

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad en Ingeniería en Electricidad y Computación^a

**Facultad de Ingeniería en Mecánica y Ciencias de la
Producción^b**

Plataforma de microistemas IoT enfocada a la sistematización de
componentes

PROYECTO INTEGRADOR

Previo la obtención del Título de:

Ingeniera en Ciencias de la Computación^{a,1}

Ingeniero en Mecatrónica^b

Ingeniera en Telemática^{a,2}

Presentado por:

Ronny Gabriel Benitez Ordoñez^b

Mercedes Natalia Mawyin Cabay^{a,1}

Milena Desiree Riquero Herrera^{a,2}

GUAYAQUIL - ECUADOR

Año: 2022

DEDICATORIA

El presente proyecto lo dedico, en primer lugar, a mi abuela por estar siempre pendiente y cuidar de mí. A mi familia, por apoyarme y ser un soporte en mi vida. Por último, a aquellos que ya no están en vida porque me ayudaron a madurar para enfrentar la nueva etapa que está por empezar. – **Milena Riquero.**

Dedico este proyecto a mis padres por su apoyo y motivación constante en todo momento. A mi novia por su amor, compañía y apoyo en este proyecto. A la Msig. Adriana Collaguazo cuya sabiduría y paciencia han sido claves en la realización de este proyecto. Por último, a mi abuelita difunta por enseñarme valores y fortaleza que hoy me guían. – **Ronny Benitez**

A mis padres, Jorge Mawyin y Mercedes Cabay, y a mi hermano, Jorge Daniel, por su amor y apoyo incondicional en mi vida. A mi tía Irene, que siempre está cuando la necesito. Y a mi abuelito, por su presencia constante. Este logro también es suyo. – **Natalia Mawyin**

AGRADECIMIENTOS

Le agradezco sinceramente a Dios por iluminar mi camino todos los días. A mis padres, por todo el esfuerzo que han hecho para que lograra tener estudios profesionales. A mis maestros por todos los conocimientos transmitidos, la confianza prestada y los consejos dados. – **Milena Riquero**

En primer lugar, agradezco a Dios por su constante guía y bendición en mi vida. También quiero agradecer a la ESPOL, por brindarme la oportunidad de adquirir conocimientos y formarme como un profesional. – **Ronny Benitez.**

Agradezco a Dios, a mis padres y a todos mis maestros por ser parte fundamental de mi educación. En particular a la Dra. Mónica Villavicencio y la Msig. Adriana Collaguazo por su guía constante a lo largo de mi formación y crecimiento como profesional. – **Natalia Mawyin**

DECLARACIÓN EXPRESA

"Los derechos de titularidad y explotación, nos corresponde conforme al reglamento de propiedad intelectual de la institución; *Ronny Gabriel Benitez Ordoñez, Mercedes Natalia Mawyin Cabay, Milena Desiree Riquero Herrera*, damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"



Ronny Benitez



Natalia Mawyin



Milena Riquero

EVALUADORES CARRERA^{a,1}

Ronald Criollo, Msig.

PROFESOR DE LA MATERIA

Mónica Villavicencio, PhD.

PROFESOR TUTOR

EVALUADORES CARRERA^{a,2}

Washington Velásquez, PhD.

PROFESOR DE LA MATERIA

Adriana Collaguazo, Msig.

PROFESOR TUTOR

EVALUADORES CARRERA^b

Bryan Puruncajas, M.Sc.

PROFESOR DE LA MATERIA

Efraín Terán, M.Sc.

PROFESOR TUTOR

RESUMEN

La educación en IoT se ve obstaculizada por la falta de componentes y complejidad en la configuración de los dispositivos, así como por la escasez de cursos dinámicos e interactivos. Estos desafíos pueden socavar el interés de los proyectos de IoT en los estudiantes, limitando su motivación.

En el presente proyecto se desarrolló una plataforma con fin educativo para interactuar con dispositivos IoT mediante escenarios, donde se pudiese presenciar en tiempo real su funcionamiento, constando de tres partes importantes: la plataforma web, los escenarios, y la comunicación entre dispositivos y plataforma.

Para saber qué dispositivos eran los más indicados, se realizó un análisis de soluciones, determinando que la solución óptima es el uso de una Raspberry Pi como controlador y ESP32 TTGO como los nodos en cada escenario. Además, en la comunicación se utilizó el protocolo MQTT para poder enviar datos hacia el controlador, que se encarga de guardar los datos en la base, adicional a ello cada placa cuenta con un servicio de actualización de código fuente a través de OTA. Por parte de la plataforma, esta envía y recibe información desde la base de datos a través de API CRUD y fue codificada en Django para el BackEnd y Angular en el FrontEnd.

Al final de la implementación se realizaron las respectivas pruebas de uso, de las cuales se pudo corroborar la escalabilidad de la plataforma, además de comprarse el tiempo promedio en la comunicación entre los nodos y el controlador, y el controlador y la base.

Palabras Clave: Casa Inteligente, Comunicación, Control de Temperatura, IoT, Plataforma Web.

ABSTRACT

IoT education is hampered by a lack of components and complexity in device configuration, as well as a shortage of dynamic and interactive courses. These challenges can undermine students' interest in IoT projects, limiting their motivation and engagement.

In this project, a platform was developed for educational purposes to interact with IoT devices through scenarios, where their operation could be seen in real time, consisting of three important parts: the web platform, the scenarios, and the communication between the devices and the platform.

In order to know which devices were the most suitable, an analysis of solutions was carried out, having that the most optimal solution is the use of a Raspberry Pi as controller and ESP32 TTGO as the nodes in each scenario. In addition to this, the communication uses the MQTT protocol to send data to the controller, which handles storing the data in the base, where each board has a source code update service through OTA. As for the platform, it sends and receives information from the database through API CRUD and was coded in Django for the BackEnd and Angular in the FrontEnd.

At the end of the implementation, the respective usability tests were performed, from which it was possible to corroborate the scalability of the platform, in addition to check the average time in the communication between the nodes and the controller, and the controller and the base.

Keywords: *communication, home automation, IoT, Temperature Control, Web Platform.*

ÍNDICE GENERAL

RESUMEN.....	I
ABSTRACT	II
ÍNDICE GENERAL	III
ABREVIATURAS.....	VI
SIMBOLOGÍA.....	VII
ÍNDICE DE FIGURAS	VIII
ÍNDICE DE TABLAS.....	X
CAPÍTULO 1.....	1
1. Introducción	1
1.1 Descripción del problema	4
1.2 Justificación del problema	5
1.3 Objetivos	6
1.3.1 Objetivo general.....	6
1.3.2 Objetivos específicos	6
1.4 Estado del arte	7
1.4.1 Los efectos perpetuos del COVID 19 en el aprendizaje	7
1.4.2 Internet de las Cosas – IoT	8
1.4.3 Uso de dispositivos inteligentes en las universidades	11
1.4.4 Uso de dispositivos inteligentes en las industrias	11
1.4.5 Comunicación publicador/suscriptor	11
1.4.6 Protocolos de comunicación	12
1.4.7 Actualización de código de manera remota	12
1.5 Fundamentos Teóricos.....	13
1.5.1 Dispositivos IoT.....	13
1.5.2 Controlador PID	14
1.5.3 Método Ziegler-Nichols	15

CAPÍTULO 2.....	17
2. Metodología	17
2.1 Alternativas de solución.....	17
2.2 Criterios de selección	17
2.3 Proceso de gestión del proyecto	19
2.4 Requerimientos del proyecto.....	20
2.5 Alcances y limitaciones	21
2.6 Riesgos y beneficios de la solución.....	22
2.7 Beneficiarios de la solución	22
2.8 Métricas de rendimiento	22
2.9 Métricas de satisfacción de usuario.....	23
2.10 Prototipado de la plataforma web.....	23
2.10.1 Evaluación de la solución	29
2.11 Diseño de la solución	29
2.11.1 Diagrama de despliegue	30
2.11.2 Hardware del proyecto.....	31
2.11.3 Arquitectura del proyecto	31
2.11.4 Diagrama esquemático o circuito.....	33
2.11.5 Software.....	38
2.11.6 Casos de uso.....	38
2.11.7 Diagrama de interacción	39
2.11.8 Diseño de la base de datos	42
2.12 Función de transferencia	43
2.13 Constantes de control.....	44
CAPÍTULO 3.....	46
3. Resultados y Análisis.....	46
3.1 Análisis	46

3.1.1	Desarrollo de la plataforma.....	46
3.1.2	Módulo de librerías	47
3.1.3	Administrador Django	48
3.1.4	Plan de implementación.....	49
3.1.5	Implementación de infraestructura.....	50
3.2	Pruebas.....	52
3.2.1	Pruebas de la plataforma	52
3.3	Análisis de costos.....	57
3.4	Escenario de casa inteligente utilizando dispositivos IoT.....	58
3.5	Escenario de control de temperatura usando control PID	58
3.6	Tiempos de ejecución.....	59
3.6.1	Comunicación MQTT	59
3.6.2	Peticiones a la base de datos	60
3.6.3	Subida de archivos a los nodos mediante OTA	61
3.7	Encuesta de satisfacción de usuario	61
CAPÍTULO 4.....		63
4.	Conclusiones Y Recomendaciones	63
4.1	Conclusiones.....	63
4.2	Recomendaciones.....	64

BIBLIOGRAFÍA

APÉNDICES

ABREVIATURAS

ESPOL	Escuela Superior Politécnica del Litoral
IoT	Internet of Things
STEM	Science, Technology, Engineering and Mathematics
TIC	Tecnología de la Información y Comunicación
MOOC	Massive Open Online Courses
OTA	Over the Air
PID	Control Proporcional Integral Derivativo

SIMBOLOGÍA

s	segundos
ms	milisegundos
μs	microsegundos

ÍNDICE DE FIGURAS

Figura 1.1 Avance tecnológico al pasar los años	1
Figura 1.2 Estadísticas del uso de la plataforma Moodle y aula virtual	7
Figura 1.3 Categoría de los dispositivos IoT	10
Figura 1.4 Mecanismo Publicador/Suscriptor	12
Figura 1.5 Diagrama de bloques de un control PID en lazo cerrado	15
Figura 2.1 Proceso de gestión de proyecto basado en metodología ágil	20
Figura 2.2 Prototipo de pantalla de inicio de sesión de la plataforma web	24
Figura 2.3 Prototipo de la pantalla de inicio de la plataforma web.....	25
Figura 2.4 Prototipo de pantalla de escenarios de la plataforma web.	26
Figura 2.5 Prototipo de pantalla de escenario de la plataforma web.	27
Figura 2.6 Prototipo de la pantalla librerías de la plataforma web.	28
Figura 2.7 Prototipo de la pantalla de librería de la página web.	28
Figura 2.8 Retrospectiva del MVP (Figma).....	29
Figura 2.9 Diagrama de despliegue.....	30
Figura 2.10 Plano arquitectónico	32
Figura 2.11 Diseño de maqueta en 3D.....	32
Figura 2.12 Diagrama de arquitectura del proyecto.....	33
Figura 2.13 Diagrama de esquemático del escenario de casa inteligente.....	34
Figura 2.14 Vista 3D en un PCB del diagrama esquemático del escenario casa inteligente.	34
Figura 2.15 Diagrama de esquemático del escenario de control de temperatura PID	35
Figura 2.16 Vista 3D en un PCB del diagrama esquemático del escenario control de temperatura PID.	36
Figura 2.17 Diagrama de esquemático de subcontrolador del escenario de casa inteligente.	37
Figura 2.18 Vista 3D en un PCB del diagrama esquemático del subcontrolador del escenario de casa inteligente	38
Figura 2.19 Diagrama de casos de uso	39
Figura 2.20 Diagrama de secuencia.....	41
Figura 2.21 Diseño de la base de datos	42
Figura 2.22 Diagrama simplificado del sistema a controlar	43

Figura 2.23 Modelo de salida simulada vs el modelo experimental.....	44
Figura 2.24 Tiempo de estabilización del modelo simulado.	45
Figura 3.1 Vista de las librerías en la plataforma.....	48
Figura 3.2 Vista de librerías Colores RGB.....	48
Figura 3.3 Vista de servidor web	49
Figura 3.4 Infraestructura del laboratorio remoto.....	51
Figura 3.5 Pantalla de inicio de la plataforma.....	52
Figura 3.6 Pruebas de conexión con 50 usuarios.....	54
Figura 3.7 Prueba de conexión con 500 usuarios	55
Figura 3.8 Prueba de conexión con 1000 usuarios	56
Figura 3.9 Tiempo de respuesta de 1000 usuarios	56
Figura 3.10 Vista de escenario Casa Inteligente	58
Figura 3.11 Vista de escenario Control PID.....	59

ÍNDICE DE TABLAS

Tabla 1.1 Descripción de los dispositivos electrónicos considerados para la solución del proyecto.....	13
Tabla 1.2 Tipo de controladores en el primer método	16
Tabla 1.3 Tipo de controladores en el segundo método.....	16
Tabla 2.1 Ponderación de criterios	18
Tabla 2.2 Prioridad de alternativas	18
Tabla 3.1 Planificación del proyecto	49
Tabla 3.2 Análisis de costos	57
Tabla 3.3 Medidas de tendencia central en la comunicación MQTT	60
Tabla 3.4 Medidas de tendencia central en las peticiones a la base de datos	60
Tabla 3.5 Medidas de tendencia central del tiempo de subida de código fuente por OTA.....	61

CAPÍTULO 1

1. INTRODUCCIÓN

Desde el 2010, han surgido grandes avances tecnológicos (Figura 1.1), y con ellos el aumento de dispositivos inteligentes, haciendo que las personas consuman productos automatizados dentro de los ambientes que los rodean. El uso del Internet de las Cosas, por sus siglas en inglés IoT (Internet of Things), aumenta y, en algunos de los casos, es casi imperceptible para el usuario. IoT se encuentra presente en varias ramas, por ejemplo, en la domótica [1], la agricultura de precisión [2], la medicina [3] y entre otras áreas más [4] [5] [6].

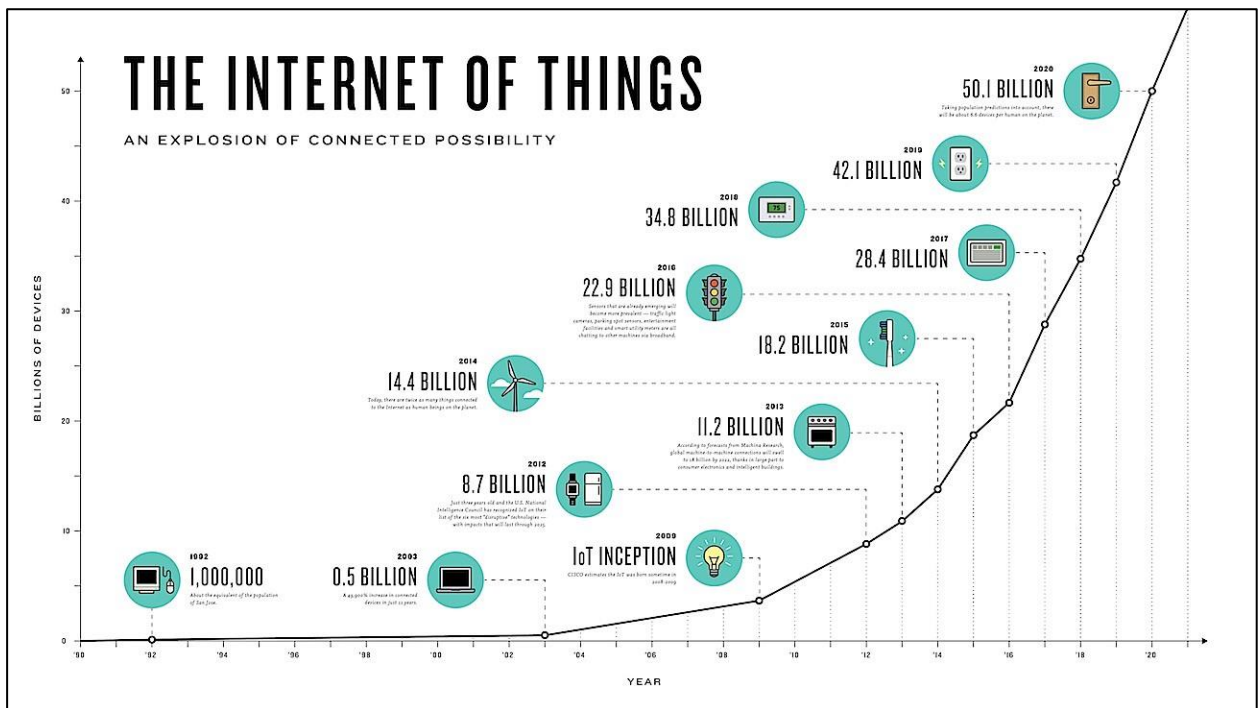


Figura 1.1 Avance tecnológico al pasar los años [7].

Adicionalmente, se espera un incremento exponencial en el uso de tecnologías IoT, teniendo una proyección de los siguientes diez años donde 125 mil millones de estos dispositivos se encontrarán conectados entre sí [8]. Se proyecta una inversión en el campo de esta tecnología de hasta 120 mil millones de dólares, según Columbus Louis de la revista Forbes, que ofrece una visión futura del impacto que va a crear la tecnología IoT dentro de las actividades diarias de las personas [9].

Otra de las áreas que se ha vuelto involucrada dentro del crecimiento del IoT es la educación. Elmqaddem [10] revela que la realidad virtual podrá ser aplicada en un futuro próximo, donde el estudiante se puede ver inmerso y usaría objetos dentro de esta a través de controladores, con ello obtendrá una experiencia de aprendizaje más interesante y fructuosa que la convencional. Sin embargo, este tema se encuentra aún en desarrollo, pero existen autores que afirman que el avance de la educación en las últimas décadas ha ido de la mano con el avance tecnológico y la inmersión de la sociedad de la información y conocimiento, la cual le ha permitido tener grandes avances en la educación, especialmente en el volumen de información que se puede llegar a manejar gracias a esta y las TIC [11], a partir de lo cual Fidalgo, ingeniero de la Universidad Politécnica de Madrid, afirma que deben existir cambios que permitan la continua mejora de los procesos de enseñanza y aprendizaje, además que estos deben ser sostenibles, transferibles, eficientes y eficaces, siendo alguno de estos cambios el resultado de la integración de las TIC, creando con ello un nuevo enfoque en la educación [12].

Es a través de las TIC que varias herramientas son utilizadas por el personal docente con la finalidad de aumentar la interacción en las clases y obtener mejores resultados de aprendizaje, siendo fundamental una continua exploración y uso de diversas fuentes y recursos educacionales que van surgiendo con el constante avance de la tecnología [11].

Tomando en cuenta la inmersión de las nuevas tecnologías dentro de la enseñanza, surge la inclusión y el aumento de la educación a nivel mundial a partir de las necesidades de las personas alrededor del mundo, la cual consiste principalmente en disminuir los obstáculos que limitan el acceso, logrando que cualquier individuo, sin importar su estatus económico, social, discapacidad, lugar, entre otros, pueda acceder a ella con facilidad, surgiendo la creación y desarrollo de instituciones parcial y completamente virtuales [13].

Por lo tanto, a partir del surgimiento de la educación virtual, sobre todo en las universidades e institutos de tercer nivel, varias de las herramientas que son utilizadas para impartir laboratorios consisten principalmente en simuladores, los cuales son eficaces para realizar un prototipo antes de poner en marcha un sistema o esquema a gran escala, evitando que se consuman recursos económicos y espacios físicos que no se poseen o que son escasos. Además, que permite a los estudiantes inferir y/o probar distintos escenarios con distintos comportamientos de un sistema [14].

Existen varios métodos que son utilizados dentro de los simuladores con enfoque realista, uno de ellos es el método VISIR (Virtual Instrument Systems in Reality) [15], el cual consiste en la interacción de los componentes físicos y el espacio virtual que engloba el laboratorio, principalmente para realizar cableado y mediciones replicando un área de trabajo real de manera remota; al igual que este existen otros métodos como son Data Acquisition Cards [16], NetLab [17] y RemotElectLab [18] que siguen en análisis con respecto a su realismo y funcionalidad [14].

Uno de los avances que se ha obtenido dentro de la educación virtual en los programas STEM (Science, Technology, Engineering and Mathematics), es la inclusión de cursos masivos conocidos como MOOC (Massive Open Online Courses), de los cuales consisten principalmente en proveer al estudiante una interacción online respecto a sus clases a través de foros; lo que permite edificar una comunidad donde se ven incluidos estudiantes, profesores y asistentes que tienen uno o más temas en común, sin embargo el aumento excesivo de usuarios interesados en estos conllevó a que se desarrollaran otros métodos masivos de enseñanza variando del contexto formal y educacional hacia uno más informal y abierto a la gran cantidad de usuarios existentes [13].

Pero a raíz de esto surge un problema de esta modalidad de educación, el cual consiste en que existen cursos que requieren la implementación o interacción con un laboratorio para el desarrollo de las habilidades del estudiante, de lo que se proponen dos posibles soluciones: la primera que se basa en un laboratorio físico real con un acceso remoto para los usuarios, y la segunda que trata principalmente en el uso de software o simuladores para realizar pruebas. En la mayoría de los casos se opta por la segunda opción debido a que requiere menos recursos físicos y tiene un mayor dinamismo a nivel de escenarios [13].

No obstante, a pesar de lo novedoso que es la educación virtual, esta presenta algunas dificultades, entre las que se pueden mencionar está la necesidad de un dispositivo con los suficientes recursos para llevar el curso ya sea un ordenador, celular o tableta, debido a que en varios de los casos las herramientas a utilizar requieren de soporte de visualización y modelado 3D, además, la mayoría de los simuladores que se utilizan en reemplazo de los equipos reales no proyectan los escenarios semejantes a los que surgen en un entorno existente, por lo que dentro de estos no existen muchos casos en los cuales algo salga mal y se puedan ver las reacciones en estos escenarios, por lo que esto orilla a la pérdida del sentido de responsabilidad y cuidado que deben

tener los estudiantes durante las prácticas. Finalmente, en la mayoría de los puestos de trabajo requieren que se haya tenido una interacción directa con los dispositivos, ya que únicamente teniendo una experiencia con equipos reales se pueden desarrollar habilidades entorno a estas [13].

Es en base a aprovechar la educación virtual y las TIC como herramientas que en el presente proyecto se presenta el desarrollo de una alternativa que permita a los estudiantes tener un encuentro real con los dispositivos IoT de manera remota, teniendo en cuenta las diferentes limitantes que pueden surgir antes y durante su desarrollo, estableciendo bases conceptuales y recolectando casos donde se hayan aplicado laboratorios virtuales desde el mismo enfoque, además, se va proceder a la estimación de recursos y obtención de los mismos para finalmente hacer una puesta en escena teniendo en cuenta un par de escenarios de microsistemas IoT, teniendo también una plataforma web como la vista del usuario para la interacción con los módulos IoT y una comunicación entre dicha plataforma y los dispositivos para llegar a tener escenarios interactivos.

En este capítulo se abarcan temas como la descripción y justificación del problema, secciones donde se describe con mayor profundidad el problema identificado, además se plantea el objetivo general del proyecto y los objetivos específicos teniendo en cuenta cada carrera implicada, para finalmente estructurar un estado del arte en torno a los temas relevantes que se relacionan con la problemática y el proyecto.

1.1 Descripción del problema

El interés de los estudiantes por el aprendizaje acerca del IoT enfocado a proyectos realizados con dispositivos electrónicos que pueden ser utilizados en varios aspectos, tanto domóticos como industriales, hace que la educación genere nuevos desafíos en el aprendizaje, tal como el uso de simuladores debido a la falta de componentes dentro de la unidad educativa para abarcar la demanda estudiantil. Cabe señalar que no todos los estudiantes tendrán un conocimiento previo del tema debido a las diferentes orientaciones académicas existentes.

Algunas de las unidades educativas que plantean una mejora continua en la educación haciendo uso del IoT han realizado como proyecto la implementación de una plataforma educativa que interactúa con el dispositivo IoT integrado en un laboratorio remoto [19].

Adicional a ello, se tienen las configuraciones reiterativas de las placas y dispositivos IoT, por lo que al ser un trabajo constante y repetitivo se pierde el enfoque del aprendizaje o no se culmina con éxito el proyecto; de ello surge la necesidad de estandarizar el funcionamiento de algunos dispositivos mediante las librerías reutilizables con la finalidad de brindar un mejor entendimiento en la configuración y reprogramación del dispositivo IoT.

Además, el avance tecnológico en la revolución industrial 4.0 se acerca más al uso de dispositivos inteligentes en sus sistemas, generando un interés en los estudiantes de diferentes especialidades de aprender sobre los dispositivos inteligentes. Para ello las instituciones educativas constan de laboratorios especializados que contienen tales dispositivos para uso educativo [20]. Sin embargo, el uso de laboratorios se verá limitado por el cambio de modalidad perdiendo así el interés del estudiante.

Teniendo como referencia que la modalidad virtual también es usada en las instituciones educativas que imparten sus cursos con una educación a distancia, una de las desventajas claras es la carencia de contenido que sirve como apoyo para el aprendizaje del estudiante [21].

Finalmente se identifica como necesario que tanto el hardware, compuesto por las placas y sensores que intervienen en los ambientes bajo la tecnología IoT, y el software, el cual consiste principalmente en la plataforma web donde se observan los dispositivos, mantengan una comunicación constante y consistente que permita la configuración remota de los sistemas, teniendo en cuenta protocolos de comunicación, que pueden ser utilizados desde los dispositivos finales como son los sensores, siendo utilizados por los sistemas embebidos para realizar el envío de información, y las diferentes herramientas que facilitan este trabajo.

1.2 Justificación del problema

El presente proyecto se enfoca en la mejora continua de la educación dándole una expectativa de progreso en la comprensión del aprendizaje por el estudiante respecto al conocimiento del funcionamiento de los dispositivos inteligentes, a través de una interfaz amigable que permita interactuar en tiempo real las acciones enviadas al laboratorio remoto.

La plataforma educativa de microsistemas IoT permitirá a los estudiantes conocer el diseño y la aplicación de proyectos inteligentes del mundo real. Esto puede ayudar a

los estudiantes a comprender mejor cómo funciona tal tecnología y como puede ser aplicada en diferentes escenarios.

Además, los microsistemas IoT también pueden utilizarse para crear módulos educativos que están disponibles a través de una plataforma de código abierto. Estos módulos permiten a los estudiantes aprender como punto de partida para sus propios proyectos IoT y seguir desarrollándolos según sus necesidades. Adicionalmente, esta plataforma ayudará a los estudiantes a adquirir una valiosa experiencia en el campo de la tecnología.

Con lo anteriormente descrito, el uso de una plataforma educativa amigable, para que el estudiante sea capaz de interactuar en tiempo real con las acciones enviadas a un conjunto de dispositivos inteligentes, es una forma de rellenar los vacíos de los estudiantes. Además, este proyecto permite ayudar al estudiante en la implementación de proyectos IoT usando como recurso clave la reutilización de librerías y así mejorar su enfoque de aprendizaje en el funcionamiento de los dispositivos inteligentes sin la necesidad de movilizarse.

1.3 Objetivos

1.3.1 Objetivo general

Desarrollar una plataforma educativa de microsistemas IoT utilizando lenguajes de programación convencionales y protocolos de comunicación para el funcionamiento de interacciones en un laboratorio remoto.

