



**Universidad
Andrés Bello®**

UNIVERSIDAD ANDRÉS BELLO

Facultad de Ingeniería

Escuela de Industrias

**“Desarrollo de sistema de control y mantenimiento de calidad de agua
dulce en grandes concentraciones mediante IoT”**

Tesis de pregrado para optar al título de Ingeniero en Automatización y Robótica

AUTOR: EDUARDO ANDRES SCHETTINO SANHUEZA

PROFESOR GUÍA: NESTOR PALOMINOS

SANTIAGO – CHILE

DICIEMBRE, 2019



**Universidad
Andrés Bello®**

FACULTAD DE INGENIERÍA

INGENIERIA EN AUTOMATIZACION Y ROBOTICA

DECLARACIÓN DE ORIGINALIDAD Y PROPIEDAD

Yo, **Eduardo Andrés Schettino Sanhueza**, declaro que este documento no incorpora material de otros autores sin identificar debidamente la fuente.

Santiago, Diciembre de 2019

Firma del alumno

AGRADECIMIENTOS

Quiero agradecer a mi madre Edith y mi padre Oscar que me han acompañado y apoyado siempre en cada decisión importante que he tomado, han sido un pilar fundamental en mi vida, ellos me han brindado la educación y los valores que hacen que hoy en día sea la persona que soy.

A mi hermana Carolina por su apoyo incondicional, comprensión y por brindarme la idea para el desarrollo de la tesis.

A mis compañeros por hacer de estos años de universidad, una estancia maravillosa e inolvidable con las juntas de estudios y asados recurrentes.

Finalmente a mis profesores y universidad por brindarme el conocimiento necesario para salir a un nuevo mundo.

RESUMEN

Una buena calidad del agua es fundamental para todo tipo de uso, ya sea industrial, para consumo o incluso para almacenamiento de agua, ya que existen variables relevantes tales como pH, que indica el grado de acidez o alcalinidad de una solución. Las variables que indican la presencia de bacterias corresponden a la concentración de cloro y el valor Redox medidos en mV.

La acidez o alcalinidad del agua pueden llegar a afectar a los materiales involucrados en instalaciones, así como también a los seres vivos si no son controladas apropiadamente, lo cual es fundamental para sustentar la vida en el planeta.

Por otro lado, la evolución de las tecnologías o desarrollo de nuevas placas como Raspberry, ha permitido la miniaturización y minimización de costos para el área I2D.

Los sensores de pH y Redox entregan un voltaje análogo entre 0 y 5 volts, los cuales serán digitalizados mediante un conversor análogo digital ADS1015 de 12 bits, los cuales serán leídos desde un código desarrollado en Python para tal efecto.

Estos datos serán almacenados en una base de datos MySQL y enviados a un bróker MQTT, además de vincular los datos a una página web, de modo de poder monitorear los valores y realizar acciones con una interfaz gráfica multiplataforma.

Por otro lado, se controlará una serie de bombas de agua utilizando la filosofía de control difuso para mantener los valores dentro del margen de buena calidad.

Para validar resultados se utilizará un estanque de 25 litros de agua como modelo

a escala y un set analizador de pH a modo de control para observar la efectividad del sistema desarrollado.

Conceptos claves: Calidad del agua, Raspberry, transmisión, IoT, sensores, actuadores, Python, MYSQL, MQTT

ABSTRACT

Water quality is essential for all types of use, whether industrial, for consumption or even for water storage, since there are relevant variables such as pH, Chlorine and Redox that can affect the materials involved as well as living

Good water quality is essential for all types of use, whether industrial, for consumption or even for water storage, since there are relevant variables such as pH, which indicates the degree of acidity or basicity of a solution, Chlorine, fundamental component for water disinfection and Redox measured in mV, values that indicate whether batteries are likely or not. These variables can affect the materials involved as well as living beings if they are not properly controlled, which is essential to sustain life on the planet.

On the other hand, the evolution of technologies or the development of new boards such as Raspberry, has allowed miniaturization and minimization of costs for the I2D area.

The pH and Redox sensors include a variable voltage between 0 and 5 volts, which will be digitized using a 12-bit ADS1015 digital analog converter, values read from a code developed in Python for this purpose.

This data will be stored in a MySQL database and sent to an MQTT broker, in addition to linking the data on a web page, in order to monitor the values and perform actions with a cross-platform graphical interface.

On the other hand, it controls a series of water pumps using the diffuse control philosophy to keep the values within the range of good quality. To validate results, a 25-liter water tank will be used as a scale model and a pH analyzer set as a control to observe the effectiveness of the system developed.

Keywords : Water quality, Raspberry, transmission, IoT, sensors, MYSQL, MQTT

INDICE DE CONTENIDOS

AGRADECIMIENTOS	II
RESUMEN.....	III
ABSTRACT	V
INDICE DE CONTENIDOS.....	1
INDICE DE FIGURAS.....	4
1 ANTECEDENTES GENERALES	7
1.1 INTRODUCCIÓN	7
1.2 OBJETIVOS.....	9
1.2.1 OBJETIVO GENERAL.....	9
1.2.2 OBJETIVOS ESPECÍFICOS	9
1.3 ORGANIZACIÓN Y PRESENTACIÓN DEL TRABAJO	10
2 MARCO TEÓRICO.....	11
2.1 ESTADO DEL ARTE.....	11
2.1.1 INTERNET DE LAS COSAS	12
2.1.2 CALIDAD DEL AGUA.....	14
2.1.3 MUESTREO DE SEÑALES.....	19
2.2 HERRAMIENTAS DEL PROYECTO.....	21
2.2.1 HARDWARE	21
2.2.2 SOFTWARE	37
2.2.3 ESTANDARES ASOCIADOS AL PROYECTO	49
2.2.4 FILOSOFIA DE CONTROL.	49
3 DESARROLLO DE LOS COMPONENTES	53
3.1 MODELAMIENTO DE LA SOLUCION	53
3.2 DISEÑO DEL SISTEMA.....	54
3.3 DIAGRAMA DE ELEMENTOS.....	54
3.4 ELEMENTOS PRINCIPALES Y FUNCIONES.....	55

3.5	APLICACIÓN ESPECIFICA	56
3.5.1	APLICACIONES GENERALES	57
3.6	CONFIGURACION RASPBERRY.....	57
3.6.1	IP ESTATICA	57
3.6.2	CONEXIÓN SSH	59
3.6.3	ACCESO A RASPBERRY DESDE WINDOWS.....	60
3.6.4	VNC VIEWER.....	61
3.6.5	INFRAESTRUCTURA LAMP	63
3.6.6	DNS DINAMICA	68
3.7	CARTA GANTT.....	70
3.8	COSTOS.....	71
3.8.1	COSTOS DEL HARDWARE	71
3.8.2	COSTOS DEL SOFTWARE	72
4	DESARROLLO Y FLUJOS DE PROGRAMA	73
4.1	PUNTO DE CONEXION	73
4.1.1	DESARROLLO CODIGO WIFI.....	73
4.1.2	FUNCIONES GATEWAY IOT.....	74
4.2	VARIABLES GLOBALES	74
4.3	COMUNICACIÓN MQTT	75
4.3.1	DESARROLLO CODIGO	79
4.3.2	CLIENTE PYTHON MQTT	79
4.4	ADQUISICION DE DATOS	81
4.4.1	TEMPERATURA	81
4.4.2	PH	83
4.4.3	REDOX.....	84
4.5	DATOS EN MYSQL	84
4.6	DATOS EN LA NUBE	86
4.6.1	THINKSPEAK.....	86
4.7	DESARROLLO PAGINA WEB.....	87
4.7.1	INTERFAZ GRAFICA	87
4.7.2	GRAFICO DINAMICO	92
4.8	CONTROL AUTOMATICO.....	93

5	RESULTADOS	96
5.1	PRUEBAS DE FUNCIONAMIENTO	96
5.2	REGULADORES DE PH.....	96
5.2.1	ALMACENAMIENTO EN MYSQL	97
5.2.2	ALMACENAMIENTO EN LA NUBE.....	98
5.3	PRUEBA ADQUISICION DE DATOS	99
5.4	DIAGRAMA DE BLOQUES.....	103
5.5	VALIDACION	104
5.6	ANALISIS DE RENDIMIENTO.....	106
6	CONCLUSION	107
	GLOSARIO	109
7	REFERENCIAS	110
8	ANEXOS	112
8.1	LECTURA Y CALIBRACION SENSOR PH.....	112
8.2	PRUEBA ADS1015	113
8.3	RESCATANDO DATOS MQTT EN RASPBERRY	114
8.4	ENVIANDO MENSAJE DESDE NODEMCU A BROKER MQTT	115
8.5	GRAFICO CON LIBRERÍA HIGHCHARTS	116

INDICE DE FIGURAS

Figura 1 Dispositivos conectados entre 2015 y 2025.....	9
Figura 2 IoT principales componentes	12
Figura 3 Requisitos del agua para recreación con contacto directo.....	15
Figura 4 Escala de pH junto ejemplos.....	16
Figura 5 Redox.....	18
Figura 6 Discretización de una señal	20
Figura 7 Puertos de conexión de la Raspberry Pi 2B.	22
Figura 8 Puertos GPIO de la Raspberry Pi 2B.	24
Figura 9 Hardware NodeMCU ESP8266.....	25
Figura 10 Pines NodeMCU ESP32	28
Figura 11 Hardware ADS1015	29
Figura 12 Sensor de PH.....	30
Figura 13 Ejemplo conexión sensor de pH	31
Figura 14 Ejemplo conexión sensor ORP	32
Figura 15 Sensor DS18B20 Termocupla	34
Figura 16 Bomba de agua sumergible	36
Figura 17 Batería Lipo.....	37
Figura 18 Trama TCP.	39
Figura 19 El modelo de publicación y de suscripción de MQTT	41
Figura 20 Jerarquización de los tópicos MQTT	41
Figura 21 Funciones Broker MQTT	42
Figura 22 Ejemplo lógica difusa	50
Figura 23 Componentes y Conexiones	54
Figura 24 Elementos arquitectura del sistema	55
Figura 25 IP Estática.....	59
Figura 26 Configuración de Raspberry Pi	60
Figura 27 Configuración Putty.....	60
Figura 28 Consola con sesión iniciada en Putty.....	61

Figura 29 VNC Viewer	62
Figura 30 Escritorio Raspberry mediante VNC	62
Figura 31 Infraestructura LAMP	63
Figura 32 Pagina web en Raspberry	64
Figura 33 PHP en Raspberry	66
Figura 34 PHPMyAdmin en Raspberry	68
Figura 35 Puertos en Router	70
Figura 36 Carta Gantt	71
Figura 37 Costos de Hardware	71
Figura 38 Costos de Software, medidos en hora/hombre	72
Figura 39 Correlación entre pH, Redox y Cloro Libre	75
Figura 40 Enviando mensaje a un Tópico MQTT	78
Figura 41 Suscrito a un tópico MQTT	78
Figura 42 Mensaje enviado desde NodeMCU (prueba)	79
Figura 43 Conexión Sensor Temperatura	82
Figura 44 Tablas en base de datos MySQL	84
Figura 45 Ejemplo Tabla agua, cantidad en mL	85
Figura 46 Ejemplo Tabla cambioph	85
Figura 47 Ejemplo Tabla sensores	85
Figura 48 Diagrama ThinkSpeak	87
Figura 49 Pagina Web (1)	88
Figura 50 Gráficos en página web	89
Figura 51 Pagina web desde Móvil	89
Figura 52 Búsqueda por Fecha, grafico Temperatura	90
Figura 53 Búsqueda por fecha, grafico pH y ORP	91
Figura 54 Ajuste Manual de pH vía Pagina web	92
Figura 55 Ventana configuración HighCharts.com	93
Figura 56 Funciones de membresía	94
Figura 57 Incrementador de pH	97
Figura 58 Reductor de pH	97
Figura 59 Datos en MySQL	98

Figura 60 Datos en ThinkSpeak.....	99
Figura 61 Temperatura Inicial	99
Figura 62 pH Inicial	100
Figura 63 ORP Inicial.....	100
Figura 64 Valores en Agua alterada.....	101
Figura 65 Resultado Final	102
Figura 66 CPU y Temperatura Raspberry Pi.....	106

1 ANTECEDENTES GENERALES

1.1 INTRODUCCIÓN

El Agua es uno de los principales elementos fundamentales para el desarrollo de la vida, todo ser vivo requiere de esta, como también para la producción de alimentos, electricidad o incluso, mantenimiento de la salud. También el agua está presente en procesos de elaboración de varios productos industriales, medios de transportes y es irremplazable para la sustentabilidad de los ecosistemas de la tierra. En países en camino de desarrollo, la gran mayoría de enfermedades se relaciona con el agua, siendo la diarrea una de las principales causas de muerte infantil. (ONU/WWAP, 2003)

Una de las variables más importantes a consideras es el nivel de pH el cual es la medida de acidez o alcalinidad en el agua la cual puede provocar daños tanto como para las personas como también para los materiales que hagan uso de ésta. Además, que un nivel de pH no adecuado podría provocar una capacidad de desinfección con menor eficacia haciendo uso de cloro en caso de agua en piscinas.

Otro factor es el potencial de oxidación-reducción (Redox) el cual está directamente vinculado con la medición de la oxidación y la actividad bacteriana en el agua.

Ambos factores, pueden ser alterados como consecuencia de condiciones meteorológicas, otro tipo de sustancias, frecuencia de uso, entre otro, pero utilizando ambas variables existe la posibilidad de mantener su calidad gracias al cloro el cual actúa como un depredador de electrones de la materia orgánica.

En la actualidad los sensores que se utilizan para el monitoreo de las variables

ya mencionadas se han reducido considerablemente en tamaño y precios, como también la conexión que se debe realizar para utilizarlos, ya que según antecedentes se deben conectar a un controlador definido ya por el sensor.

Las nuevas placas de desarrollo y sistemas integrados como lo es Raspberry Pi, dan la posibilidad de grandes oportunidades de programación y de sistemas eficientes para la comunicación de hardware ya sea local o remotamente, gracias a que permite montar fácilmente servidores o clientes. Con esta información, la evolución de los sistemas de control y de programas es inminente, proporcionando comunicación y control más eficiente. La recolección de datos producidos requiere de análisis en tiempo real, una interfaz amigable y almacenamiento, en pocas palabras, implementar tecnologías nuevas como lo es IoT.

Se ha popularizado en la actualidad el término IoT, internet de las cosas o por su sigla en inglés 'Internet of Things' el cual es un concepto que se refiere a una interconexión digital de objetos cotidianos con internet (Hadi,2015), pero que se estima que en un futuro la interconexión no será solo de objetos cotidianos como tostadoras o televisores, sino también de dispositivos industriales o de investigación.

Internet of Things - number of connected devices worldwide 2015-2025

Internet of Things (IoT) connected devices installed base worldwide from 2015 to 2025 (in billions)

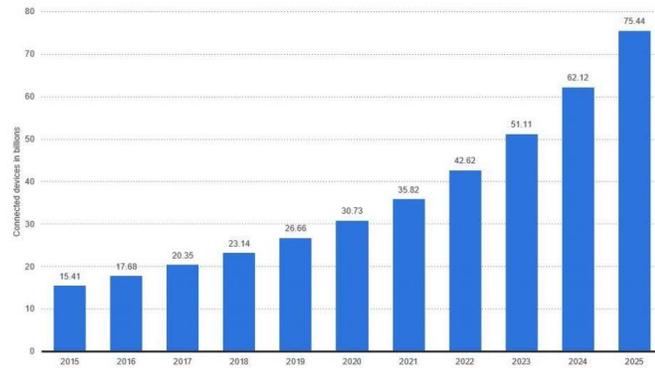


Figura 1 Dispositivos conectados entre 2015 y 2025

1.2 OBJETIVOS

1.2.1 OBJETIVO GENERAL

Desarrollar un prototipo de sistema de control e interfaz gráfica para el control de parámetros en grandes concentraciones de agua dulce, que sea capaz mediante sensores monitorear pH y Redox con el objetivo de mantener regulados los niveles utilizando una filosofía de control denominada “control difuso”, con el fin de no utilizar recursos demás y mantener el agua en la mejor calidad posible. Además, este sistema de control va a poseer la capacidad de almacenar los datos en un base de datos MYSQL a los cuales se podrá acceder mediante una página web a desarrollar, donde se podrá observar el cambio en función del tiempo, para posibles estudios en el futuro.

1.2.2 OBJETIVOS ESPECÍFICOS

- Establecer obtención de datos de los sensores mediante protocolo de mensajería MQTT, utilizando como servidor una Raspberry Pi.

- Montar un servidor web el cual reciba los datos adquiridos anteriormente para la entrega de información al usuario, mediante gráficos u otros métodos
- Implementar un control difuso con el fin de mantener estables los niveles del agua.
- Creación de interfaz de usuario multiplataforma el cual permitirá un monitoreo de los datos en tiempo real e implementar un menú desde el cual se pueda controlar remotamente.
- Analizar las ventajas y desventajas de la implementación de IoT en las industrias.

1.3 ORGANIZACIÓN Y PRESENTACIÓN DEL TRABAJO

La estructura del desarrollo del presente proyecto se encuentra distribuida de la siguiente manera:

En el capítulo I se hará una breve introducción a la problemática existente que origina este proyecto, junto con los respectivos objetivos a desarrollar.

En el capítulo II se plantea un estado del arte acerca de los conceptos claves a utilizar dentro del proyecto, como también las propiedades del agua que se tomaran en cuenta y los detalles técnicos, como los dispositivos y herramientas para implementar el prototipo de control, además de software que aportaran en la comunicación y el manejo de datos.

En el capítulo III se dará inicio a la configuración inicial y la incorporación de las herramientas necesarias para llevar a cabo el proyecto, lo cual se irá registrando paso a paso toda acción realizada.

En el capítulo IV se procede al desarrollo de sistema de control y la obtención de los respectivos datos, instaurando además un método de monitoreo y control mediante página web a la cual también se podrá acceder desde un dispositivo móvil.

En el capítulo V se expondrán los resultados finales de la implementación del prototipo en un tanque con agua, junto con su sistema, llevando a cabo varias pruebas, tanto de hardware como de software, realizando también gráficos que ilustrarán los cambios de las variables respecto al tiempo.

Finalmente, en el capítulo VI se encontrarán las conclusiones, resultados finales, discusiones y problemas al realizar el proyecto, considerando cada capítulo.

2 MARCO TEÓRICO

2.1 ESTADO DEL ARTE

En la actualidad el mantenimiento del agua ha ido evolucionando constantemente, gracias a las nuevas tecnologías que van desarrollando las empresas especialistas en el control de calidad, diseñando equipos cada vez más precisos y eficientes. Pero gran parte de ellos va destinado a usos industriales, como el tratamiento de aguas servidas o usos en laboratorio ya que para la obtención de los equipos se debe invertir gran cantidad de dinero, haciendo que, para algunos proyectos, como por ejemplo en piscinas, no sea rentable adquirir estos equipos de control.

Por otro lado, existen dispositivos de detección muy similares a los que las empresas ofrecen a un bajo costo, pero sin incluir la parte lógica que realizaría el control, estos pueden ser conectados a controladores como Raspberry, por lo tanto, se abren grandes posibilidades para el desarrollo de proyectos que pueden ser enfocados en entornos menos industriales.

2.1.1 INTERNET DE LAS COSAS

El internet de las cosas o por sus siglas en inglés, IoT Internet of Things, forma parte de unos los más importantes desarrollos tecnológicos de la última década, se considera como la posibilidad de realizar una red global interconectada entre maquinas mecánicas y digitales, objetos, animales o incluso personas que tengan un método de identificación única y la capacidad de transferir datos a través de una red, sin requerir de interacciones humano a humano o humano a computadora. (Margaret Rouse,2017).

La clave de todo está en los sistemas embebidos, es decir, chip o circuitos que cuentan con las herramientas necesarias para cumplir labores muy específicas como objetos que funcionan como sensores o como actuadores y claro, los hay que cumplen ambas funciones simultáneamente. Cual sea el caso el principio siempre será el mismo, cada uno de los objetos que estén conectados a internet tienen una IP específica y mediante esa IP se puede acceder o enviar información con instrucciones. De la misma manera, podría contactar con un servidor externo y enviar los datos que recoja. (IoT, 2014)

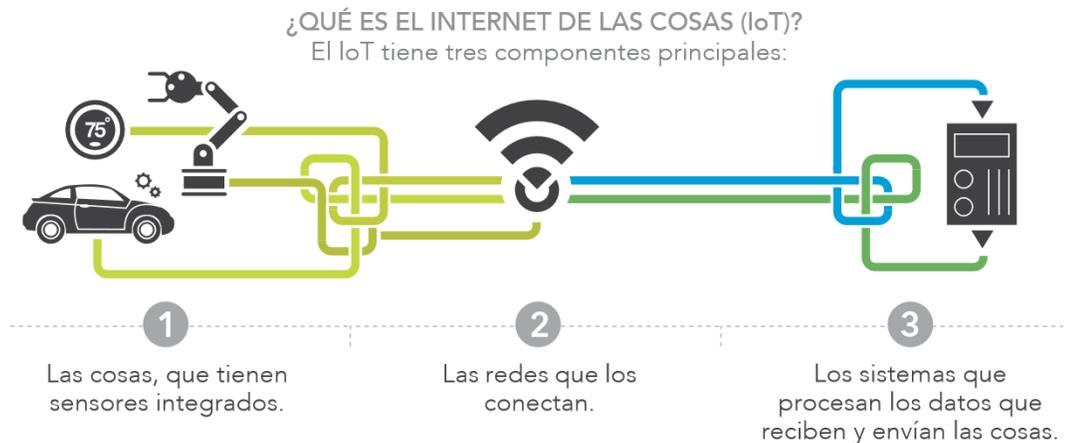


Figura 2 IoT principales componentes

Recuperado de SAS

El internet de las cosas (IoT) tomo fuerza desde el momento en el que eran

mucho más los dispositivos conectados a internet que las personas. Se estima que en el año 2030 la cantidad de dispositivos conectados internet será de más de 125.000 millones, sin incluir los dispositivos más utilizados por las personas como computadores, Smartphone y tabletas. (IHS Markit, 2018).

Con la aparición de los dispositivos IoT, surge también el término “Industria 4.0”, esta es considerada como la cuarta revolución la cual viene marcada por la masiva introducción de sistemas “ciber-físicos” (SPC, Cyber Physical System), la interconexión entre industrias y los interfaces abiertos para los servicios. En pocas palabras, la implementación del internet de las cosas al entorno industrial. La implementación del internet de las cosas dentro del entorno industrial según la Universidad Técnica de Dortmund, ubicada en Alemania, debe basarse en los siguientes principios básicos de diseño (Hermann,2015):

-Interoperabilidad: La capacidad de interconexión de todos sus elementos materiales y humanos, mediante el uso de IoT y sus servicios.

-Virtualización: Los dispositivos deben tener copias virtuales de sus datos pudiendo mostrar toda la información de sensores y sistemas, además de modelos de simulación.

-Descentralización: Los objetos conectados en entornos inteligentes deberán tener capacidad de decisión autónoma.

-Capacidad de tiempo real: Esto mediante captura de datos, análisis y toma de decisiones en tiempo real.

-Orientación al servicio: pudiendo ofrecer catálogos de servicios que permitan interacción y creación de nuevas aplicaciones a medida que el usuario o proceso final requiera.

-Modularidad: Con flexibilidad para añadir, sustraer o sustituir elementos conectados.

