



UNIVERSIDAD
NACIONAL
DE COLOMBIA

FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA Y
ELECTRÓNICA

JUAN CAMILO RAMÍREZ AYALA

TRABAJO DE GRADO

**LABORATORIO DE CONTROL PARA AU-
LA DE CLASE, CON DOS EXPERIMENTOS
INTEGRADOS AL INTERNET DE LAS COSAS**

Presentado como requisito para optar por el título de ingeniero
electricista

Universidad nacional de Colombia, sede Bogotá
2019



UNIVERSIDAD
NACIONAL
DE COLOMBIA

FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA Y
ELECTRÓNICA

DIRECTOR: M.Sc. LEONARDO ENRIQUE BERMEO CLA-
VIJO

JUAN CAMILO RAMÍREZ AYALA

TRABAJO DE GRADO

**LABORATORIO DE CONTROL PARA AU-
LA DE CLASE, CON DOS EXPERIMENTOS
INTEGRADOS AL INTERNET DE LAS COSAS**

Universidad nacional de Colombia, sede Bogotá
2019

Índice general

	Página
Índice de figuras	VII
Índice de tablas	VIII
Resumen	1
Introducción	2
Objetivos	3
1. Marco teórico	4
1.1. Internet de las cosas	4
1.2. Antecedentes en experimentos remotos	5
1.2.1. Laboratorios físicos	6
1.2.2. Laboratorios virtuales	10
2. Diseño e implementación	11
2.1. Diseño	11
2.1.1. Elementos hardware	12
2.1.1.1. Servidor	13
2.1.1.2. Controlador	14
2.1.1.3. Módulo WiFi	15
2.1.1.4. Hardware experimento Uno	16
2.1.2. Elementos software	18
2.1.2.1. Comunicación por SPI	19
2.1.2.2. Protocolo MQTT	19

2.1.2.3.	Lenguaje de programación Arduino	20
2.1.2.4.	JavaScript y Node.js	22
2.2.	Implementación	24
2.2.1.	Implementación hardware	24
2.2.1.1.	Experimento Uno	24
2.2.2.	Implementación software	25
2.2.2.1.	Servidor	26
2.2.2.2.	Módulo WiFi	28
2.2.2.3.	Controlador	28
2.2.3.	Implementación mecánica experimento 1	30
3.	Resultados	32
4.	Conclusiones	35

Índice de figuras

1.1. Arquitectura de tres niveles (IoT) [8]	5
1.2. GUI laboratorio levitador por aire [3]	7
2.1. Arquitectura general (Fuente propia)	11
2.2. Sensor láser [13]	17
2.3. Gráfico funcionamiento básico MQTT [16]	20
2.4. Esquemático, circuito acondicionamiento experimento uno (fuente propia) .	24
2.5. Shield Arduino circuito de acondicionamiento (fuente propia)	25
2.6. Arquitectura del servidor (fuente propia)	26
2.7. Interconexión del servidor (fuente propia)	27
2.8. Lazo de control experimento 1 (fuente propia)	29
2.9. Lazo de control experimento 2 (fuente propia)	30
2.10. Elemento 3D de la base del ventilador (fuente propia)	30
2.11. Elemento 3D del la base del tubo (fuente propia)	31
2.12. Elemento 3D del concentrador de aire (fuente propia)	31
3.1. Pantalla de inicio (fuente propia)	32
3.2. Página experimento 1 (fuente propia)	33
3.3. Página experimento 2 (fuente propia)	34

Índice de Tablas

2.1. Servidor [9]	14
2.2. Controlador [10]	15
2.3. Módulo WiFi [11]	16
2.4. Frameworks de node.js utilizados en el servidor	23

Resumen

El presente proyecto, cuyo objetivo es realizar un laboratorio de control con dos experimentos, se presenta como un trabajo realizado para reforzar los conceptos básicos de control enseñados en el aula de clase, permitiendo de manera interactiva la verificación tangible de lo aprendido en la asignatura. Se desarrolla una arquitectura escalable, que permite anexar más experimentos de manera modular, brindando una guía al estudiante, que le demuestra el funcionamiento de la arquitectura del servidor y el uso básico del laboratorio. Los resultados de los experimentos se grafican utilizando el navegador con el que cuenten los dispositivos móviles o computadores portátiles de los usuarios, así mismo, el servidor permite el envío de datos hacia los experimentos, realizando así una interacción completa entre el estudiante y el laboratorio.

Introducción

La enseñanza debe ir de la mano con la tecnología, es por ello que se hace vital realizar trabajos que brinden una conexión entre los nuevos tópicos y el mundo del control. El presente laboratorio, pretender ser una unión entre el internet de las cosas y el aprendizaje de control, brindando al estudiante la capacidad de interactuar desde sus dispositivos móviles, con dos experimentos en tiempo real, visualizando los resultados y las variaciones por medios gráficos desde el navegador del teléfono celular. Se estructura un servidor, que permite el uso de protocolos de comunicación escalables, esto con el fin de crear un experimento físico y uno virtual.

El usuario tiene la posibilidad de enviar cinco valores diferentes a los experimentos, tres constantes del PID, el periodo y la referencia, de esta manera se cumple el hecho de interactuar con las plantas respectivas. El laboratorio tiene la posibilidad de ser utilizado desde un sitio sin conexión a internet, ya que el servidor crea su propia red WiFi LAN, teniendo como principal obstáculo la distancia de los experimentos al servidor y la cantidad de usuarios conectados.

Objetivos

Objetivo general

Desarrollar un laboratorio con dos experimentos, integrados por medio del internet de las cosas, que permitan la interacción de los estudiantes y la comprensión de los conceptos básicos de control aplicados.

Objetivos específicos

- Desarrollar un sistema de levitación neumático, integrado con comunicación IoT e interfaz didáctica que permita el uso en el aula de clase.
- Implementar una planta virtual, simulando el control del nivel de agua en un tanque y que permita la interacción con los alumnos en el aula de clase.
- Generar una guía de uso para el estudiante, donde se expondrán los objetivos, metas y habilidades de aprendizaje obtenidos con los experimentos.
- Reforzar conceptos básicos de control en los estudiantes, como, tiempo de establecimiento, sobrepico, error y estabilidad.

Capítulo 1

Marco teórico

1.1. Internet de las cosas

El avance tecnológico brinda la posibilidad de interconectar elementos físicos a una red de internet, a este concepto se le denomina *internet de las cosas*. Formalmente se puede describir como 'Una red de elementos - cada uno integrado con sensores - que están conectados a internet [8]'. Abreviado normalmente IoT (internet of things) permite realizar el control de diversas variables a partir de elementos electrónicos tales como sensores y actuadores. El usuario puede obtener este tipo de control por medio de dispositivos que tengan la posibilidad de realizar una conexión a internet. Computadores y celulares son en general los elementos adecuados para tal fin.

Aprovechando esta tecnología y la posibilidad que brinda de interconectar elementos de medición y actuadores, se pueden generar modelos de sistemas e implementar conceptos de control para dominar las diferentes variables en estudio. La arquitectura básica de este concepto se puede tomar como una serie de tres niveles, interconectados entre si, los cuales se pueden apreciar en la siguiente figura.

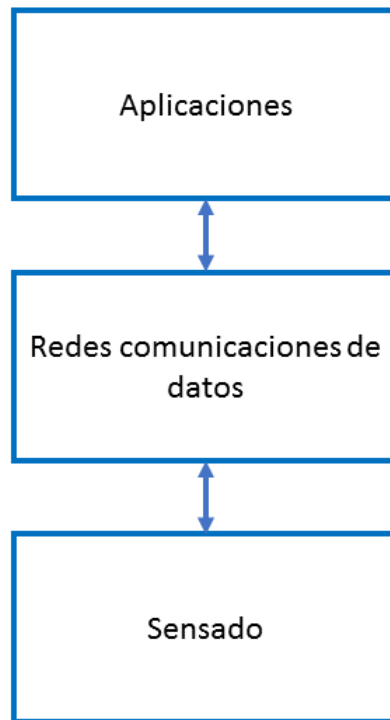


Figura 1.1: Arquitectura de tres niveles (IoT) [8]

Las aplicaciones brindan al usuario la posibilidad de interactuar con el sistema en sí, las redes de comunicaciones de datos realizan el intercambio de información entre el usuario y los elementos de sensado (en algunos casos también con actuadores) y finalmente el sensado es aquel nivel en el cual se encuentran los sensores y actuadores que permiten la interacción con el mundo físico.

1.2. Antecedentes en experimentos remotos

El uso de tecnologías web para la enseñanza en control es un tema en crecimiento, debido al aumento en las herramientas que brindan el acceso remoto de los laboratorios a distancia. De manera general, se pueden clasificar los laboratorios en dos grupos diferentes, para efectos de estudio en el presente trabajo:

- Laboratorios físicos
- Laboratorios virtuales

1.2.1. Laboratorios físicos

Los laboratorios físicos remotos brindan la posibilidad de utilizar elementos reales en sus experimentos, en estos casos los datos obtenidos se deben a variables medidas por sensores y a variaciones realizadas por actuadores en tiempo real. En la revisión de artículos, se encontraron proyectos realizados por diferentes grupos de investigación a lo largo de distintas universidades. Por ejemplo, el profesor emérito Sebastian Dormido Bencomo de la universidad nacional de educación a distancia de España, colaboró con la proyección del proyecto AutomatL@bs, una iniciativa en la cual se crea una red de laboratorios remotos para la enseñanza en automatización. En este proyecto se agrupan una serie de universidades españolas, las cuales integran sus recursos e infraestructura, brindándole al usuario la posibilidad de percibir el laboratorio como una estructura uniforme, a pesar de estar distanciados físicamente sus componentes. Se pueden desarrollar practicas que refuercen los siguientes cuatro conceptos [1]:

- Control en posición de motor mediante PID.
- Control en velocidad de motor mediante PID.
- Identificación del sistema.
- Control en el espacio de estado.

La propia universidad nacional de educación a distancia de España posee un laboratorio remoto, de tal manera que cada experimento puede ser manipulado vía web por los estudiante. Se encuentra a cargo del profesor Bencomo y cuenta con, tanques de líquidos, un sistema de rotación de helicóptero, un péndulo invertido y un sistema de roto-imán. [2].