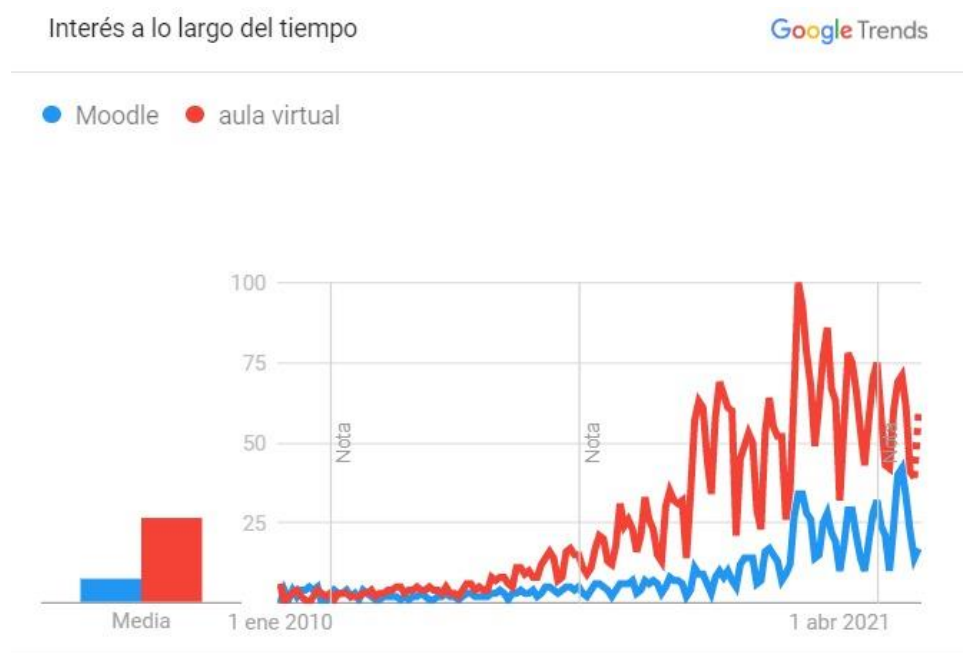
1.3.2 Objetivos específicos

1. Diseñar un escenario del laboratorio remoto para la enseñanza de un control de temperatura usando PID. (Ingeniería en Mecatrónica)
2. Diseñar un escenario de uso domótico en el laboratorio remoto enfocado a luz inteligente, monitoreo de temperatura y seguridad. (Ingeniería en Mecatrónica)
3. Implementar una comunicación entre los dispositivos IoT y una plataforma web a través de métodos de comunicación que permitan la actualización de su código fuente. (Ingeniería en Telemática)
4. Implementar una comunicación entre el controlador con la base de datos para la lectura de la información almacenada y envío de señales de accionamiento. (Ingeniería en Telemática)

5. Desarrollar una plataforma educativa como interfaz de comunicación entre el usuario y el dispositivo IoT. (Ingeniería en Ciencias Computacionales)
6. Desarrollar un módulo de gestión de librerías reutilizables en la plataforma para uso educativo en dispositivos IoT. (Ingeniería en Ciencias Computacionales)

1.4 Estado del arte

En los últimos 10 años se ha producido un incremento del uso de plataformas virtuales en el Ecuador.



Ecuador. 1/1/10 - 17/11/22. Búsqueda web.

Figura 1.2 Estadísticas del uso de la plataforma Moodle y aula virtual [22]

Con base en análisis y el resultado arrojado por Google Trends, la búsqueda de plataformas virtuales en universidades y colegios por los estudiantes hace énfasis la importancia del uso de las plataformas para gestionar y dar seguimiento al estudiante. Cabe destacar que, para preparar un mejor entendimiento de las clases, los docentes brindan materiales didácticos para una mejor comprensión.

1.4.1 Los efectos perpetuos del COVID 19 en el aprendizaje

No se puede negar que el impacto que ha tenido el COVID 19 es de escala global. La educación sufrió grandes cambios a la llegada de esta pandemia, que tuvo que forzar a las instituciones educativas y universidades a adaptarse en poco tiempo a una nueva modalidad que era poco practicada antes del 2020; la virtualidad. Esto se hizo posible

utilizando aplicaciones para reuniones digitales como Teams, Zoom o Google Meets; pero además los centros de enseñanza adaptaban sus propias plataformas educativas para este propósito. Es así como hoy en día siguen surgiendo nuevas innovaciones con respecto al aprendizaje virtual.

La aceptación general de la virtualidad depende muchas veces de la materia impartida y los recursos que se necesitan para esta. Gupta, et.al., en su investigación “Perception of BDS students of Kathmandu University on online learning during COVID-19 pandemic” [23] realizó un estudio donde se determinó que estudiantes de IT (Tecnologías de la información) tienen en su mayoría una percepción agradable de la virtualidad y están dispuestos a aplicarla en su totalidad; por otro lado, en otras investigaciones se muestran resultados desfavorables de parte de estudiantes universitarios de ingeniería mecánica, una carrera con componentes más prácticos y que normalmente requiere el uso de laboratorios físicos para poner en práctica su conocimiento. En ese estudio los estudiantes determinan que el modelo de enseñanza virtual no se ajusta a sus expectativas, sobre todo por la falta de acceso a los recursos necesarios [24].

Actualmente se está regresando progresivamente a la presencialidad, pero el efecto de la pandemia en la educación continua presente al demostrar que las instituciones están listas para regresar a la virtualidad si la situación lo requiere, sea por feriados, eventos deportivos o algún acontecimiento externo.

1.4.2 Internet de las Cosas – IoT

El incremento de las innovaciones tecnológicas en el área de IoT [25] ha implicado como resultado muchos cambios en el día a día de las personas; ya sea tanto en los hogares, como industrias, ciudades e incluso en campos como la agricultura. Uno de estos temas se puede ver en [2], donde se expone una comparativa sobre el progreso de los últimos años en la tecnología IoT para la industria agricultora.

Otra de las áreas donde más se ha involucrado el IoT es en la educación, donde cada vez surgen nuevos métodos de enseñanza que tratan de sobreponerse a las dificultades que trae el aprendizaje de las nuevas tecnologías. Entre estos se resalta [19], que presenta una metodología de enseñanza innovadora en la que se impartió un curso virtual con Teams donde los estudiantes utilizaban comandos a distancia para poder interactuar con los dispositivos a través de un servidor MQTT. En [26] también se presenta un enfoque similar al implementar un laboratorio remoto mediante el uso de

cámaras web, controladores de telepresencia y una interfaz gráfica con Processing para que los estudiantes puedan manipular interactivamente los dispositivos IoT desde sus hogares.

Los autores en [27] exponen sobre la plataforma Lab in a box (LiB), que proporciona un conjunto de sensores para diferentes aplicaciones de monitoreo ambiental. Esta arquitectura propuesta descompone la estructura de IoT en bloques funcionales y proporciona medios sencillos de módulos por capas; lo que a su vez abre las puertas a un nuevo enfoque modular que podrá ser implementado en diversos proyectos, para llevar a un sistema independiente y escalable.

1.4.2.1 Ventajas y desventajas del IoT en la enseñanza

Con el crecimiento exponencial de la tecnología y el incursionamiento en la Industria 4.0, el sector educativo busca adaptarse para mantener una educación de calidad y potenciar el rendimiento académico de los estudiantes. Es así como ingresa la tecnología IoT, que busca la conexión entre los dispositivos inteligentes y el internet [28], facilitando la tarea diaria tanto de profesores como alumnos, e.g., el uso de las pizarras digitales fomenta a la colaboración e investigación y permite el intercambio de ideas gracias al acceso a la información de todo tipo [29].

Sin embargo, el factor negativo que es común en las escuelas es la falta de recurso monetario para adquirir tales dispositivos y el desconocimiento por parte de algunos profesores y estudiantes respecto al manejo de los dispositivos digitales.

1.4.2.2 Categorización de dispositivos IoT

La clasificación general de los dispositivos IoT consiste en dispositivos restringidos, puertas de enlace, dispositivos inteligentes y plataformas IoT en la nube. Los dispositivos restringidos son pequeños dispositivos que tienen un propósito específico, contando con CPU, memoria y energía limitados. Ejemplo de estos son los microcontroladores, sensores y actuadores [30].

Las puertas de enlace son dispositivos que sirven para enrutar datos a los servidores de la nube o lidiar con los problemas de latencia de red. Estos funcionan como una puerta de enlace o conexión, de ahí el nombre, entre dispositivos, facilitando además el intercambio de recursos entre varios computadores [30].

Las plataformas IoT en la nube representan la infraestructura de software, generalmente ofreciendo servicios en la nube de tipo PAAS, que permiten la interacción con el usuario, registro de dispositivos, manejo de datos y eventos, entre otros.

Las 3 categorías son pilares claves al momento de realizar proyectos IoT, pues como se puede observar en la Figura 1.3 cada una tiene su propia funcionalidad específica, pero al momento de realizar proyectos IoT todas se acoplan y comunican sin problemas.

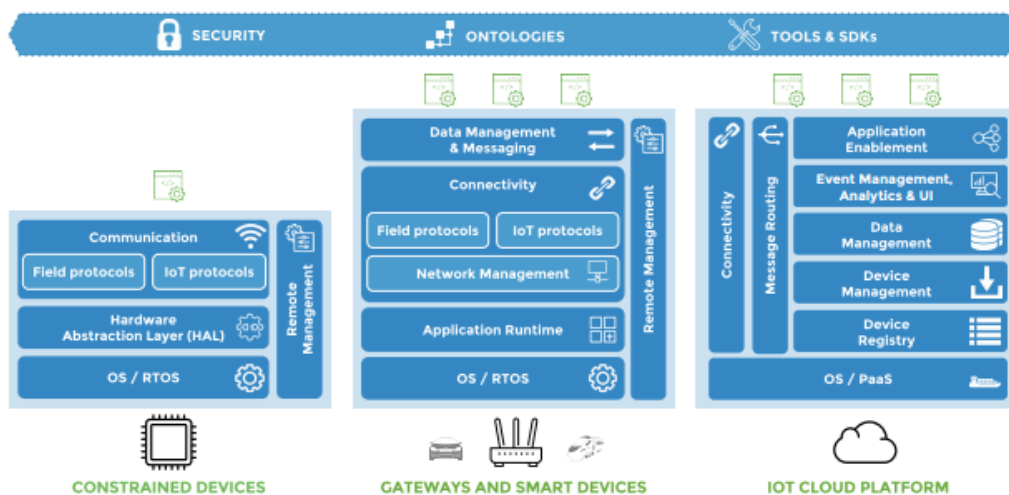


Figura 1.3 Categoría de los dispositivos IoT [30]

1.4.2.3 Aprendizaje de lenguajes de programación en IoT

El estudio de IoT implica varios factores, como la manipulación de hardware y dispositivos inteligentes, el entendimiento de protocolos de comunicación y el desarrollo de código. Es en este último donde surgen varias incógnitas al momento de empezar, ya que para un principiante los lenguajes de programación se verán muy similares y no siempre será fácil captarlo a la primera, lo que puede llevar a la frustración y a que se lleguen a rendir antes de tiempo.

Una solución a este problema es el surgimiento de los lenguajes de programación visuales (VLP), que son aplicaciones que utilizan componentes gráficos y figuras para la enseñanza de la programación. En [31] se considera el uso de los VLP aplicados al IoT, resaltando la importancia de la experiencia de usuario y como llegaría a ayudar en proyectos haciendo más fácil visualizar la lógica de la programación.

Este enfoque podría ser aplicado al laboratorio virtual y se plantea un estudio para determinar a profundidad que método de aprendizaje le conviene a estudiantes universitarios, dejando abierto a la oportunidad de aplicar las otras opciones en otros niveles, como escuela o colegio, en el futuro.

1.4.3 Uso de dispositivos inteligentes en las universidades

Las instituciones educativas han realizado estudios de la experiencia del estudiante que usan plataformas educativas para mejorar su experiencia en el entendimiento del funcionamiento de dispositivos inteligentes.

De acuerdo con [19], los estudiantes han hecho una retroalimentación exitosa cuando existe una operabilidad real con los dispositivos inteligentes. Una de las características esenciales es que el estudiante pueda visualizar en tiempo real la interacción que tiene con el conjunto de dispositivos desde su hogar mediante el uso de la plataforma educativa.

1.4.4 Uso de dispositivos inteligentes en las industrias

Los proyectos usando dispositivos inteligentes están en tendencia y en la actualidad se puede observar en diferentes áreas como en la domótica, varios ejemplos se pueden mencionar tales como el control de consumo energético inteligente, sistema contra intrusos y sistemas de detección y prevención de incendios; y para uso industrial su complejidad aumenta, pero el enfoque es el mismo, como por ejemplo el monitoreo y predicción de fallos de la maquinaria en un sistema de refrigeración [20].

1.4.5 Comunicación publicador/suscriptor

Para lograr una visualización e interacción con los componentes físicos de los microsistemas IoT a través de la plataforma web, es necesario que exista una capa de comunicación que el usuario no ve, en la cual se debe escoger el protocolo de comunicación para el intercambio de información entre ambas partes, tanto para un envío de información desde los dispositivos hacia la plataforma como viceversa.

Teniendo en cuenta esto, se identifica que uno de los métodos de comunicación óptimo para el intercambio de datos entre diferentes dispositivos es la comunicación publicador/suscriptor, el cual es mayormente usado en sistemas distribuidos, este consta de dos entes: el publicador y el suscriptor, de lo cual el suscriptor hace requerimientos al publicador a través de un nombre y definición de datos [32] [33].

El mecanismo publicador/suscriptor se basa en tres principios. El primero que consiste en el desacoplamiento espacial, lo que significa que la interacción entre el suscriptor y el publicador ocurre de manera anónima desconociendo la información del otro. El segundo recae en el tiempo desacoplado, esto significa que ambas partes no necesariamente se van a encontrar activas al mismo tiempo durante la comunicación lo

que permite que el publicador mande información a los suscriptores sin necesidad de que estos se encuentren activos. Finalmente, el tercer principio consiste en la sincronización desacoplada, es decir, que la comunicación publicador/subscriptor ocurre de manera asíncrona [33].

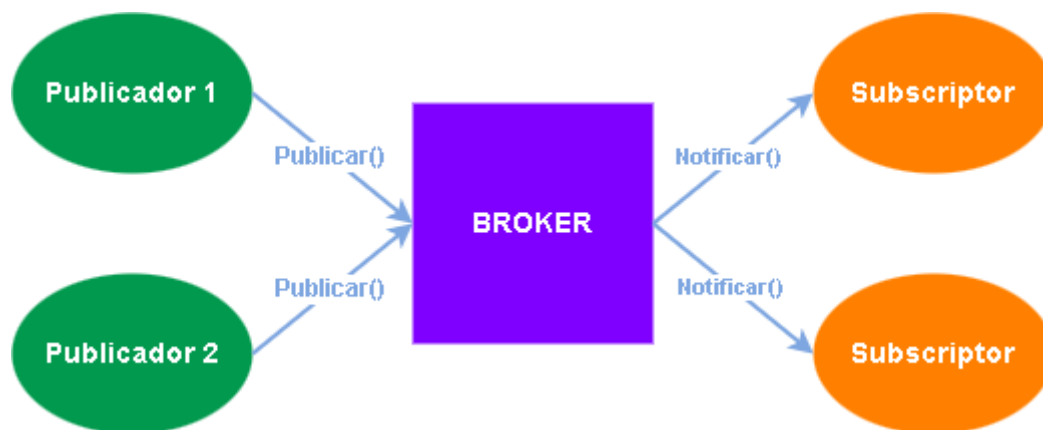


Figura 1.4 Mecanismo Publicador/Suscriptor [32]

En la Figura 1.4 se puede observar cómo se da el intercambio de información entre un publicador y un suscriptor a través del mecanismo ya mencionado.

1.4.6 Protocolos de comunicación

Existen varios protocolos de comunicación que pueden ser utilizados para intercambio de información que integren dispositivos IoT y plataformas web, el más común de todos en el servicio web es HTTP, el cual trabaja sobre el protocolo de transporte TCP con un tipo de comunicación request/response [34] y ha tenido mejoras incluidas en su versión HTTP 2.0 a nivel de parámetros de calidad de servicio como es la latencia, ancho de banda y porcentaje de paquetes perdidos, además de permitir múltiples conexiones en una sola sesión [35].

Por otra parte, para comunicar dos dispositivos IoT existen varios protocolos, pero el más conocido y que se adapta a la comunicación publicador/subscriptor es MQTT [36], el cual requiere de un nodo central llamado bróker a través del cual se comunican los publicadores y los suscriptores a través de tópicos [37].

1.4.7 Actualización de código de manera remota

Para lograr realizar una actualización de código fuente en las palcas ESP, es necesario saber mediante qué mecanismos se puede realizar dichos cambios y evitar tener una conexión física con el dispositivo en cuestión.

Bajo la idea antes explicada se escoge a la actualización mediante OTA (Over-The-Air), la cual es un tipo de programación que realiza una comunicación entre el software y el hardware del dispositivo, siendo capaz de realizar actualizaciones en los dispositivos IoT apoyándose en la conexión inalámbrica de este [38].

Esta forma de actualización de basa principalmente en el método POST del protocolo HTTP, funcionando de tal forma que mediante este se realizan las actualizaciones y una vez que esta llega al dispositivo es almacenada en la memoria flash de este. Antes de proceder a realizar la actualización se realiza una validación de la integridad, seguridad y robustes de estas para evitar inconsistencias a la hora de recibir los nuevos datos [38].

1.5 Fundamentos Teóricos

Esta sección permite introducir al lector los conceptos necesarios para su mejor comprensión de los dispositivos y herramientas utilizadas.

1.5.1 Dispositivos IoT

A continuación, se realiza una breve descripción de los dispositivos IoT que se han considerado en la solución de la problemática del proyecto, teniendo un desglose de estos en la Tabla 1.1.

Tabla 1.1 Descripción de los dispositivos electrónicos considerados para la solución del proyecto.

No.	Nombre del dispositivo	Descripción Técnica
1	ESP32	Microcontrolador de bajo costo y alta capacidad de computación, desarrollado por Espressif Systems [39]. Su doble núcleo y amplia variedad de periféricos lo hacen una herramienta ideal para aplicaciones de IoT [40]. Entre los modelos que hay de esta placa se escogió la ESP TTGO [40], debido a su particularidad para cargar el código a la placa sin accionar un botón externo.
2	Raspberry Pi	Sistema embebido de bajo costo y tamaño compacto, desarrollada por la fundación Raspberry Pi [41]. Su capacidad de procesamiento y flexibilidad la hacen adecuada para uso en aplicaciones de IoT basados en la mezcla de electrónica y computación [42].

3	Sensor fotorresistor	Dispositivo que permite medir la intensidad de luz debido a la variación de su resistencia. Este módulo es utilizado en aplicaciones como sistemas de iluminación automatizada o seguimiento solar [43].
4	Servomotor	Motor rotativo que puede ser ubicado de forma precisa en cualquier posición dentro de su rango de operación. Son controlados mediante el ancho de pulso de una señal cuadrada [44].
5	Láser KY-008	Dispositivo que emite un haz de luz laser [45]. Es usado para proyectos de detección de intruso o enviar una señal de audio usando el láser como medio de envío.
6	Receptor láser IS0203	Dispositivo capaz de detectar el haz de luz emitido por un puntero láser. Es usado en aplicaciones como detección de intrusos, obstáculos o alarmas [46].
7	Zumbador KY-012	Dispositivo que produce sonido mediante la vibración mecánica de una membrana, siendo de tipos pasivos o activos por su emisión de un solo tono o varios tonos respectivamente. [47].
8	Sensor de temperatura DS18B20	Sensor digital que se utiliza para medir la temperatura ambiental, capaz de medir temperaturas desde $-55\text{ }^{\circ}\text{C}$ hasta los $125\text{ }^{\circ}\text{C}$ [48].
9	Foco incandescente	Lámpara que produce luz y calor mediante la emisión de radiación electromagnética, cuando se calienta el filamento de tungsteno a alta temperatura [49]. Estos focos son utilizados en aplicaciones de control de temperatura, debido a su capacidad de generar calor de forma rápida y precisa [49].

1.5.2 Controlador PID

El controlador PID es un mecanismo de control que mediante un lazo de retroalimentación tiene la capacidad de controlar el tiempo de respuesta y mejorar su tiempo de precisión en la salida deseada del sistema. [50]. Este controlador consta de tres elementos: acción proporcional, integral y derivativa. De ahí proviene el nombre del controlador.

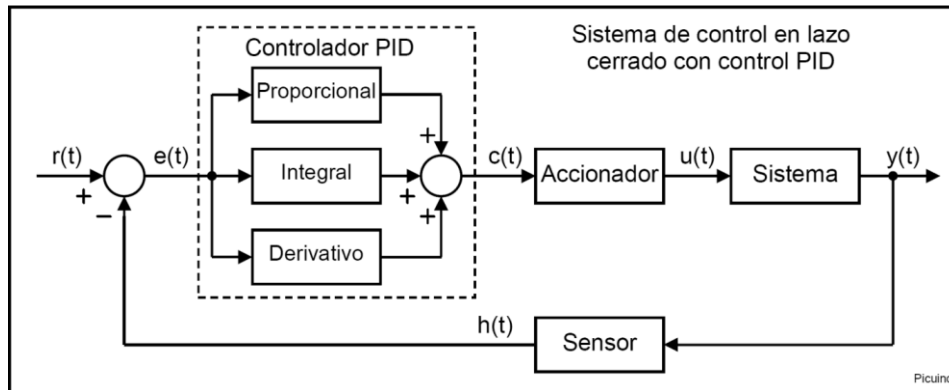


Figura 1.5 Diagrama de bloques de un control PID en lazo cerrado [41]

Los controladores PID combinan las tres ventajas que maneja cada control. La ecuación descrita en los controladores PID en un lazo cerrado se presenta de esta manera:

$$G_c(s) = k_p \left(1 + \frac{1}{\tau_i s} + \tau_d s \right) \quad (1)$$

Donde k_p es la ganancia proporcional, τ_i es la constante de tiempo integral y τ_d es la constante de tiempo derivativo [51].

1.5.3 Método Ziegler-Nichols

El método de Ziegler-Nichols es un proceso para sintonizar el control PID de manera empírica. Esto es muy útil cuando se desconoce el modelo matemático de la planta. Además, Ogata menciona dos reglas, primer y según método, de sintonía de Ziegler-Nichols [51].

1.5.3.1 Primer método

En este método el sistema recibe una entrada de escalón unitario y a partir de ella se obtiene los valores de forma experimental. Sin embargo, este método solo se puede usar cuando la respuesta de la curva tiene forma de S o sigmoideal. La función sigmoideal se caracteriza por depender de dos parámetros: el tiempo de retardo L y la constante de tiempo T cuya función de transferencia es:

$$G_p = \frac{K e^{-Ls}}{Ts + 1} \quad (2)$$

De eso, Ziegler-Nichols sugiere un ajuste según los tipos de controladores dada una respuesta escalón de acuerdo con la Tabla 1.2:

Tabla 1.2 Tipo de controladores en el primer método

Tipo de controlador	K_p	τ_i	τ_d
P	$\frac{T}{L}$	∞	0
PI	$0.9\frac{T}{L}$	$\frac{L}{0.3}$	0
PID	$1.2\frac{T}{L}$	$2L$	$0.5L$

1.5.3.2 Segundo método

En este método se fijan el parámetro derivativo e integral a cero. Luego se empieza a incrementar la constante proporcional desde 0 hasta un valor crítico que haga que el sistema oscile con una amplitud constante. Así se encontró la ganancia crítica K_{cr} y el periodo P_{cr} de forma empírica.

De ahí, Ziegler-Nichols sugiere un ajuste según los tipos de controladores basada en la ganancia y periodo críticos de acuerdo con la Tabla 1.3:

Tabla 1.3 Tipo de controladores en el segundo método

Tipo de controlador	K_p	τ_i	τ_d
P	$0.5K_{cr}$	∞	0
PI	$0.45K_{cr}$	$\frac{1}{1.2}P_{cr}$	0
PID	$0.6K_{cr}$	$0.5P_{cr}$	$0.125P_{cr}$

CAPÍTULO 2

2. METODOLOGÍA

Para la gestión del proyecto se utilizó la metodología de Scrumban, la cual es una combinación entre la metodología de Scrum y la metodología de Kanban. El uso de la metodología Scrumban permite definir las tareas dentro del proyecto para luego ejecutarlas, dándole seguimiento a la realización de cada una de las tareas establecidas y permite ver la carga de trabajo que lleva cada integrante del equipo.

2.1 Alternativas de solución

Para lograr el diseño e implementación de un laboratorio remoto se propusieron 3 alternativas de solución, las cuales cumplen con los objetivos del proyecto:

- **Alternativa A:** Uso de Raspberry PI [41] como controlador maestro usando como protocolo de comunicación MQTT.
- **Alternativa B:** Uso de ESP32 TTGO T8 [40] como controlador maestro usando como protocolo de comunicación WIFI.
- **Alternativa C:** Uso de un PLC de poco consumo como controlador maestro usando la arquitectura OPC UA.

2.2 Criterios de selección

Para discernir sobre las alternativas mencionadas y seleccionar la más conveniente, se realizó una matriz de decisión empleando los siguientes criterios:

- **Precio:** Este criterio está enlazado estrechamente con las limitaciones del proyecto debido a un fondo inicial establecido. Mientras menor sea el costo del dispositivo mayor influencia tendrá en su selección.
- **Transmisión de datos:** Este criterio se enfoca a la velocidad de transmisión de datos del dispositivo maestro al esclavo. Mientras mayor velocidad de transmisión de datos mayor será la influencia en su selección.

- **Consumo energético:** Este criterio evalúa el consumo del dispositivo maestro en estado inactivo y en ejecución. Se establece que un dispositivo de menor consumo tendrá mayor influencia en la selección.
- **Mantenimiento:** Este criterio se enfoca a la capacidad del dispositivo para recibir mantenimiento de forma remota, ya sea actualización de código o manejo de archivos. El dispositivo que puede comunicarse de forma remota o sencilla tiene mayor influencia en la selección.

Una vez realizada la ponderación, según la importancia de cada criterio, se puede observar la Tabla 2.1:

Tabla 2.1 Ponderación de criterios

Criterio	Precio	Transmisión	Consumo	Mantenimiento	ponderación
Precio		0,5	0,5	0	0,2
Transmisión	0,5		0	0	0,15
Consumo	0,5	1		0	0,25
Mantenimiento	1	1	1		0,4
				Suma	1

Según la Tabla 2.1 se define el orden que tendrán los criterios, y es el siguiente: Mantenimiento, Consumo, Precio, Transmisión.

Después se realizó la evaluación de cada criterio con las alternativas o soluciones previamente descritas. Para ello se desarrollaron cuatro tablas (Tabla A.1 - Tabla A.4) que sirvieron para encontrar la relación que tiene cada criterio con las soluciones correspondientes. Luego se resume en la Tabla 2.2, por cada alternativa de solución los pesos previamente calculados correspondientes a los criterios de aceptación. En la columna final se establece la prioridad que tiene cada solución usando de referencia la suma de todos sus pesos.

Tabla 2.2 Prioridad de alternativas

Conclusión	Precio	Transmisión	Consumo	Mantenimiento	prioridad
Solución A	0,067	0,050	0,104	0,167	1
Solución B	0,100	0,025	0,104	0,133	2
Solución C	0,033	0,075	0,042	0,100	3

Referente a la Tabla 2.2 se observó que la solución A es la más factible para efectuar la implementación del proyecto.

2.3 Proceso de gestión del proyecto

Se utiliza la metodología ágil para llevar una correcta gestión del proyecto. La metodología ágil consta de hacer retroalimentación con el cliente o realizar reuniones semanales presentando los avances establecidos, en este caso se definió los siguientes sprint:

- **Sprint 1:** Establece como objetivo final obtener la propuesta de diseño seleccionado para la implementación del proyecto, incluyendo sus requisitos y requerimientos:
- **Sprint 2:** Evaluación semanal de las propuestas de diseño del escenario y portal web. Cuando la propuesta de diseño ya no requiera cambios, se haya presentado las limitaciones y la lista de materiales para su implementación se podrá continuar, caso contrario; se replantea los diseños.
- **Sprint 3:** Se analiza la implementación semanal según su tipo, ya sea en programación del portal web, servidor, base de datos o librerías reutilizables; también es evaluado el comportamiento del control PID según la planta.
- **Sprint 4:** Evaluación semanal y con su respectiva retroalimentación de la implementación con todas las secciones unidas para un diseño final.

Todas las tareas que se comprenden en cada sprint se encuentran en mayor detalle en la Figura 2.1, teniendo dentro del sprint uno tareas relacionadas al proceso de definición de la problemática y puntos centrales del proyecto, para luego pasar al sprint dos el cual contiene tareas relacionadas al diseño de la estructura del software y hardware de la solución, para a partir de allí pasar al sprint 3 en donde se empieza con las tareas relacionadas con la implementación de la solución, para finalmente en el sprint cuatro llegar a las pruebas de funcionalidad y corrección de errores que se pudieran identificar.