En comparación con otras tecnologías más populares en la actualidad, el internet de las cosas aún no ha encontrado su punto específico en el cual generar un mayor impacto en el mercado consumidor. Ya sea porque esta tecnología aún se encuentra en desarrollo en el país o porque aún quizás grandes empresas aún no han visto la oportunidad correcta para tomarla. Aun así, grandes empresas mundiales han desarrollado dispositivos, junto con sus correspondientes aplicaciones, para la automatización de hogares con tecnologías como Home Kit (Apple) y Android @Home (Google).

2.1.2 CALIDAD DEL AGUA

EL problema de la calidad del agua es igual de relevante u importante que la escasez de esta misma, sin embargo, se le ha dado menos importancia. El concepto de calidad de agua se refiere al conjunto de parámetros que indican si el agua puede ser utilizada en los distintos propósitos que se le puede dar, como por ejemplo: doméstico, riego, recreación e industrial (Mejía Clara, 2015)

La calidad del agua se define como el conjunto de características de esta misma que pueden llegar a afectar su adaptabilidad dentro del entorno en que se usara, como también la relación entre la calidad del agua y las necesidades del usuario. Otra manera de definir la calidad del agua es mediante el análisis de su contenido (Mendoza, 1976)

La contaminación es la introducción ya sea intencional o no de materias o formas de energía que impliquen una alteración perjudicial de su calidad en relación con los usos posteriores o con su función ecológica (Gallego, 2000). Además, que la contaminación se puede generar ya sea por el uso continuo de esta o incluso también por causas naturales como podría ser el nulo uso o debido a fenómenos climáticos.

Características	Requisito
pH	6,5 a 8,3 excepto si las condiciones naturales de las aguas muestren valores diferentes, pero en ningún caso menor de 5,0 ó mayor de 9,0
Temperatura, °C, máximo	30
Claridad, mínimo ¹⁾	Visualización de discos Secchi a 1,20 m de profundidad
Sólidos flotantes visibles y espumas no naturales	Ausentes
Aceites flotantes y grasas, mg/l, máximo ¹⁾	5
Aceites y grasas emulsificadas, mg/l, máximo ¹⁾	10
Color, unidades Escala Pt-Co, máximo ¹⁾	100 Ausencia de colorantes artificiales
Turbiedad, unidades Escala Sílice, máximo ¹⁾	50
Coliformes fecales / 100 ml, máximo ¹⁾	1 000
Sustancias que produzcan olor o sabor inconvenientes	Ausentes

Figura 3 Requisitos del agua para recreación con contacto directo

2.1.2.1 PH

El pH o también conocido como la concentración relativa de iones de hidrogeno en el agua, donde un ion es una partícula cargada eléctricamente constituida por un átomo o molécula que no sea necesariamente neutra. Este valor indica si el líquido se comporta como un ácido débil o como una solución alcalina.

Esta medición es muy valiosa ya que permite interpretar los rangos de solubilidad de los componentes químicos. La actividad de los iones de hidrógenos puede afectar indirecta o directamente las actividades de otros componentes presentes en el agua por lo tanto la medida del pH compone un parámetro importante para la descripción de sistemas biológicos o químicos.

La escala del pH está dividida por 14 unidades, la cual tiene valores que van desde el cero, el valor más ácido, hasta el 14, el menos ácido a los cuales se les llama "Básicos". El agua pura se encuentra en el punto medio de la escala, es decir, en el 7 en donde se considera que el valor del líquido es Neutro, ni ácido ni básico.

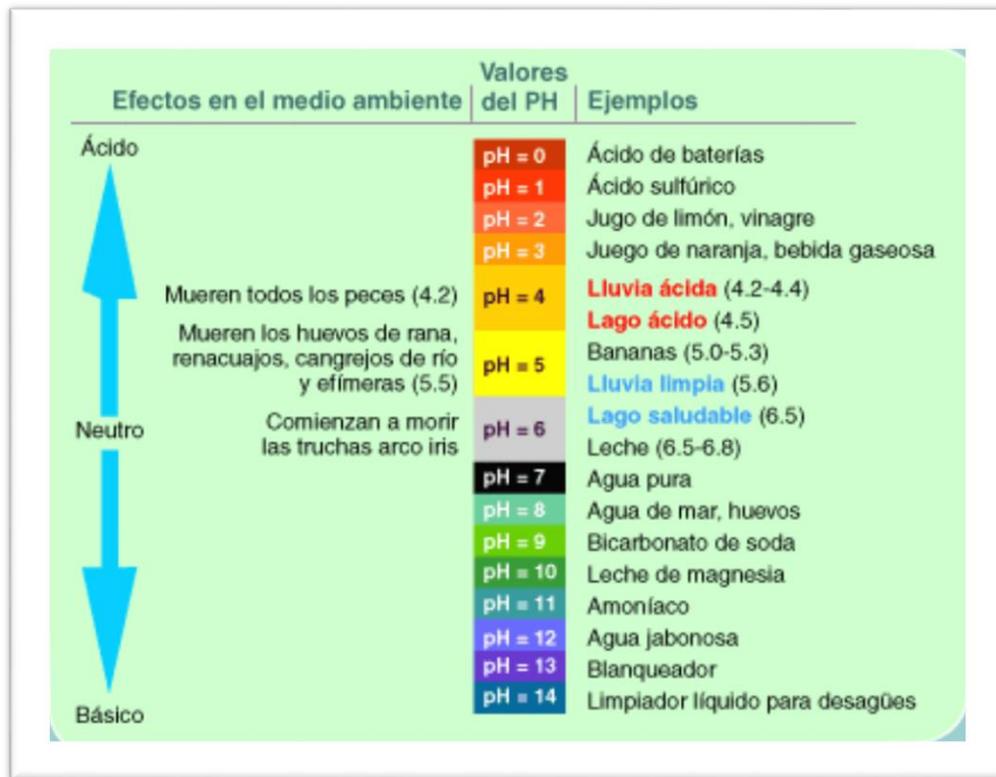


Figura 4 Escala de pH junto ejemplos
Recuperado de U.S. EPA

2.1.2.2 REDOX

El potencial REDOX o ORP, la cual es la abreviación de potencial de oxidación-reducción, es una medición muy útil para saber la actividad de los electrones presente en el agua o en sistemas acuosos. La medición se realiza de la misma manera de que se mide el pH, pero en este caso se utiliza un electrodo distinto,

el cual está constituido por metales noble, no como el electrodo del pH el cual está constituido por vidrio.

El valor obtenido por la medición Redox se calcula a partir de la diferencia de potencial entre un sistema de referencia y el sistema en donde se realice la medición, utilizando la ecuación de Nernst, el cual otorga un valor en milivoltios (mV) el cual se puede estar dentro de un rango de +/- 2000 mV siendo un valor de 650 mV un valor adecuado para el agua potable , debido a que un estudio demostró que entre los valores 650 y 700 mV variedad de bacterias desaparecen a los pocos segundos con ese voltaje, este valor depende también de la temperatura, por lo tanto es necesario indicar la temperatura en cada medición.

Ecuación de Nernst:

Ecuación 1 Ecuación de Nernst

$$E = E^{\circ} - \frac{RT}{nF} \ln(Q)$$

En donde:

- **E** es el potencial corregido del electrodo.
- **E⁰** es el potencial en condiciones estándar (los potenciales se encuentran tabulados para diferentes reacciones de reducción).
- **R** es la constante de los gases en Joules sobre Kelvin por mol.
- **T** la temperatura absoluta (escala Kelvin).
- **n** la cantidad de mol de electrones que participan en la reacción.
- **F** la constante de Faraday (aproximadamente 96500 C/mol).
- **Q** el cociente de reacción correspondiente.

Una reacción de reacción Redox (reducción-oxidación) es una reacción en la cual se produce una transferencia de electrones en donde el que pierde electrones se oxida y el que gana se reduce.

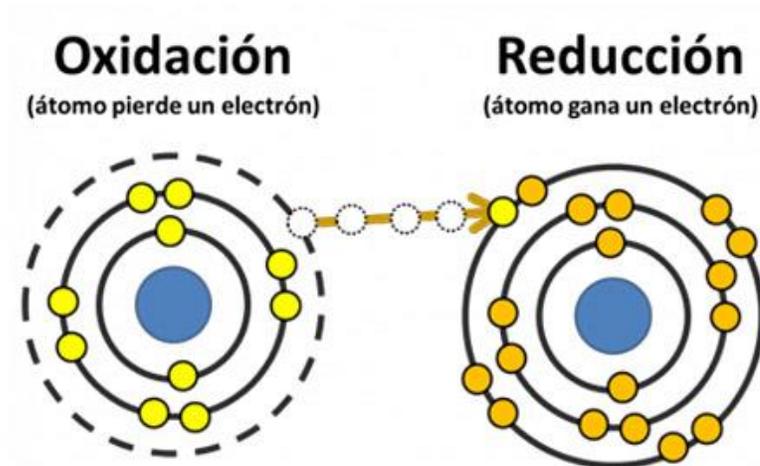


Figura 5 Redox.

2.1.2.3 CLORO

El cloro es uno de los principales agentes desinfectantes utilizados en la potabilización del agua, gracias a sus propiedades oxidantes capaces de destruir las bacterias y elementos nocivos además a su bajo costo, fácil adquisición y gran eficiencia a la hora de realizar el proceso.

Todos los productos clorados, que contienen cloro, en su tratamiento producen ácido hipocloroso (HClO) al reaccionar con el agua, este ácido es débil el cual al entrar en contacto con el agua se disocia, es decir, sus moléculas se separan en moléculas más pequeñas formando hipoclorito (ClO^-), siempre y cuando el valor de pH se encuentre en un valor óptimo.

Ya aplicado el cloro en el agua, este se divide en dos tipos:

-Cloro libre residual: Es el cloro que aún se encuentra en forma de hipoclorito o ácido hipocloroso, aun no reacciona con la materia orgánica a oxidar y por lo tanto será la proporción del elemento encargado de mantener un cierto porcentaje de protección microbiológica en el agua ante agentes contaminantes

externos. Sin embargo, la cantidad de cloro libre residual en nuestro país debe ser de a lo menos 0,2 mg/L ya que un valor mayor puede provocar problemas perjudiciales para la salud.

-Cloro combinado residual: Es el cloro que ya ha oxidado a la materia orgánica y se ha combinado con ella, y posee un nivel de capacidad desinfectante muy bajo. Se recomienda que el nivel de cloro combinado residual se encuentre en 1mg/L.

-Cloro total: Es la suma de los anteriores cloros mencionados.

El ácido hipocloroso ya mencionado es la forma activa del cloro, el cual para poder desinfectar de manera eficiente es importante que el parámetro del pH se encuentre dentro de unos márgenes ya que, si no es así, el poder desinfectante del cloro será muy bajo.

A continuación, se presenta una tabla con los valores óptimos de pH para el correcto funcionamiento del cloro.

pH	Cloro residual libre necesario (mg/L)
7.0	0.5
7.4	0.7
7.7	1.0
7.9	1.8

Tabla 1 Valores óptimos de pH y cloro

2.1.3 MUESTREO DE SEÑALES

La manera de registrar gran variedad de parámetros o información en la actualidad es mediante el muestro de señales, la cual consiste en tomar muestras de una señal cada un determinado tiempo, las cuales podrían ser cualquier perturbación física con la capacidad de ser medible que transporte información.

Estas señales son análogas las cuales constan de una amplitud que puede tomar cualquier valor, es decir, su nivel de tensión en cualquier muestra no está limitado a un conjunto finito de niveles predefinidos.

El muestreo o también llamado “Discretización de una señal” es el proceso inicial para la conversión de una señal análoga en una señal digital y consiste en obtener muestras de la señal analógica en intervalos de tiempos constante registrando la amplitud de la señal en cada uno de estos instantes.

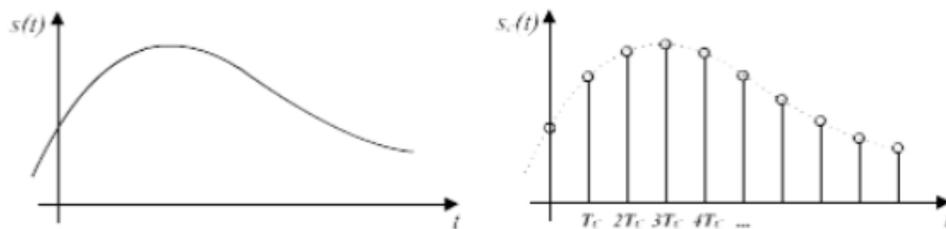


Figura 6 Discretización de una señal

2.1.3.1 TEOREMA DE NYQUIST

Este teorema trata sobre el muestreo, proceso realizado previamente al proceso de cuantificación, este último mencionado es un proceso irreversible en el cual existe pérdida de información hasta en los casos más ideales.

Las muestras discretas de una señal son valores exactos que aún no han sufrido redondeo o truncamiento alguno sobre una precisión determinada, es decir, aún no han sido cuantificadas. El teorema demuestra que la reconstrucción exacta de una señal periódica continua en base a partir de sus muestras es viable matemáticamente, siempre y cuando la señal este limitado en banda y la tasa de muestreo superior sea el doble de su ancho de banda

Por ejemplo, matemáticamente hablando Si la frecuencia más alta contenida en una señal analógica $x(t)$ con una frecuencia $F_{\max} = B$, y esta señal tiene una tasa de muestreo de $F_s > 2 \cdot F_{\max} = 2B$ entonces esta señal analógica $x(t)$ podrá ser recuperada totalmente a partir de sus muestras mediante una función de

interpolación.

2.2 HERRAMIENTAS DEL PROYECTO

A continuación, se presentan las características principales de los dispositivos o programas utilizados para el desarrollo del prototipo.

2.2.1 HARDWARE

2.2.1.1 RASPBERRY PI 2 MODEL B

2.2.1.1.1 CARACTERÍSTICAS DEL HARDWARE

La placa Raspberry 2 es la segunda generación de las placas Raspberry. Esta es una placa con sistema embebido, es decir prácticamente un computador al cual se le puede asignar diversas tareas, de bajo coste gracias a sus múltiples salidas y entradas en su hardware, como también sus puertos USB que permiten conectar periféricos, conector ethernet, entre otros...

A continuación, se listarán las características principales del hardware de las Raspberry Pi 2 modelo B:

- CPU de cuatro núcleos ARM Cortex-A7 a 900MHz
- GPU Broadcom VideoCore IV 250 MHz. OpenGL ES 2.0
- SDRAM LPDDR2 de 1 GB (Memoria Ram)
- 4 conectores hembra USB 2.0
- Conector macho de 40 pines para buses serie y GPIO
- Conector hembra de vídeo/audio HDMI 1.3 y 1.4
- Ranura para tarjetas microSD
- Conector hembra Ethernet RJ45 10/100 BaseT
- Conector DSI
- Conector CSI
- Conector Jack de 3,5 mm

- Compatibilidad completa con los anteriores modelos de Raspberry Pi
- Fuente de alimentación: +5 V a 2 A a través de conector hembra microUSB
- Dimensiones: 86 x 56 x 20 mm

Rasperry cuenta con una ranura para insertar tarjetas microSD lo que da la posibilidad de instalar un sistema operativo fácilmente en una tarjeta económica de 4 GB, a pesar de que mientras más sea la capacidad será mejor.

Una de las características principales de la placa Raspberry Pi es la capacidad de conectar distintos dispositivos auxiliares e independientes que van directamente conectados a la unidad central de procesamiento de la placa, con la finalidad de comunicarse con el exterior (entradas y salidas) o de archivar y almacenar información de sistemas (memorias auxiliares).

Dentro de ellos se encuentran los siguientes dispositivos:

- Teclado y ratón.
- Cámara.
- Salida de video (HDMI)
- Servomotores
- Puertos GPIO

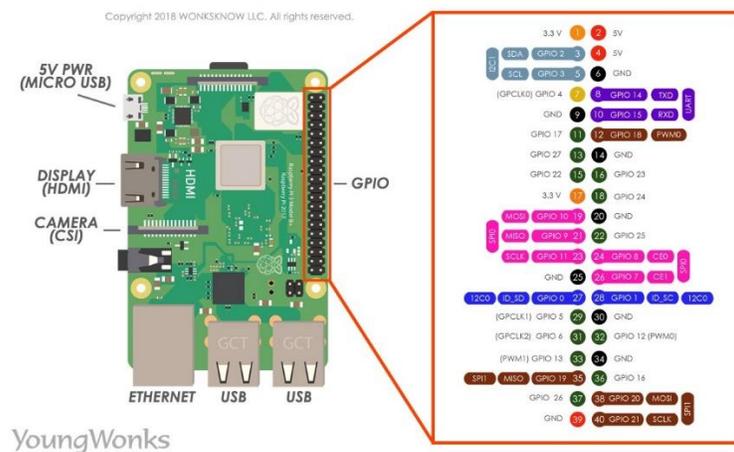


Figura 7 Puertos de conexión de la Raspberry Pi 2B.

2.2.1.1.2 CARACTERISTICAS DEL SOFTWARE

La placa de desarrollo Raspberry puede utilizar diversos tipos de sistema operativos, ya sean gratuitos o de pago, dependiendo del uso que se le dará.

Raspbian es un sistema operativo libre y gratuito basado en Debian optimizado para el hardware Raspberry Pi.

Raspbian ofrece más que un sistema operativo puro, ya que viene con más de 35.000 paquetes, softwares precompilados para una fácil instalación en Raspberry Pi cuando sea necesario utilizarlos.

Este sistema operativo se encuentra disponible para descargar en la página oficial de Raspberry, en la sección descargas.

Raspbian integra una interfaz gráfica muy similar a Windows, a la cual podremos acceder utilizando un monitor mediante un cable HDMI o por programas terceros como VNC Viewer. Principalmente se ingresará a esta mediante un servidor SSH utilizando el programa Putty.

2.2.1.1.3 PUERTOS GPIO

Los puertos GPIO (General Purpose Input/Output) se trata de un bus de expansión de 40 pines en los modelos más nuevos como el modelo 2 y 3, ya que en las versiones anteriores solo disponen de 26 pines. Permiten a las Raspberry Pi comunicarse con el exterior, ya se para activar elementos como también leer el estado de estos. La tensión de trabajo de los puertos es de 3,3 V para un 1 lógico y 0 V para un 0. Sin embargo, igual dispone de pines de tensión de 5V y

3.3V para la alimentación de dispositivos con dicha tensión.

La corriente máxima que puede proveer cada pin es de 16 mA, siendo 50 mA la corriente total suministrada por los puertos GPIO.

Es de suma importancia tener conocimiento sobre la cantidad de tensión que puede recibir cada pin, ya que una sobrecarga puede provocar que la placa se quemara y no vuelva a funcionar. Por ejemplo, los chips y circuitos de la placa utilizan solo 3,3V por lo cual, si se conecta un sensor que envíe una señal de 5V, la placa se sobrecargara.

Raspberry Pi Model A+/B+/Pi 2			
B+ J8 GPIO Header			
	Pin No.		
3.3V	1	2	5V
GPIO2	3	4	5V
GPIO3	5	6	GND
GPIO4	7	8	GPIO14
GND	9	10	GPIO15
GPIO17	11	12	GPIO18
GPIO27	13	14	GND
GPIO22	15	16	GPIO23
3.3V	17	18	GPIO24
GPIO10	19	20	GND
GPIO9	21	22	GPIO25
GPIO11	23	24	GPIO8
GND	25	26	GPIO7
DNC	27	28	DNC
GPIO5	29	30	GND
GPIO6	31	32	GPIO12
GPIO13	33	34	GND
GPIO19	35	36	GPIO16
GPIO26	37	38	GPIO20
GND	39	40	GPIO21

Figura 8 Puertos GPIO de la Raspberry Pi 2B.

2.2.1.2 NODEMCU ESP32

2.2.1.2.1 CARACTERISTICAS DEL HARDWARE

La placa NodeMCU está diseñada especialmente para trabajar en protoboard. Posee un regulador de voltaje el cual le permite alimentarse directamente desde un puerto USB

A continuación, se listarán las características principales del hardware de la tarjeta de desarrollo NodeMCU:

- CPU principal: Tensilica Xtensa 32-bit LX6
- Frecuencia de Reloj: hasta 240Mhz
- Desempeño: Hasta 600 DMIPS
- Procesador secundario: Permite hacer operaciones básicas en modo de ultra bajo consumo
- Wifi: 802.11 b/g/n/e/i (802.11n @ 2.4 GHz hasta 150 Mbit/s)
- Bluetooth:v4.2 BR/EDR and Bluetooth Low Energy (BLE)
- Xtensa® Dual-Core 32-bit LX6 microprocesador, up to 600 DMIPS
- Pines Digitales GPIO: 24 (Algunos pines solo como entrada)
- Converso analógico Digital: Dos ADC de 12bits tipo SAR, soporta mediciones en hasta 18 canales, algunos pines soportan un amplificador con ganancia programable
- UART: 2



Figura 9 Hardware NodeMCU ESP8266

2.2.1.2.2 CARACTERISTICAS DEL SOFTWARE

NodeMCU es una placa de desarrollo la cual permite mayor facilidad a la hora de realizar proyectos especialmente orientados al Internet de las cosas (IoT), La plataforma ESP32 ofrece a bajo costo capacidades wifi, bluetooth y BLE. Es la evolución del chip ESP8266 mejorando sus capacidades de comunicación y procesamiento computacional. Un chip altamente integrado, diseñado para satisfacer las necesidades de un mundo conectado. Integra un potente procesador con un Arquitectura de 32 bits.

Para el desarrollo de aplicaciones se puede elegir entre los lenguajes de Arduino y Lua. Así es, se puede trabajar dentro del entorno Arduino y hacer uso entorno de desarrollo integrado (IDE) sencillo de utilizar. Esto permite que la CPU ESP32 y sus componentes Wi-Fi sean programados como cualquier otro dispositivo Arduino.

El lenguaje de programación con el cual opera el IDE de Arduino es llamado "Wiring" el cual está basado en la plataforma Processing – C/C++. En este entorno de programación se escribe el programa que se desea, además consta con varias funciones disponibles que se encuentran en la barra de herramientas.

2.2.1.2.3 DESCRIPCION DE PINES

Esta placa de desarrollo posee un total de 30 pines, los cuales permiten el trabajo junto con el módulo ESP8266 que tiene integrado, no todos los pines son iguales o tienen la misma función, dentro de ellos se encuentran distintos tipos dentro de los cuales existen:

Pines GPIO: Al igual que en la placa Raspberry, está dispone de Pines GPIO (General Purpose Input/Output) los cuales pueden ser configurados como entradas y salidas digitales.

El Pin A0 o TOUT: Este Pin está destinado a la obtención de entradas analógicas, el cual funciona cuando recibe una señal desde un receptor hacia el PIN y este la modula, el rango de la tensión puede estar entre 0 y 5 voltios. Esta tensión es recibida y convertida en valores digitales que van desde el 0 al 1024.

Pines PWM (Pulse Width Modulation): Estos pines permiten generar salidas analógicas desde los pines digitales

Pines SPI (Serial Peripheral Interface): Son utilizados para la comunicación entre dispositivos que dispongan de un circuito integrado siguiendo un protocolo de comunicación síncrono, es decir, a la misma frecuencia del reloj de los circuitos. Un ejemplo de esta comunicación sería las pantallas LED o OLED. Entre los pines SPI se encuentran *SPICLK*, *SPIQ (MISO)*, *SPID (MOSI)* y *SPICS*.

Pines UART (Universal asynchronous Receiver / Transmitter): Son utilizados para la transmisión (TX) y recepción (RX) de datos de la placa. Los datos que se envíen a la Nodemcu nos permiten configurar la velocidad de transmisión como también el formato. Los pines utilizados para esta función serían: *TXD0 (GPIO1)*, *RXD0 (GPIO3)*, *TXD1 (GPIO2)*, *TXD2 (GPIO15)*, *RXD2 (GPIO13)*.