Es importante aclarar que los proyectos anteriores poseen un alto nivel de complejidad en su construcción y un elevado costo, sin embargo, existen alternativas mas simples y económicas, como laboratorios de un solo experimento. Las soluciones presentadas en este grupo permiten ser replicadas sin mayor dificultad, siendo uno de los principales objetivos de su desarrollo. Esto se debe en gran medida al uso de dispositivos electrónicos de bajo costo, fácil adquisición e implementación; dispositivos como, Raspberry pi, Beaglebone Black, Arduino, NodeMCU, entre otros, permiten la construcción de experimentos sencillos y económicos con la capacidad de acceder a una red de internet.

La creación de un levitador de aire para la enseñanza de control, es un proyecto desarrollado para comprender los conceptos básicos de estabilidad, sobre pico y tiempo de establecimiento. Se implementó un laboratorio con este experimento utilizando como controlador un híbrido entre una Beaglebone Black y un Arduino, realizando el servidor en la misma placa y generando una interfaz de usuario amigable con el cliente. Este sistema consta de un sensor infrarrojo que mide la distancia de levitación del elemento, el cual es un disco realizado en impresión 3D que se encuentra adentro de un tubo de polietileno transparente, así mismo se crea una planta que sirve para realizar el acondicionamiento de las señales de entrada y salida y el control del ventilador, cuya función es generar la levitación del disco. Consta también de un componente software sobre el cual se realiza el servidor (Node.js), su conexión a internet, el desarrollo de la interfaz de usuario y el programa que controla el sistema [3].

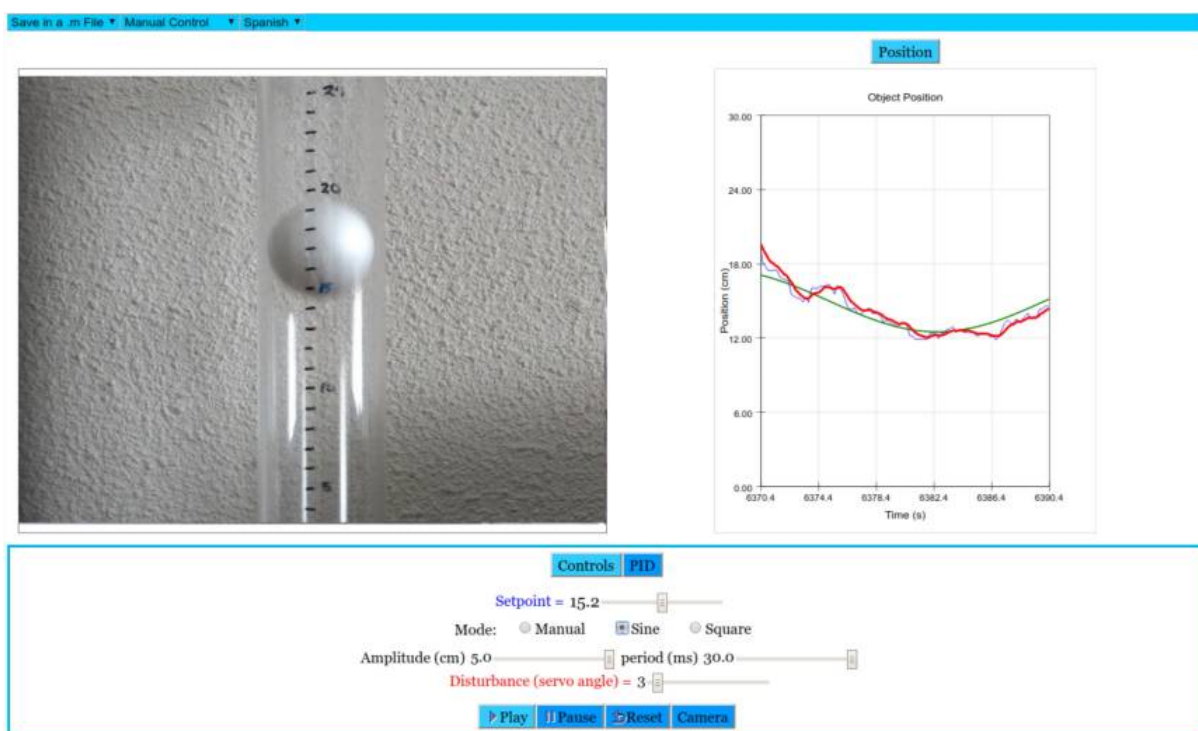


Figura 1.2: GUI laboratorio levitador por aire [3]

La escuela de ingeniería eléctrica y electrónica de la universidad de Bío-Bío en Chile posee un laboratorio de control automático, donde se incorporaron dos plantas marca Feedback, un tanque acoplado y un servomecanismo, pertenecientes al sistema de laboratorios a distancia (SLD). Para el sistema de servomecanismo se puede desarrollar:

- Identificación del motor ante entrada escalón.

- controlador proporcional (control velocidad).
- controlador proporcional-integral (control velocidad).
- Compensador en adelante (control posición).

Este sistema consta de un motor DC, cuya función de transferencia es la de un sistema de primer orden (la identificación se realiza con base en el primer ítem anterior), a partir de la cual se pueden realizar las diferentes tareas que el laboratorio permite. El sistema es accesible a partir de la página web del laboratorio y el diseño de control se realiza por medio de MATLAB y SIMULINK, teniendo de forma directa los diagramas de bloques listos para implementar sobre el servomecanismo. Los resultados se obtienen por medio de gráficas (velocidad y posición) y diagramas de Bode [5].

El departamento de arquitectura de computadores y control automático de la universidad complutense de Madrid ha diseñado una arquitectura de software nueva, de código abierto, la cual puede ser accesible desde cualquier dispositivo inteligente de manera remota, cuya principal capacidad es la flexibilidad de ser adaptada a diferentes experimentos y controladores. Para este tipo de arquitectura, la estructura básica consta de:

- Interfaz gráfica de usuario: Encargada de mostrar el comportamiento de la planta y el controlador, brinda la interacción entre el usuario y el servidor.
- Servidor: Provee acceso al experimento y maneja el intercambio de información entre la interfaz gráfica de usuario y el controlador.
- Controlador: Cierra el lazo sobre la planta del experimento. Posee sensores y actuadores, la planta es el punto físico sobre el cual se realiza el estudio del experimento.

El trabajo realizado en este proyecto se centra sobre la interfaz gráfica de usuario y el servidor, generalmente entre laboratorios remotos estos dos ítems son comunes, la variación se observa sobre la planta y el controlador, propios de cada experimento. Las herramientas utilizadas en cada una de las partes de la arquitectura del sistema se listan a continuación.

- Interfaz gráfica: JavaScript embebido en los navegadores, útil desde cualquier PC, Smartphone o tableta. Se utiliza el software Easy Java/JavaScript Simulations (EJSs).
- Comunicación cliente-servidor: Para la comunicación se utiliza el protocolo WebSocket, mas específicamente la librería socket.io de Node.js.

- Servidor: Se utiliza el entorno de ejecución de JavaScript, Node.js. También, se monta el servidor gracias a la librería Express.js del mismo entorno.

Sobre esta arquitectura se implementaron experimentos como, circuitos electrónicos, motores DC y rotores de cuadratura. Se desarrollaron controladores PID y se estudiaron temas como, variables de estado, linealización de la realimentación y control deslizante [6].

En la escuela de ingeniería eléctrica de la universidad de Belgrado se cuenta con un laboratorio denominado eLab, desarrollado con el fin de brindar la posibilidad a todos los estudiantes de ingresar a un experimento, sin necesidad de tener acceso físico a este. El servidor esta creado sobre un computador de escritorio, al cual se pueden conectar distintos clientes al tiempo, sin embargo, solamente uno puede controlar el experimento. Lo interesante de este laboratorio es que brinda al estudiante la posibilidad de acceder desde un computador en su hogar por internet o directamente desde una red LAN en el mismo sitio físico del laboratorio. eLab consta de los siguientes elementos:

- Computador: Hace las veces de servidor y se encarga de la comunicación entre la tarjeta de adquisición de datos y las redes (LAN o internet).
- Tarjeta multifunción NI (AD/DA): Tarjeta de National Instruments utilizada para adquirir datos de temperatura e interconectar estos datos con el computador. Referencia NI DAQ-6221 (AD/DA).
- Proceso de temperatura: Equipo de laboratorio, que se puede ver como la planta física del experimento educativo.
- LabView software: Software utilizado para visualizar los datos y el proceso de control, se interconecta con el navegador web.

En el laboratorio se espera trabajar sobre fenómenos básicos relacionados con transferencia de calor, tiempo de establecimiento, realimentación negativa de lazo cerrado, exactitud en medidas y efectos no lineales. Así mismo, se espera trabajar sobre dos enfoques de control diferentes, controladores PID clásicos y control predictivo, ambas herramientas utilizadas desde LabVIEW, cada una con su respectivo diagrama de bloques de proceso [7].

1.2.2. Laboratorios virtuales

Existe una variante en la cual los elementos no existen físicamente, en estos casos se trata sobre modelos matemáticos de los sistema a estudiar, esto significa que los usuarios acceden directamente a una simulación de los experimentos. De acuerdo a la naturaleza de estos laboratorios, es importante recalcar las herramientas que permiten la creación de los diferentes experimentos, especialmente los programas computacionales.

LabVIEW ofrece un ambiente de desarrollo gráfico, su lenguaje de programación, llamado G, posee una amplia variedad de librerías que permiten adquisición y tratamiento de datos. Cuenta con una herramienta denominada *LabVIEW remote panel*, que ofrece al usuario la posibilidad de utilizar sistemas en línea y convertir el desarrollo a una variante virtual, sin conocimientos profundos de programación. Se pueden modelar diferentes tipos de sistemas, y sobre estos diseñar controladores, que permiten manipular el comportamiento de la salida del experimento. LabVIEW permite la creación de sistemas mecánicos y eléctricos, mecánicos como por ejemplo, un sistema masa-resorte amortiguado, para el cual se desarrolla un controlador proporcional (P), un controlador proporcional integral (PI) y un controlador proporcional integral derivativo (PID), con el fin verificar de acuerdo al tipo de controlador comportamientos como estabilidad, sobrepico, tiempo de establecimiento, etc. Así mismo, el software entrega la posibilidad de observar el diagrama de polos y ceros, el lugar de las raíces, el diagrama de Nyquist y el diagrama de Bode. Como sistema eléctrico se puede modelar un circuito RLC serie, que brinde la posibilidad de entender los mismos conceptos del sistema mecánico, en términos de definiciones básicas de control y electrónica [4].