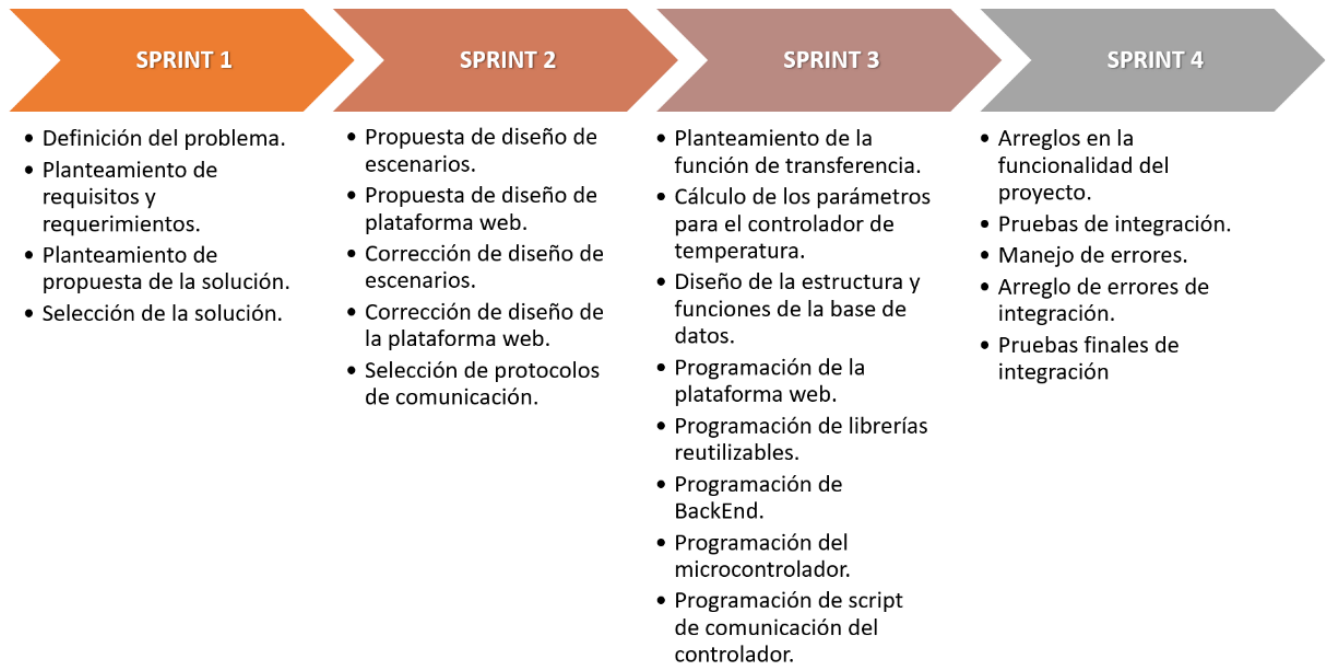


Figura 2.1 Proceso de gestión de proyecto basado en metodología ágil

2.4 Requerimientos del proyecto

Requerimientos funcionales de la plataforma web

La plataforma posee los siguientes requerimientos funcionales:

- Tener la información de los dispositivos IoT en tiempo real.
- Contar con un repositorio de librerías reutilizables que conecten con los dispositivos IoT.
- Permitir al usuario elegir un dispositivo IoT entre los que se encuentren registrados para su visualización y manipulación.
- Contar con una visualización de los dispositivos IoT en tiempo real por medio de una cámara web.
- Permitir al usuario realizar interacciones con el laboratorio remoto en tiempo real.
- Permitir al usuario administrador actualizar la información de las librerías en caso de que necesite.
- Mostrar una gráfica del comportamiento del control PID cuando el escenario esté en funcionamiento.

Requerimientos no funcionales

- El sistema debe ser accedido por estudiantes de ESPOL.

- No mostrar la interacción o cambios en el comportamiento con el laboratorio pasado los 30 segundos.
- La interacción al laboratorio enviada por la plataforma no cumpla con lo solicitado.

2.5 Alcances y limitaciones

El presente proyecto tiene por finalidad ayudar tanto al personal docente como al estudiante a lograr tener un mayor acercamiento a la automatización y dispositivos inteligentes, a través de una interfaz amigable que permita observar en tiempo real como se ven y se configuran los dispositivos IoT, sin llegar a reemplazar en su totalidad el manejo directo de estos, esperando que la plataforma a desarrollar facilite a los estudiantes a acceder a los recursos del laboratorio sin la necesidad de invertir demasiado tiempo en movilización.

Una de las limitaciones a nivel de comunicación entre los dispositivos y la plataforma web, consiste principalmente en la compatibilidad entre el módulo a utilizar y el protocolo de comunicación a escoger debido a que existen microcontroladores que no aceptan ciertos protocolos por limitaciones en sus funcionalidades.

Adicionalmente la originalidad de estos que puede llegar a intervenir en la correcta interpretación del código utilizado para llegar a ésta, por lo que uno de los mayores retos de esta parte del proyecto está fuertemente ligado a la interoperabilidad.

Por otro lado, el medio de interacción con el laboratorio presenta varias desventajas, como es el poco dinamismo de los escenarios, es decir, para diversificar los escenarios es necesario que el administrador de la plataforma realice modificaciones sobre el mismo.

Además de que es una de las opciones de enseñanza que mayor cantidad de recursos físicos y monetarios conlleva, por lo que el presente proyecto presenta dos escenarios de microsistemas IoT que serán visualizados a través de una cámara web para que los estudiantes aprendan de IoT y a su vez evitar un alto consumo de los recursos limitados que se poseen.

Es necesario enfatizar, que el desarrollo de este producto se encuentra en su primera versión y por lo tanto presentará limitaciones en cuanto a los dispositivos disponibles, utilizando dos escenarios. Además, este puede acoplarse a cualquier red Wifi existente teniendo en cuenta de que es necesario realizar configuraciones en los dispositivos, principalmente en las credenciales de la red y la dirección IP del controlador.

2.6 Riesgos y beneficios de la solución

El producto final beneficia a los estudiantes politécnicos que cuenten con poco conocimiento en IoT y deseen aprender los conceptos fundamentales. Además, que favorece directamente a la institución al brindar un recurso para el aprendizaje autónomo de materias que integren el uso de dispositivos inteligentes. La plataforma podrá, por medio de la vista en tiempo real de los escenarios, dar indicios básicos de domótica, IoT, conectividad de circuitos y sensores, además que contará con recursos educativos universitarios que podrán ser accedidos en todo momento.

Al ser un laboratorio remoto, la plataforma corre el riesgo de presentar problemas de conectividad con los dispositivos por una sobrecarga de información, al estar constantemente recibiendo datos. El modelo actual del laboratorio cuenta con una sola cámara web conectada a dos escenarios diferentes, por ello solo podrá acceder un estudiante a la vez a los recursos para su interacción; lo cual traería problemas de tráfico en la plataforma.

En un trabajo futuro se podrían agregar más cámaras, controladores y escenarios para solucionar este inconveniente.

2.7 Beneficiarios de la solución

Se busca que los usuarios que utilicen la aplicación sean los miembros de la comunidad politécnica, tanto profesores como estudiantes, quienes tendrán acceso con diferentes perfiles en la plataforma.

Los perfiles disponibles serán:

Estudiante: Estudiantes universitarios de la ESPOL podrán iniciar sesión, ver librerías, escenarios y recursos educativos. Además, podrán ingresar a un escenario y manipular los sensores, escribir reseñas y comentarios.

Administrador: Profesores y administradores que podrán iniciar sesión, ver y actualizar el código de librerías y escenarios, crear recursos educativos y publicar los mismos.

2.8 Métricas de rendimiento

En la parte de la comunicación, se considera la latencia operativa, debido a que a través de esta se puede conocer cuánto se demora el servidor en realizar cada una de las operaciones como son las conexiones hacia las bases de datos, ejecución de la interfaz de usuario, conexión hacia el controlador, despliegue de servicios, entre otras.

Para la parte de la conexión entre el controlador y los nodos, se considera el ancho de banda al utilizar un protocolo de comunicación como es MQTT. Además, que se realizará pruebas de rendimiento en función a esta comunicación para saber si es capaz de soportar cambios constantes en las variables los nodos.

2.9 Métricas de satisfacción de usuario

Para analizar el punto de vista del usuario, se pretende utilizar encuestas que abarquen una muestra de al menos cincuenta usuarios, donde se puedan incluir observaciones para conocer los puntos débiles que reconocen en el proyecto. Además de conocer la disposición del usuario en recomendar la solución planteada, considerando el cincuenta por ciento de respuestas positivas como éxito.

Entre otros aspectos que se consideran, está la facilidad de uso de la plataforma por parte del usuario, para saber qué tan intuitiva puede ser la plataforma web. Adicional a ello, en la encuesta se incluirá varias preguntas relacionadas a la utilidad de la página, para saber si la solución propuesta logrará abarcar mayor número de población y la frecuencia de uso de esta por parte de los usuarios.

2.10 Prototipado de la plataforma web

Para el prototipado de la plataforma web se utilizó la herramienta Figma, donde se describen las pantallas que darán flujo al sistema y funciona como un Producto Mínimo Viable (MVP) para el cliente.

A continuación, se presenta algunas de las pantallas más relevantes que formarán parte de la aplicación. En Figura B.1 se puede observar el prototipo completo.

Inicio de sesión: El inicio de sesión está planteado para ser un simple ingreso de correo y contraseña como se puede observar en la Figura 2.2.

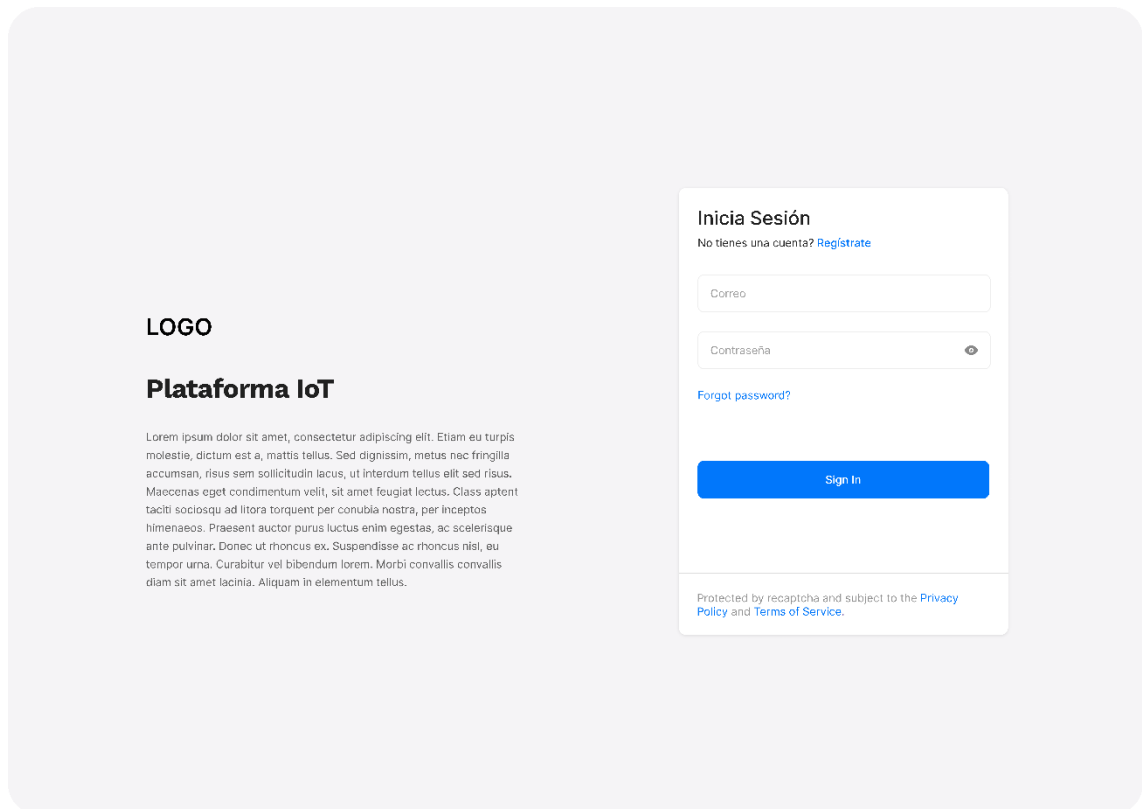


Figura 2.2 Prototipo de pantalla de inicio de sesión de la plataforma web

Dashboard: Es la primera pantalla a la que accederá el estudiante, como se ve en la Figura 2.3 cuenta con una barra lateral de opciones, como Escenarios, Librerías y Lecturas. Además, cuenta con algunas tarjetas con información adicional y de acceso rápido a otros recursos en la plataforma.

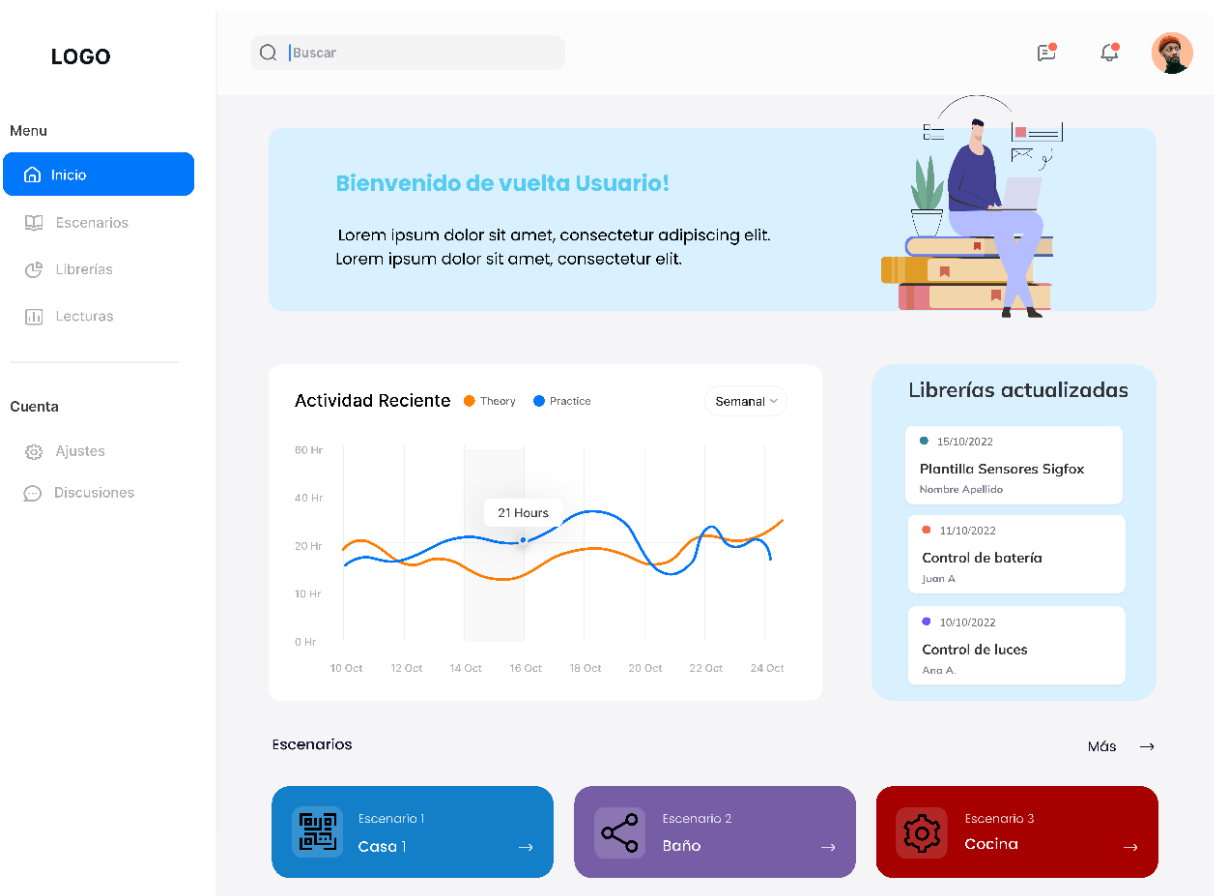


Figura 2.3 Prototipo de la pantalla de inicio de la plataforma web.

Escenarios: El usuario podrá visualizar una lista de todos los escenarios cargados en la plataforma, como está en la Figura 2.4. Lo que se visualizará es una versión compacta que muestra una imagen del escenario y el estado de la cámara. Además, se tendrá la opción de ver los escenarios revisados, que son aquellos a los que el usuario ya ha accedido previamente.

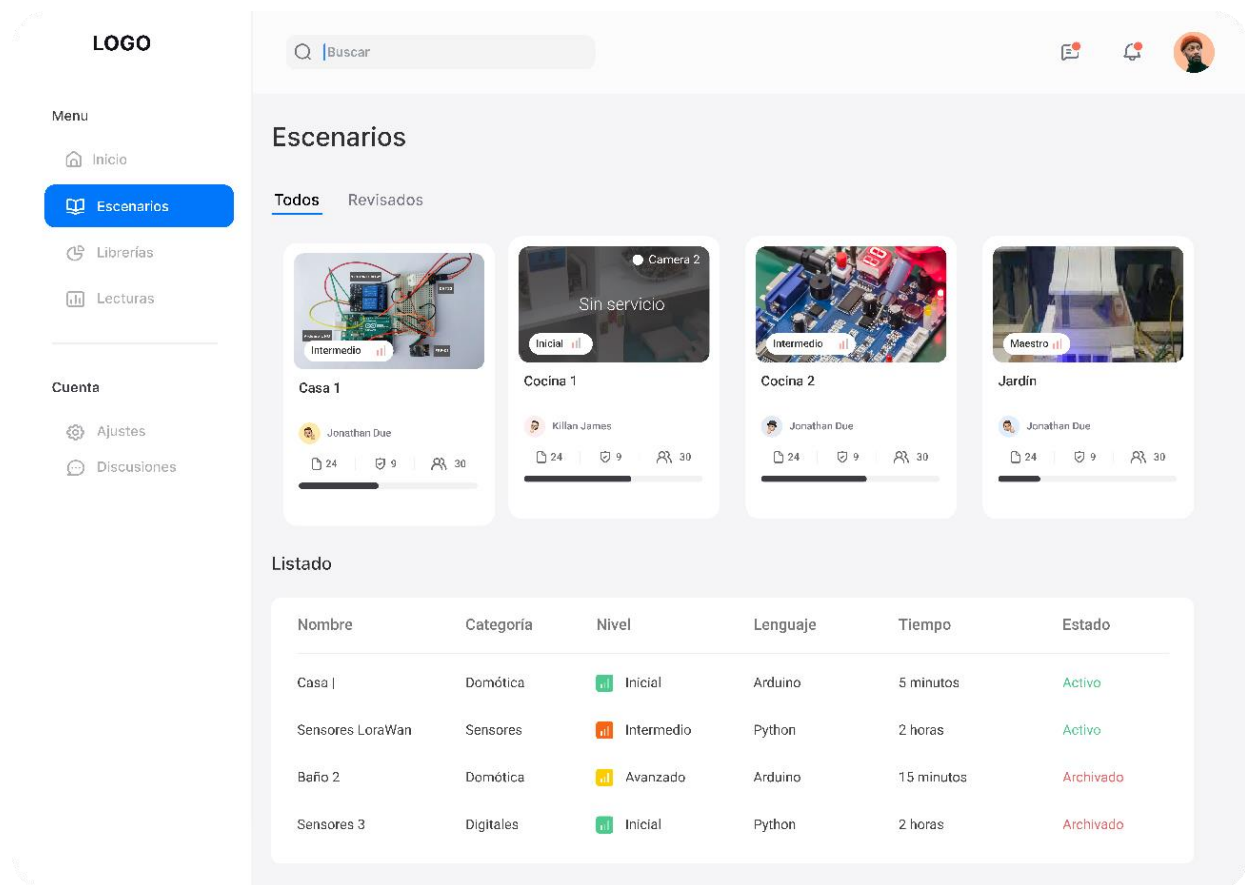


Figura 2.4 Prototipo de pantalla de escenarios de la plataforma web.

En la Figura 2.5 se muestra una cámara que brindará una transmisión en tiempo real del escenario seleccionado por el estudiante. El escenario también cuenta con un panel de control donde el estudiante tendrá la opción de enviar datos para monitorear el funcionamiento de los sensores, siempre y cuando sea el primero en acceder al escenario.

Debido a las limitaciones actuales, el estudiante solo puede ingresar a un escenario que esté enfocado por la cámara, aunque puede acceder a un escenario activo que ya esté en uso por otro estudiante; simplemente se activará el modo Observador, que tiene el panel de control bloqueado.

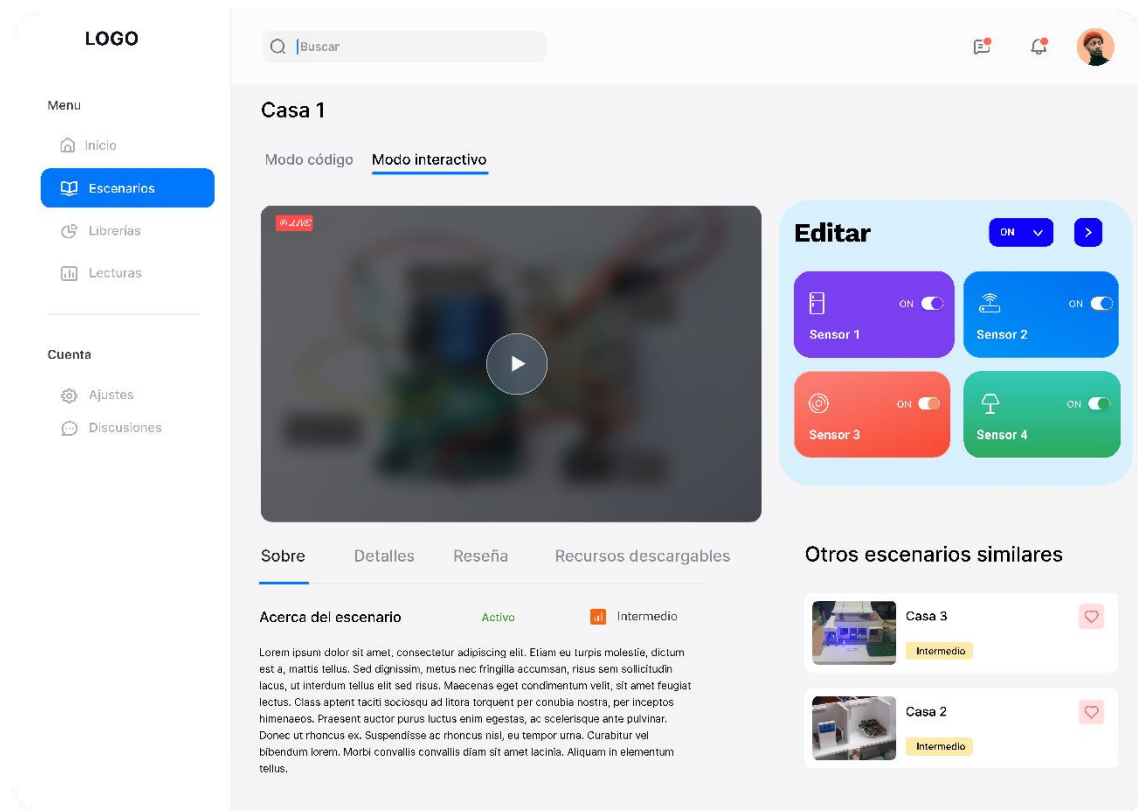


Figura 2.5 Prototipo de pantalla de escenario de la plataforma web.

Librerías: En esta opción los estudiantes podrán observar la lista de librerías cargadas en la base de datos, que están abiertas a descargas y reutilizaciones. En la vista individual se presentará el código de la librería, para visualización y descarga del mismo, pero los estudiantes que deseen pueden dejar comentarios y calificar el código.

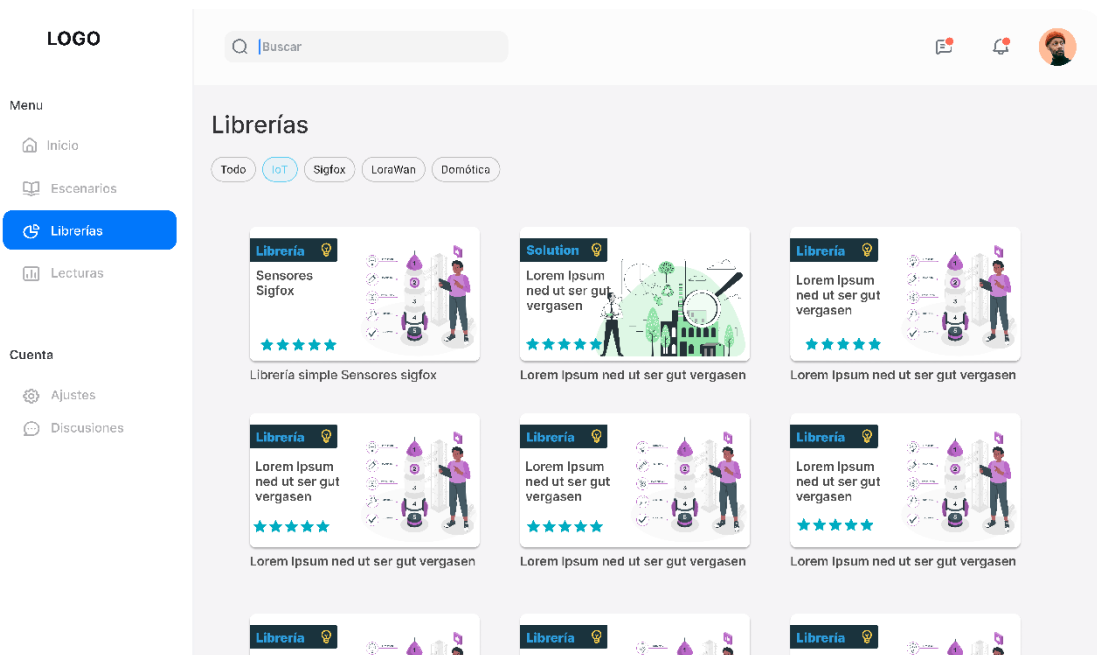


Figura 2.6 Prototipo de la pantalla librerías de la plataforma web.

Además, si se ha encontrado algún error pueden presentar un informe que será aprobado por los administradores, como se puede observar en la Figura 2.7.

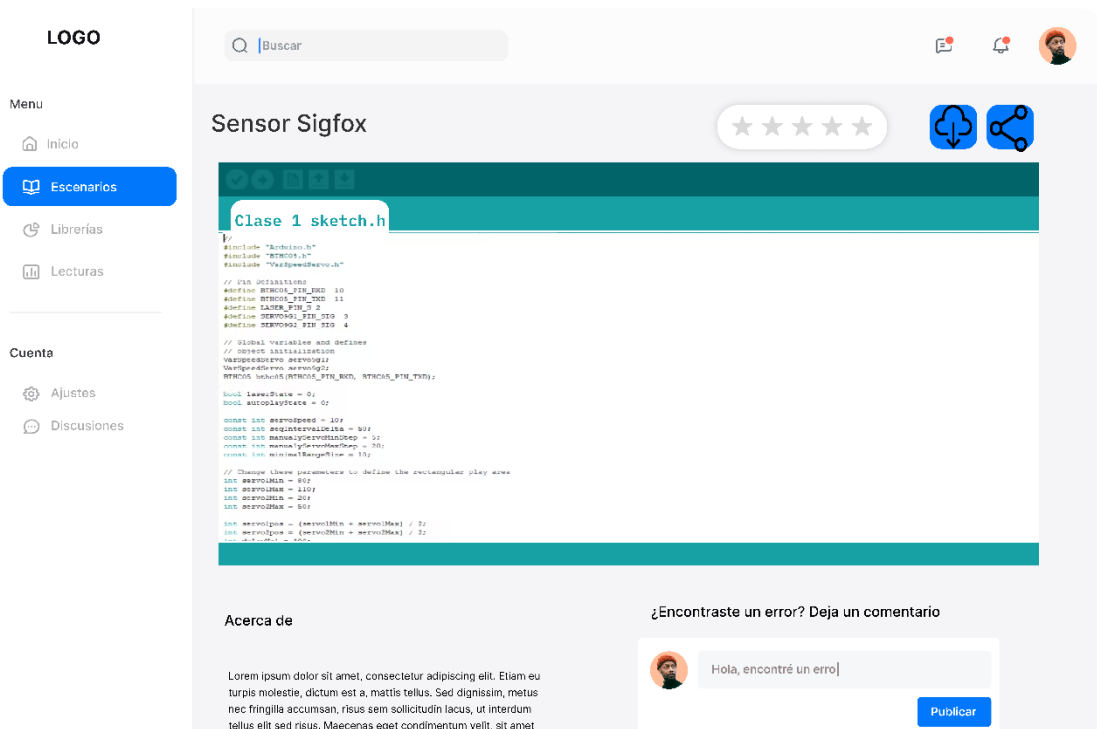


Figura 2.7 Prototipo de la pantalla de librería de la página web.

2.10.1 Evaluación de la solución

Para la evaluación se le presentó al cliente el prototipo para tener su retroalimentación activa, con ayuda de la herramienta Miro para poder obtener sus comentarios de forma remota. Tomando eso en consideración, en la Figura 2.8 se muestra una tabla de la retrospectiva del MVP elaborado en Figma. Toda esta retroalimentación se tomará en cuenta para el desarrollo de la solución final.