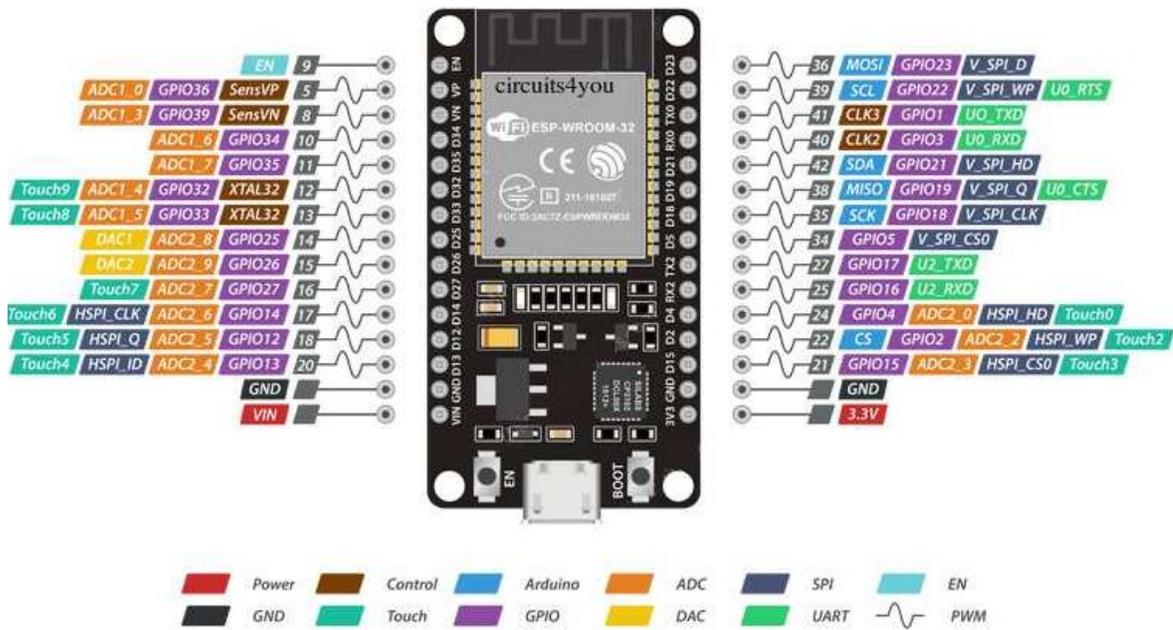


Figura 10 Pines NodeMCU ESP32

En la Figura 7 se presentan los pines GPIO con una letra y un número, ya que de esa forma es como se deben ingresar en el IDE de Arduino a la hora de programar.

2.2.1.3 ADS1015

El ADS1015 es un convertidor análogo a digital (ADC) de precisión perteneciente a la compañía de hardware de código abierto “AdaFruit”, el cual pueden utilizar los microcontroladores que no incluyen un conversor análogo a digital, como por ejemplo la Raspberry Pi.

Este conversor permite una resolución de 12 bits a 3300 muestras/segundo (SPS, samples per second). Los datos se transfieren a través de una interfaz serial compatible con I2C; Se pueden seleccionar cuatro direcciones de esclavo I2C o 4 canales simultáneamente. También incluye un amplificador de ganancias programable de hasta x16 lo que ayuda a amplificar las señales de menor

intensidad.

El ADS1015 funciona desde una única fuente de alimentación que varía de 2.0V a 5.5V.

Características del convertidor análogo a digital ADS1015:

- Amplio rango de alimentación: 2V hasta 5.5V.
- Bajo consumo de corriente: solo 150µa. Posee auto apagado para modo de lectura simple
- Data rate programable: 128sps hasta 3.3ksps
- Voltaje de referencia interno.
- Oscilador integrado.
- PGA integrado.
- Interfaz I2C: dirección seleccionable por pin.
- Cuatro entradas análogas simples o dos diferenciales.
- Comparador programable

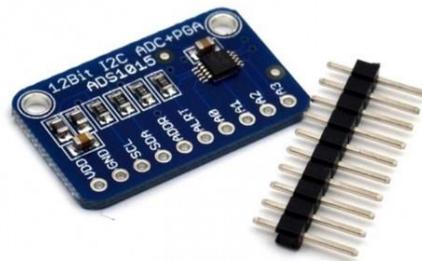


Figura 11 Hardware ADS1015

2.2.1.4 SENSOR DE PH

El sensor de PH analógico Gravity Sku Sen0161 está diseñado por la empresa

“DFROBOT”, y es especialmente para ser utilizado junto a controladores como Arduino o Raspberry, permitiendo una fácil conexión y programación para el monitoreo de PH de bajo costo.

Este dispone de un electrodo sumergible que cumple la función de adquirir la información del líquido en el cual este sumergido, el electrodo va conectado mediante un conector BNC a una placa de conversión de señal desde la cual realizando la conexión a un puerto de entrada analógica de cualquier controlador se podrá obtener los datos. Este sensor puede ser utilizado para el testeo de calidad de agua como también para la agricultura y la acuicultura.



Figura 12 Sensor de PH

2.2.1.4.1 DESCRIPCION DE PINES

1- VDD	Power supply 3.3~5.5V DC
2- GND	Ground, the negative power
3- DATA	Serial Data

Tabla 2 Pines sensor de pH

2.2.1.4.2 PARAMETROS DEL SENSOR

- Voltaje de trabajo: 3.3- 5.5V.
- Tamaño del módulo: 43 x 32mm.
- Rango de medición: 0-14pH.
- Medición de Temperatura: 0-60°C.
- Precisión: ± 0.1 pH (25°C).
- Tiempo de respuesta: ≤ 1 min.
- Sensor de pH con conector BNC.
- Ajuste de ganancia con potenciómetro.
- Indicador LED de alimentación.

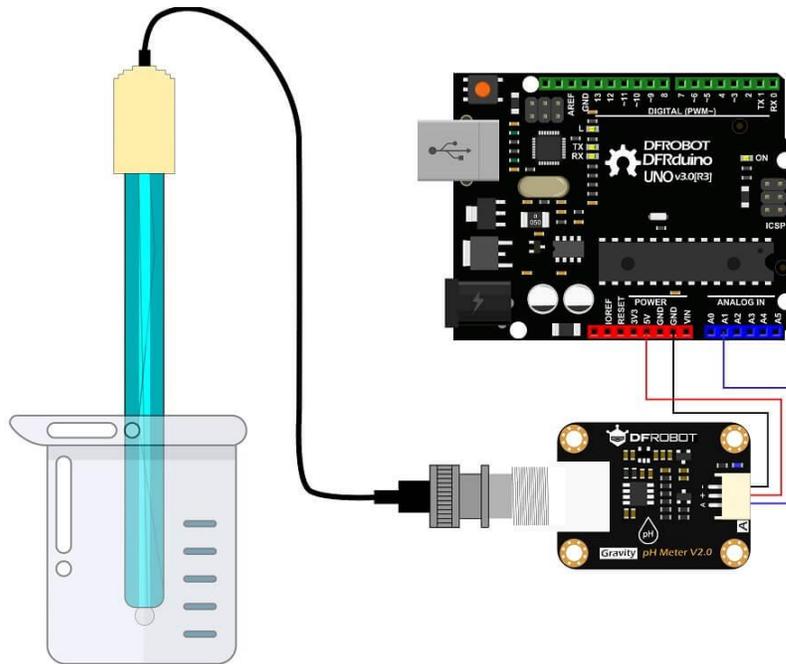


Figura 13 Ejemplo conexión sensor de pH
rescatada de <https://wiki.dfrobot.com/>

2.2.1.5 SENSOR ORP

El sensor Gravity: Analog ORP es bastante similar al sensor de pH, también posee una placa de conversión de señal junto con un electrodo, pero en este caso, el electrodo este hecho de oro o platino, por lo tanto, su precio es más elevado.

El ORP es una medida de la capacidad de oxidación y reducción de una solución acuosa. Si el potencial de reducción de la oxidación es alto, la oxidación química es fuerte, mientras que, si el potencial es menor, entonces la oxidación es más débil.

Debido a que la medición va desde -2000mv a 2000mv, si el potencial es positivo significa que la solución muestra un cierto grado de oxidación, mientras que el potencial negativo significa que la solución muestra un cierto grado de reducción.

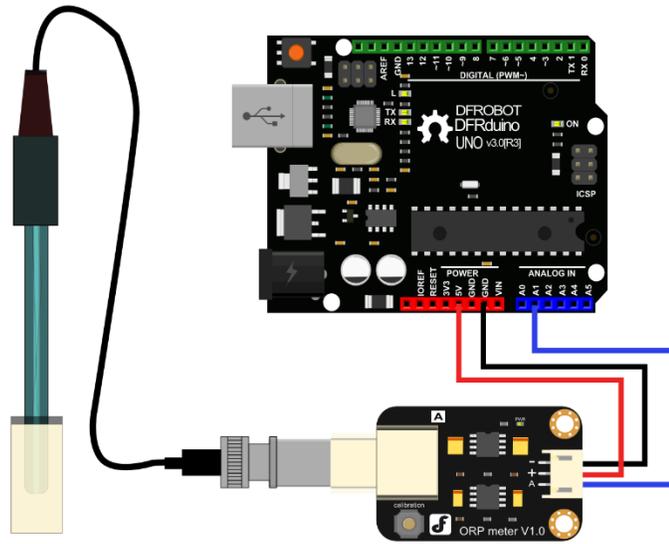


Figura 14 Ejemplo conexión sensor ORP

2.2.1.5.1 DESCRIPCION DE PINES

1- VDD	Power supply 5.5V DC
2- GND	Ground, the negative power
3- DATA	Serial Data

Tabla 3 Pines sensor ORP

2.2.1.5.2 PARAMETROS DEL SENSOR

- Potencia del módulo: + 5.00V
- Tamaño del módulo: 40 mm x 27 mm
- Rango de medición: -2000mV ~ 2000mV
- Temperatura adecuada: 5 ~ 70 ° C
- Precisión: $\pm 10\text{mv}$ (25 ° C)
- Tiempo de respuesta: ≤ 20 segundos
- Sonda ORP con conector BNC
- Interfaz PH2.0 (parche de 3 pies)
- Botón de calibración cero
- LED indicador de encendido

2.2.1.6 SENSOR DE TEMPERATURA

El sensor de temperatura a utilizar en el prototipo será el DS18B20, uno de los sensores más versátiles que se puede encontrar, es capaz de medir la temperatura en ambientes húmedos e incluso dentro del agua gracias a que existe un modelo que incorpora una termocupla.

A pesar de que utiliza el protocolo 1-Wire, que no están común, posee una variedad de librerías que hace que la programación de este sea bastante sencilla.



Figura 15 Sensor DS18B20 Termocupla

2.2.1.6.1 DESCRIPCION DE PINES

Este sensor tiene tres pines útiles VD, GND y DQ.

VD es la tensión de alimentación, es decir, que voltaje necesita para que el sensor de temperatura DS18B20 funcione correctamente. Podemos alimentar desde 3V a 5,5V

GND es la toma de tierra.

DQ es el pin de datos. Por este pin es por donde se recibirán todos los datos en el protocolo 1-Wire. Este protocolo tiene una ventaja. Como su propio nombre indica (1-Wire significa un cable en español) solo es necesario utilizar un cable para conectar varios sensores de temperatura DS18B20. Por lo tanto, se puede utilizar solo un pin para conectar múltiples sensores.

2.2.1.6.2 PARAMETROS DEL SENSOR

Característica	Valor
Voltaje de alimentación	3V a 5,5V
VDD	Voltaje de alimentación

GND	Tierra
DQ	Datos
Rango de temperaturas	-55°C a 125°C
Error (-10°C a 85°C)	±0,5°C
Error (-55°C a 125°C)	±2°C
Resolución programable	9-bit, 10-bit , 11-bit o 12-bit (default)

Tabla 4 Parámetros sensor de temperatura

2.2.1.7 PUENTE H L293D

A la hora de energizar los actuadores que en este caso serán bombas de agua, se requiere de un circuito llamado “Puente H” el cual permite el control de un motor DC en ambas direcciones. Los puentes H están disponibles en circuitos integrados como lo es el L293D o también se puede construir mediante componentes discretos.

Al mismo tiempo de proporcionar un fácil control de los motores, también previene de cortocircuitos y corrientes que puedan ir hacia la placa controladora provocando que se dañe.

El circuito integrado L293D incluye en su interior 4 drivers o medio puente H por lo tanto con solo uno se podrán controlar dos bombas de aguas independientemente, ya que en este caso solo se necesita que estas hagan circular el líquido en solo una dirección.

Algunas características del circuito integrado L293D:

- Alimentación: 45 a 36 VDC
- Corriente de salida: 600 mA.
- Corriente pico de salida: 1 A por canal.
- Encapsulado: DIP de 16 pines.
- Alta inmunidad al ruido eléctrico.

- Protección contra exceso de temperatura.
- Diodos de protección incorporados.

2.2.1.8 ACTUADORES

Para realizar el control de la calidad del agua en el prototipo se utilizarán 3 bombas de agua sumergibles, dos para agregar el químico que se encargara de aumentar y disminuir el pH del agua. Y otra para agregar cloro para la desinfección.

Características:

- Mini Bomba sumergible 2.5-6 Vdc 80-120L/H
- Voltaje: 2.5-7 Vdc
- Altura máxima: 40-110 cm / 15.75"-43.4"
- Flujo: 80-120 L/H
- Diámetro exterior de la salida de agua: 7.5 mm / 0.3"
- Diámetro interior de la salida de agua: 4.7 mm / 0.18"
- Diámetro: Aproximado 24 mm / 0.95"
- Largo: Aproximado 45 mm / 1.8"
- Altura: Aproximada 33 mm / 1.30"
- Material: plástico
- Vida de trabajo continuo de 500 horas



Figura 16 Bomba de agua sumergible

2.2.1.9 FUENTES DE PODER

Para la alimentación de sensores y de actuadores se usarán baterías lipo, un tipo de batería recargable y pequeñas, bastante útil para dispositivos que no puedan ser conectados a la red eléctrica directamente con un cable.

Características:

- Capacidad: 370mAh
- Voltaje: 2S1P / 2 Cell / 7.4V
- Descarga: 25C constante / 40C Burst
- Peso: 27g incluyendo cable, enchufe y estuche)
- Dimensiones : 63x32x9mm
- Enchufe de equilibrio: JST-XH
- Enchufe de descarga: Mini -JST conector.



Figura 17 Batería Lipo

2.2.2 SOFTWARE

2.2.2.1 TCP/IP

Siempre que uno o varios dispositivos interactúan entre sí, ya sea enviando datos o señales, lo hacen mediante ciertos protocolos, los cuales deben cumplir normas o requisitos para la correcta transmisión o recepción de datos. Dentro de estos

protocolos de encuentra uno de los más utilizado mundialmente en internet, TCP/IP , acrónimo el cual proviene de la unión de dos protocolos, TCP, Protocolo de control de transmisión o en inglés “Transmission Control Protocol” y el protocolo IP, “Internet Protocol”.

2.2.2.1.1 PROTOCOLO TCP

El protocolo de control de transmisión (TCP) es uno de los principales protocolos que se encuentran en la capa de transporte dentro del Modelo OSI, este permite que la transmisión de datos entre dispositivos puede realizarse de forma segura, ya que cuenta con un sistema de acuse de recibo con el cual se verifica que todos los datos se hayan entregado de manera correcta o que no exista ningún dato corrompido.

Para la correcta transmisión de datos, los dos dispositivos o maquinas deben establecer una conexión consensuada, es decir, una maquina solicita a la otra y esta última debe aceptar la solicitud, comenzando un entorno de Cliente-Servidor, en donde dependiendo de la función o configuración de cada una se le asigna un rol.

Este protocolo permite dividir el flujo de información en segmentos ya sea de igual o distinto tamaño, dependiendo de la cantidad de datos que contenga y la configuración adicional que se agrega, además de añadir la cabecera del mismo protocolo, este segmento luego es encapsulado en un datagrama IP para posteriormente ser enviado por la red.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
Puerto TCP origen																Puerto TCP destino															
Número de secuencia																															
Número de acuse de recibo																															
HLEN				Reservado				Bits de código				Ventana																			
Suma de verificación																Puerto de urgencia															
Opciones (si las hay)																								Relleno							
Datos																															
...																															

Figura 18 Trama TCP.

2.2.2.2 PROTOCOLO MQTT

MQTT O Message Queue Telemetry Transport es un protocolo de comunicación orientado al uso de sensores mediante principalmente redes inalámbricas, el cual permite enviar datos de telemetría (medición remota de magnitudes físicas a distancia) como mensajes, a través de redes de alta latencia. Permitiendo también la comunicación Machine to Machine e internet de las cosas (OASIS Standard, 2015)

Este protocolo está diseñado específicamente para dispositivos con recursos limitados, canales de comunicación de bajo ancho de banda y redes no confiables.

Sigue un esquema de Publish/subscribe, que es una alternativa al modelo cliente-servidor dentro de él se encuentran las siguientes entidades típicas involucradas en este protocolo que son Publisher, Subscribers y Brokers.

Los clientes que utilizan el protocolo pueden actuar enviando o recibiendo mensajes a través del servidor MQTT o Broker, este último está vinculado con los publishers y subscribers, y su función principal es filtrar los mensajes entrantes y distribuirlos apropiadamente (Mijovic, Shehu & Buratti, 2016).

Estos datos enviados al Broker por el Publishers, el cual son dispositivos de detección que recopilan los datos del entorno, se enviaran desde el Broker, a los determinados subscribers dependiendo al tópico que estén suscritos.

El principio con el cual se produce el intercambio de datos es el siguiente:

Los publishers envían los mensajes o datos a un canal llamado topic, al cual los subscribers deben estar suscritos para poder recibirlo. Los tópicos pueden estar distribuidos jerárquicamente, para poder seleccionar exactamente los datos que necesitamos y solicitamos.

Breve resumen:

1. El cliente se conecta al bróker. Puede suscribirse a cualquier "topic" de mensajería del bróker. Esta conexión puede ser una conexión TCP/IP simple o una conexión TLS cifrada para mensajes sensibles.
2. El cliente publica los mensajes en un tema, enviando el mensaje y el tema al bróker.
3. Después, el bróker remite el mensaje a todos los clientes que se suscriben a este tema.

Gracias a que los mensajes se pueden distribuir por tópicos, el desarrollador de las aplicaciones podrá determinar que clientes pueden interactuar con algunos mensajes y cuáles no. Por ejemplo, en la figura los sensores publicarán en el tópico "sensor_data" y estos mismos se suscribirán a "config_change" en el cual

una consola administradora podrá ir publicando las configuraciones del sensor.

También el cliente encargado del procesamiento de los datos se suscriba al tópic "sensor_data" para poder recibir los mensajes publicados por el sensor.

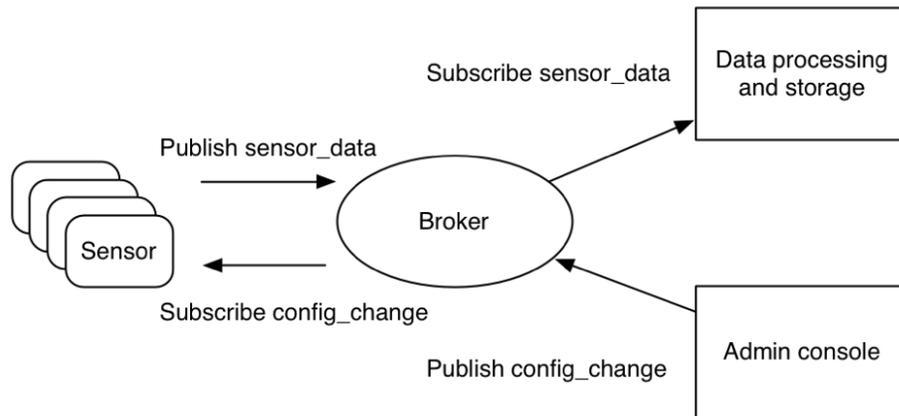


Figura 19 El modelo de publicación y de suscripción de MQTT

Recuperado de <https://www.ibm.com/>

Un ejemplo básico para entender la jerarquía que se puede aplicar en los tópicos utilizados en el protocolo MQTT es el siguiente:

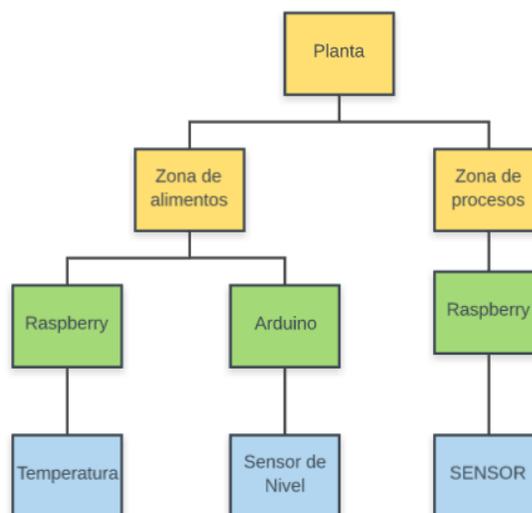


Figura 20 Jerarquización de los tópicos MQTT

En donde al suscribirse al t3pico “Planta”, se obtendr3a acceso a todos los datos enviados por los dispositivos que se encuentren por debajo de este, por otro lado, si un cliente se suscribe al t3pico “Zona de alimentos”, este tendr3a acceso solo a los dispositivos destinados a esa zona, es decir, solo obtendr3a los datos de temperatura y del sensor de nivel y no los datos de la Raspberry que se encuentra en la “Zona de procesos” debido a que es otro t3pico al cual no est3 suscrito.

2.2.2.2.1 PUBLISH

Al conectarse un cliente al servidor “Broker” tiene la posibilidad de publicar mensajes. Cada mensaje debe contener el t3pico al cual va dirigido, adem3s debe incorporar un payload, el cual contendr3 los datos en formato de bytes a ser transmitidos.

2.2.2.2.2 SUBSCRIBE

Un cliente debe suscribirse a un t3pico del Broker para comenzar a recibir datos, este mensaje es bastante simple, solo basta con un identificador y una lista de suscripciones (en caso de mas de 1 t3pico) con sus respectivos QoS, en caso de no especificar, por defecto ser3 0.

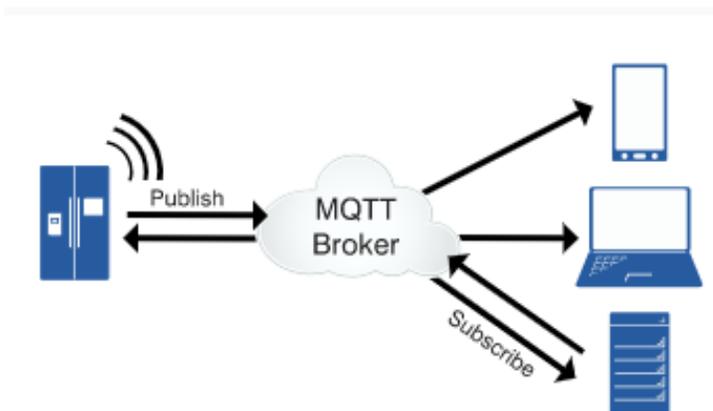


Figura 21 Funciones Broker MQTT

Existe una variedad de software que permiten convertir un dispositivo o

computadora en un Broker, ya sea de forma remota o local, uno de los más destacados es Mosquitto, debido a su versión open source, este tiene soporte para todos los sistemas operativos más utilizados en la actualidad.