Capítulo 2

Diseño e implementación

2.1. Diseño

La estructura general del laboratorio se puede dividir en 4 partes: controlador, módulo WiFi, servidor y usuario. En la siguiente figura se puede apreciar de forma clara la arquitectura desarrollada.

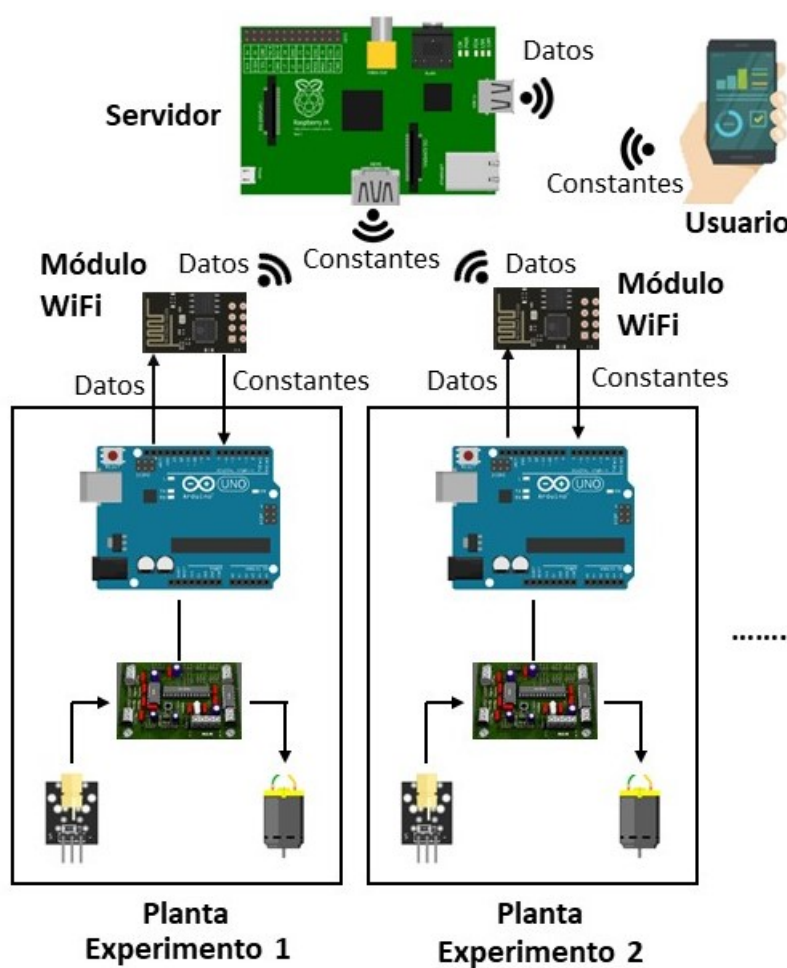


Figura 2.1: Arquitectura general (Fuente propia)

En la etapa de diseño se define que función cumple cada una de las partes nombradas anteriormente, y como estas aportan a la etapa siguiente o a la anterior.

- **Planta:** Brinda la posibilidad de control sobre el experimento, gracias al controlador diseñado y programado en la placa. Necesita de un circuito de acondicionamiento que permita el uso de aquellas señales que el controlador no puede manejar de forma directa, es decir, señales de sensores y actuadores, que necesitan ser acondicionadas antes de ingresar o salir del dispositivo.
- **Módulo WiFi:** Puente de comunicación entre el controlador y el servidor.
- **Servidor:** Eje central del laboratorio, maneja la interfaz de comunicación entre los datos provenientes de las señales inalámbricas, enviadas tanto por el usuario como por el módulo WiFi permitiendo la comunicación entre estos dos puntos.
- **Usuario:** Cliente del laboratorio, es allí donde se generan las gráficas con los datos provenientes del controlador y donde se hace el envío de constantes que alteran el experimento.

Es importante aclarar que la arquitectura esta diseñada de tal manera que si se realizan otros experimentos (con la misma estructura controlador-módulo Wifi), el laboratorio podrá ser escalable y ejecutarse para nuevos sistemas con un mínimo de agregados al servidor.

Para dar funcionamiento a cada una de las partes mostradas con anterioridad, existe una interacción entre dos tipos de elementos, componentes software y componentes hardware. En el diseño del laboratorio se definen que tipos de componentes se van a utilizar, verificando sus ventajas y sus desventajas, así como su utilización dentro del proyecto.

2.1.1. Elementos hardware

Como hardware se entenderán las partes físicas electrónicas a utilizar en el proyecto, se tienen como estos elementos:

- Sensores
- Actuadores
- Circuitos acondicionamiento

- Controlador
- Módulo WiFi
- Servidor

Los sensores, actuadores y circuitos de acondicionamiento son propios del experimento uno, por ende, serán tratados en su apartado correspondiente. En las siguientes tablas se resumen las características principales de los elementos seleccionados como controlador, módulo WiFi y servidor, indicando a su vez la función que cumplen, sus ventajas y desventajas.

2.1.1.1. Servidor

El dispositivo debe estar en capacidad de realizar las siguientes funciones:

- Generar el servidor Web al cual se van a conectar los usuarios, desplegando las funciones básicas de conexión y desconexión.
- Manejar los datos enviados por el usuario y por el módulo WiFi, brindando la comunicación entre dispositivos y el correcto direccionamiento de la información.
- Crear la red local para la conexión de los módulos WiFi y de los usuarios.
- Diferenciar los protocolos de comunicación utilizados entre dispositivos, teniendo la capacidad de manejarlos todos de forma correcta.
- Graficar en el lado del cliente los resultados expuestos para visualización.

Existen diferentes tipos de placas similares a computadores de pequeño tamaño, sin embargo, Raspberry posee la ventaja de tener un gran número de usuarios, lo cual brinda la posibilidad de un mayor y mejor soporte ante cualquier dificultad que se pueda presentar. Así mismo, a diferencia de otras placas, Raspberry contiene como sistema operativo principal Raspbian (distribución de GNU/Linux), basado en Debian, que contiene un repositorio completo para las diferentes tareas a realizar, lo cual facilita la instalación de paquetes y software útil en el proyecto.

La placa utilizada (Raspberry pi 3B+) posee una antena WiFi, capaz de ser utilizada como puerto de conexión, lo cual significa que se puede generar una red local inalámbrica, base importante en el proyecto ya que puede ser útil sin tener una red de internet comercial. La

principal desventaja de este dispositivo es su precio, claramente existen placas mas económicas e igualmente útiles como servidor, sin embargo, no poseen la antena WiFi incluida y se debe realizar la compra por separado (por ejemplo, Raspberry Pi Zero).

Características	Raspberry Pi 3B+
SoC	Broadcom BCM2837
CPU	1.4GHz 64-bit quad-core ARMv8
RAM	1 GB (compartidos con la GPU)
Puertos USB 2.0	4
Almacenamiento	SD / MMC / ranura para SDIO
Conectividad de Red	Puerto RJ-45 (Ethernet), Wi-Fi 802.11ac
Conectividad inalámbrica	Bluetooth 4.2 BLE
Periféricos de conexión	17 x GPIO
Consumo energético	800 mA (4.0 W)
Fuente de alimentación	5 V vía Micro USB
SO	Raspbian

Tabla 2.1: Servidor [9]

2.1.1.2. Controlador

El controlador debe estar en capacidad de realizar las siguientes funciones:

- Comunicarse, enviar y recibir datos de manera alámbrica del módulo WiFi, así como enviar los datos al circuito de acondicionamiento y los elementos conectados.
- Tener capacidad de procesamiento suficiente para ser parte del lazo de control, en el controlador se programa el controlador PID, se reciben las señales del sensor y se envían las señales al actuador, así que debe ser capaz de manejar estos datos sin perdida.
- Poseer código reutilizable, ya que gran parte de las funciones para el uso de los sensores y actuadores se encuentran en librerías ya creadas.

De acuerdo a las necesidades anteriores, se observa que las placas Arduino cumplen con las especificaciones solicitadas, en vista de la enorme cantidad de usuarios y foros en internet, así como las librerías para el uso de sensores y actuadores, posee ventaja en gran

medida, apoyada por el amplio soporte de los dispositivos. La programación de los dispositivos Arduino se puede hacer fácilmente por medio del IDE oficial, utilizando el lenguaje de programación propio de las placas (muy similar al lenguaje C). Las placas son económicas y se pueden comprar fácilmente en tiendas de electrónica, tanto placas oficiales de Arduino como no oficiales, teniendo en cuenta que es un proyecto OpenSource. Es por lo anterior que se selecciona la placa Arduino UNO, con las características presentadas en la tabla correspondiente a este ítem.

La principal desventaja de esta placa es la poca memoria flash que posee (donde se alojan los programas), esto limita las capacidades ya que se debe tener en cuenta que las diferentes librerías, el diseño del controlador y los protocolos de comunicación, exigen cada uno cierta porción de la memoria para su funcionamiento. Es importante recalcar que para el escalamiento del laboratorio, las restricciones en memoria del Arduino son importantes, cada experimento debe contar con su controlador propio.

Características	Arduino UNO
Microcontrolador	ATmega328P
Voltaje operación	5V
Voltaje alimentación	6-20 V
Digital I/O	14
Salidas PWM	6
Memoria flash	32 KB
SRAM	2 KB
EEPROM	1 KB
Velocidad de reloj	16 MHz

Tabla 2.2: Controlador [10]

2.1.1.3. Módulo WiFi

El módulo WiFi debe estar en capacidad de:

- Manejar los protocolos de comunicación al enviar datos, tanto de manera alámbrica como inalámbrica.
- Realizar el envío de datos de forma correcta desde el controlador al servidor, teniendo

en cuenta que no debe haber pérdida de información.

El NodeMCU permite ser programado desde el IDE oficial de arduino y con el mismo lenguaje de programación, por ello es una alternativa con una gran cantidad de usuarios en internet, también, contiene un enorme componente de librerías con distintos protocolos de comunicación, útiles para el laboratorio.

La simplicidad de funcionamiento de esta placa la hace adecuada para el proyecto a realizar, es económica, y se puede comprar con facilidad en cualquier tienda de electrónica. Su microcontrolador es veloz y posee una memoria flash de tamaño considerable, por lo cual puede realizar diferentes tareas y manejar los datos de manera correcta. Es importante aclarar que esta tarjeta es un kit de desarrollo, es un SoC montado sobre una placa que posee acceso simple a los pines de entrada y salida tanto digitales como analógicos. La antena WiFi viene incluida en el kit.