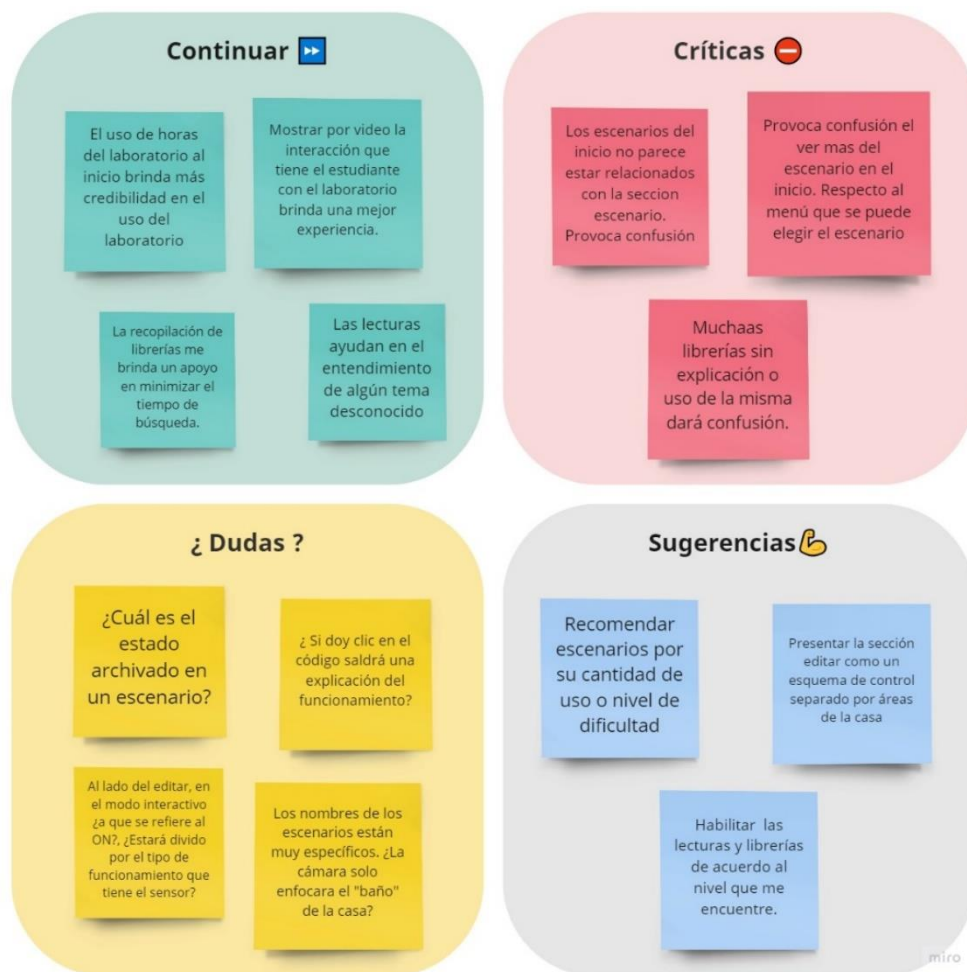


Figura 2.8 Retrospectiva del MVP (Figma)

2.11 Diseño de la solución

A continuación, se muestran varios diagramas para brindar contexto y entender la solución planteada, indicando las condiciones que realizan los actores del problema, que son el administrador y el estudiante.

2.11.1 Diagrama de despliegue

En el diagrama de despliegue se muestra la arquitectura de ejecución del sistema, describiendo los protocolos de comunicación entre cada objeto y los lenguajes que se utilizarán. En la Figura 2.9 se observa como el usuario se conecta directo a la plataforma web, donde podrá observar la página y realizar funciones como ver escenarios, librerías, enviar datos y crear comentarios. Toda esa información pasa directo al servidor, que estará realizado en Python y tiene conexión con la base de datos. El sistema hace una lectura y escritura de la base de datos para atender el requerimiento del usuario y, dependiendo del tipo de requerimiento, envía los datos al laboratorio remoto.

Profundizando en los protocolos que se encuentran implicados en cada una de las fases de comunicación, se tiene en primer lugar a HTTP, el cual es el encargado de establecer la comunicación entre el web browser con el servidor web, trabajando a través del puerto 8080.

Otro de los protocolos que se ha considerado es UART, el cual hace uso de los puertos seriales del controlador, ya que se consideró una conexión directa entre la Raspberry y la cámara que mostrará los escenarios IoT.

Se tiene también a MQTT, el cual es el protocolo más utilizado para la comunicación pub/sub entre dispositivos IoT a través del puerto 1883, para este caso se lo considera para establecer comunicación entre el controlador y los nodos.

Adicional a esto, los sensores se encontrarán conectados de manera directa a los nodos a través de los pines GPIO de la placa ESP32.

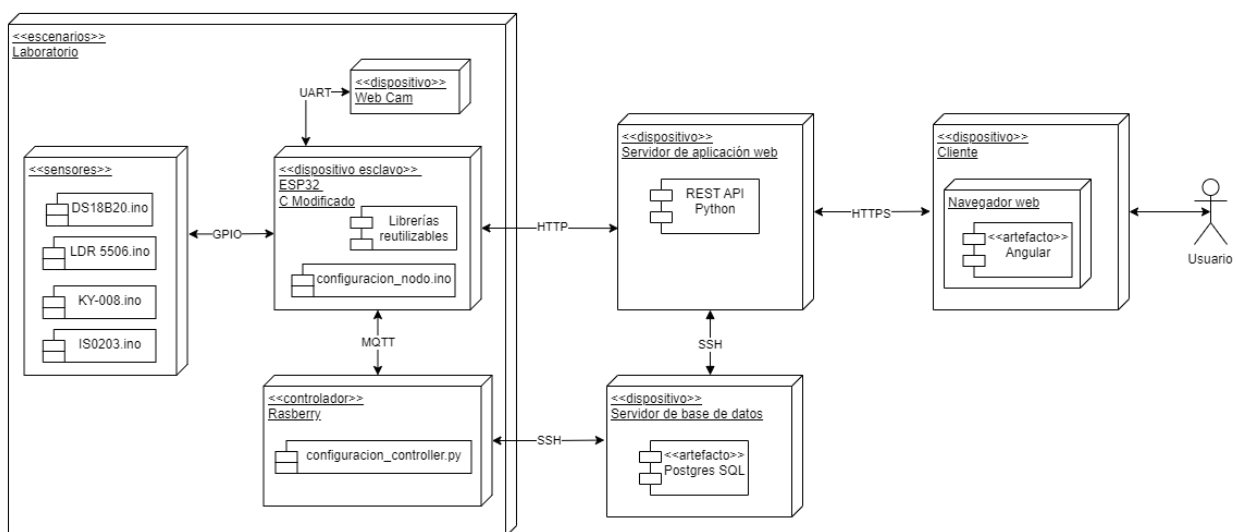


Figura 2.9 Diagrama de despliegue

2.11.2 Hardware del proyecto

Para la creación del laboratorio virtual se necesitan ciertos componentes de hardware que tendrán comunicación con la plataforma virtual. En Tabla C.1, se presenta la lista de los componentes a utilizar en el proyecto con su respectivo costo unitario.

2.11.3 Arquitectura del proyecto

Para la arquitectura se ha realizado un plano de la infraestructura a desarrollar, tal como se puede observar en la Figura 2.10, que se estableció como MVP (mínimo producto viable) debido a las limitaciones y requerimientos previos. Para esta versión del proyecto existen 2 secciones:

La primera sección se enfoca en el uso de luz inteligente, seguridad y temperatura para un enfoque domótico dividido en pequeños cuartos. Para la segunda sección, escenario de control PID se pretende enseñar el monitoreo y control de temperatura usando un controlador.

El sistema cuenta con una sola cámara que estará en rotación dependiendo de los accesos a los escenarios cargados en la plataforma, debido a las limitaciones de hardware. Además, cuenta con una zona de control capaz enviar y recibir las instrucciones mediante peticiones web o por MQTT usando la Raspberry y efectuar su funcionamiento a la infraestructura de cada escenario usando los microcontroladores. Cabe destacar que, en la zona de control existe un microcontrolador que tiene como función el control del accionamiento de un foco Led para la simulación del día y noche para el escenario de casa inteligente.

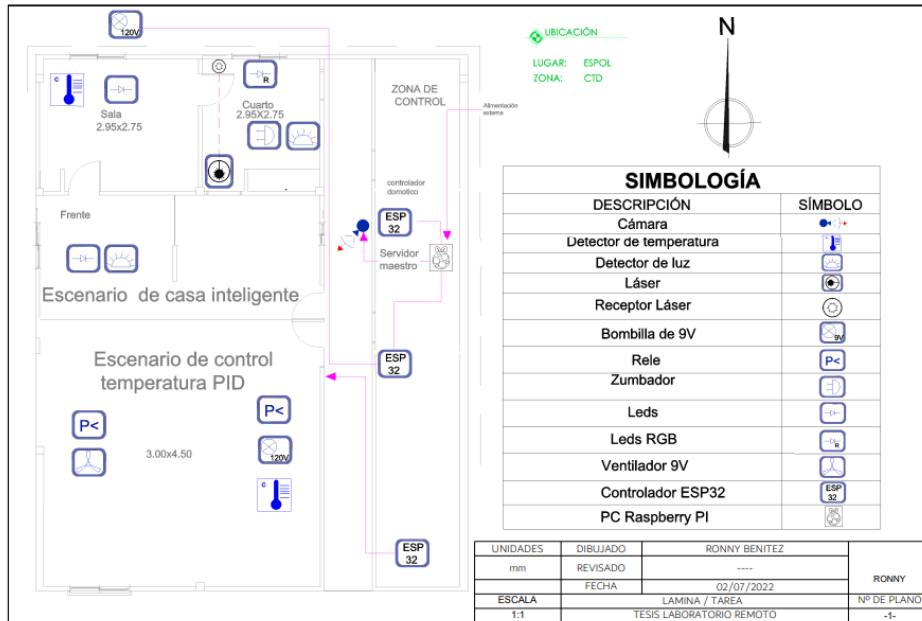


Figura 2.10 Plano arquitectónico

Se establece un diseño previo de la maqueta con su respectiva separación por escenario tal como se observó en el plano arquitectónico.



Figura 2.11 Diseño de maqueta en 3D

En la Figura 2.12 se puede observar la arquitectura del proyecto, en donde se tiene un servidor web que se encontrará alojada en la nube, este es el punto central ya que este establece una comunicación entre el cliente a través de codificación en lenguaje Python. Adicional a esto, el servidor se comunica con la base de datos, en la cual, se almacenará información referente a los usuarios que se registren en la plataforma.

Además, mantiene comunicación con el hardware a través del controlador para poder llegar hacia los nodos. El controlador es el encargado de obtener la información desde la base de datos y enviarla hacia los nodos ESP por medio de una comunicación MQTT seccionada por tópicos, y dependiendo de las acciones hechas por el usuario en la plataforma se accionan los dispositivos IoT.

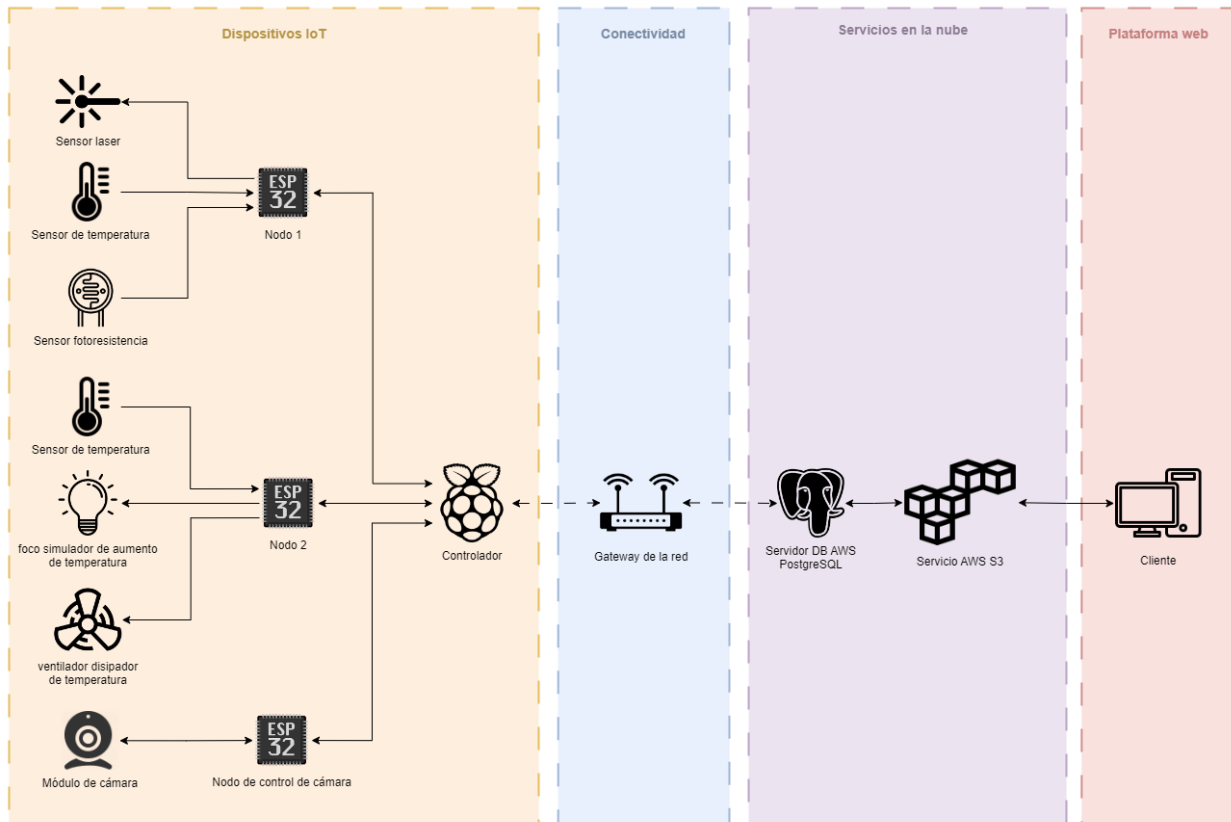


Figura 2.12 Diagrama de arquitectura del proyecto

2.11.4 Diagrama esquemático o circuito

Para la creación de los escenarios tanto para un sistema de casa inteligente y para la enseñanza de un control de temperatura usando control PID, se usa de apoyo un software para graficar circuitos usando microcontroladores y sensores.

La Figura 2.13 presenta las conexiones utilizadas para diseñar el sistema de luz inteligente, monitoreo de temperatura y seguridad inteligente para el escenario del sistema de la casa inteligente. Adicionalmente, se muestran las conexiones en un PCB para el funcionamiento de la infraestructura de este escenario como se observa en la Figura 2.14 y la pista para ser impresa en el Figura G.1.

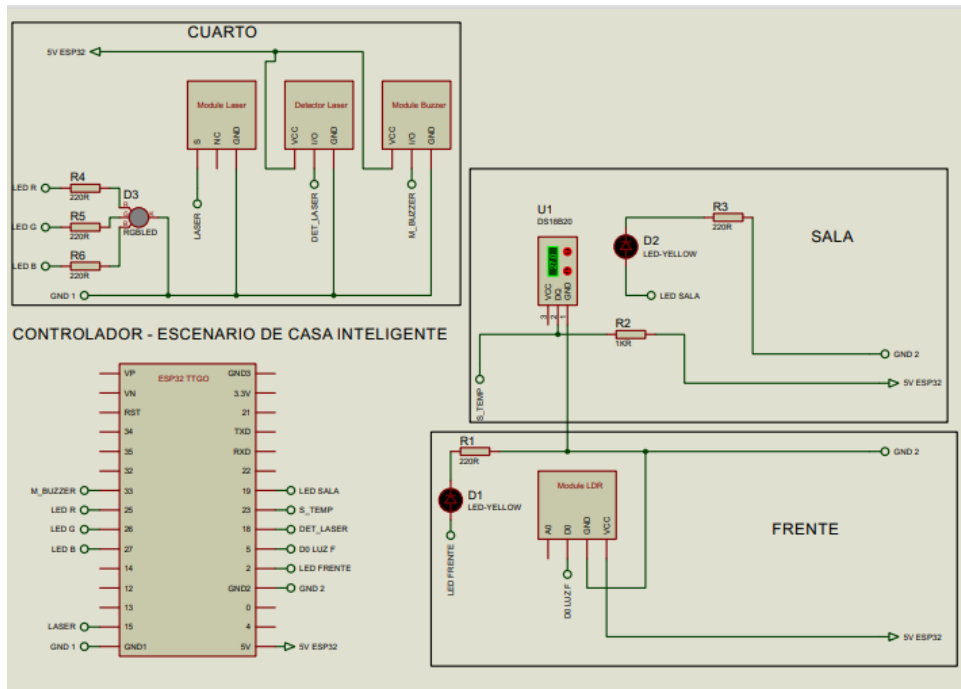


Figura 2.13 Diagrama de esquemático del escenario de casa inteligente

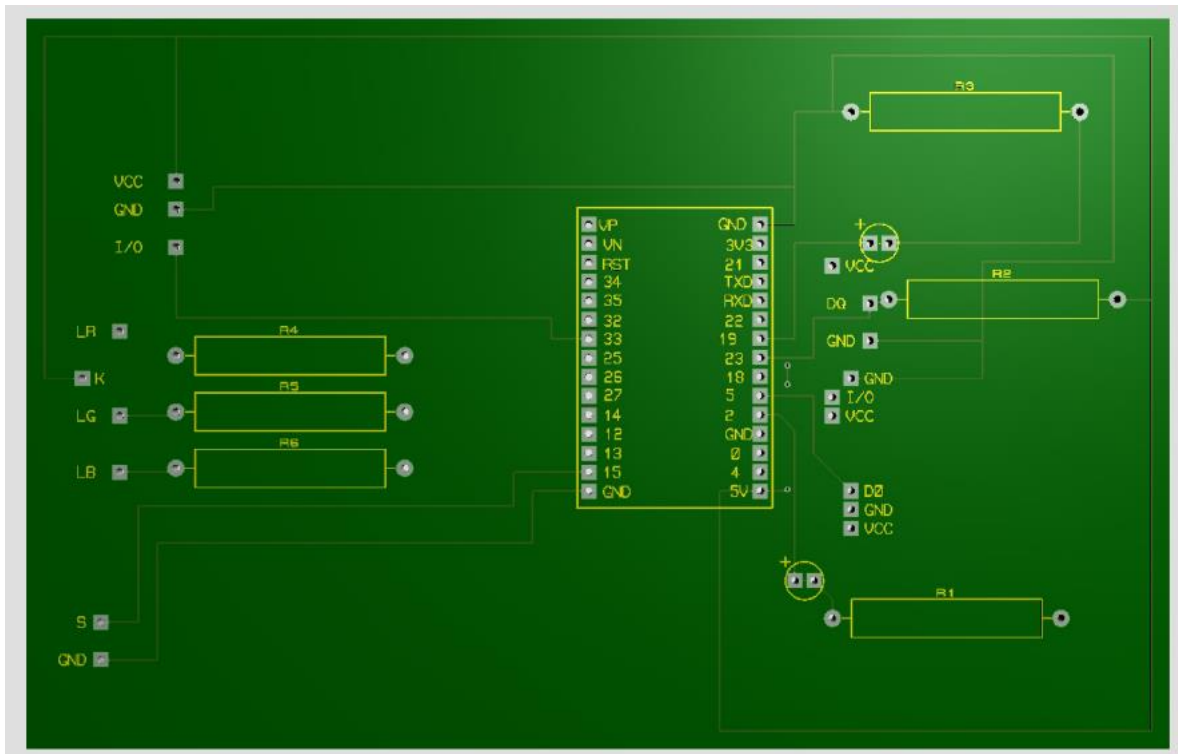


Figura 2.14 Vista 3D en un PCB del diagrama esquemático del escenario casa inteligente.

En la Figura 2.15 se presentan las conexiones utilizadas para el escenario de control de temperatura usando un controlador PID usando como fuente de calor una bombilla incandescente de 120 voltios. Adicionalmente, se muestran las conexiones en

un PCB para el funcionamiento de la infraestructura de este escenario como se observa en la Figura 2.16 y la pista para ser impresa en el Figura G.2.

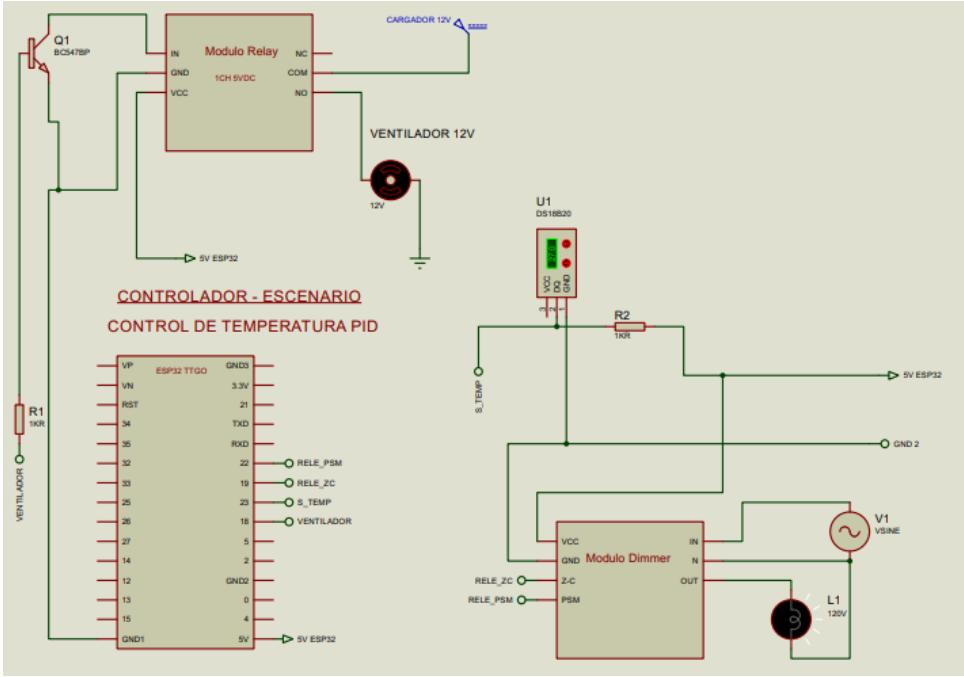


Figura 2.15 Diagrama de esquemático del escenario de control de temperatura PID

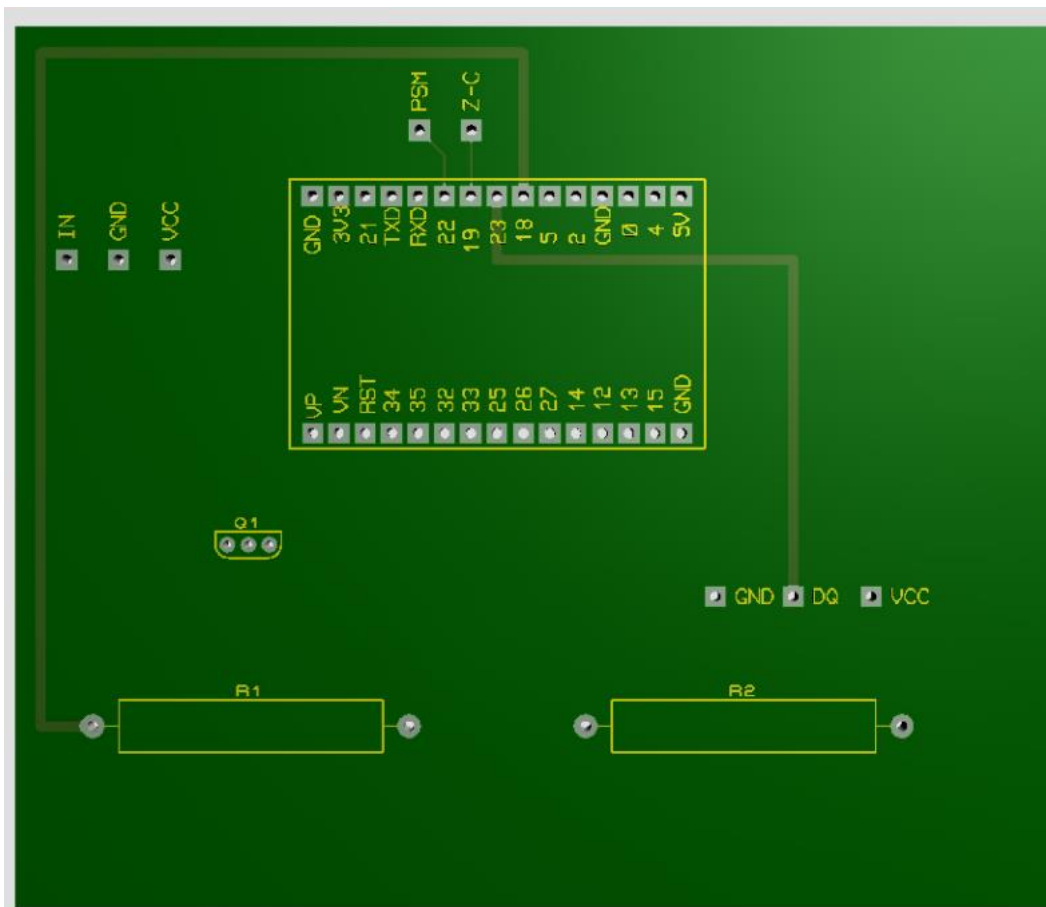


Figura 2.16 Vista 3D en un PCB del diagrama esquemático del escenario control de temperatura PID.

En la Figura 2.17 se presentan las conexiones utilizadas para llevar el control del accionamiento de un foco led que se encuentra en la parte superior del escenario domótico para la simulación del día y la noche. Adicionalmente, se muestran las conexiones en un PCB para el funcionamiento de la infraestructura de este escenario como se observa en la Figura 2.18 y la pista para ser impresa en el Figura G.3.

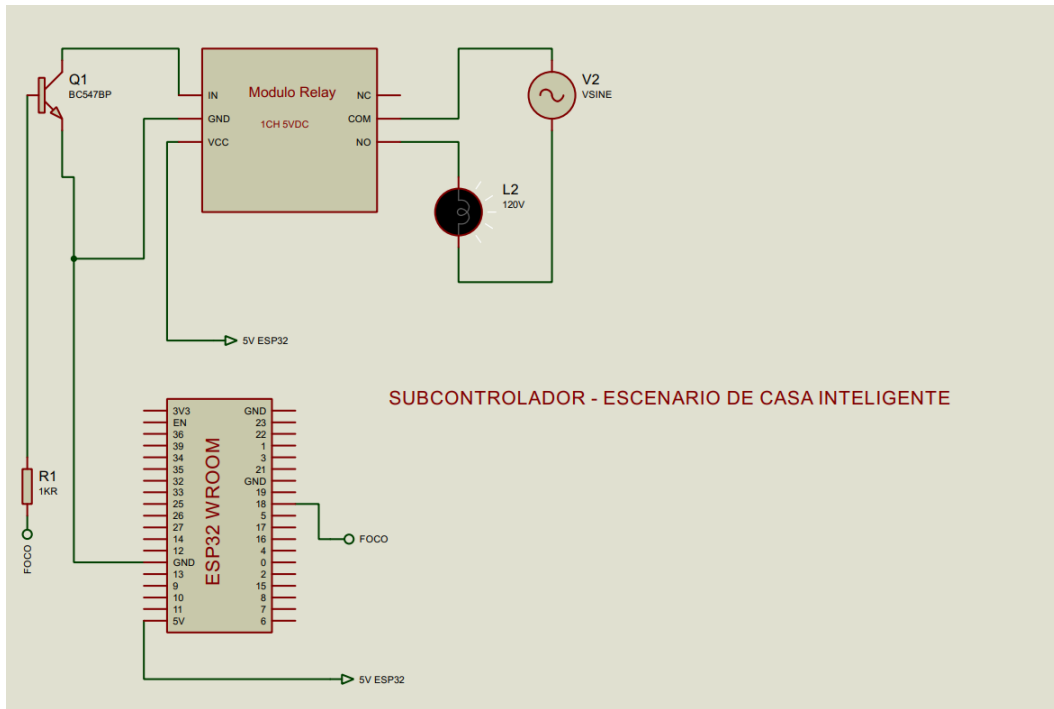


Figura 2.17 Diagrama de esquemático de subcontrolador del escenario de casa inteligente.