2.2.2.3 ECLIPSE MOSQUITTO

Eclipse Mosquitto es un servidor de mensajes de código abierto, el cual implementa las versiones 5.0, 3.1.1 y 3.1 del protocolo MQTT. Aquello proporciona un método ligero para llevar a cabo la mensajería utilizando el modelo publishers/subscribers , lo que lo hace adecuado para usar en todo dispositivo, desde computadoras de una sola placa de baja potencias, hasta servidores muy complejos.

Además, es adecuado para la transmisión de datos dentro del mundo “IoT”, como sensores de baja potencia, o dispositivos pequeños como teléfonos, sistemas embebidos o microcontroladores.

2.2.2.4 PYTHON

Python es un lenguaje de programación que esta orientado a objetos claros y potentes, similar a Perl, Ruby, Scheme o Java. Algunas de las características notables de Python son:

- Utiliza una sintaxis elegante, lo que hace que los programas que escribes sean más fáciles de leer.
- Es un lenguaje fácil de usar que simplifica el funcionamiento de su programa. Esto hace que Python sea ideal para el desarrollo de prototipos y otras tareas de programación ad-hoc, sin comprometer la mantenibilidad.
- Viene con una gran biblioteca estándar que admite muchas tareas de programación comunes, como conectarse a servidores web, buscar texto

con expresiones regulares, leer y modificar archivos.

- Se ejecuta en cualquier lugar, incluyendo Mac OS X , Windows , Linux y Unix , con compilaciones no oficiales también disponibles para Android e iOS.

2.2.2.5 HTML5 y JAVA

HTML es el acrónimo que significa en inglés “HyperText Markup Language” (Lenguaje de marcado de hipertexto), no es un lenguaje de programación ya que no dispone de la capacidad de crear una funcionalidad dinámica, es mas bien un lenguaje informativo ya que está diseñado para permitir la creación de sitios web con el fin de mostrar documentos o información dentro de una red local o para todo el internet.

La definición de HTML es HyperText Markup Language:

- HyperText es el método por el cual se navega por la web, haciendo clic en un texto especial llamado hipervínculos el cual permitirá el acceso a otra página. El hecho de que sea hiper simplemente significa que no es lineal, es decir, se puede ir a cualquier lugar de internet cuando se desee haciendo clic en los enlaces, no hay un orden establecido para hacer las cosas.
- Lenguaje de marcado: un lenguaje de marcado es un lenguaje de computadora que se utiliza para aplicar convenciones de diseño y formato a un documento de texto. El lenguaje de marcado hace que el texto sea más interactivo y dinámico. Puede convertir texto en imágenes, tablas, enlaces, etc.
- HTML es un lenguaje , ya que tiene palabras de código y sintaxis como cualquier otro lenguaje.

HTML puede ser asistido por otras tecnologías como CSS y JavaScript, ya que por si solo, no es posible construir un sitio web profesional, solo se obtendría el texto ingresado y la estructura de este.

En cambio, implementado JavaScript y CSS funciona muy bien, logrando un buen diseño web y una grata experiencia para el usuario.

- CSS es responsable de los estilos, como el fondo, los colores, los diseños, el espaciado y las animaciones.
- JavaScript le permite agregar funcionalidades dinámicas como controles deslizantes, ventanas emergentes y galerías de fotos.

HTML5 es la versión mas reciente del lenguaje HTML el cual fue anunciado el 22 de enero del 2008 y lanzado el año 2014, en la cual se encuentran dos organizaciones , una es W3C (World Wide Web Consortium) y otra es WHATWG (Grupo de trabajo de tecnología de aplicación de hipertexto web) , las cuales participaron y sigue trabajando en el desarrollo de HTML5.

HTML5 va actualizando constantemente con mas funciones para hacer que el internet sea accesible para todos, sus características son compatibles con todos los navegadores además HTML5 comienzo a admitir la API de JavaScript , como la API de geolocalización para identificar la ubicación, el soporte de aplicaciones móviles multiplataforma, entre otras.

2.2.2.6 PHP

El preprocesador de hipertexto o por su acrónimo “PHP” es un lenguaje de script de código abierto especialmente adecuado para el desarrollo de páginas web, utilizando HTML.

Lo que distingue a PHP de algo del lado del cliente como JavaScript es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá el resultado de ejecutar el script, aunque no se sabrá el código

subyacente que era. El servidor web puede ser configurado incluso para que procese todos los ficheros HTML con PHP, por lo que no hay manera de que los usuarios tengo los permisos para acceder a tal información.

Una de las características más potentes y destacables de PHP es su soporte para un amplio abanico de bases de datos. Escribir una página web con acceso a una base de datos es increíblemente simple utilizando una de las extensiones específicas de bases de datos, por ejemplo, para MySQL el cual utiliza PHPMyAdmin.

PHPMyAdmin es una herramienta de administración gratuita y de código abierto para las bases de datos MySQL y MariaDB.

Las características proporcionadas por el programa incluyen:

- Interfaz web
MySQL y gestión de bases de datos MariaDB
- Importar datos de CSV y SQL
- Exporte datos a varios formatos: CSV , SQL , XML , PDF, etc.
- Administrar múltiples servidores
- Crear gráficos PDF del diseño de la base de datos

2.2.2.7 MYSQL

MySQL es un sistema de administración de bases de datos de código abierto, el cual puede ser ejecutado en prácticamente todas las plataformas, incluidas LINUX, INUX y Windows. Puede ser utilizando en una gran cantidad de aplicaciones, pero el usa más frecuente para este sistema es el desarrollo de aplicaciones web y publicaciones en línea.

Sus principales ventajas son su bajo consumo de memoria física para poder ser ejecutado en máquinas con bajo rendimiento, su facilidad de configuración e instalación y su velocidad al realizar las operaciones necesarias.

Por seguridad, MySQL utiliza un sistema de privilegios de acceso y contraseña cifrada que permite la verificación basada en el host. Los clientes MySQL pueden conectarse a MySQL Server utilizando varios protocolos, incluidos los sockets TCP / IP en cualquier plataforma. MySQL también es compatible con varios programas de clientes y utilidades, programas de línea de comandos y herramientas de administración como MySQL Workbench.

2.2.2.8 HIGHCHARTS

HighCharts es una biblioteca de gráficos escrita netamente en JavaScript la cual permite implementar una gran cantidad de gráficos interactivos a sitios web o aplicaciones web. Funciona en todos los navegadores web, ya sea en computadores o en móviles, ya que presenta un soporte multitáctil que proporciona una muy buena experiencia al usuario. Es gratis y de código abierto, por lo cual es posible editar los gráficos en su 100%

2.2.2.9 INTERFAZ DE USUARIO

Se define como interfaz de usuario o UI por las iniciales de “User Interface” o GUI por “Graphical User Interface”, a todo el conjunto de dispositivos que permiten interacción entre la aplicación o proceso y el usuario o cliente. Dentro de esto se encuentra todo lo que sean botones, gráficos, imágenes, interruptores, etc.

Para crear una interfaz de usuario primero es tener diseñada y programada la aplicación y finalmente programar cual serán los eventos que activarán los botones o demás opciones.

2.2.2.10 APLICACIONES NATIVAS

Las aplicaciones nativas corresponden a las que son desarrolladas para un sistema o sistema operativo determinado, aprovechando así el hardware en un 100% haciendo que la experiencia con el usuario sea bastante positiva,

comparado con otras aplicaciones que no sean nativas.

Debido a que cada plataforma posee un sistema diferente, por ejemplo, Android e IOS, la misma aplicación se tendría que desarrollar, pero en dos lenguajes totalmente diferentes.

Ventajas	Desventajas
<ul style="list-style-type: none"> - Máximo rendimiento en dispositivos - Mejor experiencia con el usuario - App disponibles en tienda del sistema - Actualizaciones constantes 	<ul style="list-style-type: none"> - Alto costo de desarrollo - Tiempo de desarrollos prolongados - Uso de diferentes lenguajes de programación.

Tabla 5 Ventajas y Desventajas de aplicaciones nativas.

2.2.2.11 APLICACIONES WEB

Una aplicación Web utiliza lenguajes bastante comunes como lo son PHP, HTML, JavaScript entre otros. La principal característica de una aplicación web es que se puede programar sin restricciones según el sistema operativo en el que se utilizara, debido a que está enfocado a ser utilizada en un navegador, por lo tanto, tampoco necesita ser instalada.

Ventajas	Desventajas
<ul style="list-style-type: none"> - Compatibilidad multiplataforma - Menos requerimiento de memoria - Múltiples usuarios concurrentes 	<ul style="list-style-type: none"> - No disponible en tiendas del sistema - Diseño visual no acorde al SO - Reducido uso de hardware.

2.2.2.12 APLICACIÓN HIBRIDA

Corresponde a la combinación de ambas aplicaciones mencionadas anteriormente, diseñadas en un lenguaje de programación web, ya sea HTML, CSS o JavaScript para obtener una vista web dentro de un contenedor de una aplicación nativa, permitiendo el acceso de API's dentro del sistema. Otra gran característica es que este tipo de aplicaciones si pueden ser encontradas en tiendas del sistema.

2.2.3 ESTANDARES ASOCIADOS AL PROYECTO

2.2.3.1 NCH 409

Para que un agua sea considerada como agua apta para el consumo humano, debe cumplir una serie de requerimientos dentro de los cuales se consideran aspectos bacteriológicos como físico y químicos.

Estos requerimientos se encuentran definidos en la norma Chile NCH 409:

- Define requisitos
- Exige concentraciones menores al máximo para contaminantes químicos
- Exige contenido mínimo para desinfectante activo residual
- Exige ausencia de contaminación por bacterias fecales

2.2.4 FILOSOFIA DE CONTROL.

2.2.4.1 CONTROL DIFUSO

La lógica difusa es una generalización de la lógica estándar, en la que un concepto puede poseer un grado de verdad en cualquier lugar entre 0 y 1.

La lógica estándar se aplica solo a conceptos que son completamente verdaderos (que tienen un grado de verdad 1) o completamente falsos (que tienen un grado de verdad 0). Se supone que la lógica difusa se usa para razonar

sobre conceptos inherentemente vagos, como la "altura". Por ejemplo, se podría decir que una persona X es alta, con un grado de verdad de 0,85.

El término “difuso” significa cosas que no son muy claras o vagas. Normalmente se pueden encontrar situaciones en la que no se puede decidir si la afirmación es verdadera o falsa. En ese momento, la lógica difusa ofrece una flexibilidad muy valiosa para el razonamiento. Como también considerar las incertidumbres de la situación.

Resulta que las aplicaciones útiles de la lógica difusa no están en la inteligencia artificial de alto nivel sino en el control de máquinas de nivel inferior, especialmente en productos de consumo. Por lo general, los controladores difusos se implementan como software que se ejecuta en microprocesadores estándar. Se han construido microprocesadores de propósito que realizan operaciones difusas directamente en hardware, pero incluso estos usan señales binarias digitales (0 o 1) en el nivel de hardware más bajo. Hay algunos prototipos de investigación de chips de computadora que usan señales analógicas en el nivel más bajo, pero estos los chips simulan el funcionamiento de las neuronas en lugar de la lógica difusa.

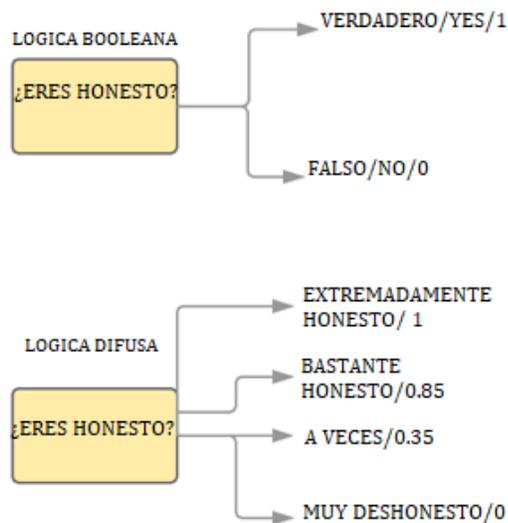


Figura 22 Ejemplo lógica difusa

La arquitectura de la lógica difusa se conforma de cuatro partes principales:

Base de la regla: Contiene todas las reglas y las condiciones si-entonces ofrecidas por los expertos para controlar el sistema de toma de decisiones.

Fuzzificación: El paso de Fuzzificación ayuda a interpretar las entradas, permite convertir números nítidos en conjuntos difusos. Entradas nítidas medidas por sensores y pasadas al sistema de control para su posterior procesamiento. Como temperatura ambiente, presión, etc.

Máquina de inferencia: Le ayuda a determinar el grado de coincidencia entre la entrada difusa y las reglas. Según el porcentaje de coincidencia, determina qué reglas necesitan implicación de acuerdo con el campo de entrada dado. Después de esto, las reglas aplicadas se combinan para desarrollar las acciones de control.

Defuzzificación: Finalmente, se realiza el proceso de defuzzificación para convertir los conjuntos difusos en un valor nítido. Hay muchos tipos de técnicas disponibles, por lo que debe seleccionar la que mejor se adapte cuando se utiliza con un sistema experto.

2.2.4.1.1 VENTAJAS

A continuación, ventajas sobre la lógica difusa:

- **Más barato :** el desarrollo de un control de lógica difusa es comparativamente más barato que el desarrollo de un controlador basado en modelos u otro controlador en términos de rendimiento.
- **Robusto :** El control de lógica difusa es más robustos que los controladores PID debido a su capacidad para cubrir una amplia gama de condiciones de funcionamiento.

- Emule el pensamiento deductivo humano : Básicamente, el control de lógica difusa está diseñado para emular el pensamiento deductivo humano, el proceso que las personas usan para inferir conclusiones de lo que saben.
- Fiabilidad : el control de lógica difusa es más confiable que el sistema de control convencional.
- Eficiencia : la lógica difusa proporciona más eficiencia cuando se aplica en el sistema de control.

2.2.4.1.2 DESVENTAJAS

A continuación, desventajas de la lógica difusa:

- Requiere muchos datos : El control de lógica difusa necesita muchos datos para aplicar.
- Útil en caso de datos históricos moderados : el control de lógica difusa no es útil para programas mucho más pequeños o más grandes que los datos históricos.
- Necesita una gran experiencia humana : este es un inconveniente, ya que la precisión del sistema depende del conocimiento y la experiencia de los seres humanos.
- Necesita actualización periódica de las reglas : las reglas deben actualizarse con el tiempo.

3 DESARROLLO DE LOS COMPONENTES

3.1 MODELAMIENTO DE LA SOLUCION

Se desarrollará un sistema de control de software y hardware para el control de la calidad del agua, junto con una interfaz gráfica para el control o monitoreo ya sea local o remotamente. Esto se logrará implementando un sistema de comunicación que corresponde a un Gateway IoT, gracias a la Raspberry Pi 2 Model B la cual permitirá el control del sistema, el acceso a los datos de los distintos sensores recopilados por la placa NodeMCU Esp32 los cuales serán almacenados localmente o en una nube en caso de disponer de internet, todo gracias a las tecnologías que otorgan la capacidad de interactuar remotamente entre un usuario a través de las interfaces multiplataformas y la comunicación Machine to Machine, a través de protocolos como Websocket y MQTT.

Este sistema podrá ser monitoreado a través de una interfaz en tiempo real y desde múltiples plataformas, diseñado en una página web, mediante el vínculo entre PHP y MySQL.

Para realizar esta solución, se realizará un prototipo funcional el cual estará basado en el uso de una Raspberry Pi 2 Model B con un adaptador Wifi.

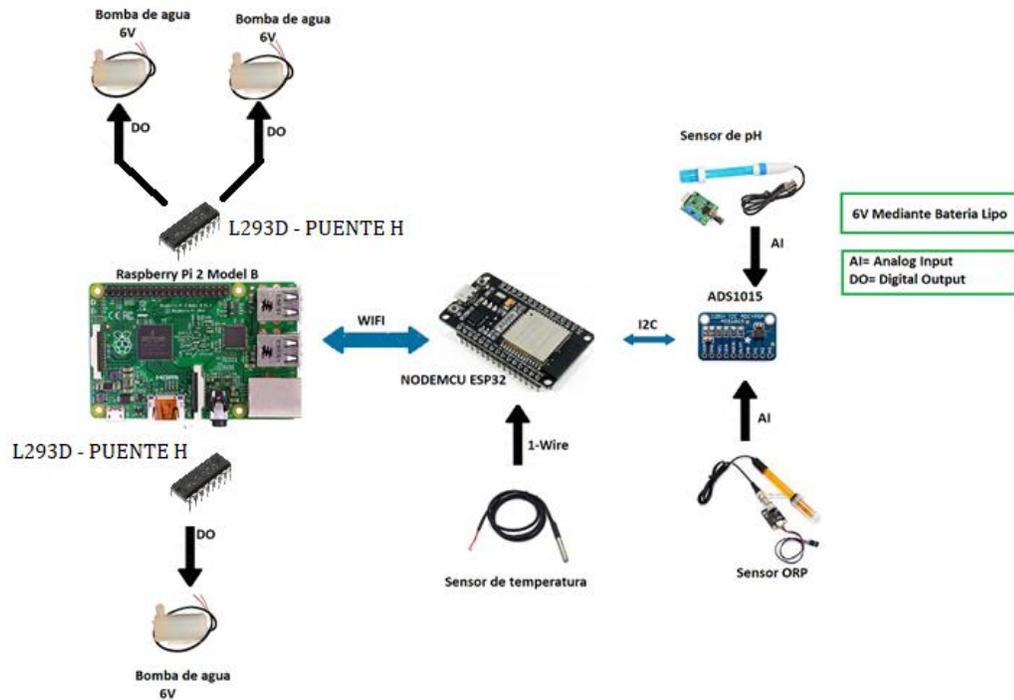


Figura 23 Componentes y Conexiones

El consumo del prototipo será de 5V para las placas embebidas y sensores, y 6V para los actuadores, los cuales serán suministrados mediante un Puente H.

3.2 DISEÑO DEL SISTEMA

El sistema por tema de facilidad de uso y rapidez será modular, es decir estará compuesto por varias partes las cuales tendrán entre si una conexión, en donde una de esas partes serán el Gateway IoT, encargado de realizar la conexión con la nube, los sensores y actuadores. Las partes ya mencionadas son elementos principales dentro de la arquitectura del internet de las cosas. Estos estarán conectados por medio de protocolos de comunicación como lo son Websocket y MQTT, además del protocolo de HTTP para realizar la solicitud de datos o para realizar acciones.

3.3 DIAGRAMA DE ELEMENTOS

Para la comunicación del dispositivo se utilizará una red local Wifi, la cual será

generada por la placa NodeMCU, además la Raspberry estará conectada mediante cable de Red a internet. Para realizar la conexión remota o fuera del área local, se utilizará un túnel DDNS llamado NGROK, el cual permite una fácil configuración, la apertura de puertos correspondientes, además de implementar seguridad a la conexión.

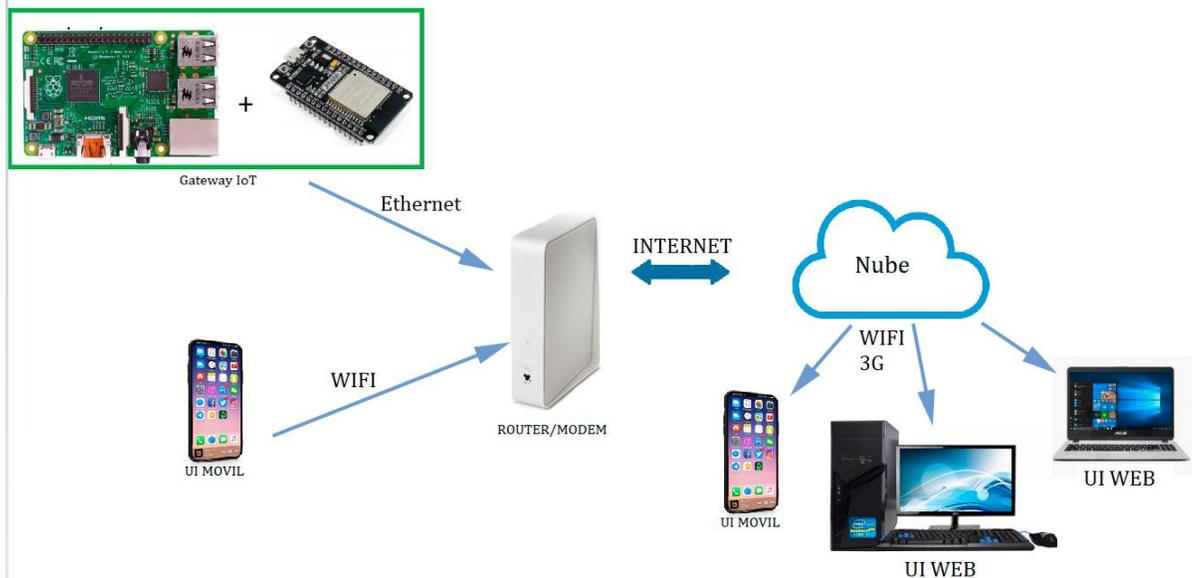


Figura 24 Elementos arquitectura del sistema

3.4 ELEMENTOS PRINCIPALES Y FUNCIONES

El elemento principal en el sistema será la Raspberry Pi, la encargada de realizar el control automático del proceso, también cumple la función de Gateway IoT permitiendo la comunicación entre los sensores y la nube. Además, esta placa tendrá la función de crear una página web, la cual será la interfaz gráfica multiplataforma, junto con su respectiva base de datos MySQL.

La placa NodeMCU será la responsable de la obtención de los datos de los sensores y el envío de estos hacia la Raspberry mediante MQTT.

Los sensores, que en este caso corresponden a la medición de pH , ORP y temperatura, obtendrán los datos para el posterior análisis y toma de decisiones del sistema.

El usuario en este sistema será una persona que pueda solicitar la interfaz o página, en donde se desplegaran una serie de gráficos con los datos del sistema, con el fin de supervisar el correcto funcionamiento y el estado óptimo del agua.

La interfaz gráfica, corresponde a una página web HTML5, la cual será alojada en la Raspberry. Esta interfaz será interactiva y puede mostrar datos o funciones disponibles para realizar, así como solicitar datos de un determinado día, la configuración de la cantidad de agua a mantener o la manipulación de los actuadores.

3.5 APLICACIÓN ESPECIFICA

El prototipo puede funcionar con sensores de laboratorio y actuadores de bajo consumo, pero perfectamente puede escalar y agregar mejor hardware. Tiene la capacidad de adquisición y almacenamiento de datos con la posibilidad de observar estos datos localmente como también desde cualquier otra parte.

Este prototipo comenzara con una cantidad baja de agua, para obtener respuestas rápidas, debido a que el cambio de pH mediante reductor o incrementador de pH tarda más cuando se trata de masivas cantidades de agua. Se realizarán funciones de medición de ORP y pH como también una estimación del cloro. También tendrá actuadores para proveer de un control de pH y la posibilidad de suministrar cloro al agua con el fin de mantener el agua en óptimas condiciones.

Con todos los datos obtenidos se podrá realizar un análisis del comportamiento de la calidad del agua, logrando comprender los grandes factores que alteran el agua, como, asimismo poder efectuar un control automático para reducir los efectos de estos.

Para hacer uso de los datos entregados por los sensores, estos serán almacenados localmente en una base de datos, a los cuales se podrán acceder en todo momento desde la página HTML.

3.5.1 APLICACIONES GENERALES

El desarrollo de este prototipo puede ser destinado al área de la desinfección del agua potable como también de otras concentraciones de agua como lo son las piscinas, en donde se necesitarán recursos menos precisos, debido a que el rango optimo es mucho mayor en ese caso.