Su principal desventaja es la gran cantidad de fabricantes existentes, entre si pueden tener variaciones del mismo kit de desarrollo, haciendo que el funcionamiento varíe en algunos casos.

Características	NodeMCU
SoC	ESP8266EX
Microcontrolador	Tensilica L106 32 bits
Velocidad de reloj	160 MHz
Conectividad de red	WiFi b/g/n 2.4 GHz
Digital I/O	9 GPIO
Memoria flash	4 MB
RAM	50 KB

Tabla 2.3: Módulo WiFi [11]

A continuación se dará una breve introducción a los elementos hardware de cada laboratorio.

2.1.1.4. Hardware experimento Uno

Como experimento uno se tiene un levitador neumático, el cual contiene un sensor láser, un ventilador (motor DC) y un circuito de acondicionamiento, compuesto de diferentes ele-

mentos electrónicos.

- **Sensor láser:** Se utiliza el sensor de distancia láser VL53L0X, cuya distancia máxima es de 2 metros y distancia mínima de 5 cm. Posee dos ventajas que lo hacen diferente a otro tipo de sensores; es capaz de trabajar en un ambiente con elevada luz (ya sea visible o infrarroja), y la medición no se verá afectada por la forma física del objeto, que en ocasiones producen eco o reflectancia del haz. El funcionamiento del sensor se basa en el principio ToF (time of light), en el cual el sensor emite un haz de luz y cuenta el tiempo que este tarda en volver [12].

El emisor láser envía una señal de luz de 940 nm, que una vez reflejada es recibida por un receptor SPAD (Single Photon Avalanche Diodes). El protocolo de comunicación que utiliza el sensor es I2C, así mismo, existe una librería desarrollada por el fabricante principal de este sensor (polulu) que facilita el uso con Arduino sin preocuparse por el método de comunicación ni tampoco por el funcionamiento interno del sensor [13].

Al ser un sensor por haz, posee una desventaja y es la dependencia del color del cuerpo, la precisión en la medición en las distancias máximas y mínima dependen del color del cuerpo, siendo mejor para colores claros. El ángulo de apertura de medición es corto, a un metro el sensor tiene un ángulo de aproximadamente 25° , lo que significa que los objetos tienen que situarse prácticamente frente al emisor/receptor [12]. En la siguiente figura se puede verificar el tamaño del sensor (aproximadamente tiene un área de 1 cm^2).

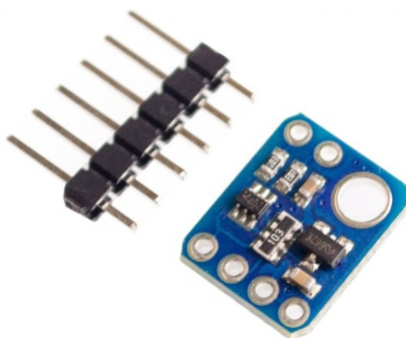


Figura 2.2: Sensor láser [13]

- **Ventilador:** Se utiliza el ventilador de referencia AVC DASA0925B2S, el cual funciona con una tensión nominal de 12 V y una corriente de 2 A. El ventilador está compuesto por un motor DC, por ende, se puede controlar la velocidad de dicho motor

por medio del cambio en la tensión de entrada, esta variación se realiza por modulación de ancho de pulso (PWM).

- **Circuito de acondicionamiento:** Para controlar la tensión de entrada del ventilador se necesita un circuito que permita el manejo de la señal proveniente de la fuente de alimentación, para ello se utiliza un transistor MOSFET IRF630B, que por medio de la conmutación que se presenta en el gate, varia la tensión en el drain, es por ello que a esta terminal se le conecta el ventilador (en antiparalelo a un diodo).

Para realizar la conmutación del transistor se utiliza el controlador (Arduino UNO), quien tiene la posibilidad de emitir en algunas de sus terminales modulación por ancho de pulso (PWM), sin embargo, esta señal emitida por el controlador no se puede conectar directamente al gate del transistor, para ello se utiliza un dispositivo optoacoplador (6N137) que desacopla los pines de salida del arduino con cualquier señal de potencia que se pueda presentar en el circuito de acondicionamiento. Entre el transistor y el optoacoplador se conecta un driver MOSFET (MC33152PG) que permite el paso a los cambios rápidos provenientes de la conmutación de la señal PWM sin la pérdida de pulsos generados.

2.1.2. Elementos software

Como software se entenderán aquellos paquetes, lenguajes de programación y elementos computacionales no físicos que permiten el desarrollo del proyecto. Las placas electrónicas utilizadas necesitan algún tipo de software para su funcionamiento, especialmente el servidor, cuyo desarrollo fue gracias a los paquetes provenientes del entorno utilizado. Se dará una breve introducción a cada uno de los ítems software, teniendo en cuenta especialmente los paquetes y librerías útiles en el desarrollo del proyecto. En este apartado no se diferencian los experimentos, ya que los elementos software utilizados son los mismos sin importar el tipo de experimento realizado. Se tiene:

- Comunicación por SPI
- Protocolo MQTT
- Lenguaje de programación Arduino
- Node.js

2.1.2.1. Comunicación por SPI

A pesar que el proyecto no se basa en este protocolo de comunicación entre dispositivos, si es importante dar una muy breve explicación sobre su funcionamiento, esto con el fin de entender como se realiza la comunicación entre el Arduino y el NodeMCU. Este protocolo funciona de manera síncrona, es decir, los datos se envían de manera ordenada de acuerdo a una señal de reloj. Es un protocolo serial utilizado en su mayoría para comunicar microcontroladores con elementos externos a corta distancia. Para llevar a cabo la comunicación entre dispositivos, se debe tener un elemento como maestro y uno o mas esclavos, donde el maestro controla las funciones de comunicación entre periféricos. Para realizar la comunicación con este protocolo se deben tener las siguientes líneas en conexión [14]:

- MISO (Master Input Slave Output): Como su nombre lo indica es la línea que permite al maestro recibir los datos que provienen de algún esclavo.
- MOSI (Master Output Slave Input): Contrario a la línea anterior, esta línea permite el ingreso de datos al esclavo que provienen del maestro.
- SCK (Reloj): Señal de reloj que permite la sincronización en el envío y recepción de datos.
- SS (Slave Select): Pin que permite al maestro seleccionar entre los diferentes esclavos.

2.1.2.2. Protocolo MQTT

MQTT es un protocolo de envío de mensajes, basado en el principio cliente/servidor, el cual utiliza subscripciones y publicaciones sobre una serie de tópicos. Estos tópicos son una especie de grupos sobre los cuales se inscriben diferentes clientes y cada vez que uno de ellos envía un mensaje a un tópico, el resto de dispositivos inscritos a este recibirán dicho mensaje. Este protocolo es ampliamente utilizado en los proyectos que utilicen el concepto de internet de las cosas, ya que brinda una escalabilidad importante, si se desea anexar un nuevo cliente simplemente se realiza la conexión de este al tópico deseado y se completaría la conexión. Para MQTT el servidor recibe el nombre de broker, este último puede ser montado sobre TCP/IP o sobre cualquier otro medio de comunicación bi-direccional, como por ejemplo WebSockets [15]. Para comprender de forma clara como es el modelo de MQTT se puede observar en la siguiente figura un breve diagrama del modo de funcionamiento sobre IOT.

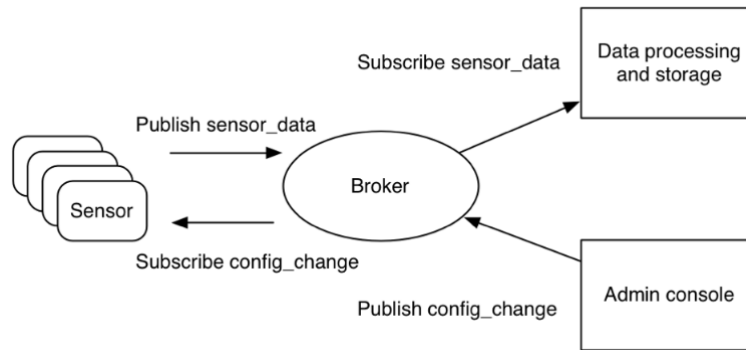


Figura 2.3: Gráfico funcionamiento básico MQTT [16]

En el presente trabajo, el uso de MQTT es vital, ya que es gracias al protocolo que el servidor (broker MQTT) puede diferenciar los experimentos realizados, conectando cada experimento a una serie de tópicos diferentes. Finalmente se entregan una serie de ventajas a tener en cuenta.

- Protocolo altamente escalable
- Transporte de mensajes transparente, reduce el tráfico en la red sobre la que se utiliza.
- Posee QoS (calidad del servicio) que brinda confiabilidad sobre el envío y la recepción de mensajes.
- Puede notificar la desconexión instantánea entre el cliente y el broker.

2.1.2.3. Lenguaje de programación Arduino

Lenguaje de programación basado en C++, que posee características propias para el desarrollo en los microcontroladores utilizados para las placas Arduino. Es un lenguaje libre cuya principal característica es la gran cantidad de librerías existentes para diferentes elementos, tales como sensores, motores, protocolos de comunicación, etc. El IDE oficial de Arduino permite la programación de diferentes placas con este lenguaje, incluyendo el kit de desarrollo utilizado para el presente proyecto como módulo WiFi (NodeMCU). A continuación se presentan las librerías utilizadas para el desarrollo del proyecto, aclarando que se pueden instalar desde el IDE oficial de arduino.

- **SPI.h y SPISlave.h:**

Librerías utilizadas para realizar la comunicación cableada entre el Arduino y el NodeMCU. En la placa arduino se utiliza la librería SPI.h, la cual hace las veces de maestro en el protocolo utilizado, es decir, es quien envía los datos al dispositivo esclavo, indicando en que momento se inicia la comunicación y en que momento se termina. En el NodeMCU se instala la librería SPISlave.h, que como su nombre lo indica trata al dispositivo como un esclavo SPI, cada vez que el maestro inicia la comunicación y envía una trama de datos, el dispositivo esclavo envía una respuesta, que en la mayoría de los casos son los datos provenientes del usuario enviados a través del servidor.