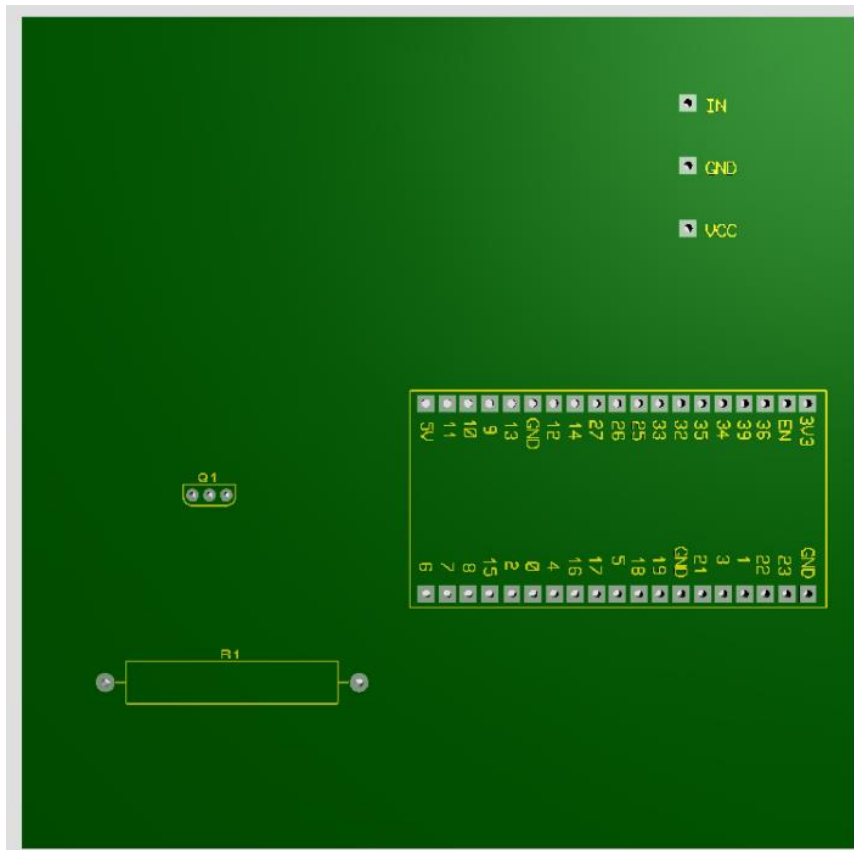


Figura 2.18 Vista 3D en un PCB del diagrama esquemático del subcontrolador del escenario de casa inteligente

2.11.5 Software

Para la creación de la plataforma web se ha planteado el uso de Angular para el FrontEnd y Django-Python para el servicio web del BackEnd, debido a que Python facilita la integración con el laboratorio remoto.

La comunicación entre el FrontEnd y el BackEnd se realizará a través de APIs. Para almacenar la información se utilizará una base de datos PostgreSQL. Además, el código de las librerías reutilizables será tanto en Python como en Arduino, dependiendo de las necesidades de cada escenario.

2.11.6 Casos de uso

Para los casos de uso se han especificado las funciones que podrán realizar en la plataforma tanto el estudiante como el administrador, detallando la interacción interna entre cada una.

El diagrama de la Figura 2.19 solo presenta información de la plataforma, más no del hardware general, ya que está mostrando todo lo que puede hacer el usuario.

Una vez inicia sesión, el usuario puede revisar y calificar las librerías, dejar reseñas, visualizar e ingresar a los escenarios propuestos y enviar datos a los dispositivos, además de ver e ingresar a recursos educativos.

Con respecto al administrador, este tiene funciones similares al usuario normal, a diferencia que este también es capaz de revisar las reseñas hechas por los usuarios, cargar y actualizar nuevas librerías y crear recursos educativos para los usuarios.



Figura 2.19 Diagrama de casos de uso

2.11.7 Diagrama de interacción

En la Figura 2.20 describe la interacción que tiene el usuario con los escenarios disponibles, incluyendo las funciones que puede realizar en ese proceso.

El usuario primero inicia sesión en la plataforma, donde podrá acceder a la lista de escenarios. Una vez escoja un escenario que le llame la atención, este podrá ingresar al mismo y a su vez se envía una petición de disponibilidad a la base de datos y al sistema.

Es importante recalcar que el controlador siempre estará recibiendo el estado de los escenarios de los nodos, por lo que, al momento de realizar la petición la respuesta debería llegar de forma casi instantánea. Si el escenario estuviese ocupado, se le indica al usuario que debe esperar hasta que se desocupe, pero caso contrario se podrá visualizar el escenario correctamente.

Es ahí donde el usuario puede decidir si enviar datos nuevos a los sensores, como apagar o prender Leds, y el controlador recibe esa petición que el usuario podrá ver reflejada en la cámara en tiempo real, en caso de no haber algún problema externo.

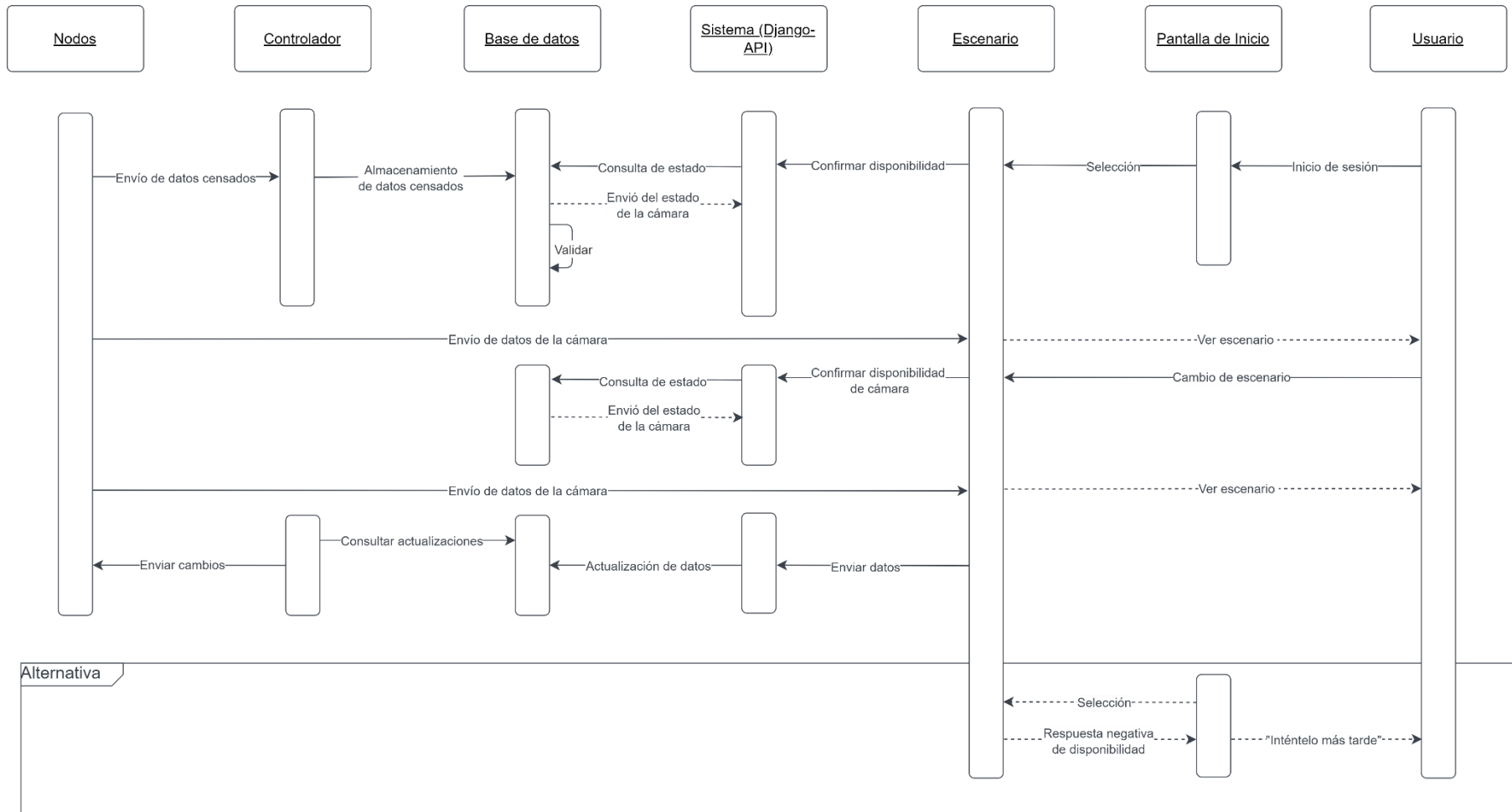


Figura 2.20 Diagrama de secuencia

2.11.8 Diseño de la base de datos

Para la base de datos se realizó el diseño entidad-relación (Figura 2.21), donde se muestra como los usuarios se relacionan entre sí y con los objetos de la plataforma, como los escenarios, librerías y recursos educativos. Además, también se ha diseñado la relación que tendrán en la base de datos los dispositivos de hardware.

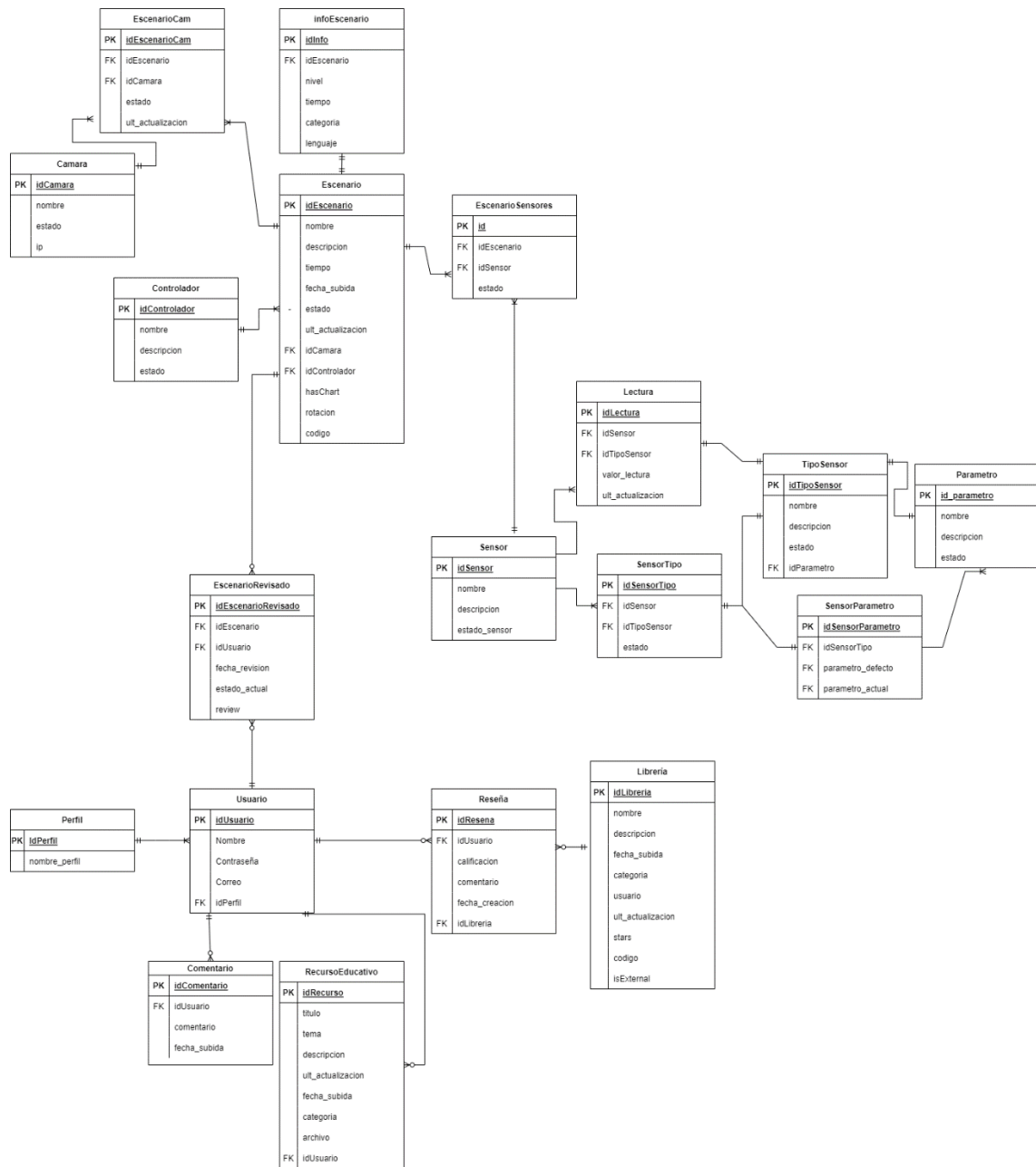


Figura 2.21 Diseño de la base de datos

2.12 Función de transferencia

En la Figura 2.22 se puede observar que debido a luz emitida por el foco existe una radiación térmica que es censada por el sensor de temperatura, esto a su vez es recopilado por el microcontrolador para así obtener valores de temperatura provenientes de la intensidad del foco.

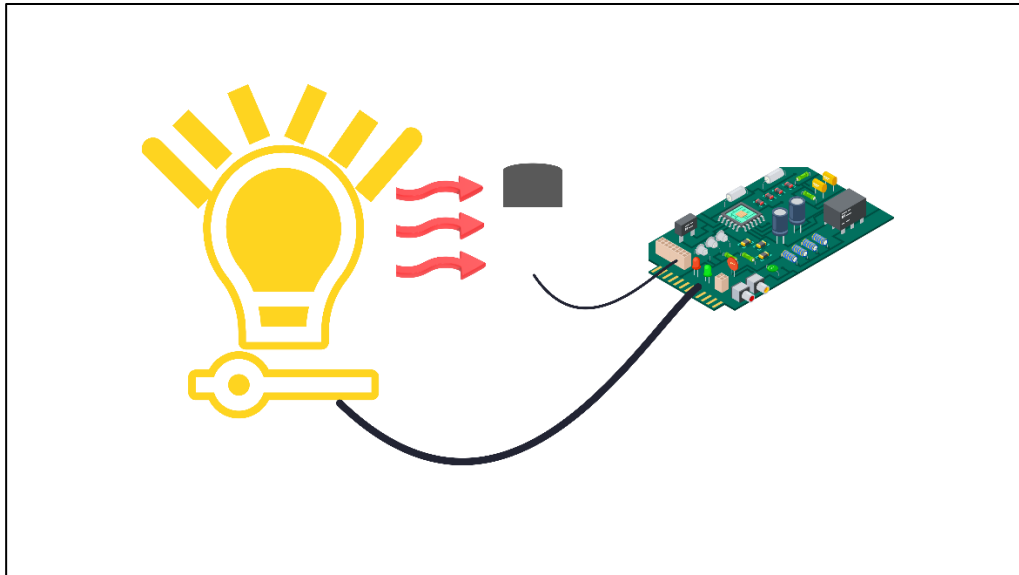


Figura 2.22 Diagrama simplificado del sistema a controlar

Para poder determinar la función de transferencia de calor que emite el foco se tomará datos de forma experimental por 5 minutos cada segundo usando como base la infraestructura del control de temperatura PID (Figura F.3), con los siguientes parámetros:

Potencia del ventilador: 0%

Potencia de la bombilla: 100%

Una vez ya terminado el experimento se procede a la recopilación de datos e ingresarlos a una función llamada `systemIdentification`; permite realizar la identificación de un sistema desconocido a un modelo definido por el usuario [52], que viene incorporada en Matlab.

Se establece como parámetro de entrada una función escalón y de salida los valores de muestreo (temperatura) obtenido previamente. Luego de establecer los polos y ceros se obtiene la función de transferencia de la planta como se muestra en la siguiente ecuación:

$$\frac{11.04s + 0.1998}{s^2 + 0.1873s + 0.00271} \quad (3)$$

Finalmente, se puede modificar los parámetros para encontrar una nueva función de transferencia con una exactitud deseable. En la Figura 2.23 muestra una exactitud de 92.23% de semejanza que tiene la función de transferencia encontrada (3) respecto a la función de entrada experimental. Como punto importante, la función resultante es la **tf3** o la gráfica de color rojo y la de color negro es la gráfica extraída de los datos experimentales.

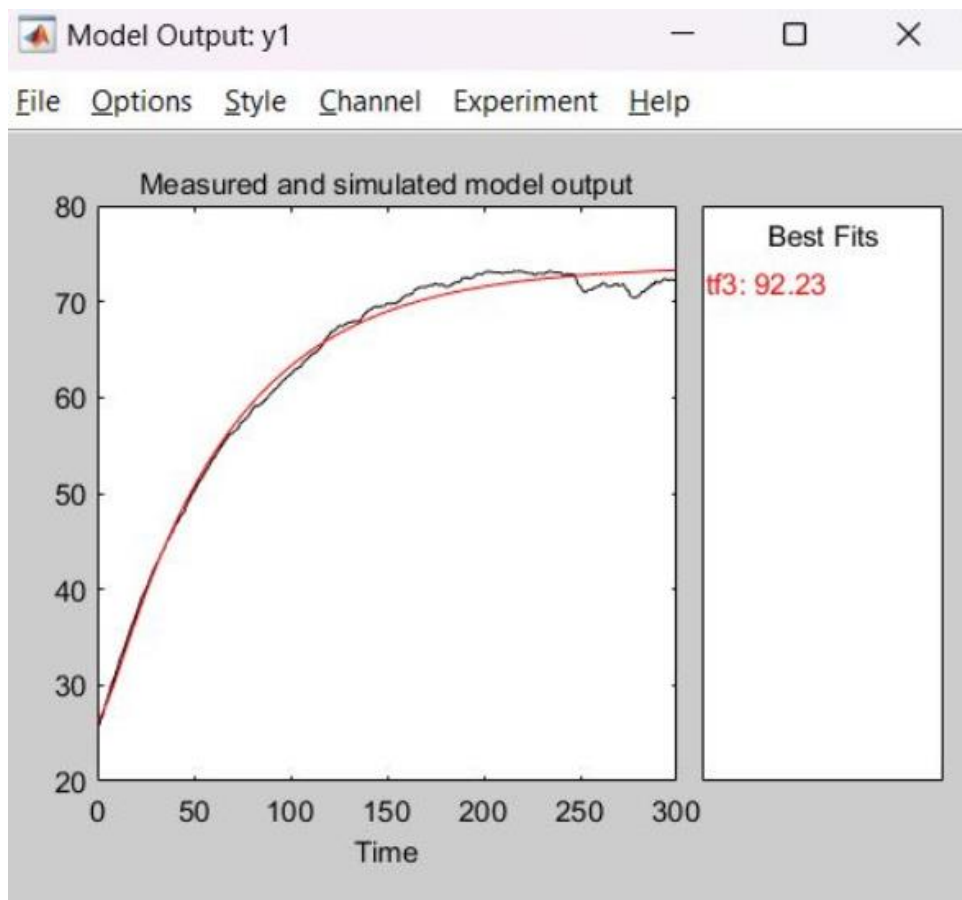


Figura 2.23 Modelo de salida simulada vs el modelo experimental.

2.13 Constantes de control

En la sección 2.12 se hizo un experimento para obtener la función de transferencia de calor de forma experimental recopilando datos de temperatura emitido por el foco durante 5 minutos censado por el sensor de temperatura. Luego para definir las constantes del control PID para estabilizar el sistema se usó una herramienta de Matlab llamada *Pid Tuner*, para posteriormente ser usadas como parámetros de control

preestablecidos en el modo automático del escenario de control de temperatura PID. El resultado fue que la constante proporcional es de 0.020703, la derivativa es de 0.062518 y la integral es de 0.00001 con un tiempo de estabilización de 123 segundos como se visualiza en la Figura 2.24.

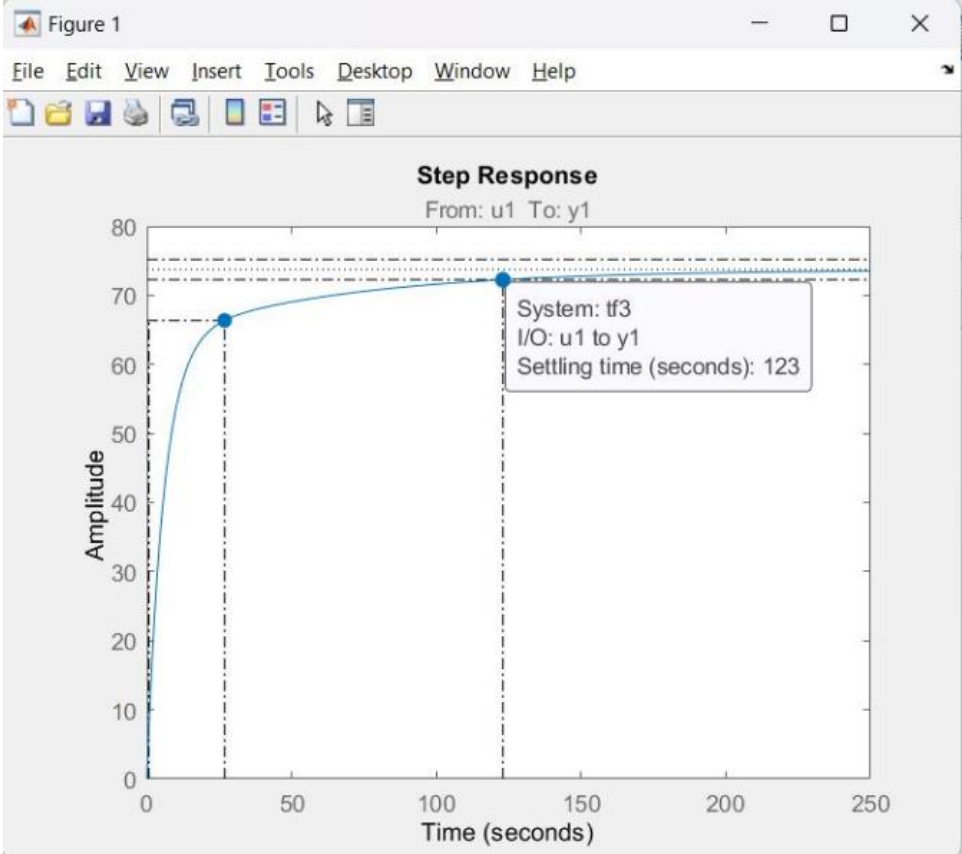


Figura 2.24 Tiempo de estabilización del modelo simulado.

CAPÍTULO 3

3. RESULTADOS Y ANÁLISIS

3.1 Análisis

Durante la ejecución del proyecto, se realizaron una serie de actividades que contribuyeron al desarrollo de la plataforma educativa. Estas actividades estuvieron enfocadas en satisfacer las necesidades y requerimientos específicos del cliente, garantizando que la plataforma cumpla con los objetivos y expectativas establecidos. El análisis detalla cada una de estas actividades, permitiendo entender cómo se logró cumplir con los objetivos del proyecto y brindar una solución efectiva a los requerimientos establecidos.

3.1.1 Desarrollo de la plataforma

Para el desarrollo de la plataforma web se dividió en 3 etapas: programación, lanzamiento a producción, y pruebas. En la fase de la programación, se mantuvo en su mayoría el flujo indicado en el prototipo, pero hubo algunos cambios significativos.

Debido a las limitaciones del proyecto, se implementaron en la plataforma dos escenarios: uno de casa inteligente y otro de control de temperatura; a los cuales se les estableció una restricción en la manipulación de los sensores, donde solo un estudiante podrá realizar cambios a los sensores a la vez, y si hay más estudiantes en el escenario, estos entran en Modo Observador, donde tendrán bloqueado el panel de control.

En los modelos de la base de datos, se ha añadido el campo "rotación" a la entidad "Escenario" para determinar el rango de rotación de la cámara. Debido a las limitaciones en hardware, solo se está utilizando una cámara para dos escenarios, por lo que es necesario conocer la cantidad de rotación que debe realizar la cámara para visualizar cada escenario adecuadamente. Además, se ha agregado un campo "dirección IP" a la entidad "Cámara", lo que permitirá transmitir los datos en tiempo real a un servidor y ser accedidos por la plataforma.

Otro cambio importante fue que se realizó un servicio de autenticación interno utilizando la base de datos en PostgreSQL y SimpleJWT en Django para manejar la autenticación por medio de tokens, por lo que se tuvo que realizar una pantalla de registro y no sacar la información de la base de datos de la universidad como estaba planeado.

El servidor seleccionado para la puesta a producción fue AWS, luego de una comparación que tomó en cuenta los límites económicos previamente establecidos, de lo que se tuvieron que hacer correcciones en los diagramas finales para adaptar esta tecnología.

La arquitectura fue cuidadosamente diseñada para aprovechar los servicios de alto rendimiento de la plataforma. Para el FrontEnd, se utilizó el servicio S3 de AWS, implementado con CloudFront para buscar una mejora en la velocidad y el rendimiento de la aplicación. Por otro lado, el backend del proyecto se implementó utilizando el servicio de ejecución de código sin servidor de AWS, Lambda, con la ayuda de la herramienta Zappa de Django.

Es importante destacar que, durante todo el proyecto, se mantuvo en la zona free tier de AWS, lo que significa que se aprovecharon los recursos gratuitos disponibles para alojar y ejecutar la plataforma. Esta elección fue motivada por la necesidad de mantener bajos los costos del proyecto sin sacrificar su calidad ni rendimiento.

Los cambios mencionados fueron aceptados por el cliente y adaptados a los requerimientos establecidos. Adicionalmente, se incluyen en Figura E.1, los manuales de implementación y de usuario de la plataforma, que brindan detalles exhaustivos sobre cómo instalar y utilizar la misma de manera efectiva.

3.1.2 Módulo de librerías

Para la gestión de las librerías, se han dividido en dos categorías: las librerías externas y las librerías de usuario. Para las librerías externas, se ha incluido un enlace al repositorio donde se encuentra la información oficial de la librería. Por otro lado, las librerías de usuario se han desarrollado inicialmente en C++ para su posterior uso en los escenarios y se da la opción para que administradores ingresen más. Ambos tipos de librerías están accesibles a través de la plataforma y pueden ser visualizadas en las Figura 3.1 y Figura 3.2.

El módulo de librerías presenta una vista general de todas las librerías disponibles, que pueden ser filtradas por categorías y tienen una imagen de portada para su identificación. Para la vista individual, se ha utilizado el paquete Highlightjs en Angular para visualizar el código en la plataforma web. Además, está disponible una sección de información sobre la librería y otra de comentarios para que los usuarios puedan dejar comentarios, sugerencias o informar sobre errores encontrados. Estas funcionalidades

permiten a los usuarios tener un control más eficiente y accesible sobre las librerías, lo que mejora la experiencia general en la plataforma.

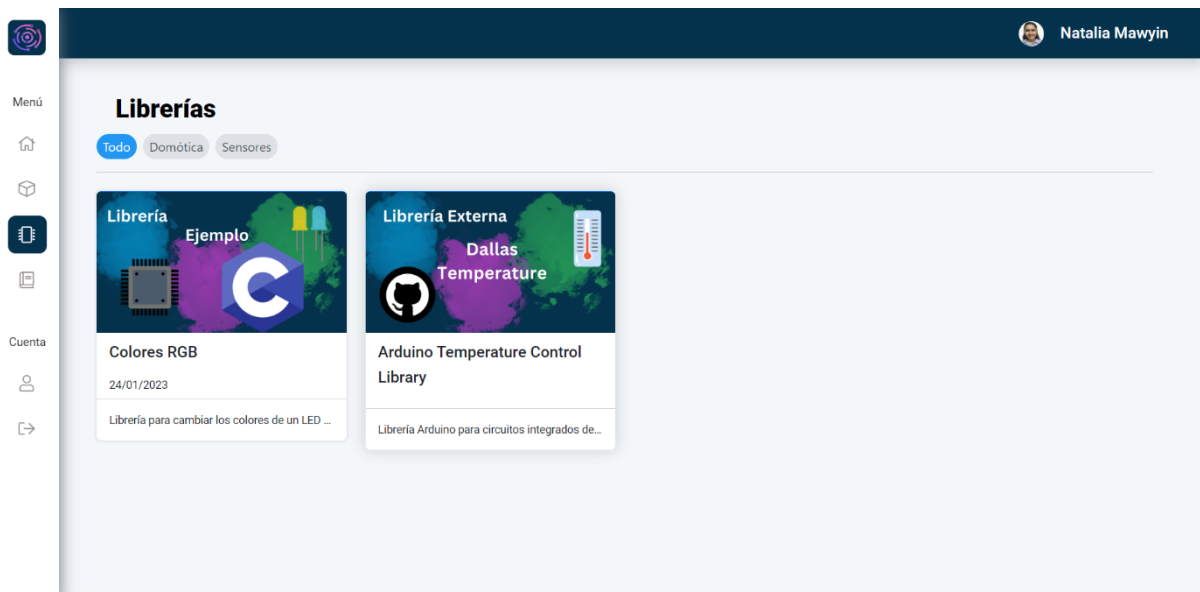


Figura 3.1 Vista de las librerías en la plataforma

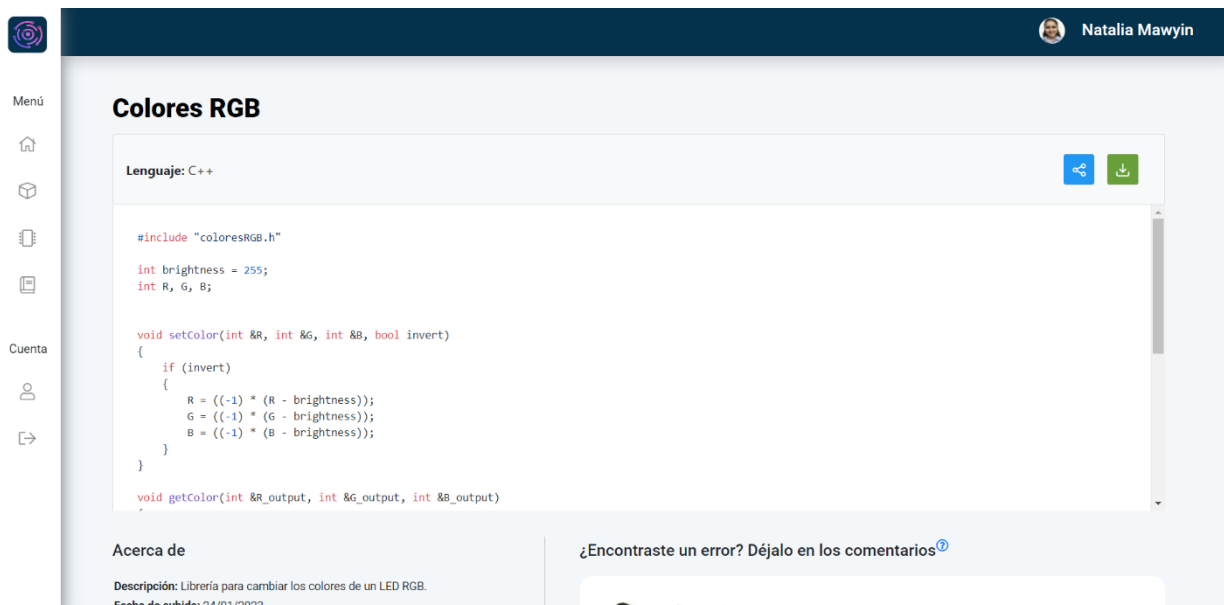


Figura 3.2 Vista de librerías Colores RGB

3.1.3 Administrador Django

Django Admin es un componente clave de la plataforma que permite a los usuarios de nivel administrativo acceder a la configuración general. Con esta herramienta, los administradores pueden generar nuevos escenarios, librerías, recursos educativos,

sensores, entre otros, en función de las necesidades específicas que vayan surgiendo. La Figura 3.3 muestra una vista previa del servidor, que permite una gestión eficiente de la plataforma. La flexibilidad y la capacidad de adaptación de la plataforma se ven reforzadas por la presencia de Django Admin, lo que permite a los administradores satisfacer sus requisitos específicos. Además, la interfaz intuitiva y fácil de usar de este servidor asegura que la administración de la plataforma sea rápida y sencilla para todos los usuarios.

