



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
Facultad de Ingeniería en Electricidad y Computación

“DISEÑO DE UN SISTEMA DE SENSORES DEDICADO AL
CONTROL Y MONITOREO DE LA CALIDAD DEL AGUA EN
PISCINAS DE CRIANZA DE CAMARONES UBICADA EN LA
ISLA SANTA ROSA - GUAYAQUIL”

INFORME DE PROYECTO INTEGRADOR

Previo a la obtención del Título de:

LICENCIATURA EN REDES Y SISTEMAS OPERATIVOS

KERWIN MIGUEL RODRÍGUEZ FREIRE
KEVIN ANTONIO VILLAMAR RODRÍGUEZ

GUAYAQUIL – ECUADOR

AÑO: 2017

AGRADECIMIENTO

A Dios por permitirnos estar vivos aún, a nuestros padres y familiares por el gran sacrificio que han hecho por nuestra educación, a nuestros compañeros de estudio, a maestros y amigos, quienes sin su ayuda nunca habiéramos podido hacer esta tesis. A todos ellos se lo agradecemos con mucho aprecio.

Kerwin Miguel Rodríguez Freire

AGRADECIMIENTO

Agradezco a los docentes, padres y familiares que me ayudaron en este proceso vida universitaria.

Kevin Antonio Villamar Rodríguez

DEDICATORIA

Dedicamos este trabajo a Dios por permitirnos estar presentes, a nuestros padres quienes nos dieron la vida, educación, apoyo y consejos para poder continuar, a nuestros amigos por ayudarnos con ciertos temas en alguna materia que estuviésemos flaqueando, a nuestros amigos de KCLC por apoyarnos en todos los momentos, ayudarnos a distraernos y botarnos de los laboratorios cuando teníamos que presentar algún proyecto no terminado.

Kerwin Miguel Rodríguez Freire.

TRIBUNAL DE EVALUACIÓN

Ing. Adriana Collaguazo Jaramillo

PROFESOR TUTOR

Ing. Robert Andrade Troya

PROFESOR COLABORADOR

DECLARACIÓN EXPRESA

"La responsabilidad y la autoría del contenido de este Trabajo de Titulación, nos corresponde exclusivamente; y damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"

.....
Kevin Villamar

.....
Kerwin Rodríguez

RESUMEN

En la isla Santa Rosa se encuentra ubicada la camaronera José Bravo, ésta tiene varias piscinas para la crianza de camarón, de las cuales se tiene siempre una pérdida considerable del 25% de la cosecha, esto como consecuencia a que el agua se pudre o malogra por no realizar el cambio de agua en el momento oportuno ya que no tienen los implementos completos para verificar el control de la calidad del agua de las piscinas como lo indicamos en el capítulo 1.

La solución planteada a esta problemática es la creación de un sistema de medición que realizará la tarea de examinar la calidad del agua periódicamente por medio del sub-sistema de sensores, acompañado de un sub-sistema de energización solar que le proporcionará la autonomía necesaria para poder realizar las mediciones y podrá presentar dichos resultados por medio de un sub-sistema de aplicación web, para que así la persona pertinente del cuidado de las piscinas se encargue de realizar el respectivo mantenimiento acorde con los valores mostrados por el sub-sistema de sensores. En el capítulo 2 presentamos el esquema general de la solución, así como la conformación de los subsistemas para tener una mejor estructura de trabajo y el capítulo 3 se puede observar con más detalle las características de los subsistemas, así como los dispositivos que lo conforman.

ABSTRACT

In the Santa Rosa Island is located the José Bravo Shrimp hatchery, it has several pools for the shrimp breeding, of which is always gotten a considerable loss of 25% of the harvest, this because of the water rots or spoils by not making the change of water at the right time, this happens because ok they do not have the full complements for verify the control of the water qualify of the pools as indicated in chapter 1.

The solution proposed to the problem is the creation of a measurement system that will perform the track of examining the water qualify periodically by means of the sensor sub-system, accompanied by a sub-system of solar energization that will provide the necessary autonomy to be able to perform the measurements and may present these results through web application sub-system, so that the relevant pool care person is responsible for carrying out the corresponding maintenance according to the values shown by the sensor sub-system. In chapter 2 we present the general scheme of the solution, as well as the conformation of the sub-systems to have a better structure of work and chapter 3 can be observed in more detail the characteristics of the subsystems, as well as the devices that conform it.