3.6 CONFIGURACION RASPBERRY

Una vez con el sistema operativo ya cargado en la raspberry mediante su tarjeta microSD, lo principal para trabajar cómodamente es realizar la comunicación mediante el protocolo de administración remota, también llamado SSH el cual utiliza la dirección IP del dispositivo para conectarse a él, por lo tanto, lo recomendable inicialmente es asignarle una IP estática a la placa.

3.6.1 IP ESTATICA

Para lograr conectarse a la Raspberry Pi, será necesario configurar su IP, ya que, si posee una IP dinámica, esta ira variando cada vez que se produzca un reinicio del dispositivo. Por lo tanto, la mejor alternativa es mantener un IP fija o estática, a la cual se podrá acceder siempre que se requiera.

Para poder configurar la IP estática se deben seguir los siguientes pasos:

Se debe ingresar a la consola de la Raspberry directamente, es decir, se le debe incorporar un teclado y un monitor e introducir el siguiente comando.

```
sudo nano /etc/dhcpd.conf
```

Dentro del fichero se encuentra una serie de líneas comentadas, las cuales son ejemplos de configuraciones de red que se pueden aplicar, y dentro de estas se

encuentra un ejemplo de IP estática. Para crear nuestra propia IP estática se debe ingresar las siguientes líneas:

```
interface wlan0
static ip_address=192.168.1.20/24
static routers=192.168.0.1
static domain_name_servers=192.168.0.1
```

En donde:

interface = Nombre de la interfaz que queremos configurar, en este caso wlan0, ya que la conexión es mediante WIFI

static ip_address = Dirección fija que se desea. (dejar el /24 al final)

static routers = Dirección del Gateway. (Ip del router)

static domain_name_servers = Dirección del servidor DND (normalmente la del router).

Finalmente se debe comprobar que efectivamente la configuración se estableció correctamente, mediante el comando

Ifconfig

```
pi@raspberrypi: ~  
Archivo Editar Pestañas Ayuda  
pi@raspberrypi:~$ ifconfig  
eth0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500  
ether b8:27:eb:a0:5a:6c txqueuelen 1000 (Ethernet)  
RX packets 0 bytes 0 (0.0 B)  
RX errors 0 dropped 0 overruns 0 frame 0  
TX packets 0 bytes 0 (0.0 B)  
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
inet 127.0.0.1 netmask 255.0.0.0  
inet6 ::1 prefixlen 128 scopeid 0x10<host>  
loop txqueuelen 1000 (Local Loopback)  
RX packets 17 bytes 1004 (1004.0 B)  
RX errors 0 dropped 0 overruns 0 frame 0  
TX packets 17 bytes 1004 (1004.0 B)  
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
vlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
inet 192.168.0.20 netmask 255.255.255.0 broadcast 192.168.0.255  
inet6 fe80::9ab1:b633:3d00:dd16 prefixlen 64 scopeid 0x20<link>  
ether 40:a5:ef:d2:f4:7b txqueuelen 1000 (Ethernet)  
RX packets 1093 bytes 94546 (92.3 KiB)  
RX errors 0 dropped 0 overruns 0 frame 0  
TX packets 1153 bytes 389491 (389.3 KiB)
```

Figura 25 IP Estática

3.6.2 CONEXIÓN SSH

SSH (Secure Shell) es un protocolo de comunicación que utiliza como puerto estándar el 22, el cual permite la conexión remota al dispositivo, es decir, un dispositivo que se encuentre en la misma red se podrá conectar a otro, siempre y cuando exista un tipo de acuerdo para poder acceder.

Con este protocolo dentro del proyecto se podrá configurar la Raspberry de forma segura desde un computador remoto, sin la necesidad de contar con un teclado y un monitor exclusivo para la placa.

Para poder activar esta función dentro de la Raspberry Pi se debe ingresar a la configuración de esta, desde el escritorio que incorpora el sistema operativo Raspbian. Es muy sencillo, solo se ingresa en configuración de Raspberry Pi, en la sección de interfaces, en donde se encuentran varias opciones de comunicación.

Se recomienda activar SSH y VNC y dejar las demás que vienen por defecto.

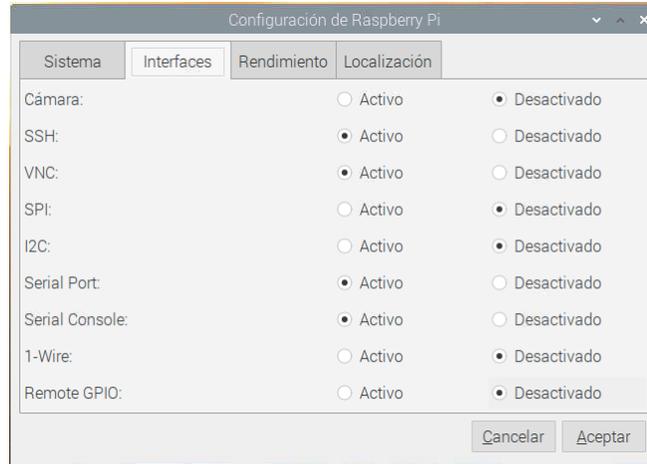


Figura 26 Configuración de Raspberry Pi

3.6.3 ACCESO A RASPBERRY DESDE WINDOWS

Para poder ingresar desde Windows u otro sistema operativo es necesario utilizar un cliente SSH, dentro de los cuales se encuentra Putty para Windows.

Putty es un software fácil de utilizar el cual se puede comportar como un cliente SSH, entre otros tipos de protocolo. Al iniciar Putty se abrirá una ventana de configuración en donde se deben ingresar los parámetros para poder realizar la conexión.

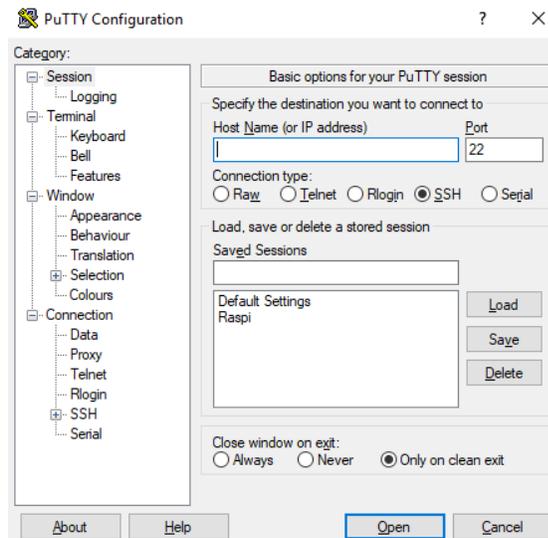
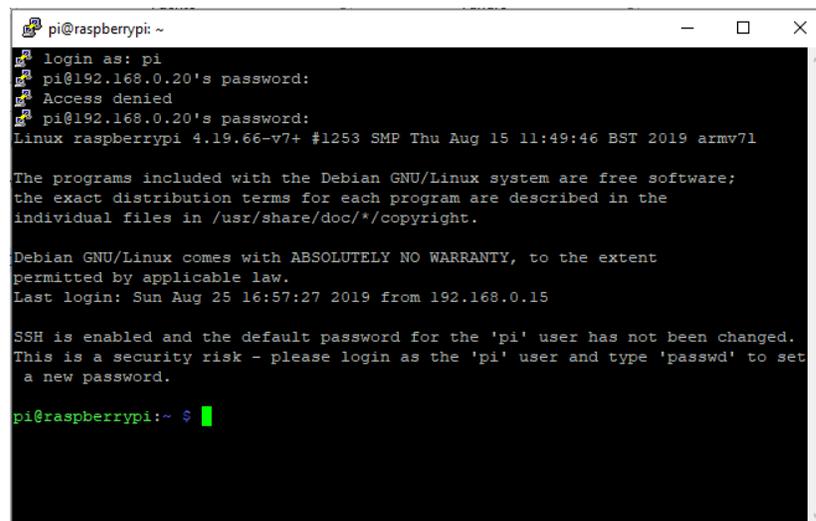


Figura 27 Configuración Putty.

Solo se debe ingresar la dirección IP del dispositivo a conectar y establecer el tipo de conexión, en este caso al presionar en SSH el puerto se asigna automáticamente y luego presionar en “Open”. Si funciona correctamente se debe abrir una consola solicitando contraseña de un usuario “pi” , en caso de no se haya modificado el nombre de la Raspberry Pi. Una vez ingresada la contraseña, se da como iniciada la sesión, dando acceso a la consola de Raspberry Pi.



```
pi@raspberrypi: ~  
login as: pi  
pi@192.168.0.20's password:  
Access denied  
pi@192.168.0.20's password:  
Linux raspberrypi 4.19.66-v7+ #1253 SMP Thu Aug 15 11:49:46 BST 2019 armv7l  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Sun Aug 25 16:57:27 2019 from 192.168.0.15  
  
SSH is enabled and the default password for the 'pi' user has not been changed.  
This is a security risk - please login as the 'pi' user and type 'passwd' to set  
a new password.  
pi@raspberrypi:~ $
```

Figura 28 Consola con sesión iniciada en Putty.

3.6.4 VNC VIEWER

VNC por sus siglas en inglés “Virtual Network Computing” es un sistema gráfico que permite compartir el escritorio de una computadora o sistema a otra computadora mediante el protocolo RFB. Permite que una computadora controle otra, copiando completamente las acciones realizadas por el teclado y ratón, y reproduciéndola en la otra, además retransmitiendo actualizaciones de pantalla gráfica mediante una red, para observar que se realiza en la otra computadora. Previamente asignamos la interfaz VNC como activa en la Raspberry Pi, por lo tanto, esta se comportará como un servidor VNC dando la posibilidad de controlar y programar la Raspberry desde el computador, además de la opción de enviar y recibir archivos a la Raspberry Pi facilitando el desarrollo.

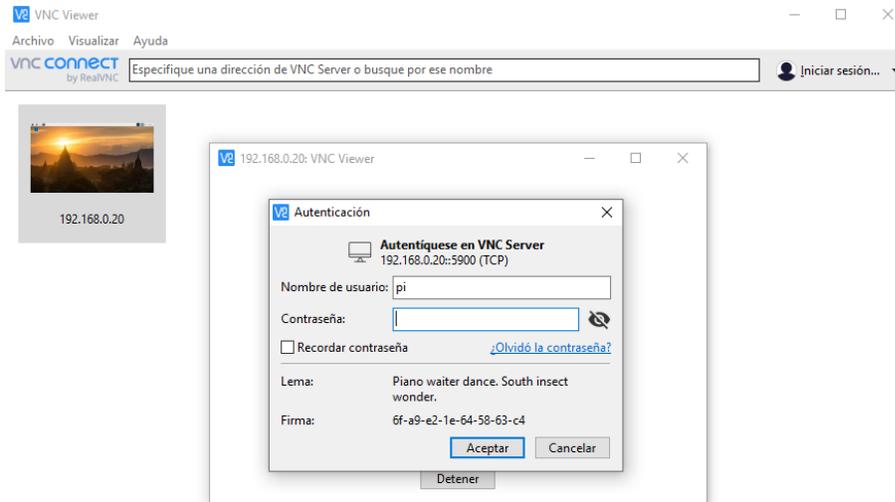


Figura 29 VNC Viewer

Al igual que con Putty, para acceder se deben ingresar ciertos parámetros para establecer la conexión, en este caso son los mismo ingresados anteriormente para la conexión SSH.

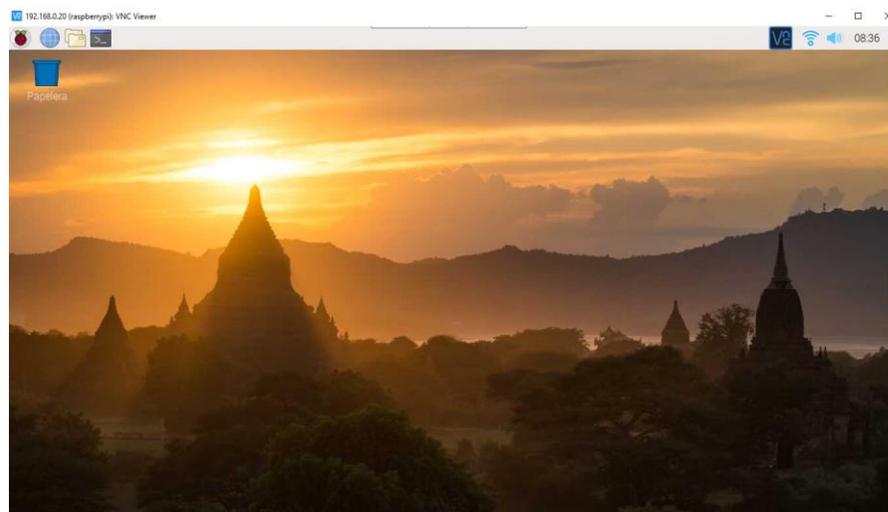


Figura 30 Escritorio Raspberry mediante VNC

Inmediatamente después de ingresar los parámetros, se abrirá una ventana la cual presenta el escritorio de Raspberry Pi, permitiendo ya el control de está remotamente.

3.6.5 INFRAESTRUCTURA LAMP

La infraestructura LAMP, es un conjunto de herramientas utilizadas para el desarrollo de un sistema de infraestructura en internet. LAMP es el acrónimo utilizado debido a que utiliza las siguientes herramientas:

L: Linux, el sistema operativo a utilizar.

A: Apache, el servidor web.

M: MySQL / MariaDB, gestor de bases de datos.

P: Perl, PHP o Python, el lenguaje de programación a utilizar.



Figura 31 Infraestructura LAMP

Este será utilizado para montar en la placa raspberry un servidor web, junto con una base de datos MySQL para guardar localmente los datos obtenidos por los sensores.

La página web diseñada será la interfaz gráfica del prototipo, por lo tanto, este tipo de arquitectura aporta una gran cantidad de herramientas que permitirán el desarrollo de esta.

Para instalar la infraestructura LAMP en la placa Raspberry, hay que ingresar los siguientes comandos en una terminal o consola del sistema.

Como de costumbre antes de realizar cualquier proceso de instalación, es recomendable buscar actualización de la Raspberry o del sistema operativo, para ello se ejecuta la siguiente orden:

```
$sudo rpi-update
$sudo apt update
$sudo apt upgrade
```

Luego lo primero será instalar el servidor Web Apache2 , utilizando el siguiente comando:

```
$sudo apt install apache2
```

Ya finalizada la instalación, se puede comprobar que haya sido instalada correctamente ingresando en un navegador la dirección IP de la raspberry. Si todo funciona según lo esperado, se podrá ver una página web como la que se muestra a continuación.

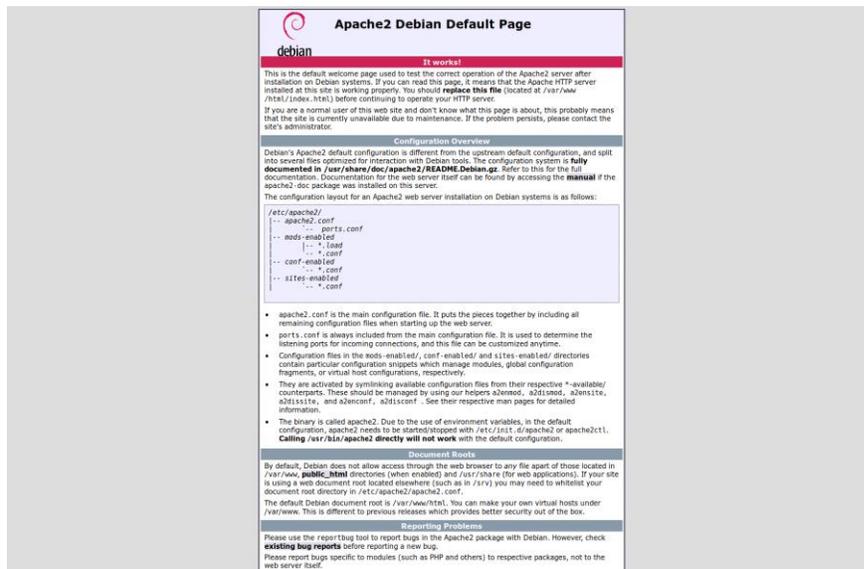


Figura 32 Pagina web en Raspberry

Antes de continuar, para facilitar el trabajo y el correcto funcionamiento del servidor Webs, se realizarán cambios al propietario del directorio HTML y de los archivos, como también sus permisos con el siguiente comando:

```
$cd /var/www/
$sudo chown -R www-data:www-data html
$sudo find html -type d -print -exec chmod 775 {} \;
```

```
$sudo find html -type f -print -exec chmod 664 {} \;
```

Y finalmente se debe añadir el usuario de la raspberry, en este caso "Pi" , al grupo www-data para que pueda operar sin programas.

```
$usermod -a -G www-data pi
```

Ahora la placa raspberry puede crear sitios Web en HTML utilizando CSS y JavaScript internamente, sin embargo, es posible que se requiera la interacción entre el usuario y el sitio, como por ejemplo necesitar que el usuario se registre.

Para instalar PHP nuevamente se utilizará el terminal o consola para ingresar el siguiente comando:

```
$sudo apt install php php-mbstring
```

Y para comprobar la correcta instalación de PHP se realiza un proceso similar a la verificación del servidor web.

Primero se debe eliminar el archivo "Index.html" en el directorio /var/www/html y luego crear un archivo "index.php" en este mismo directorio.

```
$sudo rm /var/www/html/index.html  
$echo "<? php phpinfo ();?>"> /var/www/html/index.php
```

A partir desde aquí, la comprobación es la misma que con el servidor web, ingresar a un navegador web e ingresar la dirección IP. Si funciona correctamente, se deberá tener un resultado similar a la siguiente figura.



System	Linux ajaniserveur 3.2.0-4-amd64 #1 SMP Debian 3.2.54-2 x86_64
Build Date	Dec 12 2013 08:42:50
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/apache2
Loaded Configuration File	/etc/php5/apache2/php.ini
Scan this dir for additional .ini files	/etc/php5/apache2/conf.d
Additional .ini files parsed	/etc/php5/apache2/conf.d/10-pdo.ini, /etc/php5/apache2/conf.d/20-gd.ini, /etc/php5/apache2/conf.d/20-mysql.ini, /etc/php5/apache2/conf.d/20-mysqli.ini, /etc/php5/apache2/conf.d/20-pdo_mysql.ini
PHP API	20100412
PHP Extension	20100525
Zend Extension	220100525
Zend Extension Build	API220100525 NTS

Figura 33 PHP en Raspberry

A este punto, ya se poseen las herramientas necesarias para realizar una pagina web estable y con buena apariencia con la capacidad de interactuar con un usuario, pero ahora lo mas probable que se desee hacer con PHP será almacenar información para usar en el sitio. Para este propósito las bases de datos se usan con mayor frecuencia, por lo tanto, se instalará una base de datos MySQL.

Para hacer esto se instalará mariadb-server y php-mysql , para realiza la conexión entre PHP y MySQL con el siguiente comando:

```
$sudo apt install mariadb-server php-mysql
```

Para verificar que funciona correctamente, simplemente se debe comprobar la conexión la siguiente línea de comando:

```
$sudo mysql --user = root
```

Para crear otro usuario en la base de datos se debe ingresar lo siguiente, una vez ingresado la línea anterior.

```
DROP USER 'root'@'localhost';  
CREATE USER 'usuario'@'localhost' IDENTIFIED BY 'password';  
GRANT ALL PRIVILEGES ON *.* TO 'usuario'@'localhost' WITH GRANT OPTION;
```

Entonces ya se tendrá un servidor web conectado con PHP y MySQL funcional, un componente extra que no es obligación instalar es PHPMyAdmin que incorporara una interfaz amistosa para el uso y configuración de la base de datos.

Dispone de una fácil y rápida instalación:

```
$sudo apt install phpmyadmin  
$sudo phpenmod mysqli  
$sudo /etc/init.d/apache2 restart
```

Finalmente, para verificar y usar PHPMyAdmin, se debe acceder desde un navegador ingresando la dirección IP de la raspberry mas el directorio específico de la siguiente manera: 192.168.x.x/PHPMyAdmin y cargara una pagina similar a la de la siguiente figura.



Figura 34 PHPMyAdmin en Raspberry

3.6.6 DNS DINAMICA

Una DNS dinámica corresponde al nombre de dominio estático, el cual estará fijado a una dirección IP pública dinámica, con dinámica se refiere a que se va modificando cada cierto periodo de tiempo. Gracias a este servicio será posible el control, monitoreo o acceso a nuestra placa raspberry desde fuera de la red de área local.

Para obtener este servicio se utilizará NO-IP, un proveedor de DNS dinámicas para servicios de pago y gratuitos. Para ello primero se debe registrar en la página

web “<https://www.noip.com/>” para posteriormente instalar el servicio en la raspberry haciendo uso de los siguientes comandos:

```
$mkdir no-ip
$cd no-ip
$wget http://www.no-ip.com/client/linux/noip-duc-linux.tar.gz
$tar -zxvf noip-duc-linux.tar.gz
$cd noip-2.1.9-1/
$make
$sudo make install
```

Durante el proceso de instalación requerirá ingresar la cuenta que se ha registrado en la pagina web de NO-IP para poder proseguir con la instalación.

Ya finalizada la instalación se debe crear un archivo que autorizara el arranque automático de No-IP al momento de encender la Raspberry.

```
$sudo nano /etc/init.d/noip2
```

Y dentro del archivo ingresar la siguiente línea:

```
sudo /usr/local/bin/noip2
```

Luego de esto es necesario agregarlo a la cola de arranque de las Raspberry.

```
sudo chmod +x /etc/init.d/noip2
sudo update-rc.d noip2 defaults
```

Ya terminada con la instalación en la Raspberry, se debe realizar una configuración en el router, es necesario configurar la DDNS, ingresando los puertos que se utilizaran, en este caso por ser un servicio Web se utilizaran los puertos por defecto que es el 80. Los pasos por seguir para la configuración del puerto varían según el router que se utilice.

Local		Exterior			
Dirección IP	Rango de puertos	Rango de puertos	Protocolo	Habilitado	Borrar
0.0.0.0	0-0	-0	UDP	<input type="checkbox"/>	<input type="checkbox"/>
192.168.0.22	80-80	80-80	TCP	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Figura 35 Puertos en Router

3.7 CARTA GANTT

A continuación, la carta Gantt de las actividades realizadas a lo largo del año, para el desarrollo del proyecto.

grandes, no era factible realizar debido a que hubiese quedado demasiado pesado, lo que produciría que se hundiera, ingresando agua a la placa de los sensores y a la misma NodeMCU haciendo que estos se estropearan.

3.8.2 COSTOS DEL SOFTWARE

La siguiente tabla corresponde a un estimado de horas que podría llevar a cabo el desarrollo de una interfaz para la interacción con el cliente y un control automático para el proyecto.

ELEMENTOS	HORAS HOMBRE
DESARROLLO PROGRAMACION	30
DESARROLLO WEB	15
CREACION BASE DE DATOS	10
PROTOTIPO IMPRESIÓN 3D	10
INTERFAZ GRAFICA	15
HORAS TOTALES	80

Figura 38 Costos de Software, medidos en hora/hombre

4 DESARROLLO Y FLUJOS DE PROGRAMA

4.1 PUNTO DE CONEXION

Para establecer una comunicación directa entre la NodeMCU ESP32 y la Raspberry, sin uso de internet se configurará la NodeMCU como router, es decir emitirá una señal Wifi, a la cual se conectará la Raspberry, por este medio también se llevará a cabo la comunicación MQTT.

