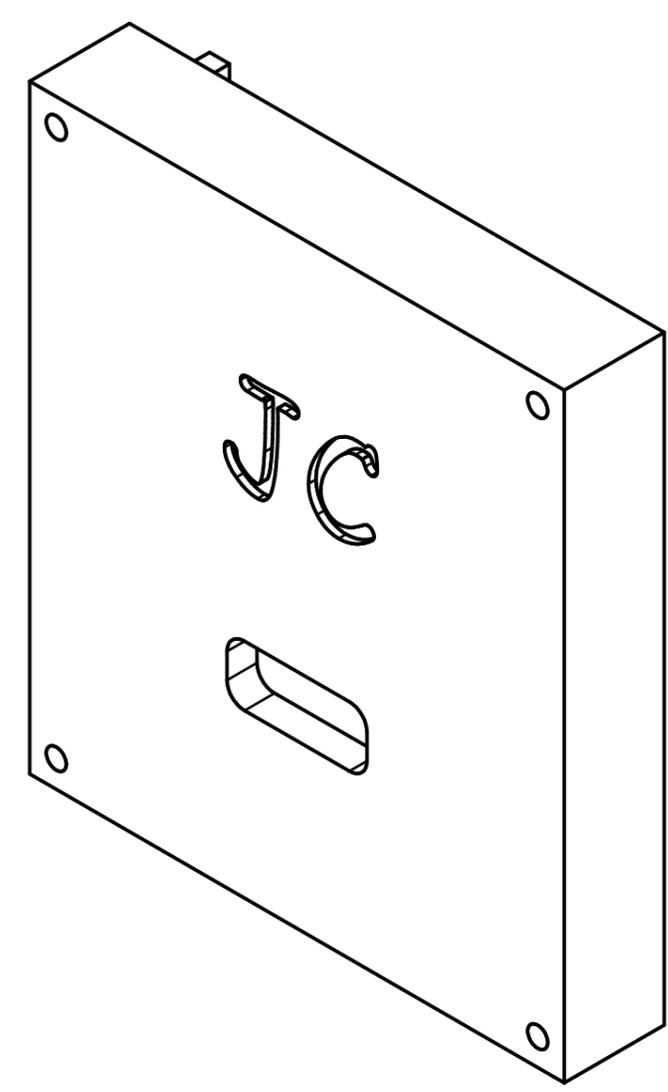
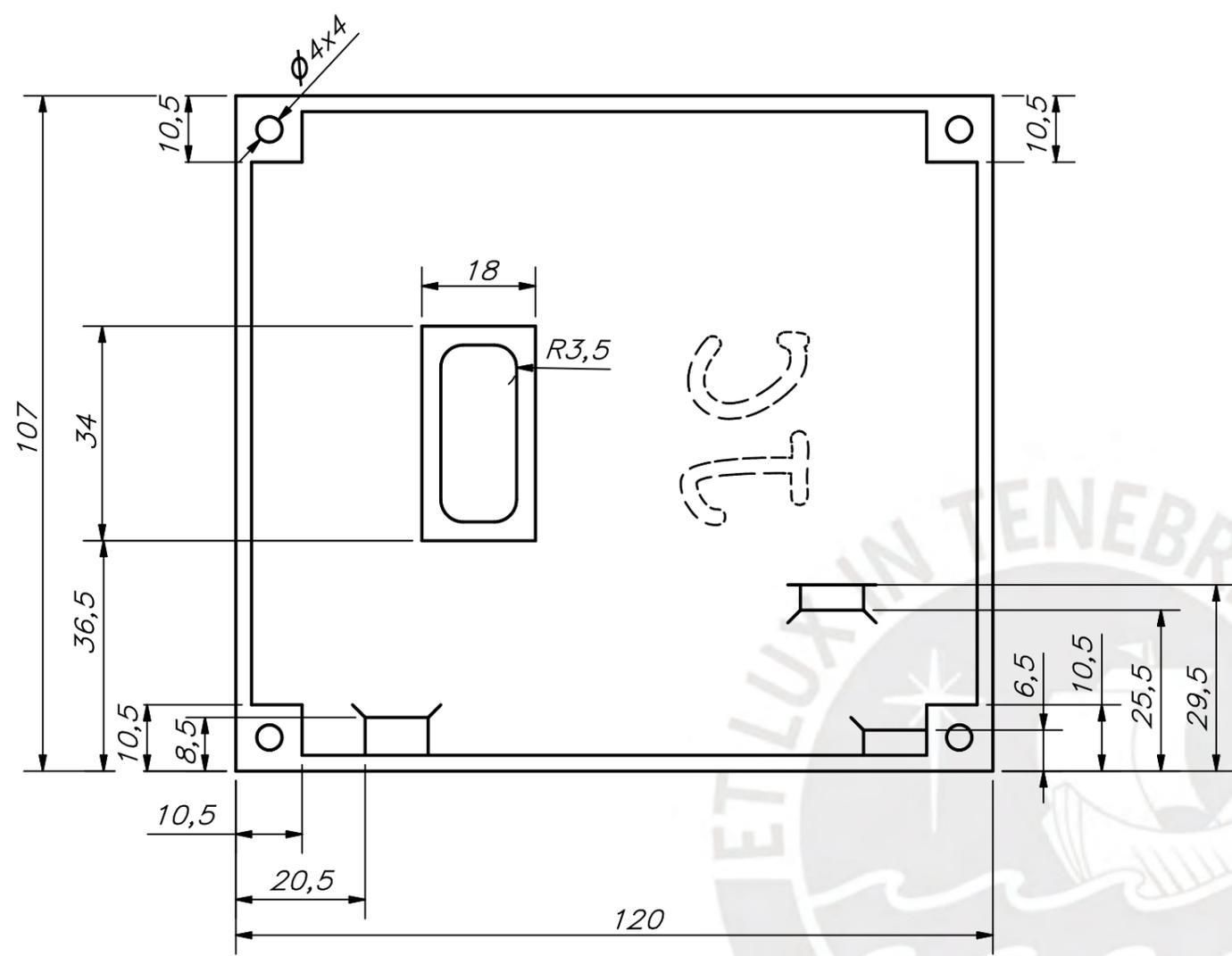