ÍNDICE GENERAL

RESUMEN	I
ABSTRACT	II
ÍNDICE GENERAL	III
ABREVIATURAS.....	V
SIMBOLOGÍA.....	VI
ÍNDICE DE FIGURA.....	VII
ÍNDICE DE TABLAS	VIII
CAPÍTULO 1	1
1. ANÁLISIS DEL PROBLEMA.....	1
1.1 Tasa de Mortalidad elevada.....	2
1.2 Objetivos.....	3
1.2.1 Objetivo Principal	3
1.2.2 Objetivo Específico	3
1.2.3 Justificación	4
CAPÍTULO 2	5
2. SISTEMA DE SENSORES DE MEDICIÓN Y VISUALIZACIÓN REMOTA.....	5
2.1. Subsistema de monitoreo	7
2.1.1. Sensor de Temperatura DS18B20.....	7
2.1.2. Sensor de PH.....	8
2.1.3. Sensor de Oxígeno disuelto.....	8
2.1.4. Sensor de Turbidez SEN0189	9
2.1.5. Módulo Wireless ESP01- ESP8266	10
2.1.6. Direccionamiento IP	11
2.2. Subsistema de aplicación web.....	12
2.2.1. Comunicación de los sensores	12
2.3. Subsistema de provisión de energía	17
CAPÍTULO 3	19
3. IMPLEMENTACIÓN DEL SISTEMA.....	19
3.1. Conexión de sensores	20
3.1.1. Sensor de Temperatura	20

3.1.2.	Sensor de PH.....	24
3.1.3.	Sensor de Oxígeno Disuelto	25
3.1.4.	Sensor de Turbidez.....	26
3.2.	Implementación de Servidor web en Raspberry Pi 3	27
3.3.	Implementación de servidor web Apache con PHP y MySQL.....	31
3.4.	Implementación de alertas por correo electrónico	33
3.5.	Implementación de seguridades en el servidor web	34
3.5.1.	Cambio de directorio por defecto	34
3.5.2.	Evitar búsqueda de DNS.....	35
3.5.3.	No mostrar versión del servidor	35
3.5.4.	Petición HTTP de seguimiento.....	35
3.5.5.	Inhabilitar las Etags.....	35
3.5.6.	Denegar el acceso al directorio raíz.....	36
3.5.7.	Bloquear los ataques DoS HTTP	36
3.5.8.	Apagar el módulo status	36
3.6.	Configuración de modo puente en dispositivo Mikrotik	36
3.6.	Configuración de router.....	38
3.7.	Proveedor del servicio de internet.....	38
CAPÍTULO 4		40
1.	CRONOGRAMA DE TRABAJO Y PRESUPUESTO.....	40
4.1.	Plan de trabajo.....	40
4.2.	Presupuesto.....	41
CONCLUSIONES Y RECOMENDACIONES		42
BIBLIOGRAFÍA		43
ANEXOS		44

ABREVIATURAS

CSV	Comma separated values
DHCP	Dynamic Host Configuration Protocol,
DNS	Domain Name System
DoS	Denial of Service
FET	Field-Effect Transistor
FSL	Free Space Loss
GND	Ground
HDMI	High-Definition Multimedia Interface
HDPE	High-Density Polyethylene
HSP	Host Signal Processing
HTTP	Hypertext Transfer Protocol
IDE	Integrated Development Environment
IP	Internet Protocol
LAN	Local Area Network
LCD	Liquid Crystal Display
Mbps	Megabits per second
MicroSD	Micro Secure Digital
NTU	Nephelometric Turbidity Unit
OS	Operative System
PHP	PHP Hypertext Preprocessor
PoE	Power over Ethernet
RAM	Random Access Memory
SMS	Short Message Service
SMTP	Simple Mail Transfer Protocol
SQL	Structured Query Language
USB	Universal Serial Bus

SIMBOLOGÍA

\$	Dólar
°C	Grados Centígrados
A	Amperio
dBi	Decibels-isotropic
dBm	decibles-milliwatts
g	Gramo
GB	Giga Bytes
GHz	Giga Hercios
Km	Kilometro
KΩ	Kilo-Ohmio
Lt	Litro
m	Metro
mA	Miliamperios
mg	Miligramo
mg/litro	Miligramo sobre litro
ms	Milisegundos
PH	Potencial de Hidrógeno
V	Voltaje
W	Watts