■ **ESP8266WiFi.h:**

Es la librería para el funcionamiento básico del NodeMCU, permite el uso de la antena WiFi y su conexión a un red de internet inalámbrica, así como la asignación de una IP estática y otras funciones de red. Gracias a este paquete de funciones se puede utilizar el lenguaje de programación de Arduino en el NodeMCU.

■ **PubSubClient.h:**

Librería utilizada en el NodeMCU, permite el uso de MQTT en el kit de desarrollo. Esta librería entrega de forma clara una serie de funciones que brindan la conexión al broker y a diferentes tópicos creados en el mismo, también posee una función que indica si ocurre una desconexión e intenta reconectar el cliente nuevamente cada cierto periodo de tiempo (indicado por el programador). Por medio de un callback (función asíncrona, significa que no se sabe en que momento se va a ejecutar en el ciclo normal del programa) recibe los mensajes provenientes del broker, en cuyo caso lo hace de acuerdo al tópico conectado.

■ **Librería sensor láser y PWM:**

También se utilizan un par de librerías para el primer experimento. La librería entregada por polulu para el sensor láser brinda una serie de funciones de recepción de datos, incluyendo dentro de si el funcionamiento necesario para el protocolo I2C, así como las configuraciones básicas del sensor. Simplemente permite seleccionar dos opciones de medición distintas, una para mayor precisión pero menor velocidad u otra para mayor velocidad pero menor precisión. Por medio de una función se reciben los datos y se guardan en una variable de punto flotante para su posterior uso.

En el caso del PWM, se maneja por interrupciones, es por ello que se necesita el uso de una librería que permita interrupciones por software, ya que es necesario que cada cierto tiempo el flujo del programa se detenga, ejecute el pwm y siga su ciclo normal. Esto con el fin de no utilizar retrasos (delay) que paran totalmente el microcontrolador y no permiten el uso de otras funciones en un determinado tiempo. El PWM utilizado es de 10 bits, que permite 1024 valores diferentes de salida.

2.1.2.4. JavaScript y Node.js

■ JavaScript:

JavaScript es un lenguaje de programación que permite ejecutar actividades y funciones complejas en una página web. Estas actividades brindan una interacción dinámica por parte del usuario con la página, dando la posibilidad de visualizar gráficas, interactuar con botones, slides, etc. JavaScript como lenguaje de programación, es asíncrono ejecutado en el navegador, esto significa que el código cliente alojado en el servidor no se está ejecutando allí, solamente se guarda, este código en realidad será ejecutado en el navegador del usuario que ingrese a la página web. Una pista importante para el uso de JavaScript es que este es utilizado sobre HTML, lo que significa que el código se utiliza sobre un script que es lanzado por HTML.

Para el presente proyecto, JavaScript es el lenguaje de programación base utilizado sobre el servidor, sin embargo, como se comentó en el párrafo anterior, JavaScript corre de forma nativa sobre el cliente en el navegador, para poder solventar esto se utiliza Node.js. Finalmente, para el funcionamiento del laboratorio, es importante aclarar que este lenguaje es asíncrono, lo que significa que el ciclo de ejecución común no se realiza de manera lineal, ciertas funciones en JavaScript son ejecutadas en el momento en que son llamadas por el navegador mas no en el orden en el cual se encuentran en el código [17].

■ Node.js:

Node.js se puede ver como un entorno de programación que fue creado para ser utilizado con JavaScript. Su principal función es desarrollar aplicaciones para la red basadas en eventos asíncronos, de ahí su importancia como entorno de programación web, ya que no posee un hilo de programación concurrente con entradas y salidas que puedan generar paradas en

la ejecución del programa, problema que en el ámbito de las aplicaciones para internet es muy común.

El lanzamiento de las funciones de los servicios de node.js se hace por eventos, donde por medio de callbacks se busca la ejecución de estos modelos predefinidos. En un caso de programa común de node.js, al momento de correr la aplicación, se inicia el bucle de eventos que se detiene una vez los callbacks del programa se han ejecutado por completo [18].

Para el presente proyecto se utiliza node.js como entorno de programación para JavaScript, se tomo esta decisión por el hecho de contar con una gran cantidad de librerías y frameworks que facilitan el trabajo de programación del lado del servidor, a continuación se enlistan los utilizados y se entrega una breve explicación de sus funciones.

Framework	Utilidad
bodyParser	Analiza gramaticalmente los request hechos por el usuario, prepara los datos enviados por el navegador para su posterior uso
chart.js	Utilizado para realizar gráficas en el lado del cliente, permite mostrar datos de manera simple con diferentes adecuaciones
ejs	Permite visualizar templates de JavaScript de forma embebida, genera diferentes páginas de acuerdo a la url indicada con HTML
express	Crea servidores de manera simple, permite el direccionamiento de url's de forma sencilla, se puede interconectar con WebSockets para envío de datos
mosca	Framework necesario para utilizar la Raspberry pi como un broker mqtt, genera una dirección IP propia
mqtt	Por medio de este paquete se puede realizar la conexión de los clientes mqtt a los topics creados en el broker
socket.io	Permite utilizar WebSockets en el servidor, esto con el fin de mantener una comunicación estable y permanente entre el cliente y el servidor para el envío y recepción de datos
path	Facilita el enrutado de archivos dentro del mismo servidor, es decir, facilita el llamado de archivos que están en las diferentes carpetas internas

Tabla 2.4: Frameworks de node.js utilizados en el servidor

2.2. Implementación

De acuerdo a los elementos hardware y software definidos en la sección de diseño, se procede a la realización tanto del servidor como los experimentos. A continuación se indicará de forma muy breve como se construyo el proyecto con los ítems nombrados anteriormente, teniendo en cuenta que los códigos y las guías estarán alojadas en GitHub donde se entenderá de manera mas clara lo realizado.

2.2.1. Implementación hardware

La principal implementación hardware del laboratorio se aprecia en la construcción de los circuitos de acondicionamiento, ya que el servidor, el módulo WiFi y el controlador, son manejados por medio de los programas software diseñados para su uso.

2.2.1.1. Experimento Uno

En el apartado anterior se explicaron los componentes que incluye el circuito de acondicionamiento y su función principal, sin embargo, no se indicaron las conexiones entre estos elementos ni los detalles de su construcción. En la siguiente figura se puede apreciar el esquemático del circuito de acondicionamiento, realizado en el software de diseño PCB KiCad.

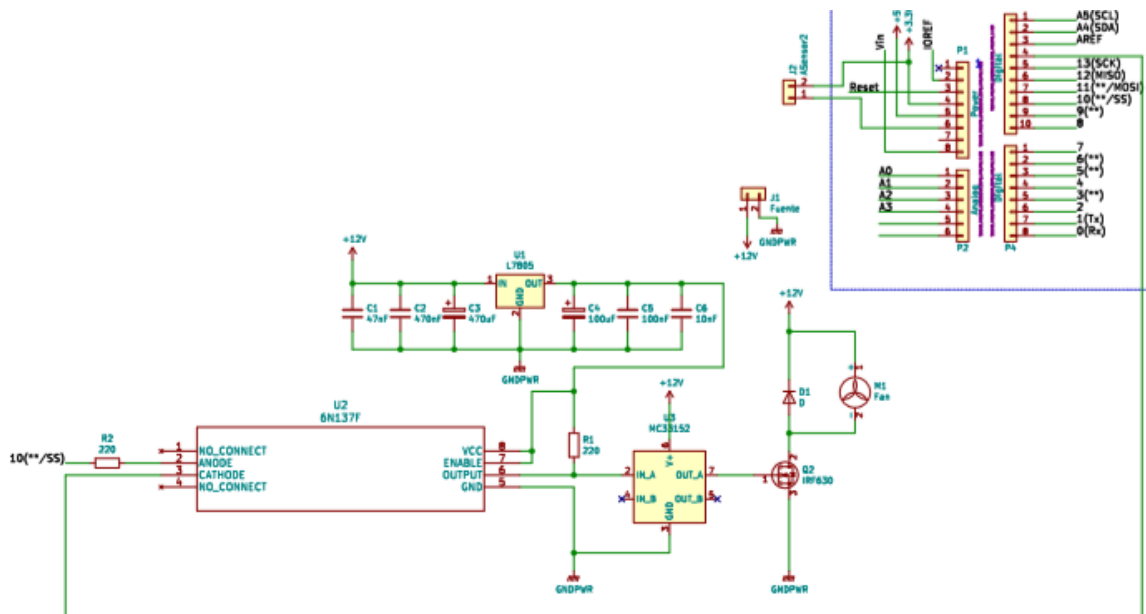
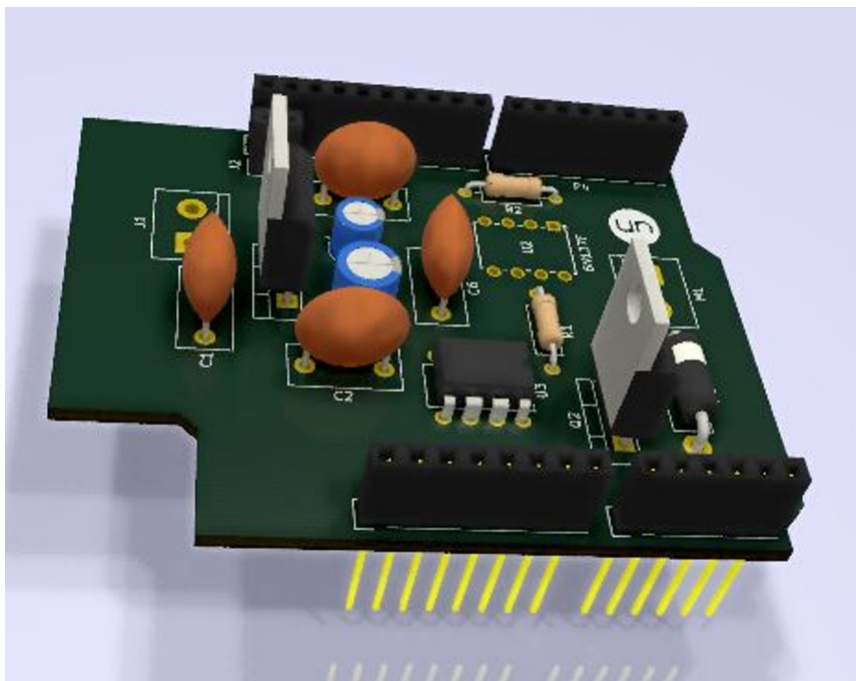


Figura 2.4: Esquemático, circuito acondicionamiento experimento uno (fuente propia)

El esquemático estará disponible en GitHub, así como los archivos necesarios para poder imprimir el circuito en su totalidad. El circuito de acondicionamiento se alimenta con 12 V, ya que como se indico anteriormente esta es la tensión nominal del ventilador. Se inicia con la salida PWM del arduino, el pin número 10 es el encargado de esto. Este GPIO se conecta al ánodo del optoacoplador, el cual esta alimentado por medio de un circuito reductor de tensión, disminuye la tensión de 12 V de la fuente a 5 V. La salida del optoacoplador se conecta a una resistencia de 220 Ohm y al driver MOSFET, que permite la rápida conmutación de la señal PWM sin la perdida de pulsos, el pin de salida del driver se conecta al gate del transistor MOSFET, el cual permite la variación de la tensión en el drain, que a su vez contiene el ventilador y un diodo en antiparalelo, esta variación de la tensión producto de la señal PWM, genera un cambio en la velocidad del ventilador, recordando que esta compuesto por un motor DC.