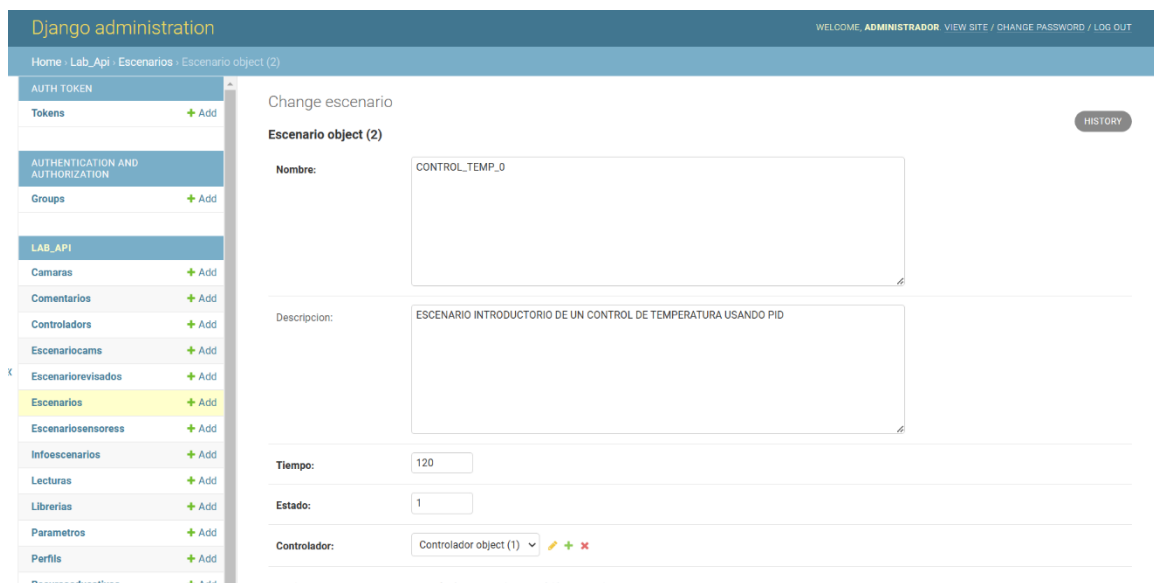


Figura 3.3 Vista de servidor web

3.1.4 Plan de implementación

Se presenta un resumen del plan de implementación establecido. La planificación fue realizada al inicio del proyecto, pero tuvo que ser adaptada a lo largo del desarrollo del proyecto. Además, como se utilizó una metodología Scrumban, se separó la revisión de los resultados en módulos:

Tabla 3.1 Planificación del proyecto

No.	Descripción	Fecha inicio	Fecha fin	Participantes
1	Documentación	12/10/2022	12/12/2022	Computación, Mecatrónica y Telemática
2	Recopilación de información acerca de protocolos de comunicación.	31/10/2022	7/11/2022	Telemática
3	Configuración de bróker MQTT en el controlador.	25/11/2022	27/11/2022	Telemática

4	Configuración de protocolos de comunicación en placas ESP.	28/11/2022	4/12/2022	Telemática
5	Configuración de comunicación MQTT en el controlador.	28/11/2022	4/12/2022	Telemática
6	Base de datos	1/12/2022	3/12/2022	Computación y Mecatrónica
7	Backend	3/12/2022	14/1/2023	Computación
8	FrontEnd	3/12/2022	14/1/2023	Computación
9	Diseño del código para el funcionamiento de los escenarios.	3/12/2022	14/1/2023	Mecatrónica
10	Comunicación entre el controlador y la base de datos.	12/12/2022	16/12/2022	Telemática
11	Diseño de la infraestructura de los escenarios	17/12/2022	20/1/2023	Mecatrónica
12	Unificación del Backend, FrontEnd, base de datos e infraestructura.	3/1/2023	14/1/2023	Computación, Mecatrónica y Telemática
13	Login/registro	9/1/2023	14/1/2023	Computación
14	Subida a producción	15/1/2023	20/1/2023	Computación
15	Pruebas	16/1/2023	20/1/2023	Computación, Mecatrónica y Telemática.

3.1.5 Implementación de infraestructura

La infraestructura se ha desarrollado usando como base el prototipo 3D (Figura 2.11) y la arquitectura del proyecto (Figura 2.10). En la infraestructura existe una separación que permite al estudiante tener una interacción y visualización independiente por cada escenario. En la Figura 3.4 se puede observar el escenario de control de temperatura PID y el exterior del escenario de la casa inteligente. Cabe destacar que debido a la separación que se tiene el interior del escenario de la casa inteligente es mostrado en el Figura F.1 y la zona de control en el Figura F.2.



Figura 3.4 Infraestructura del laboratorio remoto

3.2 Pruebas

3.2.1 Pruebas de la plataforma

La plataforma se ha desarrollado para la enseñanza y aprendizaje de los estudiantes en escenarios IoT, a través de los escenarios y librerías reutilizables. En la plataforma, se puede acceder a la vista de los sensores que tiene cada escenario. También se puede enviar datos en tiempo real haciendo uso de los protocolos de comunicación configurados en los dispositivos IoT y descargar las librerías reutilizables que están guardadas en el servidor. En la Figura 3.5 se puede observar la vista del login de la plataforma.

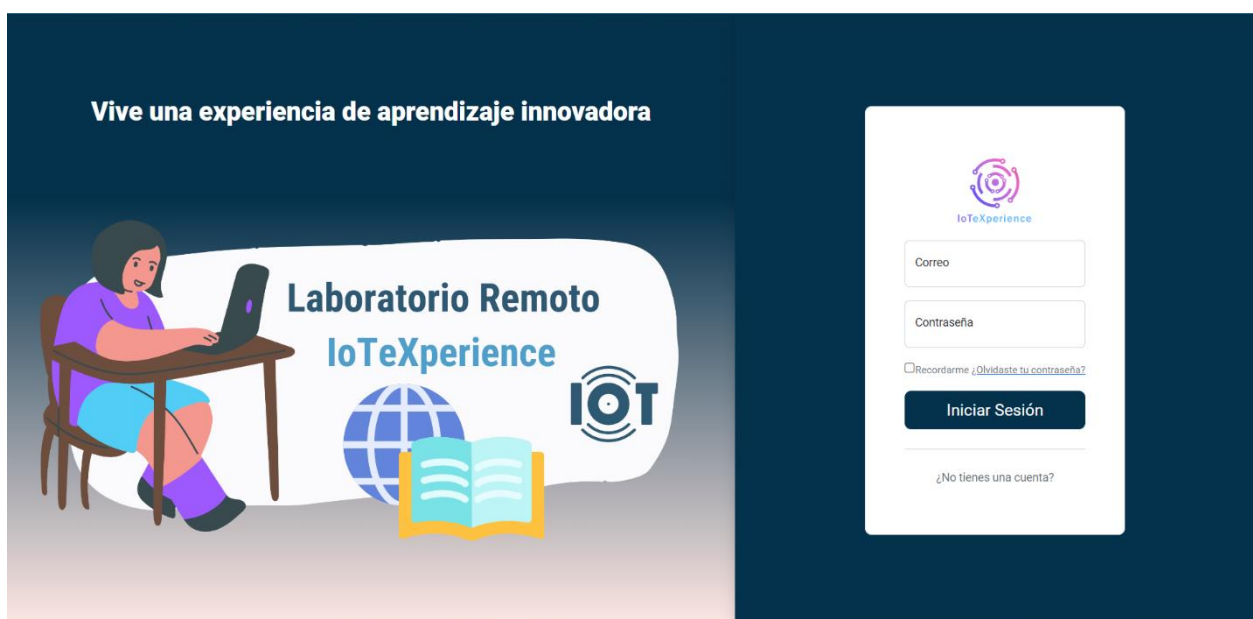


Figura 3.5 Pantalla de inicio de la plataforma

Las pruebas consistieron en ejecutar diferentes casos que miden el tiempo de demora de los endpoints, la latencia y el rendimiento de cada uno. A través de la ejecución de los diferentes casos de prueba, se recopiló información tanto cualitativa como cuantitativa que permitió evaluar el desempeño de la plataforma, determinar la efectividad de sus procesos e identificar cualquier error o falla en su funcionamiento. De esta manera, se obtuvo una comprensión más profunda y detallada de su rendimiento.

3.2.1.1 Descripción del ambiente de pruebas

Las pruebas fueron hechas en una computadora con las siguientes especificaciones:

Ambiente de prueba:

- CPU= AMD Ryzen 7 4700U with Radeon Graphics 2.00 GHz

- RAM= 16,0 GB
- Sistema operativo= Microsoft Windows 10
- Internet= 45 Mbps

3.2.1.2 Pruebas de carga

Los casos de prueba presentados incluyen la evaluación de la conectividad de la plataforma con tres cantidades diferentes de usuarios, 50, 500 y 1000, respectivamente. Estas cantidades fueron seleccionadas con el fin de analizar la escalabilidad de la plataforma, ya que, como plataforma educativa, se espera que sea utilizada por grupos de estudiantes simultáneamente.

Las pruebas de carga se llevaron a cabo utilizando el software Apache Jmeter, que es una herramienta libre y de código abierto que permite realizar pruebas de carga y rendimiento de manera sencilla y efectiva.

Pruebas de carga con 50 usuarios

La prueba de conectividad de la plataforma se llevó a cabo con una muestra de 50 usuarios, como se muestra en la Figura 3.6. Durante las pruebas, no se observaron inconvenientes con ningún requerimiento ni problemas relacionados con la capacidad de la plataforma para manejar la cantidad de usuarios.

Sin embargo, se notó una latencia intermitente en los requerimientos, con fluctuaciones en los tiempos de respuesta, donde el promedio fue de 388ms. A pesar de esto, todos los usuarios tuvieron acceso a la plataforma sin ningún problema y no se registraron solicitudes fallidas. Es importante destacar que, aunque la latencia puede ser un factor que considerar en el desempeño de la plataforma, el hecho de que todos los usuarios pudieron acceder sin problemas es un indicador positivo de su capacidad y estabilidad.

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(ms)
1	09:47:54.314	Thread Group 1-1	Home page	380	✔	11642	429	380	177
2	09:47:54.384	Thread Group 1-2	Home page	411	✔	11642	429	411	174
3	09:47:54.454	Thread Group 1-3	Home page	422	✔	11642	429	422	172
4	09:47:54.503	Thread Group 1-4	Home page	411	✔	11642	429	411	178
5	09:47:54.573	Thread Group 1-5	Home page	426	✔	11642	429	426	175
6	09:47:54.641	Thread Group 1-6	Home page	422	✔	11642	429	422	179
7	09:47:54.710	Thread Group 1-7	Home page	404	✔	11642	429	404	167
8	09:47:54.778	Thread Group 1-8	Home page	378	✔	11642	429	378	179
9	09:47:54.847	Thread Group 1-9	Home page	418	✔	11642	429	418	177
10	09:47:54.905	Thread Group 1-10	Home page	406	✔	11642	429	406	170
11	09:47:54.982	Thread Group 1-11	Home page	403	✔	11642	429	403	164
12	09:47:55.053	Thread Group 1-12	Home page	369	✔	11642	429	369	178
13	09:47:55.129	Thread Group 1-13	Home page	426	✔	11642	429	426	181
14	09:47:55.205	Thread Group 1-14	Home page	367	✔	11642	429	367	171
15	09:47:55.279	Thread Group 1-15	Home page	365	✔	11642	429	365	178
16	09:47:55.350	Thread Group 1-16	Home page	372	✔	11642	429	372	190
17	09:47:55.426	Thread Group 1-17	Home page	434	✔	11642	429	434	194
18	09:47:55.497	Thread Group 1-18	Home page	380	✔	11642	429	380	191
19	09:47:55.548	Thread Group 1-19	Home page	444	✔	11642	429	444	181
20	09:47:55.604	Thread Group 1-20	Home page	420	✔	11642	429	420	173
21	09:47:55.672	Thread Group 1-21	Home page	428	✔	11642	429	428	188
22	09:47:55.771	Thread Group 1-23	Home page	368	✔	11642	429	368	170
23	09:47:55.719	Thread Group 1-22	Home page	441	✔	11642	429	441	182
24	09:47:55.837	Thread Group 1-24	Home page	358	✔	11642	429	358	162
25	09:47:55.890	Thread Group 1-25	Home page	430	✔	11642	429	430	176
26	09:47:55.968	Thread Group 1-26	Home page	378	✔	11642	429	378	182

Scroll automatically?
 Child samples?
No of Samples: 50
Latest Sample: 337
Average: 398
Deviation: 33

Figura 3.6 Pruebas de conexión con 50 usuarios

Pruebas de carga con 500 usuarios

La prueba de carga de la plataforma con 500 usuarios de la Figura 3.7 resultó en un desempeño sólido y estable. Durante el ensayo, la conexión no se vio afectada en ningún momento y todos los usuarios pudieron acceder a la plataforma sin interrupciones. Sin embargo, se identificó un aumento en la latencia por cada requerimiento, lo que resultó en fluctuaciones en los tiempos de respuesta. Estos resultados se encuentran en línea con los resultados de la prueba con 50 usuarios.

Además, la prueba con 500 usuarios es significativamente más exigente que la primera, y el hecho de que la plataforma haya pasado con éxito esta prueba muestra su potencial para manejar una gran cantidad de tráfico de manera eficiente.

Es importante continuar monitoreando y mejorando la latencia para garantizar una experiencia óptima para los usuarios, lo que se puede hacer con soluciones de monitoreo existentes, como CloudWatch de AWS que permite supervisar y recopilar datos de rendimiento de la plataforma en tiempo real.

Sample # ↑	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(ms)
1	09:50:13.502	Thread Group 1-2	Home page	352	✔	11642	429	352	196
2	09:50:13.568	Thread Group 1-3	Home page	413	✔	11642	429	413	169
3	09:50:13.436	Thread Group 1-1	Home page	563	✔	11642	429	563	261
4	09:50:13.619	Thread Group 1-4	Home page	406	✔	11642	429	406	162
5	09:50:13.684	Thread Group 1-5	Home page	404	✔	11642	429	404	169
6	09:50:13.751	Thread Group 1-6	Home page	439	✔	11642	429	439	178
7	09:50:13.815	Thread Group 1-7	Home page	410	✔	11642	429	410	169
8	09:50:13.920	Thread Group 1-9	Home page	421	✔	11642	429	421	167
9	09:50:13.985	Thread Group 1-10	Home page	359	✔	11642	429	359	171
10	09:50:14.058	Thread Group 1-11	Home page	400	✔	11642	429	400	164
11	09:50:13.868	Thread Group 1-8	Home page	613	✔	11642	429	613	170
12	09:50:14.126	Thread Group 1-12	Home page	423	✔	11642	429	423	174
13	09:50:14.195	Thread Group 1-13	Home page	423	✔	11642	429	423	176
14	09:50:14.249	Thread Group 1-14	Home page	403	✔	11642	429	403	172
15	09:50:14.316	Thread Group 1-15	Home page	421	✔	11642	429	421	173
16	09:50:14.373	Thread Group 1-16	Home page	365	✔	11642	429	365	164
17	09:50:14.440	Thread Group 1-17	Home page	379	✔	11642	429	379	167
18	09:50:14.508	Thread Group 1-18	Home page	364	✔	11642	429	364	171
19	09:50:14.576	Thread Group 1-19	Home page	347	✔	11642	429	347	157
20	09:50:14.631	Thread Group 1-20	Home page	350	✔	11642	429	350	164
21	09:50:14.688	Thread Group 1-21	Home page	346	✔	11642	429	346	157
22	09:50:14.753	Thread Group 1-22	Home page	409	✔	11642	429	409	169
23	09:50:14.818	Thread Group 1-23	Home page	396	✔	11642	429	396	193
24	09:50:14.891	Thread Group 1-24	Home page	392	✔	11642	429	392	172
25	09:50:14.945	Thread Group 1-25	Home page	450	✔	11642	429	450	195
26	09:50:15.001	Thread Group 1-26	Home page	430	✔	11642	429	430	227

Figura 3.7 Prueba de conexión con 500 usuarios

Pruebas de carga con 1000 usuarios

Los resultados de la prueba de carga con 1000 usuarios de la Figura 3.8 muestran que el aumento en la cantidad de usuarios ha tenido un impacto negativo en la conectividad de la plataforma. Algunos usuarios experimentaron problemas al acceder a la plataforma y se registró un aumento significativo en la latencia en cada requerimiento.

Esto puede ser un indicativo de que el uso del plan gratuito de los servicios de AWS puede ser una limitación para la plataforma a medida que crece la cantidad de usuarios. Para mejorar esta situación, será necesario considerar la opción de actualizar a un plan de pago de los servicios de AWS, que brinde una mayor capacidad de manejo de usuarios y una mejor respuesta en términos de latencia. Además, es importante monitorear constantemente los resultados de las pruebas de carga y evaluar opciones para mejorar la conectividad y optimizar la experiencia de los usuarios.

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(ms)
10	04:42:25.497	Thread Group 1-20	Home page	340	⊗	522	429	340	174
11	04:42:25.577	Thread Group 1-21	Home page	325	⊗	522	429	325	165
12	04:42:25.654	Thread Group 1-22	Home page	379	⊗	522	429	379	171
13	04:42:25.732	Thread Group 1-23	Home page	369	⊗	522	429	368	168
14	04:42:25.813	Thread Group 1-24	Home page	372	⊗	522	429	372	165
15	04:42:25.891	Thread Group 1-25	Home page	373	⊗	522	429	373	166
16	04:42:25.969	Thread Group 1-26	Home page	379	⊗	522	429	379	166
17	04:42:26.046	Thread Group 1-27	Home page	320	⊗	522	429	320	166
18	04:42:26.127	Thread Group 1-28	Home page	388	⊗	522	429	388	167
19	04:42:24.261	Thread Group 1-1	Home page	2272	⊕	11642	429	2272	277
20	04:42:26.286	Thread Group 1-30	Home page	325	⊗	522	429	325	165
21	04:42:26.205	Thread Group 1-29	Home page	424	⊕	11642	429	424	181
22	04:42:26.365	Thread Group 1-31	Home page	419	⊕	11642	429	419	172
23	04:42:26.445	Thread Group 1-32	Home page	385	⊕	11642	429	385	186
24	04:42:26.527	Thread Group 1-33	Home page	419	⊕	11642	429	419	164
25	04:42:26.609	Thread Group 1-34	Home page	409	⊕	11642	429	409	163
26	04:42:26.696	Thread Group 1-35	Home page	385	⊕	11642	429	385	170
27	04:42:26.864	Thread Group 1-37	Home page	322	⊗	522	429	322	161
28	04:42:26.778	Thread Group 1-36	Home page	427	⊕	11642	429	427	171
29	04:42:26.944	Thread Group 1-38	Home page	365	⊕	11642	429	365	165
30	04:42:27.030	Thread Group 1-39	Home page	368	⊕	11642	429	368	167
31	04:42:27.113	Thread Group 1-40	Home page	405	⊕	11642	429	405	166
32	04:42:27.198	Thread Group 1-41	Home page	358	⊕	11642	429	358	168
33	04:42:27.283	Thread Group 1-42	Home page	351	⊕	11642	429	351	165
34	04:42:27.368	Thread Group 1-43	Home page	364	⊕	11642	429	364	164
35	04:42:27.532	Thread Group 1-45	Home page	227	⊗	522	429	227	179

Scroll automatically?
 Child samples?
No of Samples: 1000
Latest Sample: 544
Average: 318
Deviation: 1133

Figura 3.8 Prueba de conexión con 1000 usuarios

Para observar más a detalle la conexión de los primeros usuarios se presenta la Figura 3.9 que muestra el tiempo de respuesta de cada requerimiento y como a medida que pasa a cierta cantidad de usuarios se va nivelando.

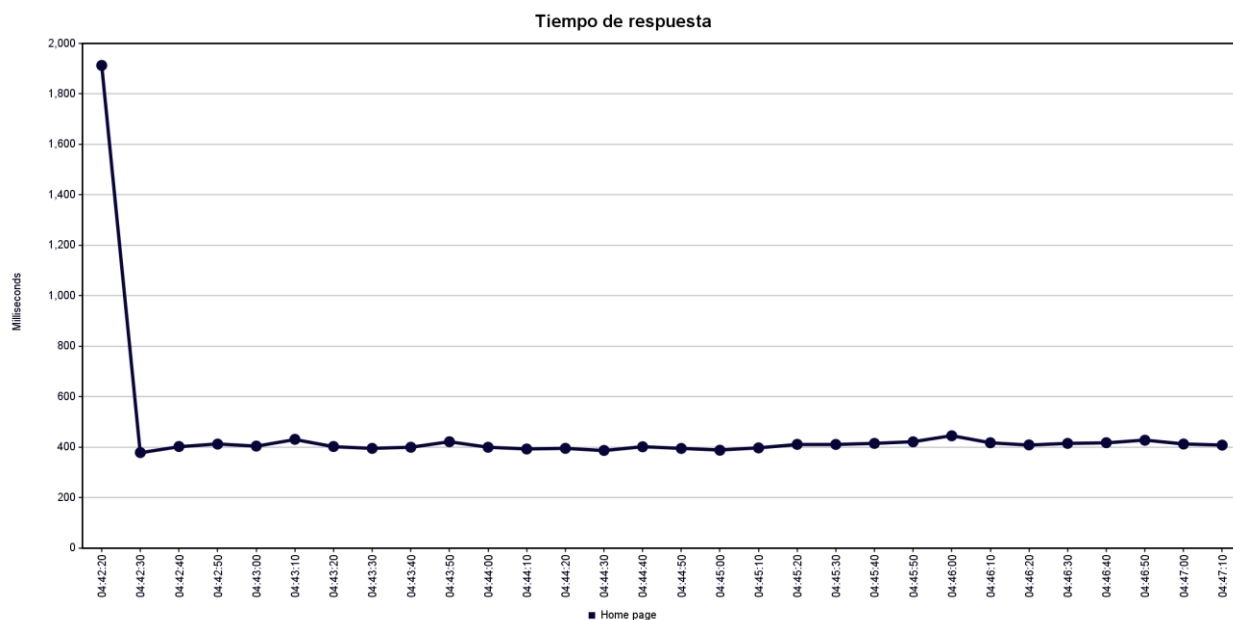


Figura 3.9 Tiempo de respuesta de 1000 usuarios

3.3 Análisis de costos

Para el análisis de costes final del proyecto se consideraron los siguientes conceptos: compra de materiales de hardware, materiales para la maqueta, servidores en la nube, y mano de obra.

A continuación, se detalla el costo de cada uno de los conceptos descritos. El detalle de gastos de cada uno se lo puede encontrar en el Tabla C.1.

Tabla 3.2 Análisis de costos

No.	Elemento	Uso	Valor
1	Raspberry PI	Maestro y controlador principal	\$200,00
2	Dispositivos IoT para escenarios	Elementos que usan para el funcionamiento de los escenarios	\$192,30
3	Maqueta de Plywood	Prototipo de maqueta funcional del laboratorio. Incluye mano de obra	\$150,00
4	Trípode	Para movimiento de la cámara	\$20,00
5	Servicio de AWS S3	Recurso Computacional	\$100,00
6	Personal para desarrollo del proyecto	Estimación de costo de un proyecto realizado en 2 meses con jornadas laborales de 8 horas. Nota: \$5xhora	\$1.600,00
7	Recurso humano para inspección de código y servicios	Estimación de costo por inspección de código y servicios en la nube de manera trimestral.	\$400,00
8	Costos adicionales	Asumiendo costos adicionales según los cambios por Sprint dada la metodología ágil	\$200,00
		Subtotal	\$2.862,30
		IVA 0%	\$0,00
		Total	\$2.862,30

El costo de implementación de este proyecto está enfocado a baja escala, considerando los dos escenarios realizados y servidores en la nube gratuitos. Se desarrolló la plataforma para ser escalable, por lo que en caso de crearse más escenarios no supondría ningún problema.

Los costes de posibles nuevos escenarios si pudiesen variar, donde cuenta el costo de los dispositivos y el diseño de la maqueta. Es notable recalcar que cualquier sensor nuevo que se agregue a los dispositivos deberá ser cargado de igual manera en la base

de datos. Al ser una plataforma educativa que quedará para los servicios de ESPOL, se mantendrá un plan de uso gratuito.

El costo de inversión fue de \$236 para la compra de los dispositivos que constan en los escenarios. Luego se asumió la implementación como un servicio de tecnología IoT. Exceptuando el personal humano, los costos de la fabricación de la maqueta y los otros costos descritos en la tabla. Se observa que la inversión dada para la compra de dispositivos entra en el rango de aceptación.

3.4 Escenario de casa inteligente utilizando dispositivos IoT.

El escenario de la Figura 3.10 mostrado en la plataforma web permitió al usuario interactuar con el laboratorio según la acción seleccionada. En la plataforma se ve reflejada la acción, por ejemplo: para encender el foco led del frente debe accionar el switch de la plataforma. La comunicación que tiene el escenario con el dispositivo maestro que controla las acciones de los microcontroladores secundarios permite que el cliente tenga una retroalimentación de su acción en tiempo real. Cada acción reflejada en el escenario casa inteligente (Figura F.1) es mostrada en la pestaña “vista de la cámara”.

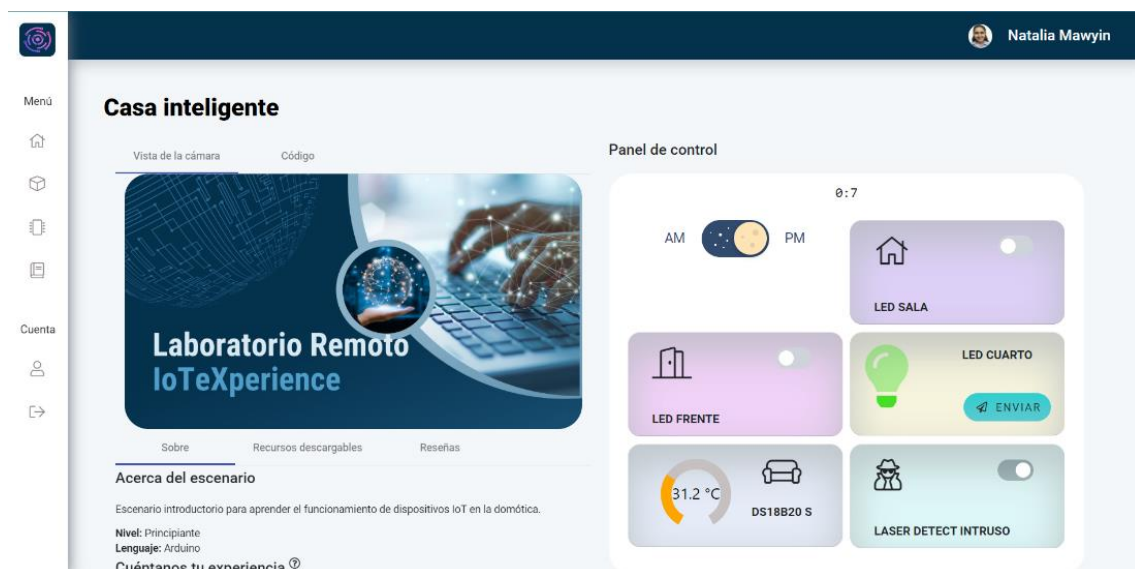


Figura 3.10 Vista de escenario Casa Inteligente

3.5 Escenario de control de temperatura usando control PID

El escenario ilustrado en la Figura 3.11 permite a los usuarios visualizar una gráfica de la temperatura en tiempo real en el laboratorio. Además, mediante la interacción con los botones relevantes, se pueden habilitar o deshabilitar opciones para modificar las constantes PID del controlador. La retroalimentación se brinda a los

usuarios a través de una gráfica de la temperatura versus el tiempo, mientras realizan cambios en las constantes del controlador PID. Y en la pestaña “vista de la cámara” se ve cada acción reflejada del escenario control de temperatura PID (Figura F.3).