Todos los bordes interiores son de 2mm y los exteriores son de 2.5 mm

TOLERANCIAS DIMENSIONALES SEGÚN DIN ISO 2768-1				
GRADO DE EXACTITUD	Más de 0,5 hasta 3	Más de 3 hasta 6	Más de 6 hasta 30	Más de 30 hasta 120
BASTO	±0,2	±0,3	±0,5	±0,8

ACABADO SUPERFICIAL ---	TOLERANCIA GENERAL SEGÚN DIN ISO 2768-1 BASTO	MATERIAL PLA
PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ CIENCIAS E INGENIERÍA - MECATRÓNICA		
MÉTODO DE PROYECCIÓN 	CONTENEDOR - TOMACORRIENTE	ESCALA 1:1
20125767	JAIMES CHACON, ARMANDO CHRISTOFER	FECHA: 2018. 11. 12
		LÁMINA: TOM-01

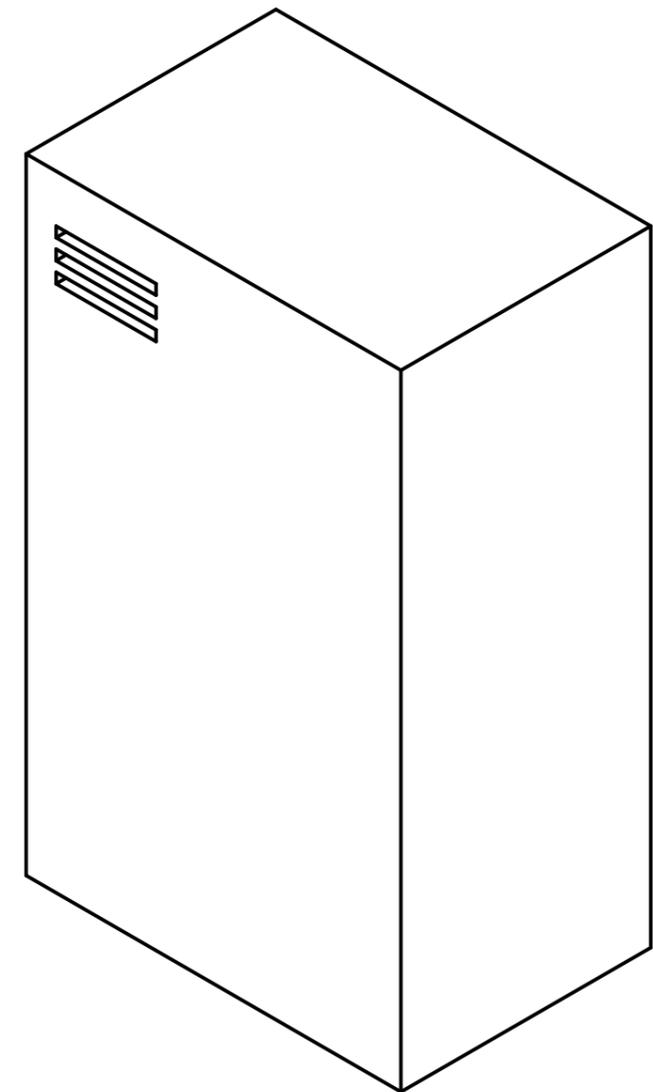
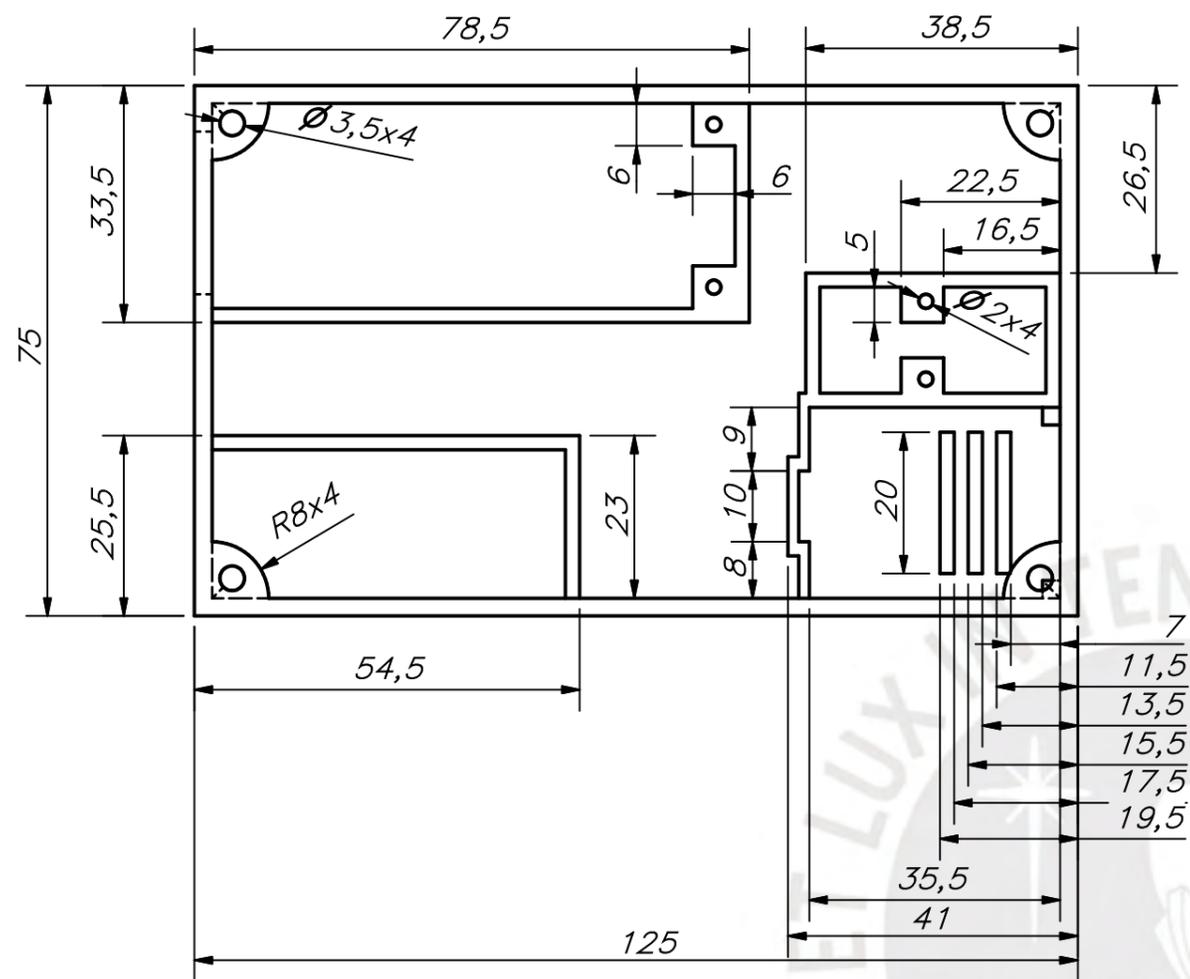


Todos los bordes interiores son de 2mm
y los exteriores son de 2.5 mm

**TOLERANCIAS DIMENSIONALES
SEGÚN DIN ISO 2768-1**

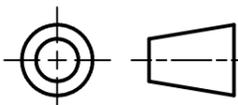
GRADO DE EXACTITUD	Más de 0,5 hasta 3	Más de 3 hasta 6	Más de 6 hasta 30	Más de 30 hasta 120
BASTO	±0,2	±0,3	±0,5	±0,8