La Raspberry dependiendo de la situación en que se encuentre podrá estar conectada mediante un cable de red a internet, permitiendo que esta se comporte como un Gateway IoT, dando acceso a múltiples funciones, como guardar los datos en la nube.

Por otro lado, si la Raspberry no se encuentra conectada a internet, el sistema podrá ser monitoreado y controlado desde cualquier tipo de dispositivo con la capacidad de acceder a la red Wifi y así acceder a la página web local, ingresando la IP de la raspberry en un navegador.

4.1.1 DESARROLLO CODIGO WIFI

Para crear la Red Wifi, se utilizó el IDE de Arduino, ya que es compatible con la placa NodeMCU esp32, y también posee una amplia cantidad de librerías que facilitan la configuración de la red wifi como también así la de sensores, entre otros.

El código utilizado es el siguiente, en donde se importa la librería, se configura el nombre de la red, la contraseña y se crea el punto de acceso.

```
#include Wifi.h
const char* ssid = "PUNTO DE ACCESO ESP32";
const char* password = "eduardo123";
WiFiClient espClient;
WiFi.softAP(ssid, password); //Crea punto de acceso
delay(500);
Serial.print("Dirección IP:"); //Muestra dirección IP
```

```
Serial.println(WiFi.softAPIP()); //DIRECCION IP
```

4.1.2 FUNCIONES GATEWAY IOT

La función que le corresponde a la placa Raspberry y que son necesarias para el manejo de diferentes funciones que el Gateway IoT o servidor IoT debe poseer son las siguientes:

- Montar un servidor para el desarrollo de páginas web y la capacidad de solicitar datos, por lo tanto, es necesario un servidor HTTP.
- El servidor será capaz de entregar la información correspondiente solicitada por el cliente.
- Se podrá comunicar con los dispositivos mediante protocolo MQTT, actuando la raspberry como Broker.
- Disponer de una base de datos para guardar localmente todos los datos recibidos de los sensores, como también contar con una base de datos de seguridad o en la nube.
- Sera necesario monitorear cada un periodo de tiempo las variables correspondientes, enviar los datos a la base de datos y analizarlos para el posterior control de la variable manipulable.

4.2 VARIABLES GLOBALES

A la hora de desarrollar el control del sistema se debe tener claridad sobre los valores a los cuales se aspira a llegar, para que la calidad sea optima, como también indicar los valores cuando sea mala. Estos valores no están definidos como solo un número, sino como un rango de valores en donde es aceptable.

PH= [6.5 a 8.3]

Dentro de esos valores se considera que el pH del agua es el indicado para poder ser utilizada en contacto directo con personas, plantas, animales, etc. Además, también es el rango perfecto para que el cloro pueda ser 100% eficiente a la hora de desinfectar.

La variable temperatura puede tomar cualquier valor, ya que no es una variable que se vaya a controlar, sino que beneficia la obtención de las demás variables, pH y Redox, debido a que estas varían con la temperatura del líquido

La variable Redox idealmente se debe encontrar entre los valores 650 y 700 mV (milivolts) , rango en que las bacterias comienzan a desaparecer, este valor depende mucho del valor de PH. Por lo tanto, es función del valor medido, se ajusta el nivel de cloro gracias a las bombas de agua para modificar esta variable.

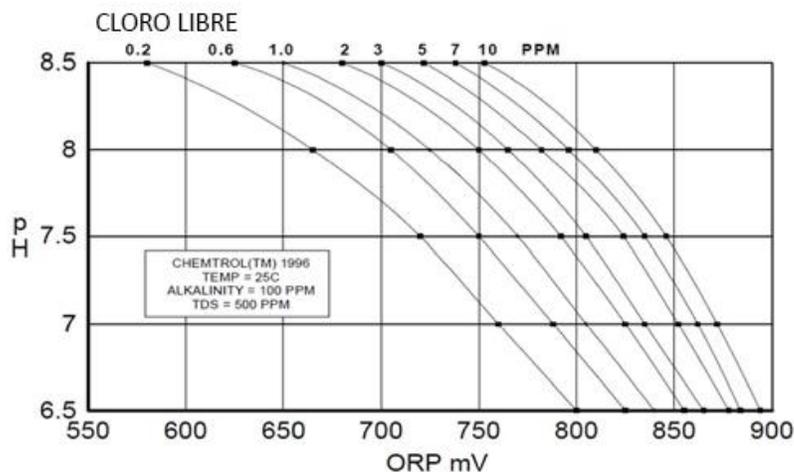


Figura 39 Correlación entre pH, Redox y Cloro Libre

4.3 COMUNICACIÓN MQTT

Para comenzar una comunicación mediante el protocolo MQTT es necesario establecer un Broker, en este caso, se manipulará una Raspberry por lo tanto se debe proceder a realizar la instalación de este servicio. El servidor Eclipse

Mosquito se instala de la siguiente forma:

Primero que nada, la Raspberry debe poseer conexión a internet para obtener los archivos necesarios. Ya conectada se procede a abrir un terminal o consola e ingresar el siguiente comando:

```
$sudo wget http://repo.mosquitto.org/debian/mosquitto-repo.gpg.key
```

Luego, se debe añadir una clave para autorizar un paquete que se descargara más adelante.

```
$sudo apt-key add mosquitto-repo.gpg.key
```

Ya añadida la clave, se debe dirigir a una carpeta para descargar allí el repositorio de Mosquitto.

```
$cd /etc/apt/sources.list.d/  
sudo wget http://repo.mosquitto.org/debian/mosquitto-stretch.list
```

Finalmente se verifica si existe alguna actualización de paquete y se procede a instalar

```
$Sudo -i  
$apt-get update  
$apt-get install mosquitto
```

Ya con esos pasos estará preparado el servidor Broker Mosquitto Instalado en la Raspberry PI. Ahora se instalará el cliente para realizar pruebas.

```
apt-get install mosquitto-clients
```

Para probar que el servidor Broker esta correctamente instalado, se deben utilizar dos comandos principales que utilizara el cliente, estos son : mosquitto_pub y

mosquitto_sub.

El comando mosquitto_sub le permite al cliente suscribirse a un tópico, en el cual podrá “escuchar” y mostrar todos los datos que estén siendo enviados a ese tópico.

```
$mosquitto_sub -h BROKER -t TOPIC
```

Donde:

- -h: indica que lo que viene después es el host. El host se refiere a la dirección IP o nombre de la máquina en la red del Broker Mosquitto. Si no se indica este parámetro se tomará como localhost
- BROKER: dirección IP o nombre de la máquina en la red del Broker Mosquitto.
- -t: indica que lo que viene después es el tópico al que se desea suscribir.
- TOPIC: nombre del tópico a suscribir.

Y el comando mosquitto_pub permite al cliente publicar mensajes simples en el tópico especificado.

```
$mosquitto_pub -h BROKER -t TOPIC -m MENSAJE
```

Donde:

- -m: indica que lo que viene después es el mensaje.
- MENSAJE: el mensaje que se desea enviar. Tiene que ir entre comillas dobles. Por ejemplo “Temperatura = 25°C”.

```
pi@raspberrypi: ~  
login as: pi  
pi@192.168.0.24's password:  
Linux raspberrypi 4.19.83-v7+ #1277 SMP Mon Nov 11 16:30:44 GMT 2019 armv7l  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Tue Dec 10 15:24:49 2019 from 192.168.0.15  
pi@raspberrypi:~ $ mosquitto_pub -h localhost -t tesis -m "Hola Mundo"  
pi@raspberrypi:~ $ █
```

Figura 40 Enviando mensaje a un Tópico MQTT

```
pi@raspberrypi: ~  
login as: pi  
pi@192.168.0.24's password:  
Linux raspberrypi 4.19.83-v7+ #1277 SMP Mon Nov 11 16:30:44 GMT 2019 armv7l  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Fri Dec 6 15:01:44 2019  
pi@raspberrypi:~ $ mosquitto_sub -h localhost -t tesis  
Hola Mundo  
█
```

Figura 41 Suscrito a un tópico MQTT

4.3.1 DESARROLLO CODIGO

Lo anteriormente visto es un ejemplo de comunicación MQTT, pero ejecutada manualmente, lo que se necesita para este proyecto es el envío automático de datos recibidos desde los sensores y ser enviados inmediatamente o cada cierto periodo de tiempo al servidor Broker.

Para ello se dispone de la placa NodeMCU la cual recibe la información de cada sensor. Para este proyecto el tópico será solo uno , denominado “sensores”, a pesar de que estos se pueden bifurcarse en varios brazos, como por ejemplo “sensores/ph” “sensores/redox”, para realizar un código mas sencillo y comprensible visualmente.

Para enviar la información al servidor Broker se utilizará una librería llamada “PubSubClient” disponible en las librerías en el IDE de Arduino. Básicamente permite a la placa NodeMCU comportarse como un cliente MQTT, es decir, con la posibilidad de suscribirse a uno o varios tópicos y publicar mensajes en estos mismos.

```
pi@raspberrypi:~ $ mosquitto_sub -h localhost -t sensores
Temperatura: 26.875000 - pH: 0.000000 - ORP: 650 mV
█
```

Figura 42 Mensaje enviado desde NodeMCU (prueba)

4.3.2 CLIENTE PYTHON MQTT

Para poder recibir los datos enviados por el cliente, se implementó un código Python para poder manipular estos datos, como, por ejemplo, realizar el control automático y enviarlos a una base de datos.

Para ello se utilizó la librería “MQTT Paho” la cual permitira la conexión con nuestro control principal a través de este protocolo. Primero se debe importar las librerías a utilizar.

```
Import paho.mqtt.client as paho
Import time
```

Luego para realizar la conexión al servidor BROKER se debe indicar el host junto con el tópico que se desea suscribir.

```
client = paho.Client()
client.on_subscribe = on_subscribe
client.on_message = on_message
client.connect('192.168.4.10')
client.subscribe('sensores')
```

La función `on_subscribe` se ejecuta al suscribirse a un tópico, este caso imprimirá un mensaje cuando se haya conectado.

```
def on_subscribe(client, userdata, mid, granted_qos):
    print('Conectado al Broker!')
```

Y la función `on_message` se ejecuta cuando se recibe un mensaje, en esta parte se logra separar por partes el mensaje recibido para así obtener los valores asociados a cada sensor, utilizando métodos de división de Python como el `Split`.

```
def on_message(client, userdata, msg):
    mensaje = str(msg.payload)
    print(mensaje)
    div= mensaje.split("-")
    #TEMPERTURA
    temp=div[0]
    tempValue=temp[13:20]
    print(tempValue)
    #PH
    ph=div[1]
    phValue=ph[5:10]
```

```
print(phValue)
#ORP
orp=div[2]
orpValue=orp[6:9]
print(orpValue)
```

El objetivo final del código realizado anteriormente es obtener cada valor registrado por los sensores en una variable distinta (phValue, tempValue, orpValue) para que la manipulación de estas sea mucho más sencilla.

4.4 ADQUISICION DE DATOS

4.4.1 TEMPERATURA

La temperatura dentro del sistema es importante tener en cuenta, ya que esta al variar, provoca cambios considerables en la medición de las demás variables . El conexionado realizado para la conexión del sensor es el siguiente.

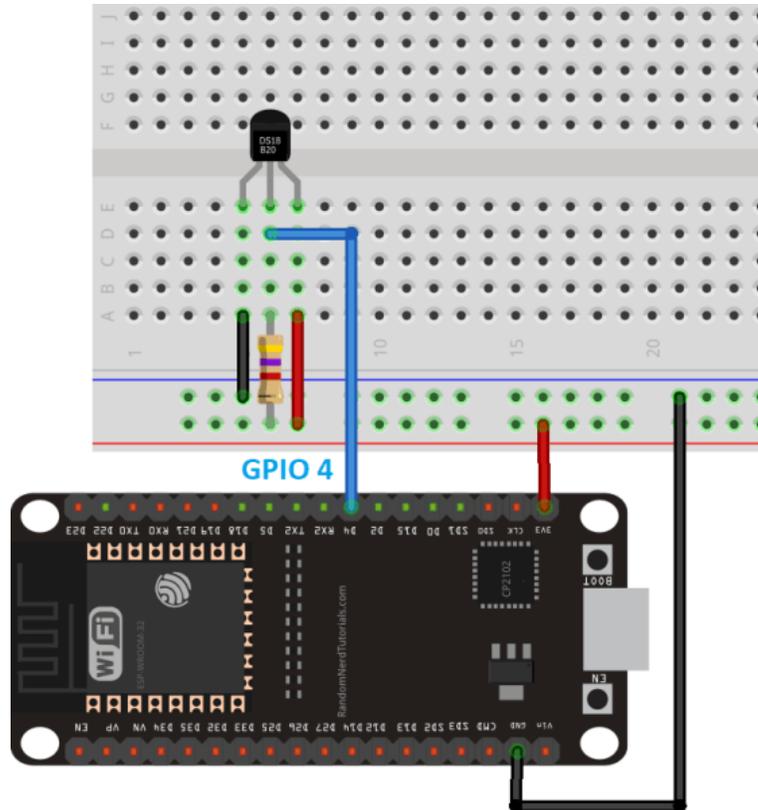


Figura 43 Conexión Sensor Temperatura

Existen dos métodos de conexión para el sensor de temperatura, pero en esta ocasión se utilizará el método normal. Por otro lado, para la programación existen dos librerías llamadas “OneWire” y “DallasTemperature”.

```
#include <OneWire.h>
#include <DallasTemperature.h>
```

Luego se establece a que pin ira conectado el cable de datos del sensor en la placa (PIN 4) y se declaran dos objetos de clases OneWire y DallasTemperature.

```
const int oneWireBus = 4;
OneWire oneWire(oneWireBus);
DallasTemperature sensors(&oneWire);
```

Finalmente se da inicio a la comunicación 1-Wire, junto con iniciar la comunicación con el monitor serie indicando la velocidad de transmisión para

poder observar los datos que el sensor entregue.

```
void setup() {
  // Inicia la comunicación serie
  Serial.begin(9600);
  // Iniciamos el bus 1-Wire
  sensors.begin();
}
void loop() {
  sensors.requestTemperatures();
  float temperatureC = sensors.getTempCByIndex(0); //SOLICITA LA TEMPERATURA EN
  °C
  float temperatureF = sensors.getTempFByIndex(0); //SOLICITA LA TEMPERATURA
  EN °F
  Serial.print(temperatureC);
  Serial.println("°C");
  Serial.print(temperatureF);
  Serial.println("°F");
  delay(5000);
}
```

4.4.2 PH

El sensor de pH de la empresa DFROBOT, dispone de su propia placa de conversión de señal, por lo que no será un inconveniente adaptarlo con la ADS1015, además de poseer librerías que permiten la calibración del sensor, utilizando soluciones buffer estándar de 4.0 y 7.0 pH.

Al calibrar una vez, los datos registrados serán guardados en la memoria EEPROM de la placa, por lo tanto, al reiniciarla no se tendrá que volver a calibrar.

Adicionalmente las librerías utilizadas se encargan de adaptar el pH según la temperatura registrada en el líquido.

4.4.3 REDOX

El sensor Redox al ser de la misma empresa fabricante que el sensor de pH, también posee una placa de conversión de señal, permitiendo que las mediciones se realicen entre -2000 y 2000 mV sin necesidad de agregar algún otro componente para el correcto funcionamiento.

Dentro de la página del fabricante se encuentra un código disponible para el uso del sensor y su calibración, el cual fue utilizado y adaptado para el desarrollo de este proyecto.

4.5 DATOS EN MYSQL

Una vez recibido la información de los sensores se desarrolló una base de datos en la cual se podrán guardar los datos localmente, indicando la hora y fecha en que se guardan como también un Valor ID. La base de datos permite que luego a la hora de desarrollar la página web, se puedan recoger los registros y puedan ser visualizados, como también generar gráficos de estos.

Para modificar la base de datos se utilizó PHPMyAdmin, desde la cual se crearon 3 tablas.



Figura 44 Tablas en base de datos MySQL

La tabla de agua contiene registros guardados sobre los cambios de la cantidad de agua (en mililitros) en que se va a implementar el proyecto, estos valores son ingresados manualmente desde una página web y es tomado por el programa encargado del control automático, para de tal forma estimar cuanto serán los recursos necesarios para utilizar.

←T→				ID	1	cantidad		
<input type="checkbox"/>		Editar		Copiar		Borrar	3	30000
<input type="checkbox"/>		Editar		Copiar		Borrar	2	85000
<input type="checkbox"/>		Editar		Copiar		Borrar	1	10000

Figura 45 Ejemplo Tabla agua, cantidad en mL

La tabla “cambioph” contiene registros sobre los cambios realizados manualmente desde la página la web, estos quedan inscritos junto con la fecha y hora de aquella modificación con el fin de informar al cliente, en qué momento se realizó cada acción.

←T→				ID	1	opcion	fecha
<input type="checkbox"/>		Editar		Copiar		Borrar	1 0.1 2019-11-24 23:53:29
<input type="checkbox"/>		Editar		Copiar		Borrar	2 0.4 2019-11-24 23:56:41
<input type="checkbox"/>		Editar		Copiar		Borrar	3 -0.3 2019-11-24 23:56:49
<input type="checkbox"/>		Editar		Copiar		Borrar	4 -0.3 2019-11-24 23:58:59
<input type="checkbox"/>		Editar		Copiar		Borrar	5 -0.5 2019-12-03 12:05:22

Figura 46 Ejemplo Tabla cambioph

Y la última tabla sensores, es la que contiene los registros de todos los sensores utilizados dentro del proyecto junto con un ID y la fecha y hora en que fueron tomados los datos.

Tiempo	1	pH	Temperatura	ORP	valor
2019-12-11 16:36:16	4.075	27.875	1841.76	47	
2019-12-11 16:36:12	4.075	27.9375	1842.28	46	
2019-12-11 16:36:08	4.085	27.9375	1844.72	45	
2019-12-11 16:36:03	4.075	27.9375	1842.66	44	
2019-12-11 16:35:59	4.075	28	1851.66	43	
2019-12-11 16:35:55	4.075	28	1842.92	42	
2019-12-11 16:35:51	4.075	27.8125	1842.92	41	
2019-12-11 16:35:47	4.075	27.9375	1825.06	40	

Figura 47 Ejemplo Tabla sensores

Esto es posible gracias a la librería utilizada en Python llamada "PyMysql", con la cual es posible realizar una conexión con la base de datos y realizar todo tipo de acciones que se podrían realizar desde la consola o terminal del servidor MySQL solo indicando el host, usuario y contraseña y la base de dato a la que se desea acceder.

```
Import pymysql
    try:
        connection =
pymysql.connect(host="192.168.0.24",user="edu",passwd="schettino2323",databas
e="tesis" )
        print("Conectado a la Base de datos!")
    except:
        print("No se logró conectar a la base de datos")

    insert1= "INSERT INTO `sensores`(`Tiempo`, `pH`, `Temperatura`, `ORP`,
`Cloro Libre`) VALUES (current_timestamp(), '%s', '%s', '%s', '0' );"
%(phValue,tempValue,orpValue)
    cursor = connection.cursor()
    cursor.execute(insert1)
    connection.commit()
    connection.close()
```

4.6 DATOS EN LA NUBE

4.6.1 THINKSPEAK

ThinkSpeak es una plataforma de IoT que permite recoger diversos datos y almacenarlos en la nube, posibilitando el desarrollo de aplicaciones IoT. Esta plataforma también ofrece aplicaciones que permiten analizar y visualizar datos en Matlab. Todos estos datos pueden ser enviados desde una Raspberry o Arduino, como también de datos recopilados desde un formulario en una página web por ejemplo.

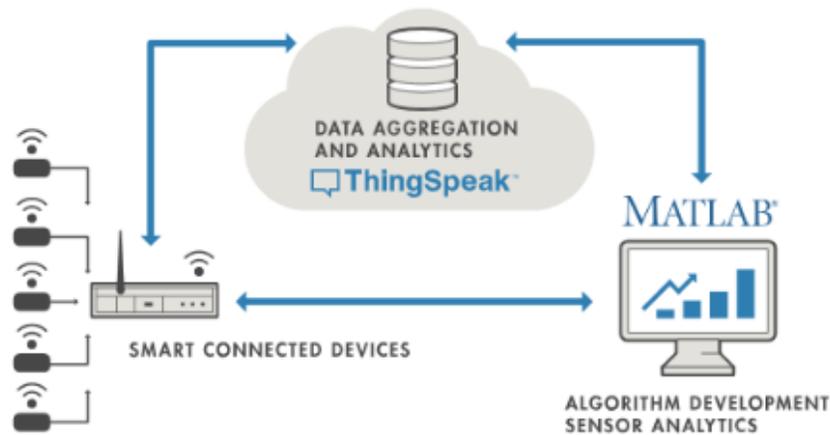


Figura 48 Diagrama ThinkSpeak

4.7 DESARROLLO PAGINA WEB

El desarrollo de una página web fue fundamental debido a que será la vía de comunicación entre el cliente y el sistema, es en donde se podrá visualizar los datos y además de realizar algunas acciones. Fue diseñada utilizando solo programación en PHP, HTML, CSS y JavaScript, no se utilizó ningún otro complemento que facilitara la programación.

4.7.1 INTERFAZ GRAFICA

Para una interfaz gráfica amigosa se realizó un menú fácil de comprender y que presente los valores obtenidos por los sensores inmediatamente utilizando gráficos. Dentro del denominado "index.php" se encuentran los gráficos de temperatura, pH y ORP , como también la cantidad de agua que está siendo monitoreada.



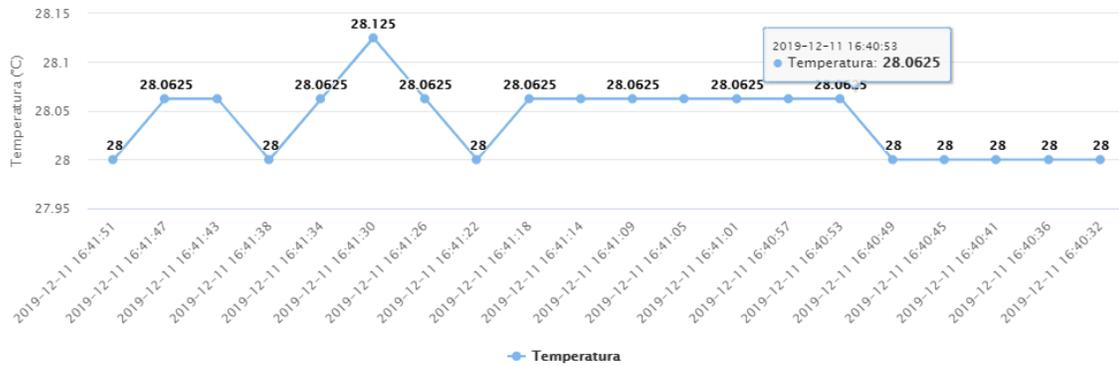
MONITOREO DE SENSORES

Cantidad de agua: 30000 ml

- BUSCAR EN BASE DE DATOS
- REINICIAR PAGINA
- CAMBIAR CANTIDAD DE AGUA
- AUMENTAR PH
- BAJAR PH
- AGREGAR CLORO

Sensor de Temperatura

Ultimos 20 valores registrados



Temperatura

Highcharts.com

Sensor de pH

Ultimos 20 valores registrados

Figura 49 Pagina Web (1)

Una de las opciones dentro del menú es la búsqueda en base de datos, en donde se puede buscar la información correspondiente al día que se indique, así se podrá observar todo el monitoreo realizado durante el día, ver las subidas o bajas, buscar patrones y analizar a que se deben, entre otros usos...