En la siguiente imagen se puede apreciar la forma física del circuito de acondicionamiento, recalcando que se hizo en forma de Shield de Arduino.



apartado no se explicaran de manera detallada los códigos desarrollados, simplemente se mostrara la estructura utilizada en la programación de estos elementos, teniendo en cuenta los lenguajes de programación de cada uno.

2.2.2.1. Servidor

Para comprender de forma clara la arquitectura del servidor, se puede apreciar la siguiente figura, donde se diferencian los tipos de archivos, carpetas y su nivel dentro de la carpeta general del servidor.

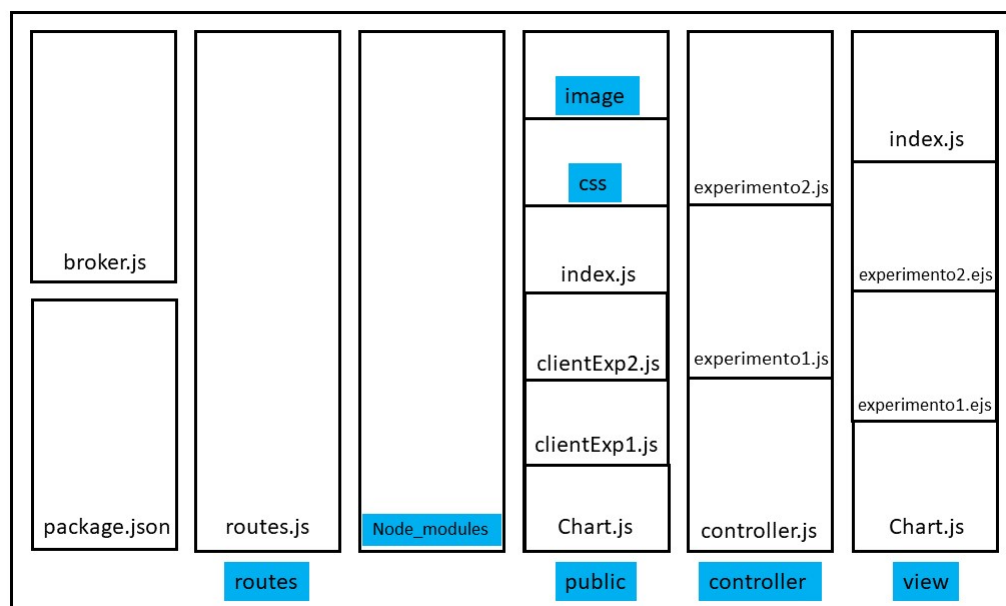


Figura 2.6: Arquitectura del servidor (fuente propia)

Se aprecia que los archivos con extensión `.js` son aquellos escritos en lenguaje JavaScript, los que poseen un fondo en azul claro simbolizan el nombre de carpetas, no de archivos, finalmente, existe un archivo especial de nombre `"package.json"`, el cual contiene información relevante sobre el servidor, como su nombre, creador, paquetes utilizados, etc. El servidor posee interconexión entre los diferentes archivos que le componen, es así como se genera una dependencia entre ellos, como se puede apreciar a continuación.

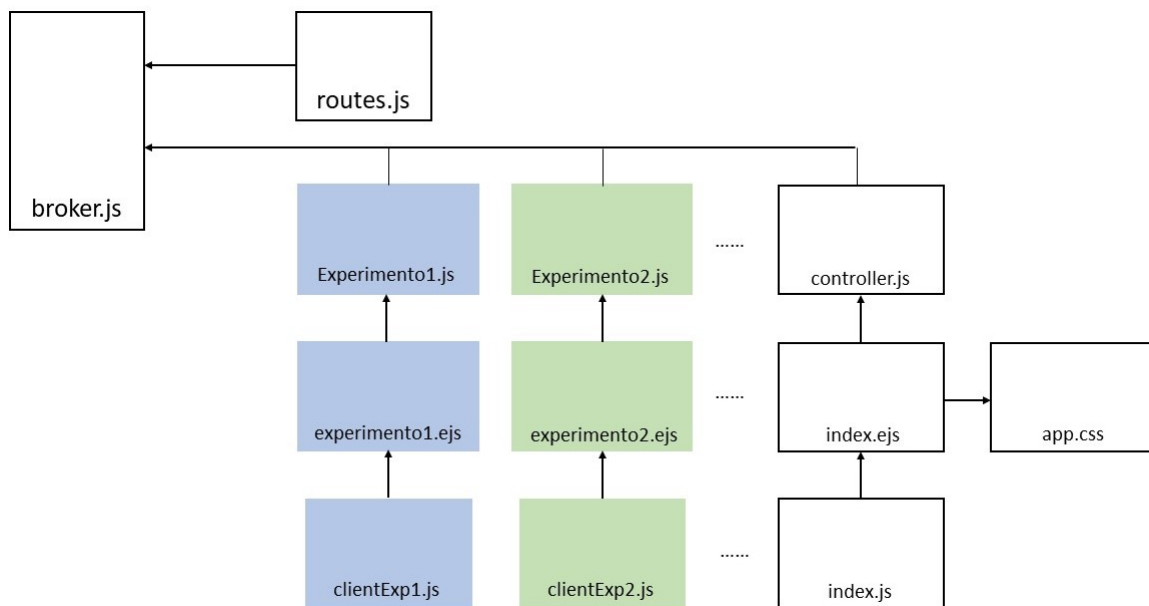


Figura 2.7: Interconexión del servidor (fuente propia)

El archivo `broker.js` es el principal, en este se crea el broker mqtt, se inicia el protocolo WebSockets y el servidor con `express.js`, es el archivo que se ejecuta una vez se desea iniciar el servidor. Para crear un servidor de forma transparente, la mejor manera de realizar la conexión entre páginas es por medio de un archivo de enrutado, el cual se encuentra en la carpeta `routes`, este archivo se conecta directamente con el broker indicándole en que dirección url se puede realizar conexión.

La carpeta `public` contiene aquellos archivos que son visibles al usuario, como los códigos que se ejecutan del lado del navegador, en este caso las gráficas y los botones de interacción con el cliente. Es importante apreciar que para cada experimento se tiene su propio archivo JavaScript del lado del cliente. La carpeta `controller` maneja los archivos que permiten la carga de las vistas, es decir, contiene un archivo por cada experimento, donde cada archivo contiene el llamado a lo que se va a mostrar una vez se ingresa a la URL de la página definida. Finalmente, la carpeta `view` contiene aquellos archivos en código HTML que se ejecutan una vez se ingresa a la URL indicada. Las URL son transparentes al usuario, esto significa que se puede ingresar a ellas directamente, sin embargo, es mas claro el funcionamiento ingresando a la página principal y en ese punto dando click al botón donde se indica el experimento que se desea visualizar.

La carpeta `Node_modules` aloja todos las carpetas concernientes a los paquetes utilizados en el servidor, la carga de estos paquetes se hace de manera automática en los archivos

JavaScript. Si se desea escalar el servidor, se deben añadir los archivos correspondientes a cada experimento en las carpetas indicadas con anterioridad; sin embargo, se tiene una guía donde se explica de manera detallada este proceso de escalabilidad, también en GitHub.

2.2.2.2. Módulo WiFi

Las librerías utilizadas en el módulo NodeMCU permiten el uso simple de las funciones de conexión, envío y recepción de datos, la complejidad en este elemento se puede dar por la confusión que se presenta a la hora de verificar que funciones son las que cumplen los objetivos deseados.

2.2.2.3. Controlador

El controlador no tiene comunicación directa con el servidor, esto se hace a través del módulo WiFi, lo que implica simplicidad y modularidad. Sin importar el experimento, el controlador posee la librería encargada de realizar comunicación por SPI como maestro, esta librería es clara en su utilización, simplemente se llama la función `send()` para enviar los datos necesarios, que en realidad se envían por medio de una trama (arreglo) de 33 bytes y para la lectura de los datos provenientes del NodeMCU se utiliza la función `esp.readData()`, donde se entregan los datos igualmente en un arreglo de 33 bytes. A continuación se explica de manera breve el código de control utilizado en cada experimento.

- **Experimento uno:** Como se explico en el párrafo anterior, es necesario el uso de la librería `SPI.h`, adicionalmente se utiliza la librería encargada del manejo del sensor láser, denominada `VL53L0X.h`, que junto con la librería I2C de Arduino (`Wire.h`) permiten la medición y comunicación con el sensor. El loop de control y la referencia se ejecutan cada cierto tiempo, por ello es necesario tener control sobre la cantidad de milisegundos entre ejecuciones, para ello se utiliza la librería `elapsedMillis.h`.

Una vez se definen las librerías se ingresa con la definición de las variables utilizadas en el programa, existen variables pertenecientes al lazo de control, al tiempo de muestreo, constantes del PID, constantes de calibración y variables de estado para el filtro utilizado en la medición del sensor. El código realiza la medición de distancia por medio del sensor láser, cuyo valor es filtrado con el fin de obtener un resultado lo mas suave posible (en términos de la señal graficada resultante, con el valor de la medición). Se genera la señal de referencia, la cual indica a que valor de distancia se desea

llegar con el objeto levitante, esta señal posee valores tanto positivos como negativos, es un escalón de cierto periodo, indicado gracias a la librería `elapsedMillis.h`.