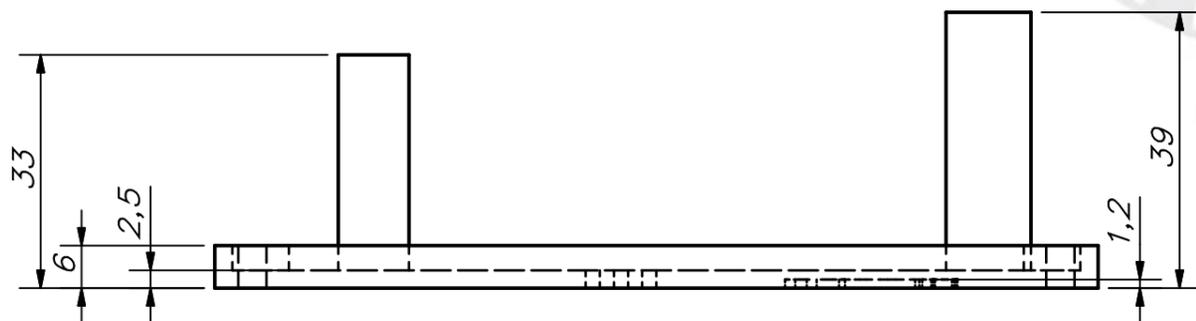
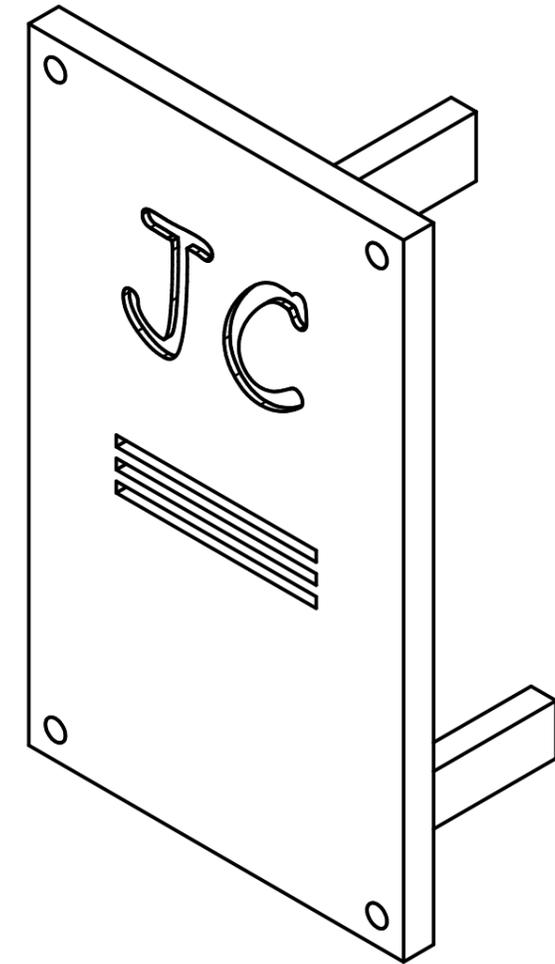
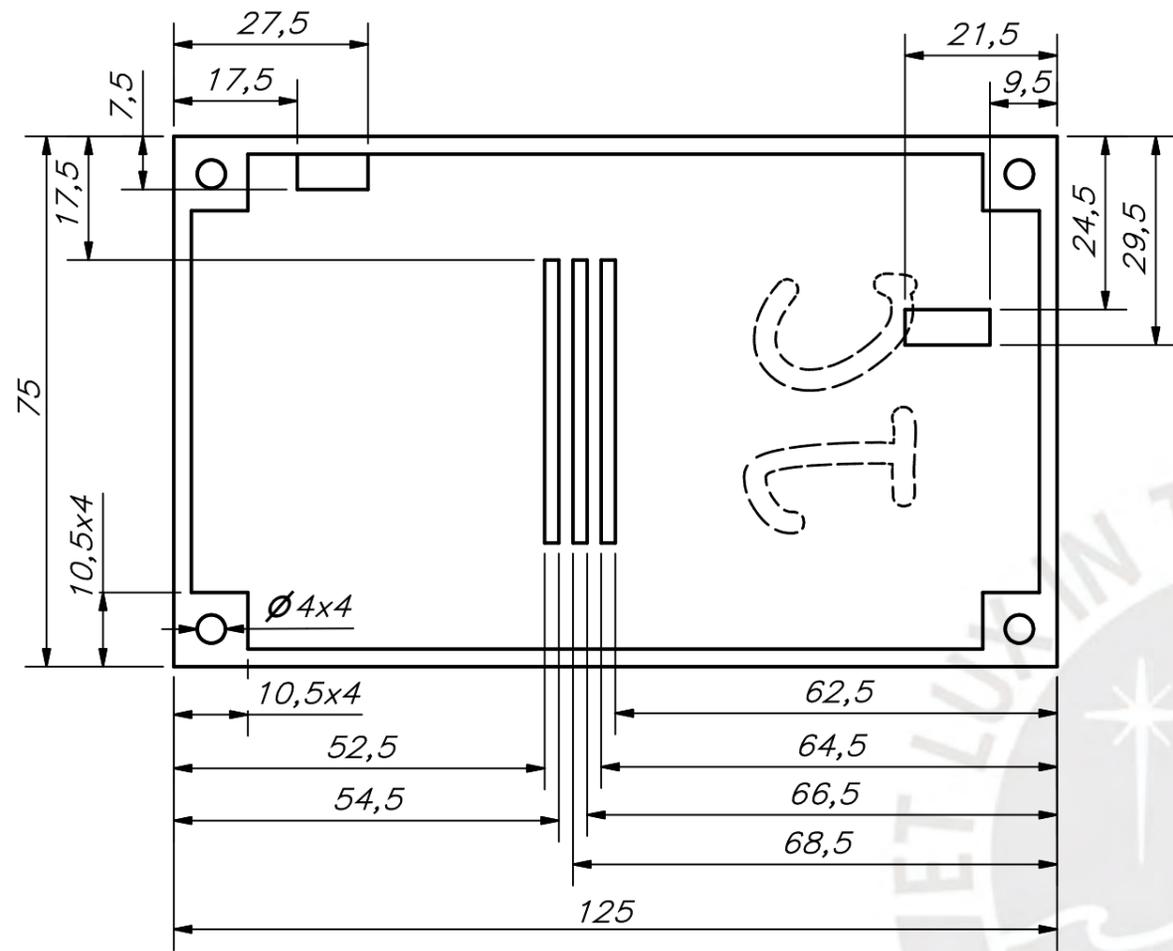
ACABADO SUPERFICIAL ---	TOLERANCIA GENERAL SEGÚN DIN ISO 2768-1 BASTO	MATERIAL PLA
PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ CIENCIAS E INGENIERÍA – MECATRÓNICA		
MÉTODO DE PROYECCIÓN 	TAPA – TOMACORRIENTE	ESCALA 1:1
20125767	JAIMES CHACÓN, ARMANDO CHRISTOFER	FECHA: 2018.11.20
		LÁMINA: TOM-02