Busqueda por Fecha



Ingrese una fecha valida:

11 - 12 - 2019

Buscar Datos

Volver a pagina principal

Sensor de Temperatura



Figura 52 Búsqueda por Fecha, grafico Temperatura

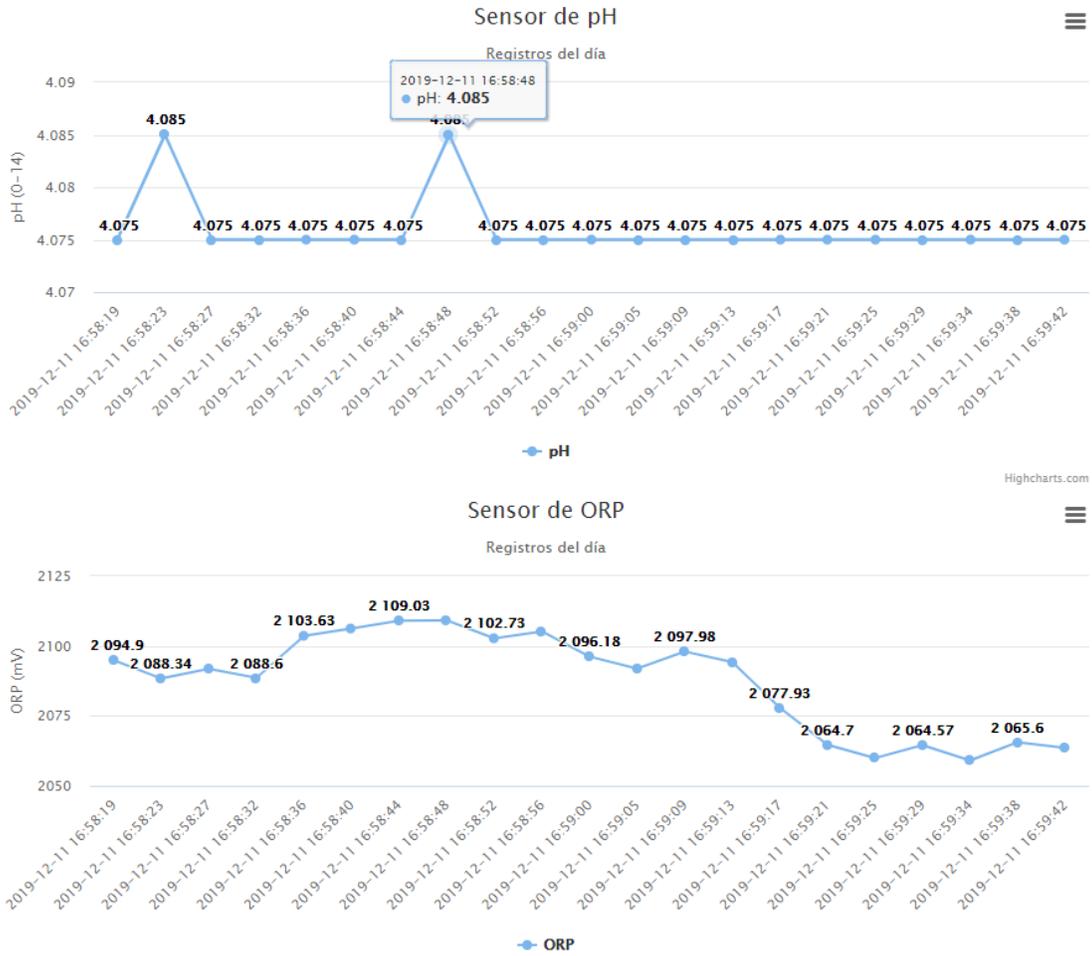


Figura 53 Búsqueda por fecha, grafico pH y ORP

También cuenta con un control manual de pH, en donde el cliente podrá elegir la cantidad que desea variar y observar un historial de los cambios realizados últimamente.



El pH actual del agua es de: 4.075

Subir 0.4 ▾ Enviar Volver

APLICAR CAMBIO

Esperar entre 15 a 20 min

Ultimas modificaciones realizadas:

Fecha	Cambio:
2019-12-06 14:30:47	-0.4
2019-12-06 14:30:39	0.1
2019-12-06 14:30:24	0.1
2019-12-06 14:17:19	0.1
2019-12-06 14:17:08	-0.1
2019-12-06 14:15:49	0.1

Figura 54 Ajuste Manual de pH vía Pagina web

4.7.2 GRAFICO DINAMICO

El desarrollo de los gráficos fue realizado gracias a la librería HighCharts combinado con PHP para solicitar la información de la base de datos MySQL. Dentro del código se pueden configurar los títulos del gráfico, el grosor de las líneas, lo puntos, como también el color. Es una herramienta muy útil a la hora de mostrar datos ya que tanto en computador como en un dispositivo móvil , al hacer contacto con cada punto sale detallada toda su información del eje X e Y.

HighCharts cuenta con una basta cantidad de tipos de gráficos, como lineales, por área, pie, de barras etcétera y todos con la posibilidad de automatizar el proceso de introducir los valores en el.

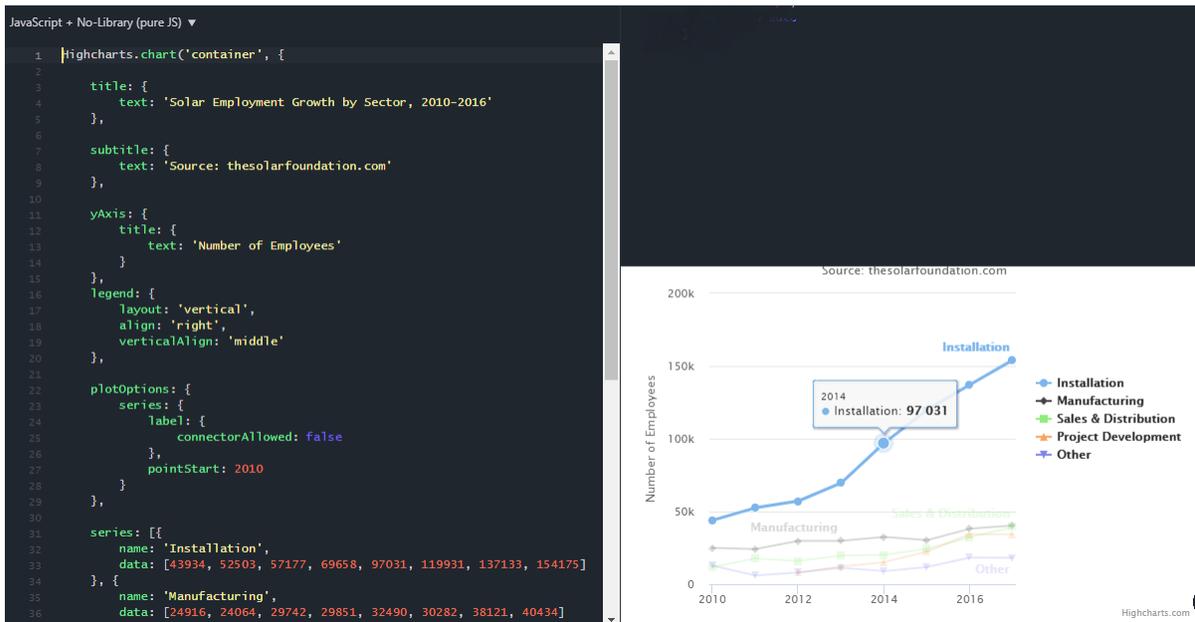


Figura 55 Ventana configuración HighCharts.com

4.8 CONTROL AUTOMATICO

El control automático del sistema se llevó a cabo gracias a la librería skfuzzy con la cual se logró diseñar las funciones de membresía necesarias con sus respectivas reglas para lograr controlar la variable pH utilizando las bombas de agua. Al obtener un valor numérico y no un 1 o un 0, es posible calcular la cantidad de solución necesaria para estabilizar el pH dependiendo de la cantidad de agua.

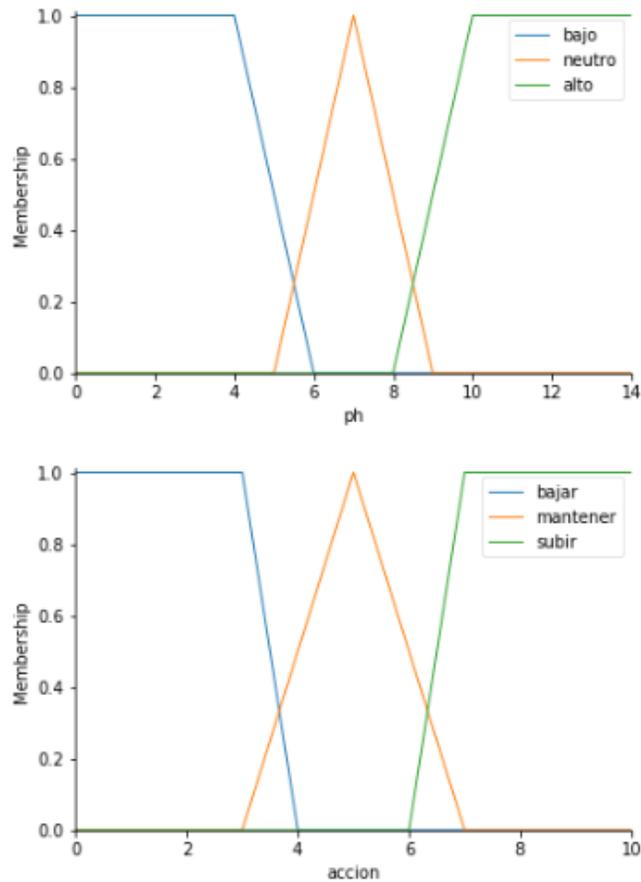


Figura 56 Funciones de membresía

Primero se deben crear las variables a utilizar, junto con el respectivo rango que tendrá.

```
ph = ctrl.Antecedent(np.arange(0, 15, 1), 'ph')
accion = ctrl.Consequent(np.arange(0, 11, 1), 'accion')
```

Luego se deben establecer las funciones de membresía a utilizar.

```
#Funcion de memebresia PH
ph['bajo'] = fuzz.trapmf(ph.universe, [0,0,4,6])
ph['neutro'] = fuzz.trimf(ph.universe, [5,7,9])
ph['alto'] = fuzz.trapmf(ph.universe, [8,10, 14, 20])
#Funcion de membresia salida
accion['bajar'] = fuzz.trapmf(accion.universe, [0,0,3,4])
accion['mantener'] = fuzz.trimf(accion.universe, [3,5,7])
```

```
accion['subir'] = fuzz.trapmf(accion.universe, [6,7, 14, 20])
```

Establecer las reglas necesarias y accionar el control.

```
rule1 = ctrl.Rule(ph['bajo'] , accion['subir'])  
rule2 = ctrl.Rule(ph['neutro'] , accion['mantener'])  
rule3 = ctrl.Rule(ph['alto'] , accion['bajar'])
```

```
accion_ctrl = ctrl.ControlSystem([rule1, rule2, rule3])  
accion_simulador = ctrl.ControlSystemSimulation(accion_ctrl)
```

```
accion_simulador.input['ph'] = ph  
accion_simulador.compute()  
valor=accion_simulador.output['accion']  
control=round(valor,2)  
print(control)
```

Del código visto anteriormente tiene como salida un número entre 0 y 10, dependiendo de cómo se encuentre el pH este número subirá o bajará. Y con este número será posible determinar cuando los actuadores deben energizarse para subir una cantidad grande de pH o pequeña.

5 RESULTADOS

5.1 PRUEBAS DE FUNCIONAMIENTO

La alimentación de la placa Raspberry se realizó mediante su cargador oficial, el cual suministra de 5V en corriente continua. La corriente debe ser mayor o igual a 2A, aunque puede depender de la cantidad de periféricos conectados que vaya a utilizar la Raspberry

La placa NodeMCU fue alimentada desde una batería Lipo recargable, de 7V y desde la misma placa se alimentaba los sensores utilizados los cuales solo consumían 3.3V.

Para el accionar de las bombas de agua mediante un control difuso se utilizó un puente H, al cual se le suministro un voltaje de 7V para alimentar las bombas de 5V, pero el Puente H absorbe 2, por lo tanto el voltaje estará bien para la bomba. El consumo promedio de la NodeMCU junto con los sensores es de 100mA, junto con la batería de 7V 370mAH, según cálculos revisados, las horas estimadas de duración de la batería será de 2,59 horas , las cuales podrán ser aumentadas cambiando la capacidad de la batería, debido que para el proyecto se eligió las más económica.

La prueba realizada consistió en variar el pH manualmente desde la pagina web realizada, cuando se registre un cambio significativo de este, se activó el control automático el cual regulara el pH. El control no esta activo durante todo el proceso, ya que el cambio de pH al introducir un agente químico externo tarda un tiempo en mezclarse con el agua, por lo cual el control se ejecutaba cada 2 minutos.

5.2 REGULADORES DE PH

Para regular la variable pH se hizo uso de dos agentes concentrados para aumentar o reducir el pH, se debe tener bastante precaución a la hora de

manipular los líquidos de cada recipiente debido a que el contacto de este con la piel puede llegar a producir quemaduras graves.



Figura 57 Incrementador de pH



Figura 58 Reductor de pH

5.2.1 ALMACENAMIENTO EN MYSQL

El resultado del almacenamiento en MySQL se puede visualizar desde la página web en buscar por fecha, desde el PHPMyAdmin y también desde el terminal o consola de la raspberry. Para visualizar los datos a través de la terminal de Raspberry se deben ingresar los siguientes comandos:

```
$sudo mariadb  
>use tesis  
>select * from sensores
```

```
pi@raspberrypi: ~
```

Tiempo	pH	Temperatura	ORP
2019-12-12 18:10:55	8.187	19.9375	111.374
2019-12-12 18:11:14	7.838	19.875	144.911
2019-12-12 18:11:17	8.016	19.8125	195.795
2019-12-12 18:11:28	8.456	19.9375	191.298
2019-12-12 18:16:45	7.983	19.875	5239.31
2019-12-12 18:16:56	8.209	20	5031.92
2019-12-12 18:17:07	8.424	19.875	4826.97
2019-12-12 18:17:18	7.999	20	4648.88
2019-12-12 18:17:29	8.198	19.9375	4452.03
2019-12-12 18:17:40	8.01	20	4270.72
2019-12-12 18:17:51	8.026	19.9375	4088.9
2019-12-12 18:18:02	8.016	19.8125	3965.67
2019-12-12 18:18:13	8.123	19.9375	3798.24
2019-12-12 18:18:24	8.123	19.9375	3639.94
2019-12-12 18:18:35	8.445	19.9375	3432.55
2019-12-12 18:18:46	8.026	19.9375	3278.35
2019-12-12 18:18:57	8.053	20.0625	3124.29
2019-12-12 18:19:09	8.182	20	2948.25
2019-12-12 18:19:20	8.467	20	2766.94
2019-12-12 18:19:31	8.445	19.8125	2556.34
2019-12-12 18:19:42	8.037	19.875	2407.8

Figura 59 Datos en MySQL

Los datos fueron guardados sin problema, realizando 321 registros en la base de datos.

5.2.2 ALMACENAMIENTO EN LA NUBE

Los datos ingresados en la Nube, utilizando la plataforma ThinkSpeak, se realizó en el mismo momento en que se ingresaban a la base de datos MySQL. En esta plataforma se logró enviar los datos a unos gráficos automáticos utilizando una librería en Python y solo una línea de código, importante resaltar que para enviar los datos a la Nube es necesario disponer de una conexión a internet, si no existe tal conexión el programa seguirá guardando sin problema los datos localmente.

```
Import requests
```

```
Enviar=requests.get("https://api.thingspeak.com/update?api_key=SK0GD2A5FODYLN  
OK&field1="+phValue+"&field2="+tempValue+"&field3="+orpValue)
```

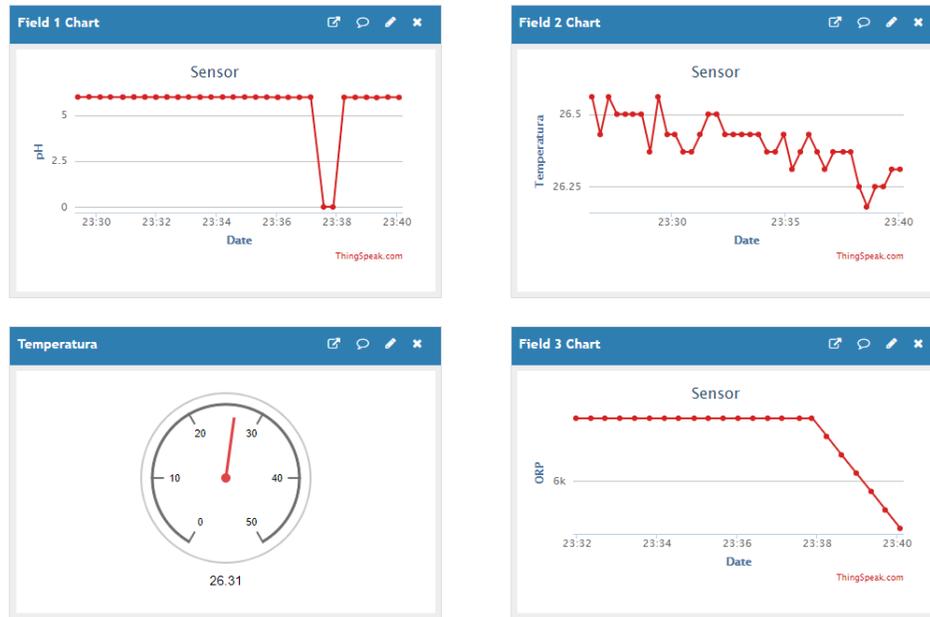


Figura 60 Datos en ThinkSpeak

5.3 PRUEBA ADQUISICION DE DATOS

Los datos obtenidos por el sensor de temperatura fueron correctos, funciono perfectamente tanto fuera del agua como en su interior en todas las pruebas realizadas, en cambio los sensores de pH y ORP presentaban valores que variaban demasiado, destacando el sensor ORP, que es el más delicado.

A continuación en la siguiente figura se observa los datos obtenidos al iniciar la prueba, la cual fue realizada en un estanque plástico que soporta 25 litros de agua.

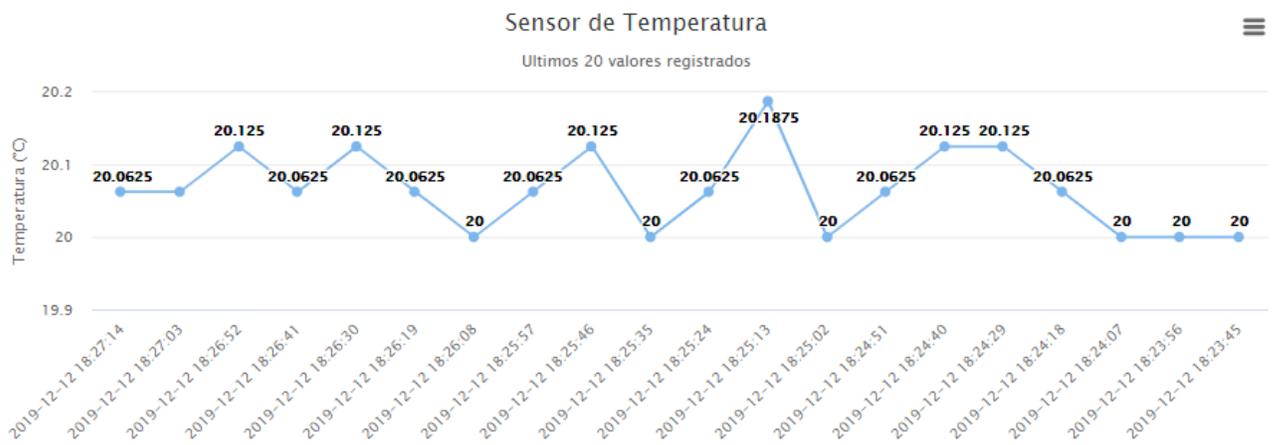


Figura 61 Temperatura Inicial

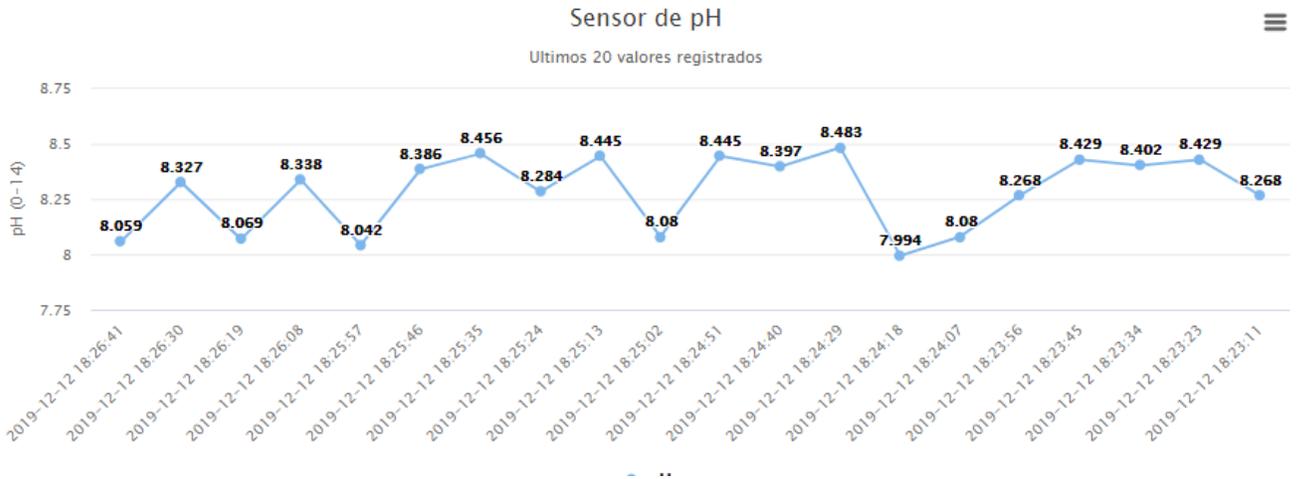


Figura 62 pH Inicial

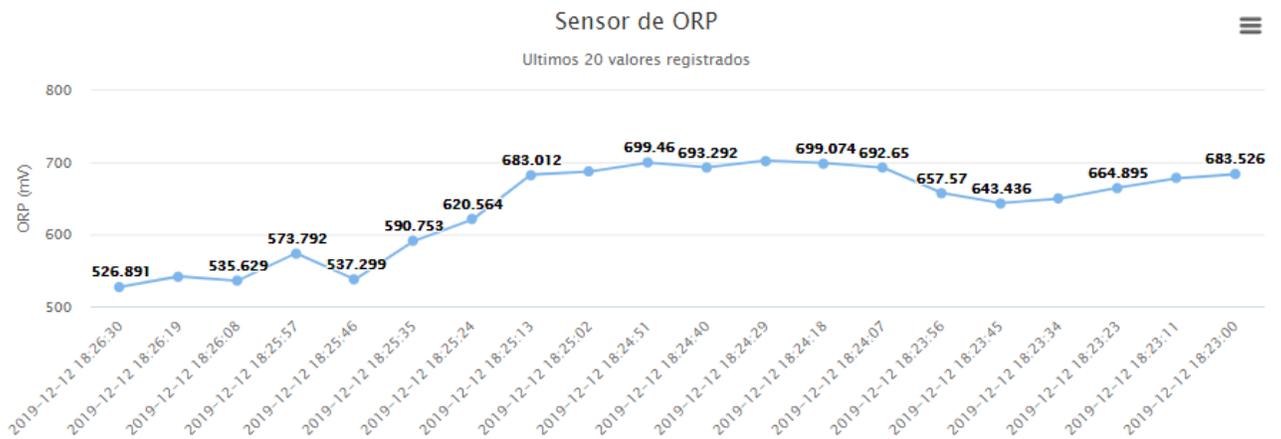


Figura 63 ORP Inicial

La variación del sensor ORP y de pH se debe a que son sensores que no están diseñados para el constante monitoreo, sino están destinados para el uso en laboratorios, pero para la ejecución del control automático, este no se vio afectado.