Esto se puede activar a través del modo manual en el que los usuarios pueden modificar las constantes del controlador PID. El modo automático utiliza las constantes previamente calculadas con *PID Tuner* en Matlab. Además, se comprobó de forma experimental que el tiempo que se tarda en estabilizar la temperatura real al objetivo en el modo automático fue de 180 segundos como se observa en el Figura F.4. Es importante destacar que el usuario siempre tiene la capacidad de cambiar la temperatura objetivo.

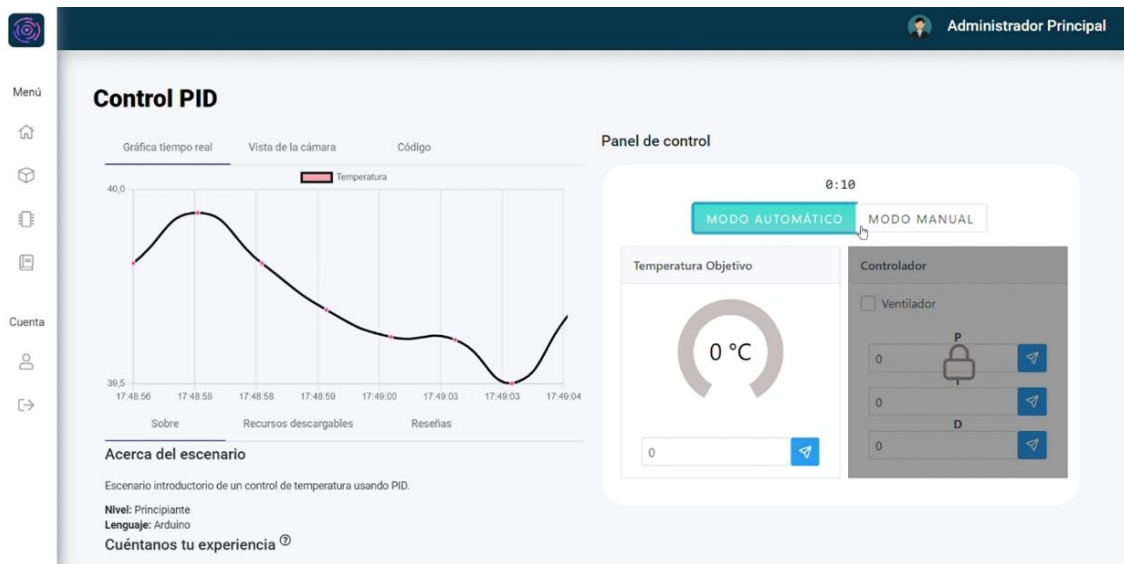


Figura 3.11 Vista de escenario Control PID

3.6 Tiempos de ejecución

3.6.1 Comunicación MQTT

Se realizó un análisis de los tiempos de ejecución de la comunicación MQTT desde las placas ESP TTGO, para ello se utilizó una función llamada *micro* que retorna los valores del tiempo de ejecución que ha transcurrido en escala de microsegundos.

Para tener un análisis de tiempos más preciso se tomó una muestra de cuarenta datos de lo cual se obtuvo que el tiempo promedio de ejecución es de 1549 μ s, con una desviación estándar de 78 μ s, lo que permite establecer un rango de valores aceptables que va desde los 1471 μ s hasta los 1627 μ s, tal como se lo observa en la Tabla 3.3.

Tabla 3.3 Medidas de tendencia central en la comunicación MQTT

Muestra	Tiempo promedio de ejecución	Desviación estándar	Rango de aceptación
40	1549 μ s	78 μ s	[1471, 1627] μ s

Bajo las medidas de tendencia central mencionadas se tiene que el 55% de los tiempos obtenidos se encuentran dentro de un rango de valores aceptable, un 25% de tiempos se encuentran debajo de los valores esperados, y un 20% por sobre los tiempos esperados. Para una mayor compresión gráfica, se puede consultar el Figura D.1.

A partir de los valores obtenidos se puede decir que los tiempos de ejecución en la comunicación MQTT se encuentran dispersos, esto pasa por los comandos que se encuentran ejecutando antes de que lleguen los mensajes. Además, que en el código fuente de las placas ESP TTGO se encuentra con un tiempo de espera que tienen un efecto ralentizador en la comunicación, esto se pudo comprobar al ver que los valores más altos de tiempo se encontraban en cuadros de código con comandos más complejos.

3.6.2 Peticiones a la base de datos

Otro punto importante en la comunicación es la ejecución de los queries que son llamados desde el controlador, para obtener los datos de tiempos de petición de los datos de la base al controlador se usó la librería *time* de Python, que retorna el tiempo de ejecución en el que se encuentra el programa.

Se hizo una toma de cuarenta muestras de tiempos en la escala de los microsegundos, de lo que se obtuvo que el tiempo promedio que se demora el controlador de pedir y recibir de la base de datos es de 95395 μ s, con una desviación estándar de 3514 μ s, datos que permiten establecer un intervalo de valores aceptados, siendo este desde los 91881 μ s hasta los 98909 μ s, tal como se lo puede observar en la Tabla 3.4.

Tabla 3.4 Medidas de tendencia central en las peticiones a la base de datos

Muestra	Tiempo promedio de ejecución	Desviación estándar	Rango de aceptación
40	95395 μ s	3514 μ s	[91881, 98909] μ s

A partir de estas medidas de tendencia central se tiene que el 80% de los tiempos tomados se encuentran dentro del rango de valores aceptados, el 5% se encuentra por

debajo del rango estimado, y un 15% sobre los tiempos esperados de ejecución. Para una mayor comprensión gráfica, se puede consultar el Figura D.2.

De estos resultados se puede decir que los tiempos de ejecución de la base de datos se encuentra poco dispersos en función de su media, lo que refleja que la base de datos no se ve afectada por cuadro de código o conexión, teniendo una respuesta bastante idónea ya que si se analiza su tiempo promedio de respuesta en una escala mayor, siendo de 98,9 ms, el cual es un tiempo bastante bajo que evita que el controlador tenga problemas para obtener las acciones de la plataforma y enviarlas hacia los nodos.

3.6.3 Subida de archivos a los nodos mediante OTA

Debido a que las placas tienen un servicio de carga de código de manera remota a través de una comunicación OTA, se vio pertinente analizar el tiempo de subida de código fuente hacia las placas, para ello se usó un cronómetro sincronizado a la subida del archivo binario.

Se hizo una toma de cuarenta tiempos de carga de código hacia las placas, de lo cual se obtuvo que el tiempo promedio de carga es de 13.48 segundos, con una desviación estándar de 0.53 segundos, de lo cual se establece un intervalo de valores aceptables desde los 12.95 segundos hasta los 14.02 segundos.

Tabla 3.5 Medidas de tendencia central del tiempo de subida de código fuente por OTA

Muestra	Tiempo promedio de ejecución	Desviación estándar	Rango de aceptación
40	13.48 s	0.53 s	[12.95, 14.02] s

A partir de estas medidas de tendencia central se obtuvo que el 65% de los tiempos obtenidos se encuentran dentro del intervalo establecido, el 17.5% se encuentra por debajo de la cota mínima de tiempo de ejecución, y el 17.5% se encuentra sobre de los tiempos de carga promedio de 14.02s. Para una mayor comprensión gráfica, se puede consultar el Figura D.3.

3.7 Encuesta de satisfacción de usuario

Para analizar el nivel de satisfacción que tiene el público objetivo se realizó una encuesta que consistía en nueve preguntas cerradas y una abierta donde podían ingresar comentarios respecto a la plataforma mostrada. Se tuvo un total de 51 respuestas, de las cuales se obtuvo que el 82.4% de los encuestados creen que el IoT

es un tema de interés y que debería ser impartido en colegios y universidades. Además, el 98% de las personas consideran útil tener una herramienta que muestre el funcionamiento real de un microsistema IoT.

Por otra parte, el 70.6% de los encuestados consideran importante que haya flexibilidad en el horario de clase, un 76.5% consideran pertinente la implementación de una herramienta que permita tomar laboratorios remotos con flexibilidad en su horario.

Desde el punto de vista del funcionamiento de la plataforma web, el 47,1% de usuarios creen que la plataforma es bastante interactiva, un 39,2% dicen que es medianamente intuitiva y un 10% de usuarios la consideran poco intuitiva.

Adicional a ello un 64.7% de las personas encuestadas muestran gran interés en la solución propuesta, mientras que un 25.5% se encuentran indiferentes y un 9.8% no se encuentran interesados en la plataforma.

Algunos de las recomendaciones que se obtuvo por parte de las personas encuestadas fueron, aumentar la explicación de los funcionamientos de la plataforma, considerar el aumento de escenarios con diferentes tipos de sensores IoT, considerar la implementación de un foro para agregar dudas por parte de los estudiantes, y finalmente tener un recorrido para los nuevos usuarios donde se expliquen las secciones y acciones que se pueden realizar dentro de la plataforma.

CAPÍTULO 4

4. CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

En el presente proyecto se desarrolló una plataforma web, que se nombró como IoTeXperience, utilizando Angular como el framework de FrontEnd y alojada en AWS, con un servidor en Django y una base de datos en PostgreSQL, en la cual se pueden visualizar escenarios en tiempo real, enviar datos a dispositivos IoT, descargar librerías reutilizables y leer los recursos educativos disponibles, brindando una experiencia de aprendizaje óptima y eficiente a estudiantes y profesores.

Adicional a ello, se desarrolló un módulo de librerías reutilizables precargado con librerías iniciales que fueron creadas basadas en los dispositivos IoT utilizados. Estas mismas librerías fueron usadas dentro de los escenarios cargados en la plataforma por lo que se demuestra el uso correcto de las mismas.

Por otra parte, en la comunicación MQTT del controlador principal a los secundarios permite una interacción con los escenarios como en tiempo real. Siendo así, la comunicación usada para enviar datos por segundo a la página web y el estudiante observe la gráfica.

Además, se realizaron pruebas que permitían analizar los tiempos de ejecución de la comunicación en donde se pudo encontrar que, en la comunicación MQTT un 55% de los datos medidos están dentro de los tiempos normales. Este porcentaje es bajo debido a que existen bloques de código que se anteponen a los comandos que se encargan de establecer la comunicación, por lo que es fundamental que se realice una modularización de estos y priorizar dicha comunicación por medio de interrupciones.

Desde el lado de las consultas hacia la base, el 80% de los tiempos medidos están dentro de los valores normales de ejecución, por lo que se puede decir que los requerimientos hacia la base se comportan de manera correcta, sin embargo, estos se podrían mejorar optimizando los queries que son utilizados para las consultas para así disminuir el peso de estos y los requerimientos sean respondidos de mejor manera.

Por parte de la actualización del código de manera remota, los tiempos tomados que están dentro de un rango correcto es del 65%, cuyo porcentaje es aceptable pero

no el ideal, esto es debido a que el método en cuestión hace una escritura sobre la memoria flash por lo que va a llegar a un punto donde el microcontrolador empiece a presentar fallas o ralentización a la hora de cargar el código.

Desde la parte de la implementación de un prototipo físico de microsistemas, se desarrolló una infraestructura sólida que se acopla a la plataforma como escenario asegurando una interacción entre usuario y laboratorio más real, teniendo la implementación de un control PID en un microcontrolador para una fuente de calor no simulada, la cual muestra un aumento en el tiempo de estabilización aproximadamente del 50%. Además, la observación del comportamiento de temperatura en la gráfica prevé el aumento del interés del estudiante.

Adicional a la implementación, se llevó a cabo rigurosas pruebas de carga para evaluar la capacidad de IoTExperience para soportar varias solicitudes de usuarios simultáneamente. Los resultados obtenidos demostraron con seguridad que la plataforma cumple con ser capaz de soportar una alta carga de usuarios, lo que la hace escalable y apta para ser utilizada en un entorno de salón de clases en tiempo real.

Finalmente, la estructura de la plataforma tiene la capacidad de soportar más microsistemas de proyectos IoT ya sean pequeños o a escala, esto es muy importante ya que para trabajos futuros se pueden acoplar proyectos más específicos, por ejemplo: edificios inteligentes o proyectos relacionados al IoT que quieran incorporar dentro los laboratorios de la materia. Cabe resaltar que, los proyectos deben regirse con la configuración y el modelo estructurado que ya posee el sistema.

4.2 Recomendaciones

Se pueden destacar las siguientes mejoras futuras para el desarrollo de la plataforma:

- Implementar un sistema de login directo por medio de los servicios de ESPOL, para que así los estudiantes no necesiten registrarse a la plataforma educativa, sino que ingresen con las credenciales de ESPOL.
- Desarrollar un módulo de carga de archivos en la plataforma para los administradores, así se puedan enviar modificaciones en tiempo real a los escenarios y no haga falta operaciones manuales para realizar los cambios.
- Subida a servidor de los envíos realizados del control PID, para que en vez de ser MQTT pueda ser accedido en forma HTTP en cualquier lugar. Actualmente solo se puede acceder si se está conectado en la misma red, por lo que se sugiere

que se suba a un servidor en la nube y se pueda acceder desde cualquier ubicación.

- Mejorar el mecanismo de soporte para una cámara de mayor peso que puede cargar un servo motor SG90.
- Establecer el laboratorio en una zona con condiciones ambientales no muy variables, para obtener un mejor comportamiento en la estabilización de la temperatura obtenida hacia el objetivo. También, se puede modificar dentro del código los límites permitidos para su estabilización.
- Usar las debidas protecciones para la implementación de la infraestructura, a su vez; usar cinta aislante por cada conexión realizada sin protección para evitar sobrecargas, explosiones o corto.

4.3 Trabajos futuros

El presente trabajo está abierto a realizar mejoras y/o cambios a nivel de prototipo físico, comunicación, y funcionalidades en la plataforma web. Uno de los posibles cambios, es el aumento de microescenarios donde se vean otros tipos de aplicaciones con la implementación de otros protocolos de comunicación, además que se puede mejorar la estética de los microescenarios ya existente.

Desde el punto de la comunicación, queda apertura para implementar otros tipos de protocolos de comunicación en los escenarios ya establecidos, además del uso de soluciones en la nube para comunicar dispositivos IoT que trabajen con otros protocolos como LoRaWAN.

Además, la plataforma web, al encontrarse desarrollada en código abierto, es capaz de adaptar nuevas funcionalidades. Teniendo en cuenta algunos de los comentarios de los usuarios encuestados, se puede agregar una sección de foro para que estudiantes y docentes tengan una interacción dentro de la plataforma sin necesidad de medios externos.

Finalmente, se la puede incluir dentro de una investigación donde se analicen a mayor detalle la funcionalidad y comunicación establecida en la plataforma, teniendo un mayor número de usuarios encuestados, a los cuales los pongan dentro de un escenario real donde se imparta una clase remota de laboratorio a través de IoT eXperience.

BIBLIOGRAFÍA

- [1] A. A. Zaidan y Zaidan B. B., «A review on intelligent process for smart home applications based on IoT: coherent taxonomy, motivation, open challenges, and recommendations,» *Artif Intell Rev*, vol. 53, pp. 141-165, 2020.
- [2] M. S. Farooq, S. Riaz, A. Abid, T. Umer y Y. B. Zikria, «Role of IoT Technology in Agriculture: A Systematic Literature Review,» *Electronics (Basel)*, vol. 9, nº 2, pp. 1-41, 2020.
- [3] S. Salagare y R. Rasad, «An Overview of Internet of Dental Things: New Frontier in Advanced Dentistry,» *Wirel Pers Commun*, vol. 110, p. 1345–1371, 2019.
- [4] A. Tavizon-Salazar, C. Laines y T. Guajardo, «IOT, el internet de las cosas y la innovación de sus aplicaciones,» pp. 2013-2040, 2016.
- [5] J. F. Fernández Sosa, «Utilización de dispositivos móviles como herramienta de sensado en aplicaciones de IoT,» 2021.
- [6] J. Camarena, L. Contreras, K. Moreno, M. Rodriguez y C. Salazar, «Aplicaciones del IoT para el control de congestión vehicular,» *3er Congreso Internacional de Ciencias y Tecnologías para el Desarrollo Sostenible 2018*, pp. 90-95, 2018.
- [7] Baz, «IoT – Now and in Ten Years!,» Huawei Forum, 28 noviembre 2021. [En línea]. Available: <https://forum.huawei.com/enterprise/en/iot-now-and-in-ten-years/thread/794825-893?page=1>. [Último acceso: 12 febrero 2023].
- [8] S. Nižetić, P. Šolić, D. López-de-Ipiña González-de-Artaza y L. Patrono, «Internet of Things (IoT): Opportunities, issues and challenges towards a smart and sustainable future,» *Journal of Cleaner Production*, vol. 274, nº 122877, pp. 1-32, 2020.
- [9] L. Columbus, «10 Charts That Will Challenge Your Perspective Of IoT's Growth,» Forbes, 6 enero 2018. [En línea]. Available: <https://www.forbes.com/sites/louiscolombus/2018/06/06/10-charts-that-will->

- challenge-your-perspective-of-iots-growth/?sh=1cfac063ecce. [Último acceso: 2022].
- [10] N. Elmqaddem, «Augmented Reality and Virtual Reality in Education. Myth or Reality?,» *i-JET Short Paper*, vol. 14, nº 3, 2019.
- [11] R. Pérez, P. Mercado , M. Martínez, E. Mena y J. Partida, «La sociedad del conocimiento y la sociedad de la información como la piedra angular en la innovación tecnológica educativa,» *Revista Iberoamericana para la Investigación y el Desarrollo Educativo*, vol. 8, nº 16, 2018.
- [12] Á. Fidalgo, «Innovación educativa en la sociedad del conocimiento,» *Teoría de la Educación. Educación y Cultura en la Sociedad de la Información*, vol. 13, nº 5, pp. 1-3, 2014.
- [13] V. Potkonjak, M. Gardner, V. Callaghan, P. Mattila, C. Guetl, V. M. Petrović y K. Jovanović, «Virtual laboratories for education in science, technology, and engineering: A review,» *Comput Educ*, vol. 95, pp. 309-327, 2016.
- [14] M. Habibi, «Work in Progress: Applications of Internet of Things (IoT) in Distance Lab Checkoff,» *American Society for Engineering Education*, 2018.
- [15] C. Kreiter, D. Cosic y T. Klinger, «A Virtual Reality Prototype for the VISIR Remote Lab,» de *New Realities, Mobile Systems and Applications*, 2022, p. 874–884.
- [16] K. Chen, H. Chen, J. Huang, F. Lanni, S. Tang y W. Wu, «A Generic High Bandwidth Data Acquisition Card for Physics Experiments,» *IEEE Transactions on Instrumentation and Measurement*, vol. 69, nº 7, p. IEEE Transactions on Instrumentation and Measurement, 2020.
- [17] A. Nafalski, M. Milosz, H. Considine y Z. Nedić, «Overseas Use of the Remote Laboratory NetLab,» *IEEE Global Engineering Education Conference (EDUCON)*, pp. 568-573, 2020.
- [18] J. F. Martins, «Matriz de Interconexão Eletrônica Programável para Prototipagem Rápida e Experimentação Remota,» *TCC(graduação) - Universidade Federal de Santa Catarina. Campus Araranguá. Engenharia da Computação.*, pp. 42-44, 2020.
- [19] G. Rath y M. Payr, «A Remote Laboratory for Teaching Internet of Things in Online Education,» *11th Mediterranean Conference on Embedded Computing (MECO)*, pp. 1-4, 2022.

- [20] D. Mourtzis, J. Angelopoulos y N. Panopoulos, «Design and development of an IoT enabled platform for remote monitoring and predictive maintenance of industrial equipment,» *Procedia Manuf*, vol. 54, pp. 166-171, 2021.
- [21] D. Rodríguez y R. Alberto, «La educación virtual universitaria como medio para mejorar las competencias genéricas y los aprendizajes a través de buenas prácticas docentes,» *Tesis Doctorals en Xarxa*, 2016.
- [22] «Comparativa de plataformas virtuales en el Ecuador,» Google Trends. [En línea]. [Último acceso: 2022].
- [23] A. Gupta, R. M. Shrestha, S. Shrestha, A. Acharya y N. Pandey, «Perception of BDS students of Kathmandu University on online learning during COVID-19 pandemic,» *Orthodontic Journal of Nepal*, vol. 10, nº 2, pp. 20-28, 2020.
- [24] K. Syauqi, S. Munadi y M. B. Triyono, «Students' Perceptions toward Vocational Education on Online Learning during the COVID-19 Pandemic,» *International Journal of Evaluation and Research in Education*, pp. 881-886, 2020.
- [25] Y. B. Zikria, S. W. Kim, O. Hahm, M. K. Afzal y M. Y. Aalsalem, «Internet of Things (IoT) Operating Systems Management: Opportunities, Challenges, and Solution,» *Sensors*, vol. 19, nº 8, p. 1793, 2019.
- [26] M. Mehrtash, K. Ghalkhani y I. Singh, «IoT-based Experiential E-Learning Platform (EELP) for Online and Blended Courses,» *International Symposium on Educational Technology (ISET)*, pp. 252-255, 2021.
- [27] S. Maitra, A. Abdelgawad y K. Yelamarthi, «Lab in a Box : A Rapidly Deployable Environmental Monitoring IoT System,» *Midwest Symposium on Circuits and Systems*, pp. 472-475, 2019.
- [28] H. Al-qozani y A. Aleryani, «The Impact of IoT on the Higher Education (Review Study),» *Saba Journal Of information Technology And Networking (SJITN)*, vol. 6, nº 2, pp. 38-48, 2018.
- [29] A. Kırbaş, «Student views on using smart boards in turkish education,» *Universal Journal of Educational Research*, vol. 6, nº 5, pp. 1040-1049, 2018.
- [30] Eclipse IoT Working Group, «The Three Software Stacks Required for IoT Architectures IoT software requirements and how to implement them using open source technology,» *Eclipse Foundation*, 2016.

- [31] P. P. Ray, «A Survey on Visual Programming Languages in Internet of Things,» *Sci Program*, vol. 2017, pp. 1-6, 2017.
- [32] I. D. Wahyono, I. Fadlika, A. R. Taufani y H. Elmunsyah, «Research in Ubiquitous Network: Communication Optimization using Publish-Subscribe,» *Electrical Power, Electronics, Communications, Controls and Informatics Seminar (EECCIS)*, pp. 202-206, 2019.
- [33] I. M. Wirawan, I. D. Wahyono, G. Idfi y G. R. Kusumo, «IoT Communication System Using Publish-Subscribe,» *International Seminar on Application for Technology of Information and Communication*, pp. 61-65, 2018.
- [34] C. Severance, «Roy T. Fielding: Understanding the REST Style,» *Computer (Long Beach Calif)*, vol. 48, nº 6, pp. 7-9, 2015.
- [35] M. Belshe, R. Peon y M. Thomson, «Hypertext Transfer Protocol Version 2 (HTTP/2),» *Internet Engineering Task Force (IETF), RFC 7540*, 2015.
- [36] A. AntoniĆ, M. Marjanović, P. Skočir y I. P. Žarko, «Comparison of the CUPUS middleware and MQTT protocol for smart city services,» *13th International Conference on Telecommunications (ConTEL)*, pp. 1-8, 2015.
- [37] N. Maynard, «MQTT and IBM MessageSight: Secure, reliable communications for the next generation of resilient mobile applications,» *IBM*, 2015.
- [38] N. Nikolov, «Research Firmware Update Over the Air from the Cloud,» *XXVII International Scientific Conference Electronics*, 2018.
- [39] Espressif Systems, «ESP32 Series Datasheet,» 2022. [En línea]. Available: <https://www.espressif.com/en/support/download/documents>. [Último acceso: 2022].
- [40] Espressif Systems, «ESP32 Technical Reference Manual Version 4.7,» 2022. [En línea]. Available: <https://www.espressif.com/en/support/download/documents>. [Último acceso: 2022].
- [41] Raspberry Pi Foundation, «Raspberry Pi,» Raspberry Pi Foundation, 2022. [En línea]. Available: <https://www.raspberrypi.org/>. [Último acceso: 2022].
- [42] Electronics Hub, «101+ Raspberry Pi Projects For Electronics Students,» Electronics Hub, 15 10 2015. [En línea]. Available: <https://www.electronicshub.org/raspberry-pi-projects/>. [Último acceso: 2022].

- [43] Naylamp Mechatronics, «MÓDULO SENSOR LDR,» Naylamp Mechatronics, [En línea]. Available: <https://naylampmechatronics.com/sensores-luz-y-sonido/135-modulo-sensor-ldr.html>. [Último acceso: 2022].
- [44] P. Aguayo, «Servo motor con Arduino,» Arduino.cl, 2009. [En línea]. Available: <https://arduino.cl/servo-motor-con-arduino/#:~:text=Un%20motor%20servo%20es%20un,mantenerse%20estable%20en%20dicha%20posici%C3%B3n>. [Último acceso: 01 12 2022].
- [45] UNIT Electronics., «Sensor Laser Módulo KY-008,» UNIT Electronics., 06 05 2019. [En línea]. Available: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwiC34_ls6r8AhVOnIQIHb-CO8QFnoECBoQAQ&url=https%3A%2F%2Fuelectronics.com%2Fproducto%2Fmodulo-ky-008-sensor-laser%2F&usg=AOvVaw0JveREjjkx5iOYFfiOXYu. [Último acceso: 01 12 2022].
- [46] Naylamp Mechatronics, «MÓDULO RECEPTOR LASER,» Naylamp Mechatronics, [En línea]. Available: <https://naylampmechatronics.com/sensores-luz-y-sonido/329-modulo-receptor-laser.html>. [Último acceso: 01 12 2022].
- [47] UNIT Electronics, «Sensor Zumbador Activo Módulo KY-012,» UNIT Electronics, [En línea]. Available: <https://uelectronics.com/producto/modulo-ky-012-sensor-zumbador-activo/>. [Último acceso: 15 12 2022].
- [48] UNIT Electronics, «Sensor de temperatura Digital DS18B20,» UNIT Electronics, [En línea]. Available: <https://uelectronics.com/producto/sensor-de-temperatura-digital-ds18b20/>. [Último acceso: 01 12 2022].
- [49] R. Vera Crespo, «Análisis de vibraciones en elementos funcionales del faro izquierdo del automovil New Beetle 2.5 Versión USA (NB-NAR),» 2006.
- [50] C. M. Pardo, «Controlador PID,» Picuino, 2018. [En línea]. Available: <https://www.picuino.com/es/control-pid.html>. [Último acceso: 01 12 2022].
- [51] K. Ogata, Ingeniería de control moderna, 3era ed., Mexico: Pearson Educación.
- [52] Mathworks, «App System Identification,» Mathworks, [En línea]. Available: <https://la.mathworks.com/products/sysid.html>. [Último acceso: 01 12 2022].

APÉNDICES

APÉNDICE A

Tabla A.1 Tabla de evaluación del criterio de precio.

Solución B > Solución A > Solución C

Precio	Solución A	Solución B	Solución C	Σ+1	Ponderación
Solución A		0	1	2	0,3333333333
Solución B	1		1	3	0,5
Solución C	0	0		1	0,166666667
			Suma	6	1

Tabla A.2 Tabla de evaluación del criterio de transmisión.

Solución C > Solución A > Solución B

Transmisión	Solución A	Solución B	Solución C	Σ+1	Ponderación
Solución A		1	0	2	0,3333333333
Solución B	0		0	1	0,166666667
Solución C	1	1		3	0,5
			Suma	6	1

Tabla A.3 Tabla de evaluación del criterio de consumo

Solución A = Solución B > Solución C

Consumo	Solución A	Solución B	Solución C	Σ+1	Ponderación
Solución A		0,5	1	2,5	0,416666667
Solución B	0,5		1	2,5	0,416666667
Solución C	0	0		1	0,166666667
			Suma	6	1

Tabla A.4 Tabla de evaluación del criterio de mantenimiento.