Todos los bordes interiores son de 2mm
y los exteriores son de 2.5 mm

TOLERANCIAS DIMENSIONALES SEGÚN DIN ISO 2768-1				
GRADO DE EXACTITUD	Más de 0,5 hasta 3	Más de 3 hasta 6	Más de 6 hasta 30	Más de 30 hasta 125
BASTO	±0,2	±0,3	±0,5	±0,8

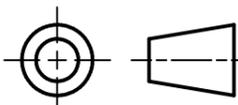
ACABADO SUPERFICIAL ---	TOLERANCIA GENERAL SEGÚN DIN ISO 2768-1 BASTO	MATERIAL PLA
PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ CIENCIAS E INGENIERÍA - MECATRÓNICA		
MÉTODO DE PROYECCIÓN 	CONTENEDOR - ILUMINACIÓN	ESCALA 1:1
20125767	JAIMES CHACON, ARMANDO CHRISTOFER	FECHA: 2018.11.21
		LÁMINA: ILU-01



**TOLERANCIAS DIMENSIONALES
SEGÚN DIN ISO 2768-1**

GRADO DE EXACTITUD	Más de 0,5 hasta 3	Más de 3 hasta 6	Más de 6 hasta 30	Más de 30 hasta 125
BASTO	±0,2	±0,3	±0,5	±0,8

Todos los bordes interiores son de 2mm y los exteriores son de 2.5 mm

ACABADO SUPERFICIAL ---	TOLERANCIA GENERAL SEGÚN DIN ISO 2768-1 BASTO	MATERIAL PLA
PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ CIENCIAS E INGENIERÍA - MECATRÓNICA		
MÉTODO DE PROYECCIÓN 	TAPA - ILUMINACIÓN	ESCALA 1:1
20125767	JAIMES CHACON, ARMANDO CHRISTOFER	FECHA: 2018.11.22
		LÁMINA: ILU-02