Con el valor de distancia del sensor láser, y la señal de referencia, se ingresa al PID, comenzando con los cálculos de las partes proporcional y derivativa. Gracias a la suma de los valores P, I y D se crea la señal de control, que se utiliza como generador del PWM, para controlar la velocidad del motor DC y por ende, el flujo del ventilador. En la tabla continua se muestran las funciones principales, utilizadas en el código Arduino y se indica también el PID y su orden respectivo dentro del código así como una breve explicación de las líneas mas importantes. En la figura se indica el lazo de control realimentado, programado en la placa.

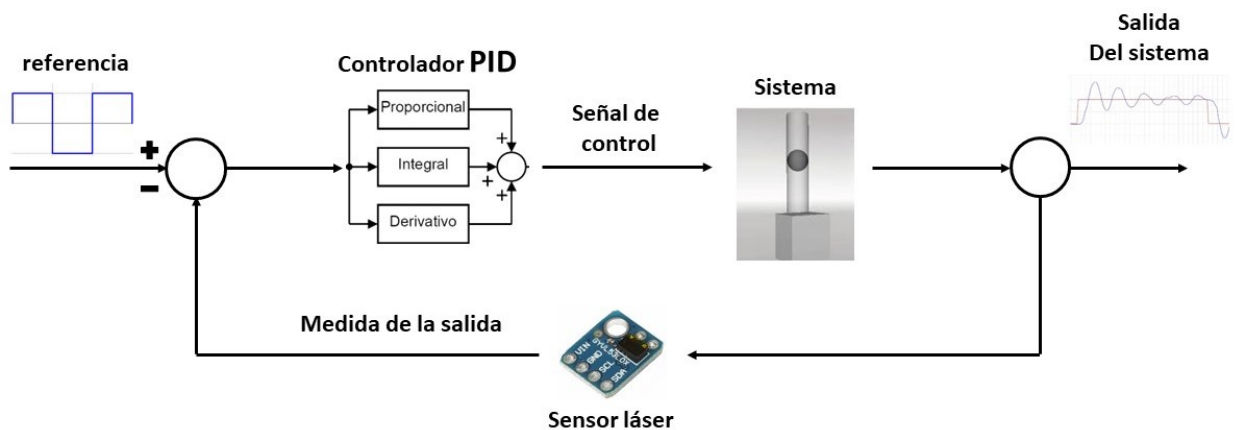


Figura 2.8: Lazo de control experimento 1 (fuente propia)

- **Experimento 2:** El experimento dos se realiza como una simulación del llenado de agua en un tanque, es por ello que este experimento es netamente virtual y programado en la placa. Se realiza un controlador PID, con ciertas diferencias puntuales, respecto al realizado en el experimento uno. Simplemente se modela el sistema en variables de estado y se realiza el controlador con base en dichas variables. En la siguiente figura se puede apreciar el lazo de control del sistema.

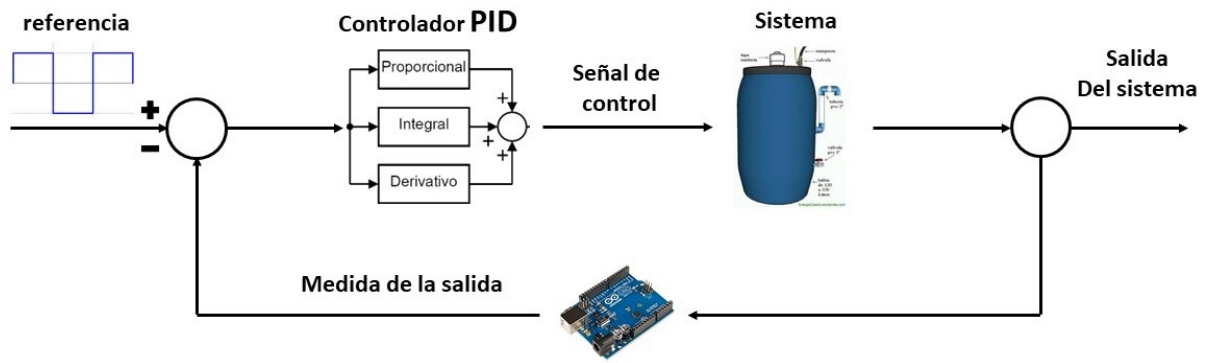


Figura 2.9: Lazo de control experimento 2 (fuente propia)

2.2.3. Implementación mecánica experimento 1

El experimento necesita una planta robusta mecánicamente, lo que indica que es necesario tener elementos que permitan el uso adecuado en respuesta de los dispositivos electrónicos. La base del tubo acrílico, la base del ventilador y el concentrador de aire se realizan en impresión 3D, los archivos .stl serán anexados al repositorio del laboratorio. En las siguientes figuras se aprecian los diseños de estos tres elementos.

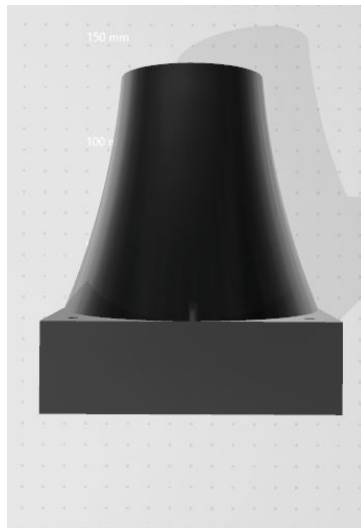


Figura 2.10: Elemento 3D de la base del ventilador (fuente propia)

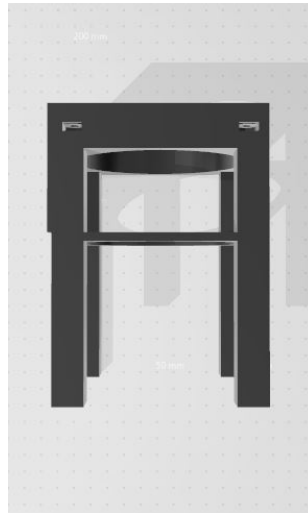


Figura 2.11: Elemento 3D del la base del tubo (fuente propia)

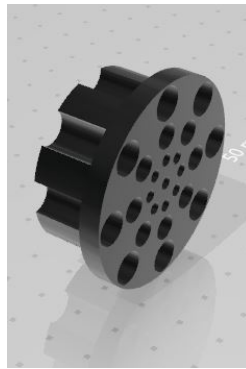


Figura 2.12: Elemento 3D del concentrador de aire (fuente propia)

La base del tubo debe tener forma cónica redondeada, esto con el fin de permitir el flujo de aire sin devoluciones, ya que si se dejan aristas, el aire las golpea y se devuelve, afectando la cantidad de viento que llega al tubo acrílico. El ventilador se atornilla internamente a la base del tubo, y a este conjunto se le desarrolla una base cuadrada con una altura aproximada de 9 cm, que permite unir las dos partes y anexar el Arduino en un lado. Finalmente, el concentrador es un elemento circular de una altura aproximada de 4 cm con agujeros de diferentes tamaños, que posee el ancho del tubo acrílico, se diseña con el fin de crear un flujo lineal de aire.

El dispositivo levitante se recrea con legos, ya que es necesario un elemento ligero y que permita el paso del aire

Capítulo 3

Resultados

Para utilizar el servidor es necesario acceder desde un dispositivo que permita realizar una conexión a internet. Se pueden utilizar navegadores como chrome, mozilla o edge (preferiblemente no utilizar el navegador que viene por defecto en el móvil). Para poder acceder al laboratorio, se debe ingresar a la siguiente URL desde la barra de navegación.

192.168.4.1:3000

Una vez allí, se observara la siguiente pantalla de bienvenida.

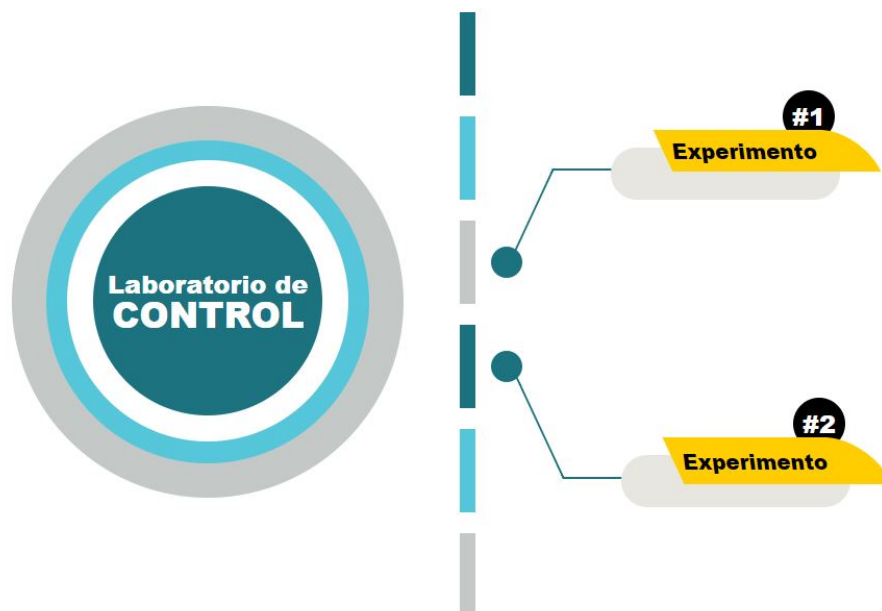


Figura 3.1: Pantalla de inicio (fuente propia)

En dicha pantalla se puede hacer la selección del experimento deseado, ya sea el levitador

neumático o el simulador de tanque (experimento uno y experimento dos respectivamente). Si se ingresa a uno de los experimentos, sin importar cual sea, el otro será automáticamente bloqueado, eliminando el botón de selección, esto con el fin de no crear un doble camino de datos y que haya problemas entre los experimentos. Si se selecciona el experimento uno se observa la siguiente imagen.