Luego de efectuar la prueba, el pH logro volver a un estado neutro, la temperatura se mantuvo constante, pero el ORP decayó drásticamente, cabe resaltar que la medida de ORP puede ser controlada mediante cloro, pero debido a los variables valores que entrega el sensor, se decidió no realizar debido a que podría llegar a alterar drásticamente el agua.



Figura 64 Valores en Agua alterada

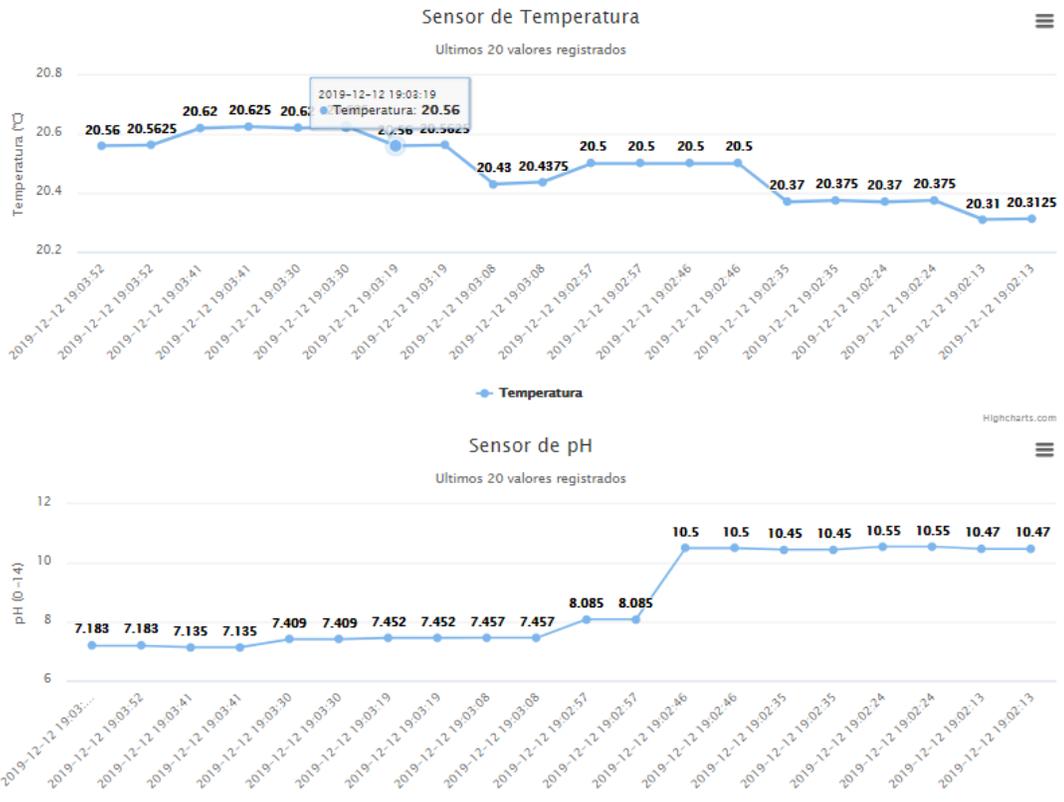


Figura 65 Resultado Final

5.4 DIAGRAMA DE BLOQUES

A continuación se presente un diagrama de bloques el cual representa el código utilizado para el monitoreo y control de la variable pH dentro del sistema acuoso.

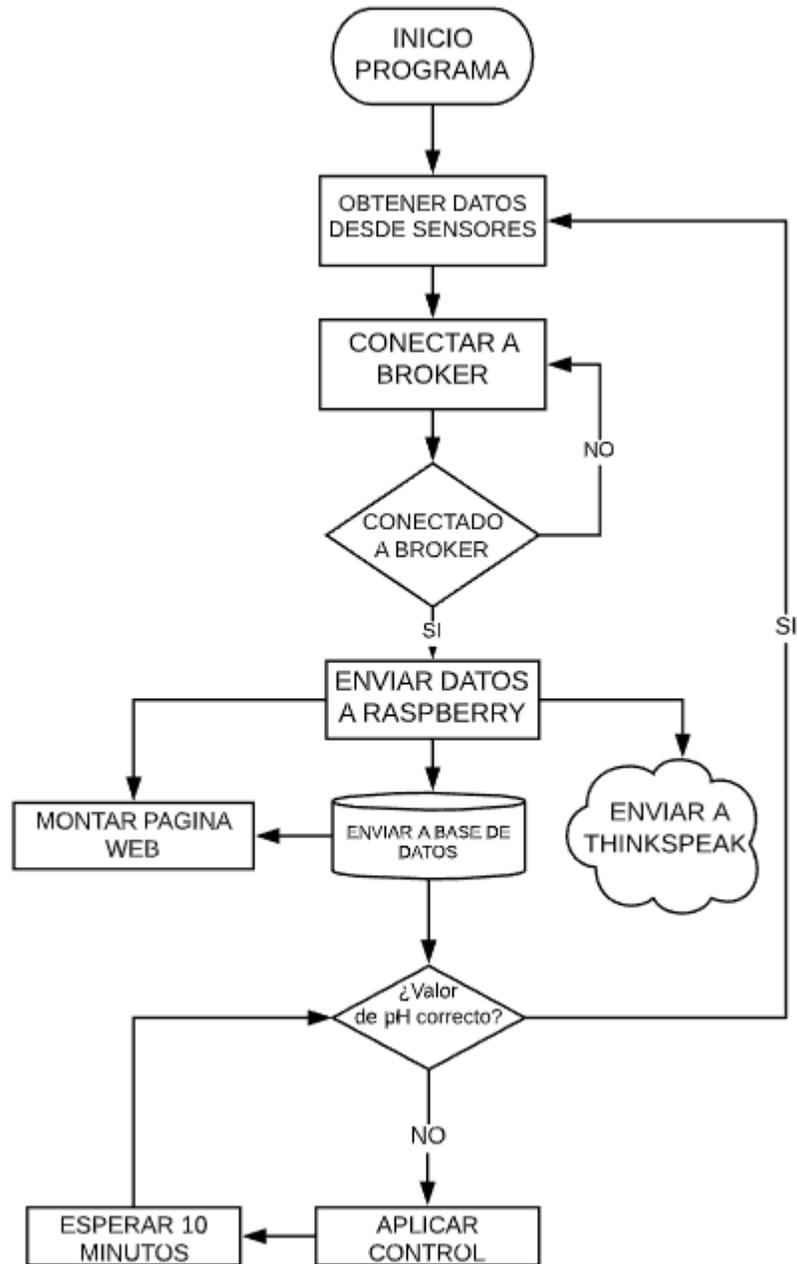


Figura 66 Diagrama de Bloques

5.5 VALIDACION

Para comprobar el correcto funcionamiento del control automático, se utilizó un set analizador de pH y cloro, el cual funciona tomando una cierta cantidad de agua y añadiéndole 4 gotas de una solución especial. Esta al entrar contacto con el agua tomara un color el cual indicara el que valor se encuentra el pH y el Cloro.

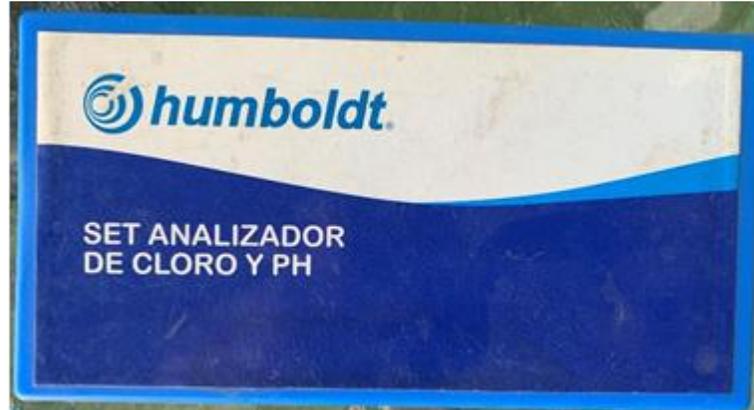


Figura 67 Set analizador de Cloro y PH

Los primeros resultados obtenidos fueron los siguientes, en donde se puede observar un correcto color, en este caso, para indicar que el pH se encuentra dentro del rango deseado, al igual que el cloro.

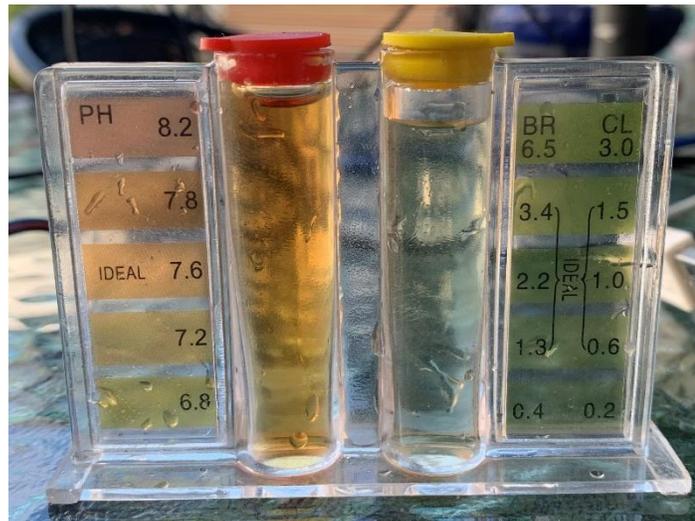


Figura 68 Set analizador

Este análisis se realizó con el agua que se utilizó para la prueba de funcionamiento, luego al variar el pH manualmente mediante la página web se

logró obtener el siguiente color, indicando claramente que el valor pH no corresponde a una medida deseada.



Figura 69 Medida pH modificado, Set analizador

Luego de activar el control automático se logró regular el pH, obtenido un color más cercano a lo ideal, pero con una pequeña turbidez en el agua.



Figura 70 Resultado Obtenido

5.6 ANALISIS DE RENDIMIENTO.

Durante el proceso de midió el rendimiento de la CPU de la raspberry y su temperatura, la cual estaba encargada se mantener el servidor web, comportarse como Broker, recibir los datos vía MQTT, enviar datos a la nube y a la base de datos y además realizar los cálculos respectivo para el control difuso. Varias de estas funciones ocurrían en segundo plano por lo tanto el uso de la CPU variaba en 3% a 6% , pero cuando llegaba a un peak de 21% era cuando se realizaba los cálculos para el control difuso por solo un par de segundos. Esto producía que la placa se ralentizará un momento.

Por otro lado la temperatura alcanzo los 41 grados centígrados, un valor bien recibido debido a que la placa Raspberry puede trabajar sin fallos con temperaturas entre 32 y 50 grados centígrados.



Figura 71 CPU y Temperatura Raspberry Pi

6 CONCLUSION

Los nuevos avances tecnológicos dentro de la categoría de sistemas embebidos, sumado a la adaptación de lenguajes de programación y su capacidad de comunicarse o conectar con estos dispositivos desde una red local o remotamente vía internet, demuestran que existe un gran potencial a la hora de crear sistemas inteligentes, demostrando una gran efectividad, además con costos muy por lo bajo comparados con los grandes controladores que se pueden encontrar en el mercado hoy en día.

La posibilidad de realizar conexiones que permiten la comunicación con dispositivos localmente o remotamente vía internet, la autonomía y el bajo consumo de estas placas, abre las puertas al mundo del internet de las cosas, el cual actualmente solo está siendo enfocado en dispositivos de uso cotidiano pero a medida que el hardware vaya mejorando, permitiendo controlar por ejemplo, equipos industriales, será un herramienta indispensable en un futuro, tener la posibilidad de controlar y analizar todos los datos directamente desde un dispositivo, literalmente desde la palma de la mano.

Con respecto al proyecto el resultado específico fue el desarrollo de un Gateway IoT, el cual cuenta con la capacidad de monitorear en tiempo real un sistema creador a partir de un software y hardware abierto, además de contar con la capacidad de permitir la comunicación entre dispositivos puede almacenar los datos obtenidos de estos, ya sea localmente como en una nube en internet, dando un respaldo de sus datos al cliente.

Gracias al bajo consumo de la placa Raspberry y su capacidad de realizar múltiples funciones se hizo posible la creación de un servidor web y una base de datos, el cual permitió la conexión de múltiples dispositivos para el monitoreo de los sensores utilizados en tiempo real, como también la visualización de los datos registrados en MySQL y un control sobre los actuadores.

El sistema de control difuso resultó eficaz a la hora de controlar la variable pH, pero no se contó con que los agentes concentrados utilizados para el incremento o reducción del pH incorporarían otro tipo de contaminación al agua, en este caso el agua se tornó turbia, lo que da cuenta de que para mantener la calidad del agua se requiere otro tipo de dispositivos como un filtro.

En un futuro gracias a las constante actualizaciones y su gran escalabilidad que poseen estas tecnologías, se podría implementar un sistema completo de desinfección del agua, siempre y cuando existan los sensores y actuadores indicados para el trabajo, con la posibilidad de interactuar o visualizar fácilmente con todos los dispositivos que podrán estar conectados entre sí.

GLOSARIO

API: Application Programming Interface

API: Application Programming Interface

UX: User Experience

TCP: Transmission Control Protocol

IP: Internet Protocol

CPU: Central Processing Unit

SO: Sistema operativo

HTML: Hyper Text Markup Language

LAN: Local Area Network

WAN: Wide Area Network

HTTP: HyperText Transfer Protocol

MQTT: Message Queue Telemetry Transport

FTP: File Transfer Protocol

VNC: Virtual Network Computer

SSH: Secure Shell

IOT: Internet Of Things

WIFI: Wireless Fidelity

3G: Tercera Generacion

HDMI: High Definition Multimedia Interface

USB: Universal Serial Bus

SD: Secure Digital

GPIO: General Purpose Input Output

URL: Uniform Resource Locator

DNS: Domain Name System

7 REFERENCIAS.

[1] Vasileios Karagiannis, Periklis Chatzimisios, Francisco Vazquez-Gallego, Jesus Alonso-Zarate, "A Survey on Application Layer Protocols for the Internet of Things" *Tecnologic de Telecomunicacions de Catalunya*, 2015

[2] Antti Luoto, Kari Systs, "IoT Application Deployment Using Request-response Pattern with MQTT", Tampere University of Technology, Tampere, Finland

[3] Fundación EOI. Las tecnologías IoT dentro de la industria conectada 4.0. Escuela de Organización Industrial, España, 2015

[4] ¿Qué es y cómo funciona el Internet de las cosas? (2014). Retrieved from <https://hipertextual.com/archivo/2014/10/internet-cosas/>

[5] Margaret Rouse. (2017). Internet de las cosas (IoT). 2019, de TechTarget Sitio web: <https://searchdatacenter.techtarget.com/es/definicion/Internet-de-las-cosas-IoT>

[6] Mejía Clara, M. (2015). Análisis de la calidad del agua para consumo humano y percepción local de las tecnologías apropiadas para su desinfección a escala domiciliaria, en la microcuenca El Limón, San Jerónimo, Honduras. Turrialba, Costa Rica.

[7] Cambra, P. (2008). Origen y Futuro de MySQL. Obtenido de Cámbrico: <http://cambrico.net/mysql/origen-y-futuro-de-mysql>

[8] Electronicwings. (s.f.). NodeMCU Basics | NodeMCU. Retrieved 6 junio, 2019,

[9] Rose, K., Eldridge, S., & Chapin, L. (2019). *LA INTERNET DE LAS COSAS—*

[10] Yuan, M. (2018). *Coconino MQTT*. [online] Ibm.com. Obtenido de : <https://www.ibm.com/developerworks/ssa/library/iot-mqtt-why-good-for-iot/index.html>.

[11] Wiki.dfrobot.com. (n.d.). *PH_meter_SKU__SEN0161_-DFRobot*. [online] Available at: https://wiki.dfrobot.com/PH_meter_SKU__SEN0161_#target_2

[12] gallego, M. 2000. El agua, vehículo de contaminación. Página electrónica en (línea) Turrialba, Costa Rica. Consultado junio 2019. www.badad.com/no01/agua.html

[13] Luis Vivas, Horacio Muñoz Abbate, Mauro Cambarieri, Nicolás García Martínez, Marcelo Petroff, Arquitectura de Software con websocket para aplicaciones web multiplataforma

[14] UNA BREVE RESEÑA. Internet Society. Obtenido de: <https://www.internetsociety.org/wp-content/uploads/2017/09/report-InternetOfThings-20160817-es-1.pdf>

8 ANEXOS

8.1 LECTURA Y CALIBRACION SENSOR PH

```
#include <Arduino.h>
#include "DFRobot_ESP_PH_WITH_ADC.h"
#include "OneWire.h"
#include "DallasTemperature.h"
#include "Adafruit_ADS1015.h"
#include "EEPROM.h"

#define ONE_WIRE_BUS 2
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);

DFRobot_ESP_PH_WITH_ADC ph;
Adafruit_ADS1115 ads;

float voltage, pHvalue, temperature = 25;

float readTemperature()
{
  //add your code here to get the temperature from your temperature sensor
  sensors.requestTemperatures();
  return sensors.getTempCByIndex(0);
}

void setup()
{
  Serial.begin(115200);
  EEPROM.begin(32); //needed EEPROM.begin to store calibration k in eeprom
  ph.begin();
  sensors.begin();
  ads.setGain(GAIN_ONE);
  ads.begin();
}

void loop()
```

```

{
  static unsigned long timepoint = millis();
  if (millis() - timepoint > 1000U) //time interval: 1s
  {
    timepoint = millis();

    voltage = ads.readADC_SingleEnded(1) / 10; // read the voltage
    Serial.print("voltage:");
    Serial.println(voltage, 4);

    temperature = readTemperature(); // read your temperature sensor to execute
temperature compensation
    Serial.print("temperature:");
    Serial.print(temperature, 1);
    Serial.println("^C");

    pHValue = ph.readPH(voltage, temperature); // convert voltage to pH with
temperature compensation
    Serial.print("pH:");
    Serial.println(pHValue, 4);
  }
  ph.calibration(voltage, temperature); // calibration process by Serial CMD
}

```

8.2 PRUEBA ADS1015

```

#El siguiente código permite la lectura de las 4 entradas analógicas que
dispone el ADS1015
import time
import Adafruit_ADS1x15
adc = Adafruit_ADS1x15.ADS1115()
GAIN = 1

print('Leyendo valores, press Ctrl-C to quit...')

print('| {0:>6} | {1:>6} | {2:>6} | {3:>6} |'.format(*range(4)))

```

```

print('-' * 37)

while True:

    values = [0]*4
    for i in range(4):
        values[i] = adc.read_adc(i, gain=GAIN)

    print('| {0:>6} | {1:>6} | {2:>6} | {3:>6} |'.format(*values))

    time.sleep(0.5)

```

8.3 RESCATANDO DATOS MQTT EN RASPBERRY

```

import ssl
import sys
import paho.mqtt.client
import pymysql

def on_connect(client, userdata, flags, rc):
    print('connected (%s)' % client._client_id)
    client.subscribe(topic='sensor/ph', qos=2)

def on_message(client, userdata, message):
    print('-----')
    #print('topic: %s' % message.topic)
    #print('Mensaje: %s' % message.payload)
    #print('qos: %d' % message.qos)
    x= str(message.payload)
    div= x.split(";")
    ph=div[0]
    pH=ph[6:13]
    print(pH) #SENSOR PH PARA ENVIAR A BASE DE DATOS
    redox=div[1]
    REDOX=redox[8:15]
    print(REDOX)
    #ABRIR BASE DE DATOS
    try:

```

```

        connection
pymysql.connect(host="localhost",user="root",passwd="",database="datos" )
    except:
        print("No se logró conectar a la base de datos")

    insert1= "INSERT INTO `tabla` (`ph`, `redox`, `cloro`, `fecha`) VALUES
('%s', '%s', '3', current_timestamp());" %(pH,REDOX)
    cursor = connection.cursor()
    cursor.execute(insert1)
    connection.commit()
    connection.close()

def main():
    client = paho.mqtt.client.Client(client_id='Sensor', clean_session=False)
    client.on_connect = on_connect
    client.on_message = on_message
    client.connect(host='192.168.0.20', port=1883)
    client.loop_forever()
if __name__ == '__main__':
    main()

```

8.4 ENVIANDO MENSAJE DESDE NODEMCU A BROKER MQTT

```

#include <PubSubClient.h>

//CONFIGURACION MQTT
const char* mqttServer = "192.168.4.10";
const int mqttPort = 1883;
const char* mqttUser = "pi";
const char* mqttPassword = "schettino2323";

// INICIO Y MENSAJE A ENVIAR
PubSubClient client(espClient);

```

```

char mensaje[80];
void ConectarBroker()
{
    //Conexión a broker
    client.setServer(mqttServer, mqttPort);
    while (!client.connected())
    {
        Serial.println("Conectando a broker MQTT...");
        if (client.connect("sensores", mqttUser, mqttPassword ))
        {
            delay(500);
            Serial.println("Conectado!");
        }
        else
        {
            Serial.println("FALLO EN LA CONEXION! - Estado: ");
            Serial.print(client.state());
            delay(2000);
        }
    }
}

sprintf(mensaje, "Temperatura: %f - pH: %f - ORP: %f mV", t2, pHValue, orpValue);
Serial.print("Mensaje enviado: ");
Serial.println(mensaje);
client.publish("sensores", mensaje);
Serial.println("Mensaje enviado con exito!");

```

8.5 GRAFICO CON LIBRERÍA HIGHCHARTS

```

<script type="text/javascript">
    Highcharts.chart('container2', {
    chart: {
        type: 'line'
    },
    title: {
        text: 'Sensor de pH'
    }

```

```

    },
    subtitle: {
        text: 'Ultimos 20 valores registrados'
    },
    xAxis: {
        categories: [
            <?php
                $servername='localhost';
                $username='root';
                $password='';
                $dbname='tesis';

                $conexion=mysqli_connect("$servername", "$username", "$password", "$dbname");

                $query="SELECT * FROM `sensores` ORDER BY valor DESC LIMIT 20";
                $res=mysqli_query($conexion,$query);
                while ($data=mysqli_fetch_array($res)) {
                    $tiempo= $data['Tiempo'];
                    ?>
                    "<?php echo $tiempo?>",
                <?php } ?>
            ]
        },
    yAxis: {
        title: {
            text: 'pH (0-14)'
        }
    },
    plotOptions: {
        line: {
            dataLabels: {
                enabled: true
            },
            enableMouseTracking: true
        }
    }

```

```

    },
    series: [{
        name: 'pH',
        data: [
            <?php
                $servername='localhost';
                $username='root';
                $password='';
                $dbname='tesis';

                $conexion=mysqli_connect("$servername","$username","$password","$dbname");

                $query="SELECT * FROM `sensores` ORDER BY valor DESC LIMIT 20";
                $res=mysqli_query($conexion,$query);
                while ($data=mysqli_fetch_array($res)) {
                    $pH= $data['pH'];
                    ?>
                    <?php echo $pH?>,
                <?php } ?> ]
        ],
    },
    ],
    responsive: {
        rules: [{
            condition: {
                maxWidth: 400
            },
            chartOptions: {
                legend: {
                    layout: 'horizontal',
                    align: 'center',
                    verticalAlign: 'bottom'
                }
            }
        }
    ]
}

```

},.);