Solución A > Solución B > Solución C

Mantenimiento	Solución A	Solución B	Solución C	Σ+1	Ponderación
Solución A		1	0,5	2,5	0,416666667
Solución B	0		1	2	0,3333333333

Solución C	0,5	0		1,5	0,25
			Suma	6	1

APÉNDICE B

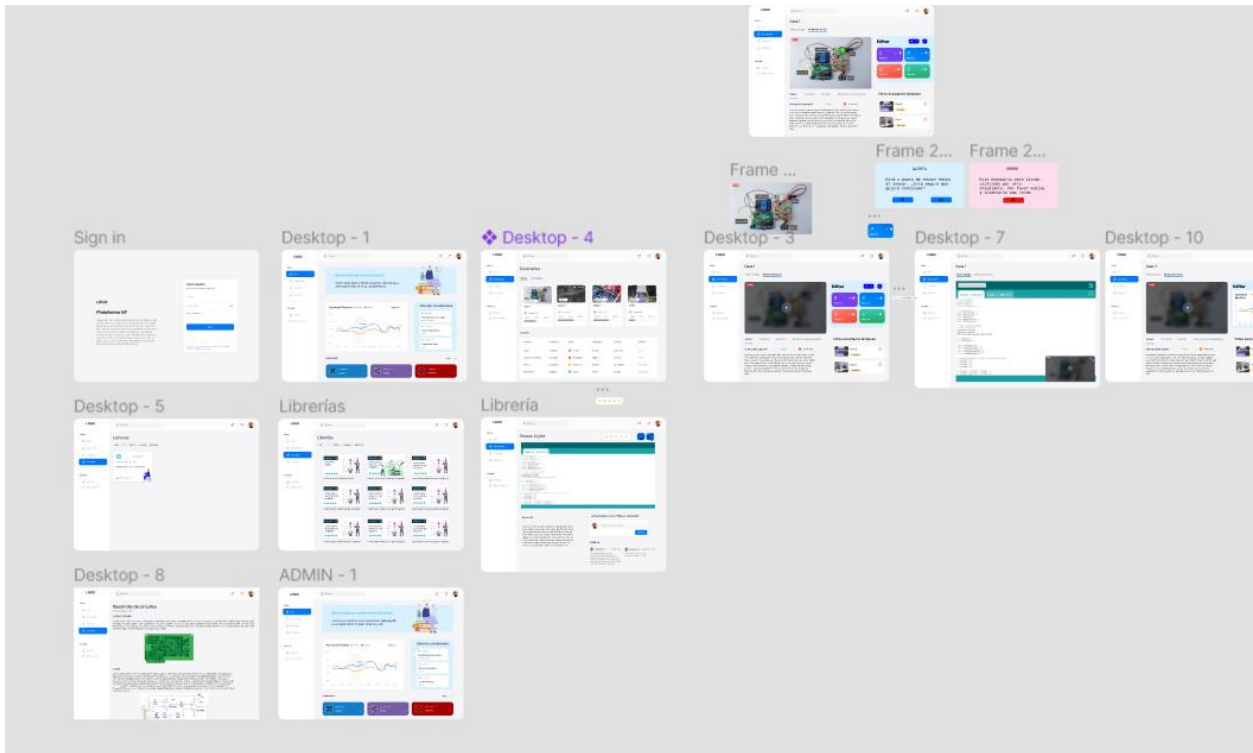


Figura B.1 Prototipo de la plataforma realizado en Figma

Url: <https://www.figma.com/proto/iDCuTKVss7VCxFZK68gM77/MI-plataforma-web?node-id=5%3A2&scaling=min-zoom&page-id=0%3A1&starting-point-node-id=5%3A2>

APÉNDICE C

Tabla C.1 Dispositivos utilizados en el prototipo físico.

Descripción	Cantidad	Valor Unitario	Valor Total
ESP32 TTGO T8	2	\$25,30	\$50,60
Módulo Sensor de Temperatura	2	\$2,40	\$4,80
Jumper-macho-20cmx40	1	\$3,75	\$3,75
Jumper-mixto-20cmx40	1	\$3,75	\$3,75
Módulo Sensor Fotorresistencia	2	\$2,40	\$4,80
Led de color primario	6	\$0,10	\$0,60
Led de colores en uno solo	4	\$0,25	\$1,00
Resistencia de 1 K ohmios	5	\$0,10	\$0,50
Resistencia de 10 K ohmios	5	\$0,10	\$0,50
Resistencia de 220 ohmios	5	\$0,10	\$0,50
Resistencia de 320 ohmios	5	\$0,10	\$0,50
Placa de prototipado	2	\$3,50	\$7,00
Módulo zumbador	1	\$1,60	\$1,60
Capacitor de 10 microfaradios	2	\$0,50	\$1,00
Batería no recargable 9V con broche	1	\$2,10	\$2,10
Ventilador de 12V DC	1	\$3,00	\$3,00
Transistor	2	\$0,50	\$1,00
Mosfet	1	\$3,60	\$3,60
Servo motor 13 Kg	1	\$10,90	\$10,90
Servo motor	1	\$5,00	\$5,00
Módulo de sensor láser	1	\$2,50	\$2,50
Módulo relé de 5V	2	\$2,50	\$5,00
Módulo de cámara web para Arduino	1	\$7,50	\$7,50
Cable de conexión a la red eléctrica	2	\$2,50	\$5,00
Cargador de 12V 1ª	2	\$4,00	\$8,00
Módulo de carga para placa de prototipado 5V	2	\$2,50	\$5,00
Módulo de regulador de intensidad 110V	1	\$26,00	\$26,00
Nodemcu ESP32 wroom32	1	\$9,80	\$9,80
Diac	1	\$0,25	\$0,25
Triac bt139	1	\$1,25	\$1,25
Foco incandescente	1	\$1,50	\$1,50
Foco led	1	\$1,00	\$1,00
Rosetón para foco	2	\$1,50	\$3,00

Módulo receptor láser	1	\$10,00	\$10,00
		Subtotal	\$192,30
		IVA 0%	\$0,00
		Total	\$192,30

APÉNDICE D

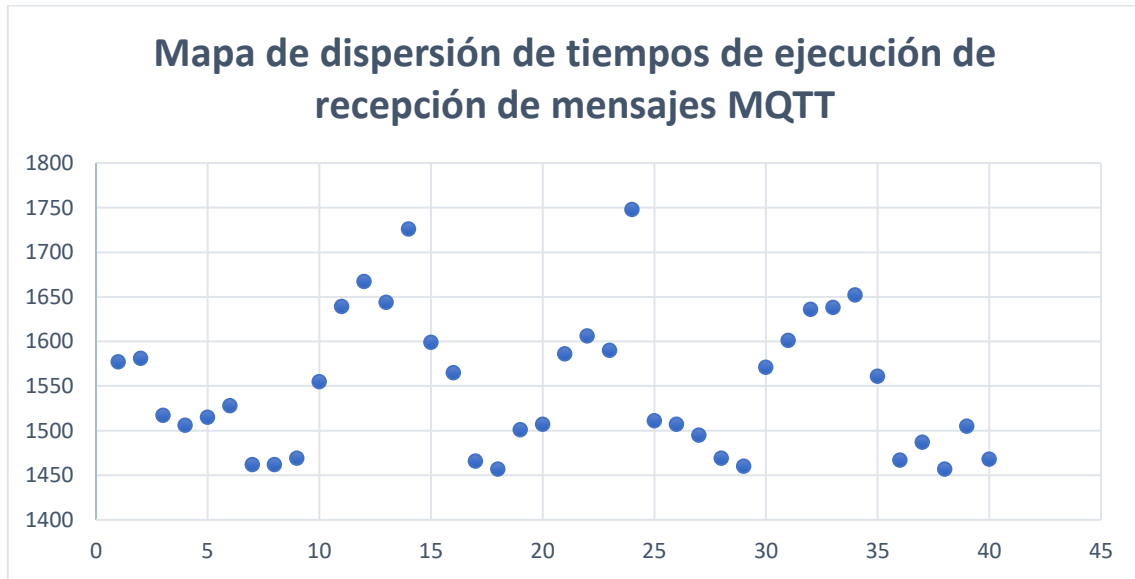


Figura D.1 Gráfica de dispersión de tiempos de ejecución en la comunicación MQTT

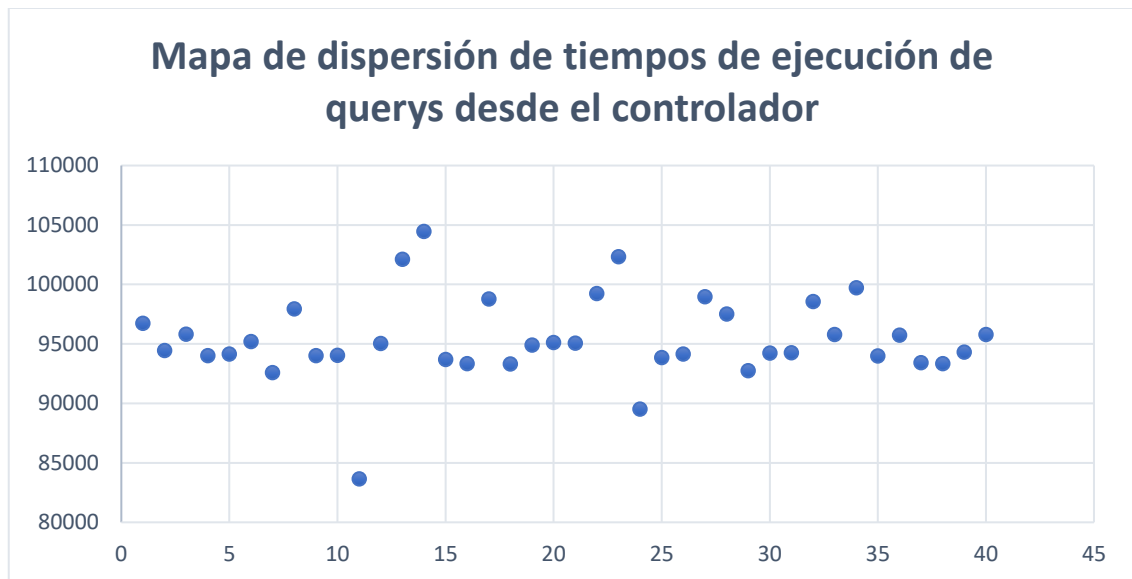


Figura D.2 Gráfica de dispersión de tiempos de petición de datos hacia la base de datos.

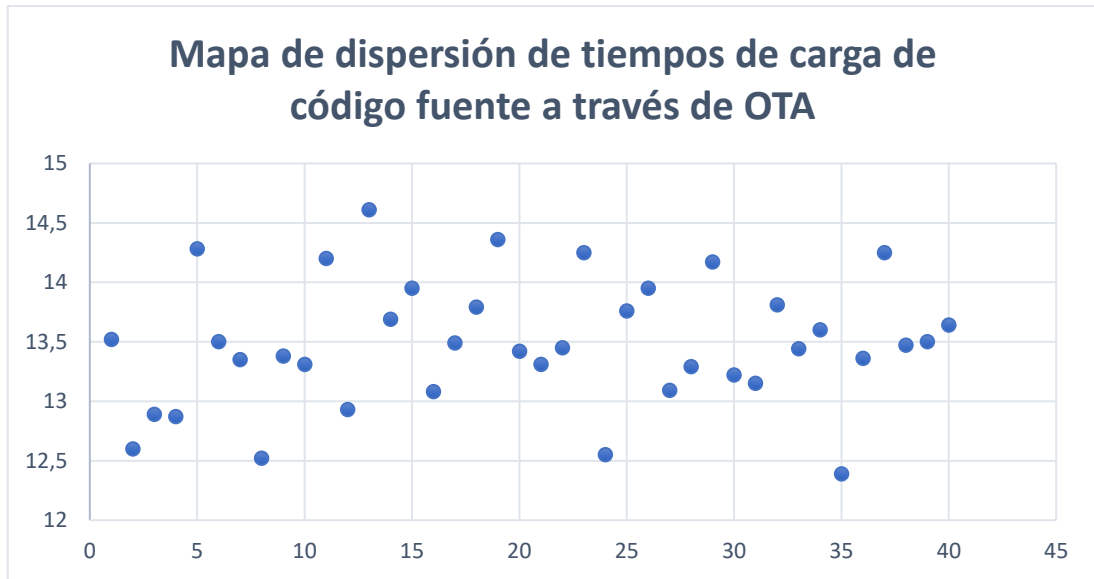


Figura D.3 Gráfica de dispersión de tiempos de la carga de código a las ESP TTGO a través de OTA

APÉNDICE E

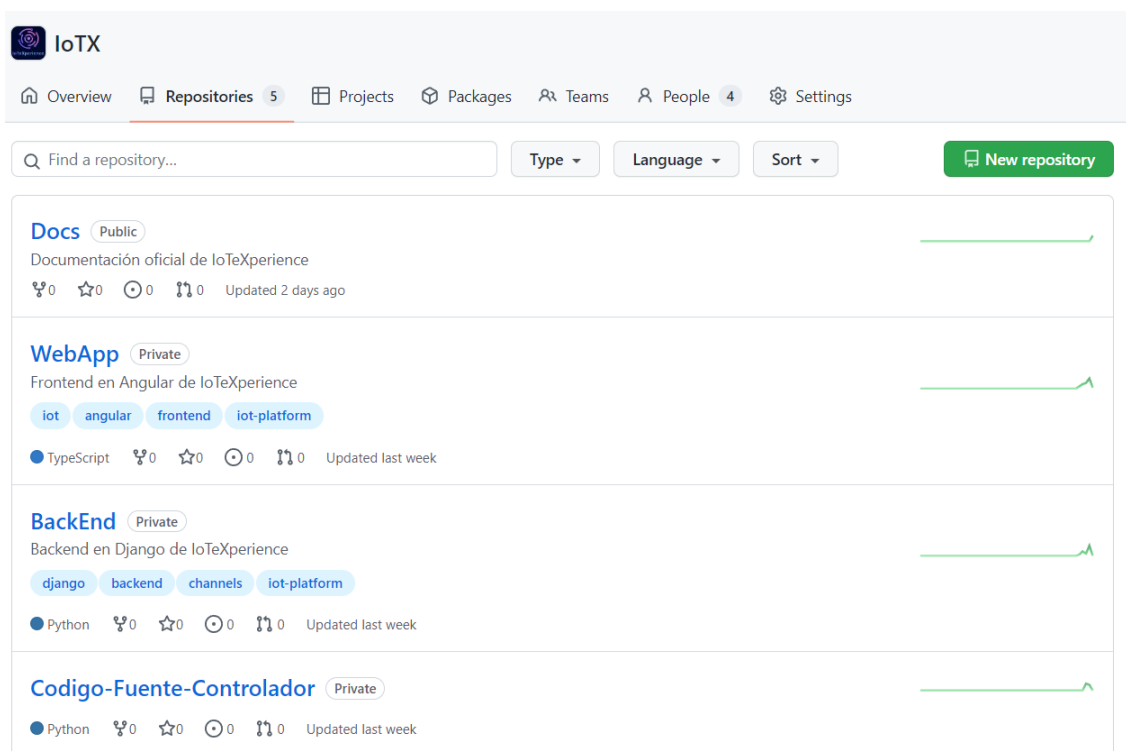


Figura E.1 Repositorio de GitHub IoTeXperience.

URL: [\[https://github.com/AcademyIoT/Docs\]](https://github.com/AcademyIoT/Docs)

APÉNDICE F



Figura F.1 Infraestructura del escenario de la casa inteligente

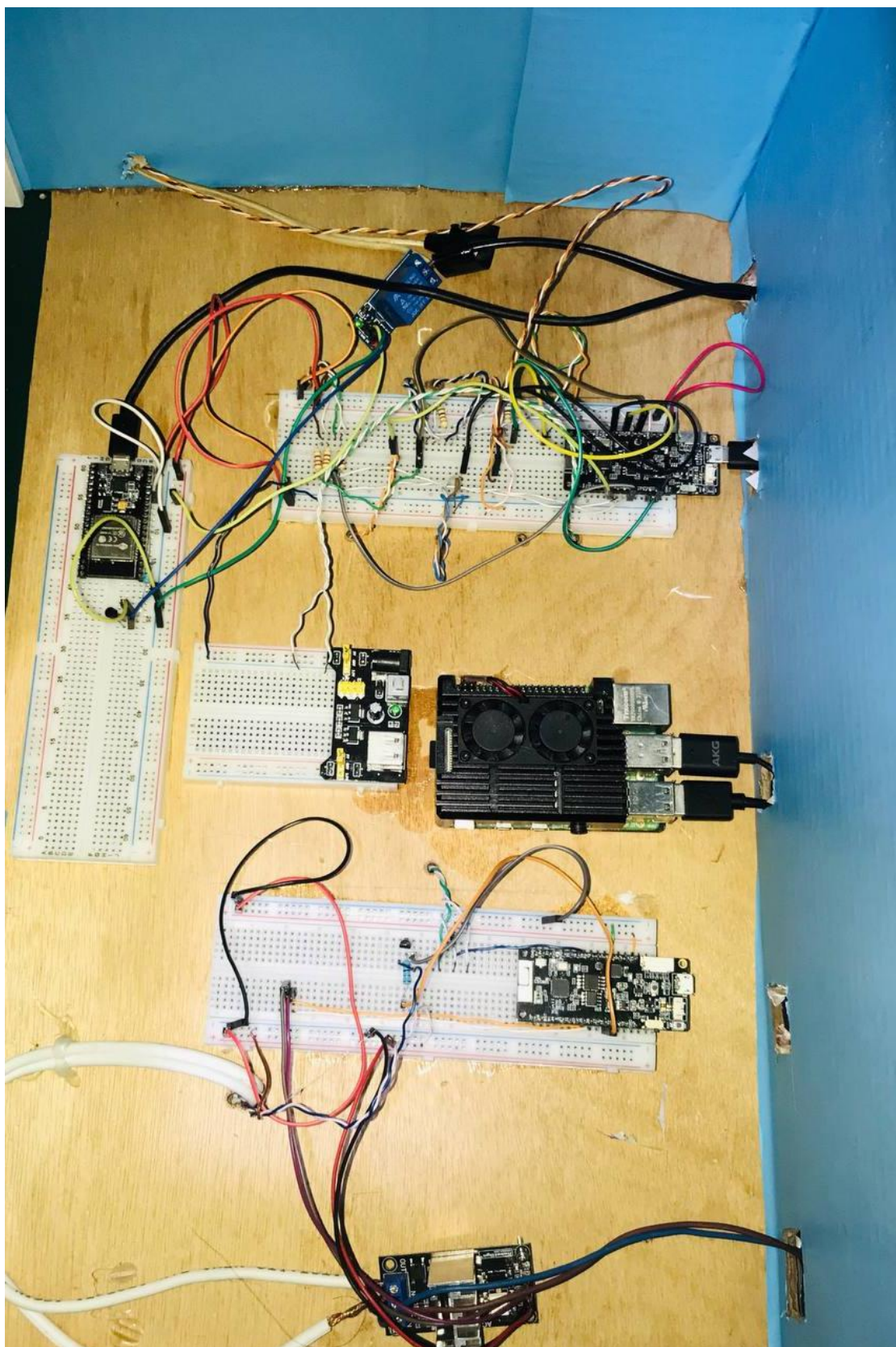


Figura F.2 Infraestructura de la zona de control



Figura F.3 Infraestructura del escenario de control de temperatura PID



Figura F.4 Fragmento de la gráfica temperatura vs tiempo luego de 180 segundos

APÉNDICE G

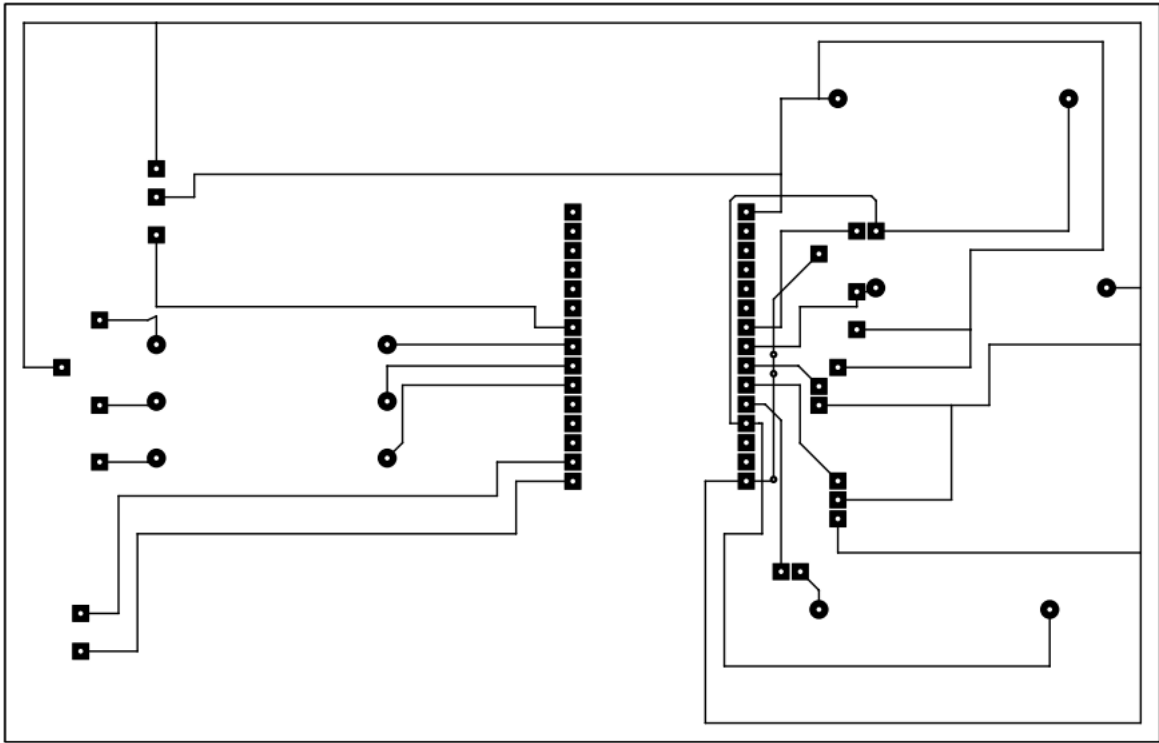


Figura G.1 Pista del diagrama esquemático del escenario casa inteligente en el PCB

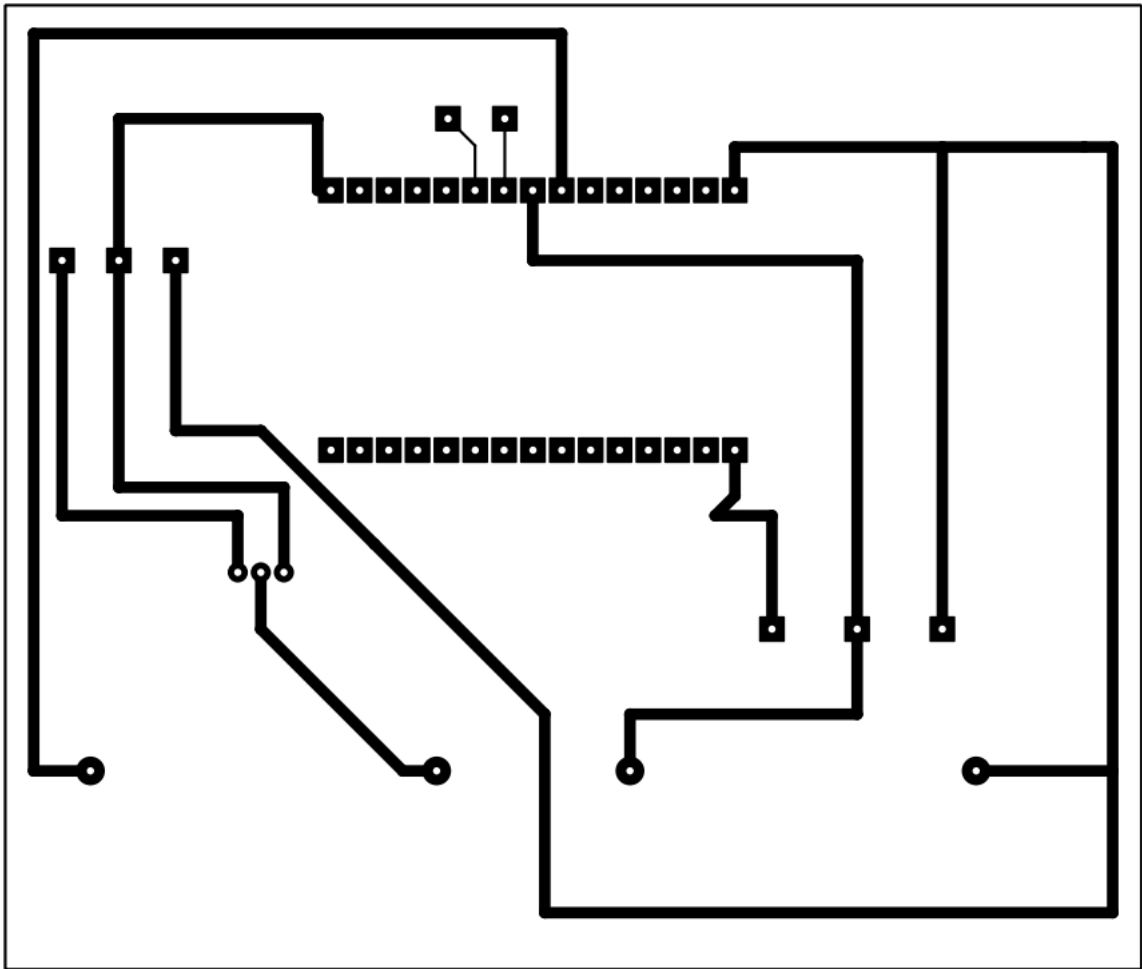


Figura G.2 Pista del diagrama esquemático del escenario control de temperatura PID en el PCB

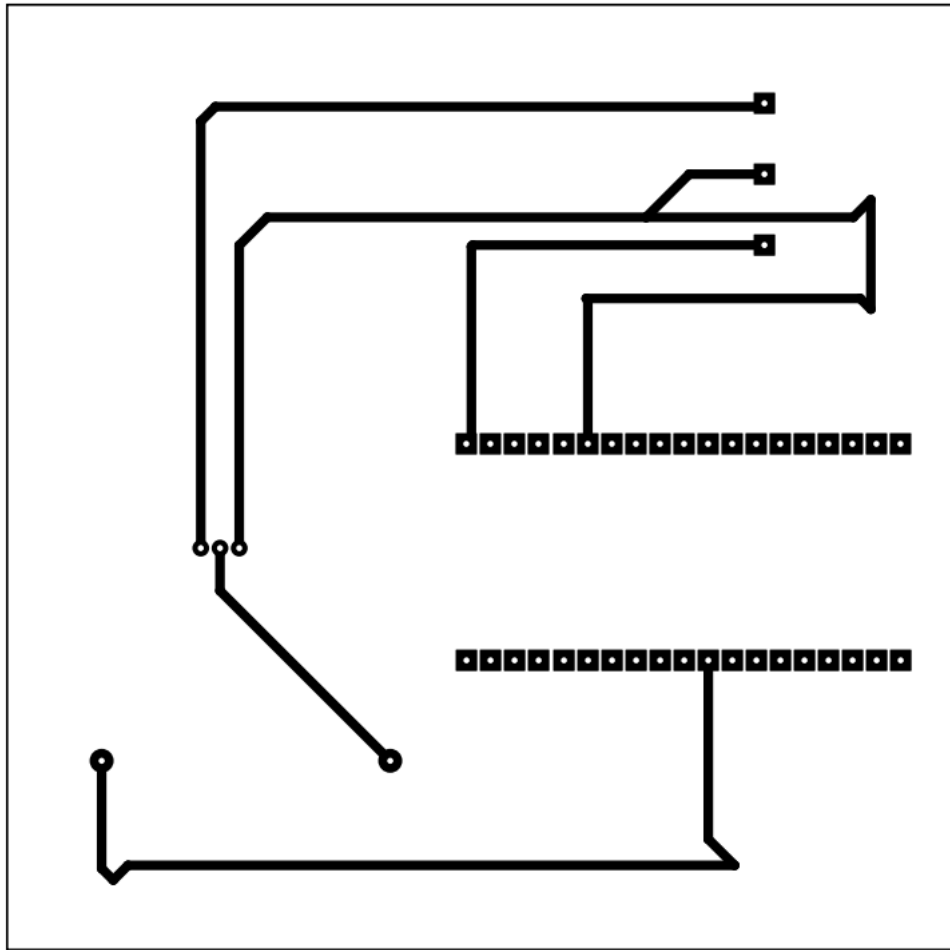


Figura G.3 Pista del diagrama esquemático del subcontrolador del escenario de casa inteligente en el PCB