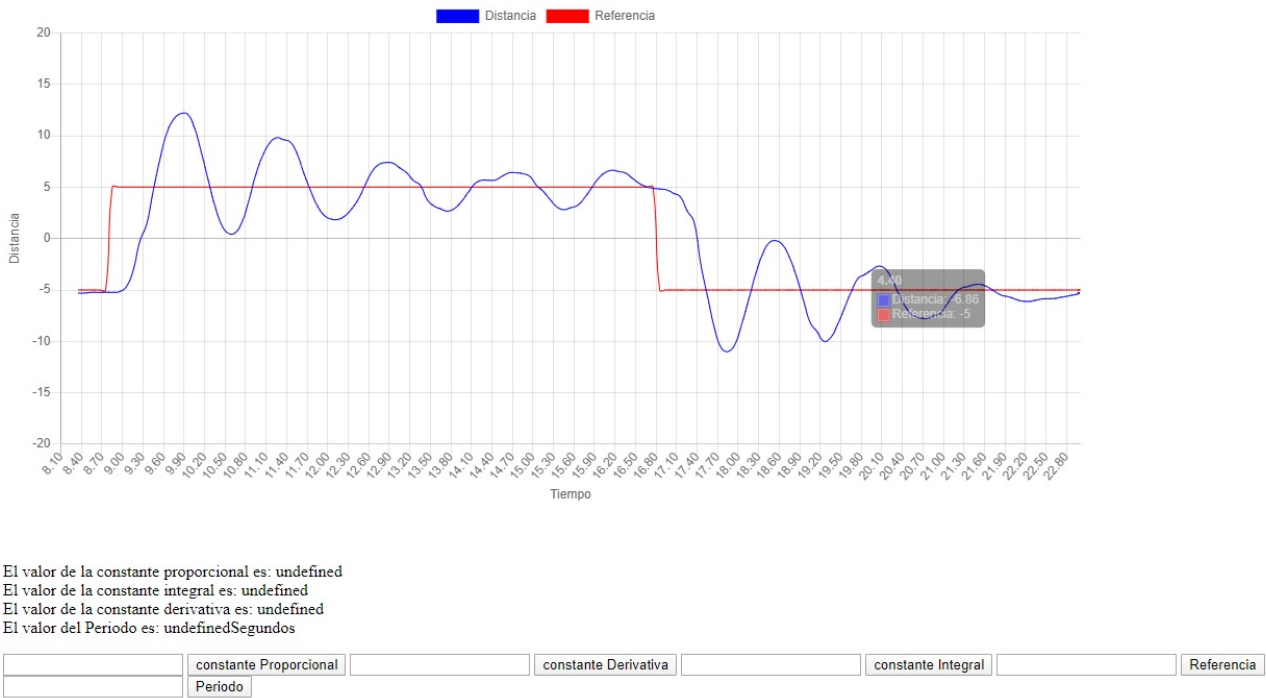


Figura 3.2: Página experimento 1 (fuente propia)

Como se observa en la figura, hay un panel donde se gráfica la medición obtenida por el sensor y el tiempo que ha transcurrido desde que se ingreso al experimento. En la parte inferior hay una serie de indicadores que muestran el valor de las constantes del PID, y el valor del periodo en ese instante. Por debajo de estos avisos, hay un botón y un cuadro de texto, cada uno referente a las constantes allí indicadas, es en estas casillas donde se cambia el valor de las constantes, del periodo y de la referencia.

Para el experimento dos se tiene la misma interfaz, observada en la siguiente figura.

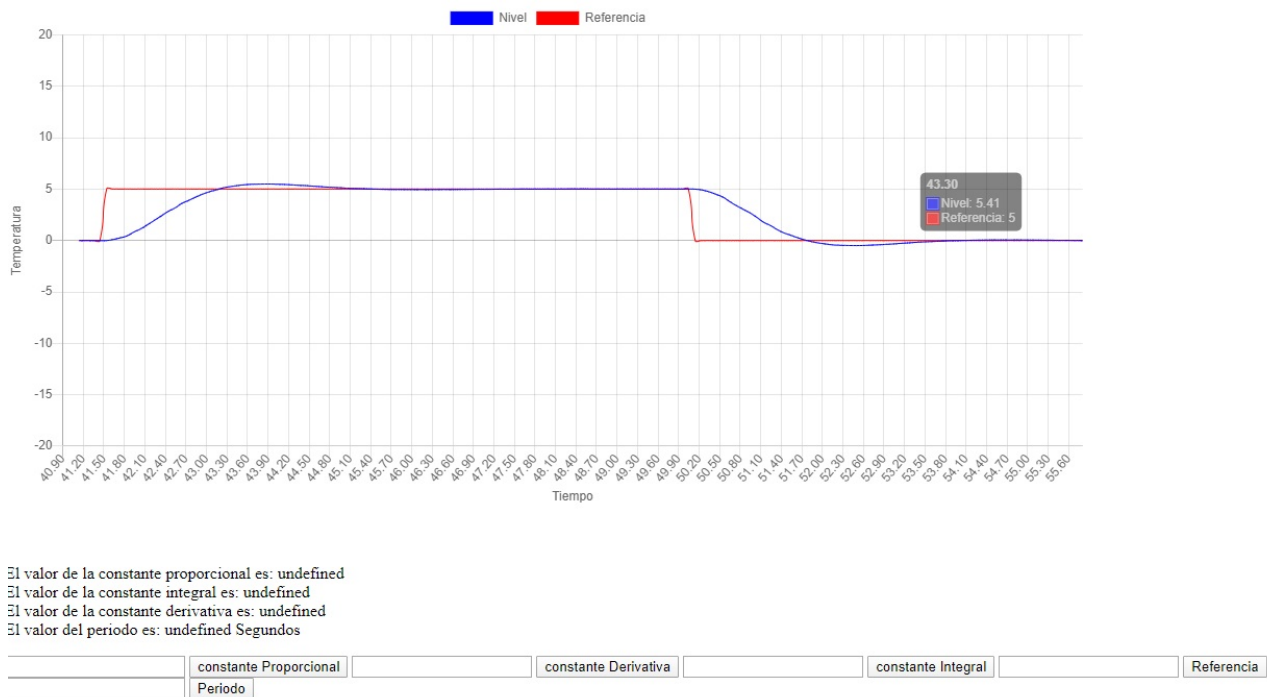


Figura 3.3: Página experimento 2 (fuente propia)

En esta caso se gráfica el valor del nivel y el tiempo transcurrido desde que se ingreso al experimento.

Para volver a la página inicial, se tiene un botón en la parte superior de cada experimento, indicado como "volver al home". Lo expuesto anteriormente indica que elementos se obtiene como resultado en la interfaz gráfica del servidor, lo ideal es que el usuario interactué con las dos plantas, enviando valores y verificando las variaciones que se obtienen de acuerdo al controlador instalado en cada experimento. Para mayor información se tiene la guía del laboratorio, donde se detalla un poco mas acerca de las pruebas que se pueden realizar sobre los experimentos, esta guía se encuentra en el repositorio del proyecto.

Capítulo 4

Conclusiones

- Utilizar el internet de las cosas a favor de la enseñanza de control, brinda la posibilidad de interacción entre usuarios y sistemas en tiempo real
- Reafirmar los conceptos visto en clase por medio de plantas físicas ayuda a conceptualizar la teoría, visualizando en la realidad las variaciones realizadas en las constantes de los experimentos.
- La creación de una red LAN con conexión inalámbrica, en el dispositivo servidor, favorece el uso del laboratorio en cualquier sitio, sin necesidad de tener una conexión estable a una red de internet.

Bibliografía

- [1] ARVC. *Descripción AutomatL@bs*. Recuperado de: <https://arvc.umh.es/proyectos/AUTOMATL@BS/index.php?type=proy&dest=inicio&lang=es&vista=normal&idp=AUTOMATL@BS&ficha=on>. Consultado el día 24 de octubre del 2019.
- [2] UNED. *Laboratorio remoto*. Recuperado de: https://canal.uned.es/video/5a6f2607b1111f2d4c8b45db?track_id=5a6f2608b1111f2d4c8b45de. Consultado el día 24 de octubre del 2019.
- [3] J. CHACON, J.SAENZ, L. DE LA TORRE, J. M. DIAZ, F. ESQUEMBRE. *Design of a low-cost air levitation system for teaching control engineering*. MDPI-journal sensors, 2017.
- [4] V. NAGARGOJE, P. BHAGAT. *Virtual laboratory of control system*. International conference on I-SMAC (I-SMAC 2017). IEEE, 2017.
- [5] C. R. FIGUEROA, E. RUBIO, I. SANTANA, J. ROHTEN. *Speed and Position Control Practices through the remote laboratory: SLD-UBB*. ICA-ACCA 2018, Gran Concepción Chile. IEEE, 2018.
- [6] J. B. ORTEGA, E. B. PORTAS, J. L. OROZCO, J. M. DE LA CURZ. *A New Open-Source and Smart-Device Accessible Remote Control Laboratory*. 4th Experiment@ International Conference, University of Algarve, Faro Portugal. IEEE, 2017.
- [7] G.S. KVASCEV, Z.M. DJUROVIC, B.D. KOVACEVIC, S.M. VUJNOVIC. *eLab as a Tool for Remote Laboratory in Process Control*. 11th International Conference on Remote Engineering and Virtual Instrumentation, 2014. Polytechnic of Porto, Portugal. IEEE, 2014.

- [8] TELECOM ITALIA S.P.A. *Towards a definition of the internet of things (IoT)*. IEEE internet initiative, revisión 1, 2015, página 10.
- [9] Recuperado de: <https://www.xataka.com/ordenadores/raspberry> Consultado el día 14 de enero 2020.
- [10] Recuperado de: <https://store.arduino.cc/usa/arduino-uno-rev3>. Consultado el día 14 de enero del 2020.
- [11] Recuperado de: <https://programarfacil.com/podcast/nodemcu-tutorial-paso-a-paso/>. Consultado el día 14 de enero del 2020.
- [12] *ST life.augmented*. VL53L0X. Datasheet sensor láser.
- [13] Recuperado de: <https://www.pololu.com/product/2490>. Consultado el día 15 de enero del 2020.
- [14] Recuperado de: <https://www.arduino.cc/en/reference/SPI>. Consultado el día 15 de enero del 2020.
- [15] Recuperado de: <http://www.tst-sistemas.es/mqtt/>. Consultado el día 2020.
- [16] Recuperado de: <https://www.ibm.com/developerworks/ssa/library/iot-mqtt-why-good-for-iot/index.html>. Consultado el día 15 de enero del 2020.
- [17] Recuperado de: https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/Qu%C3%A9_es_JavaScript. Consultado el día 15 de enero del 2020.
- [18] Recuperado de: <https://nodejs.org/es/about/>. Consultado el día 15 de enero del 2020.