ÍNDICE DE FIGURA

Figura 1.1. Visualización satelital camaronera José Bravo	1
Figura 2.1. Diseño general de la solución	5
Figura 2.2. Diseño topológico sistema de sensores	6
Figura 2.3. Cuadro de información de piscina	12
Figura 2.4. Perfil de radio enlace entre estación y Sub-sist. Monitoreo 1	16
Figura 2.5. Perfil de radio enlace entre estación y Sub-sist. Monitoreo 3	16
Figura 2.6. Perfil de radio enlace entre estación y Sub-sist. Monitoreo 19	17
Figura 2.7. Perfil de radio enlace entre estación y Sub-sist. Monitoreo 15	17
Figura 3.1. Diseño subsistema de monitoreo	19
Figura 3.2. Conexión sensor hacia Arduino UNO	20
Figura 3.3. Esquemático del sensor DS18B20	21
Figura 3.4. Conexión Sensor Oxígeno Disuelto hacia Arduino UNO	26
Figura 3.5. Conexión de Sensor de Turbidez hacia Arduino UNO	27
Figura 3.6. Descarga de Imagen Ubuntu MATE	29
Figura 3.7. Pantalla principal Win32 Disk Imager	30
Figura 3.8. Pantalla principal Ubuntu MATE	31
Figura 3.9. Configuración PHP	33
Figura 3.10. Configuración WinBox	36
Figura 3.11. Configuración nueva interface	37
Figura 3.12. Configuración puerto bridge	38
Figura 4.1. Plan de trabajo	40

ÍNDICE DE TABLAS

Tabla 1.1. Condiciones necesarias del agua.....	2
Tabla 2.1. Distribución de subsistema de monitoreo.....	6
Tabla 2.2. Características del sensor de temperatura.....	7
Tabla 2.3. Característica del sensor de PH.....	8
Tabla 2.4. Característica de sensor oxígeno disuelto.....	8
Tabla 2.5. Característica de sensor de turbidez.....	9
Tabla 2.6. Característica de Arduino UNO.....	9
Tabla 2.7. Característica de pantalla LED.....	10
Tabla 2.8. Características módulo Wireless ESP8266.....	11
Tabla 2.9. Direccionamiento de IP para sensores y servidor web.....	12
Tabla 2.10. Características de Raspberry Pi 3.....	13
Tabla 2.11. Características de antena omnidireccional.....	14
Tabla 2.12. Consumo de corriente de cada sub-sistema de monitoreo.....	18
Tabla 2.13. Características de regulador de corriente.....	18
Tabla 3.1. Referencia de resistencia con longitud de cable.....	22
Tabla 4.1. Costos de sistema.....	41

CAPÍTULO 1

1. ANÁLISIS DEL PROBLEMA

A partir de la información proporcionada por las personas que trabajan en la camaronera José Bravo, ubicada en la isla Santa Rosa provincia del Guayas, pudimos conocer la problemática que padecen actualmente.

La camaronera posee 6 piscinas naturales en las que se cría la larva de camarón para luego de su madurez poder comercializarla como camarón desarrollado, dichas piscinas son de tamaño irregular como se lo puede observar en la Figura 1.1, cada piscina tiene una profundidad de 1 metro aproximadamente y la camaronera tiene un perímetro total de 3.481,85 metros y área de 328.884,70 metros cuadrados. Para poder hacer el cambio del agua de las piscinas, dentro de la camaronera tiene un canal que transporta el agua del exterior hacia las distintas piscinas y a su vez sirve de desagüe para el agua turbia o utilizada por las piscinas.



Figura 1.1. Visualización satelital camaronera José Bravo

El tipo de camarón que se cosecha es el *litopenaeus vannamei* [1], que se cultiva en agua salada y tiene un tiempo de cultivación de entre 60 y 90 días dependiendo del peso que se desea obtener en la cosecha. Normalmente en una producción se realiza siembra de 2`000 000 de larvas de camarón. Para que el camarón tenga un crecimiento normal, la calidad del agua debe tener unas condiciones necesarias como se la puede ver en la Tabla 1.1., ya que teniendo dichos rangos de parámetros el molusco puede alimentarse y crecer de manera apropiada. Dichos parámetros son los siguientes: Oxígeno disuelto, turbidez, temperatura, PH.

CARACTERÍSTICAS TÉCNICAS	VALOR DE REFERENCIA
Temperatura Promedio (°C)	25 – 35
Nivel de Acidez (PH)	6.5 – 9
Turbidez (NTU)	20
Oxígeno Disuelto (mg/litro)	5 – 8

Tabla 1.1. Condiciones necesarias del agua

Actualmente para realizar las mediciones de calidad del agua usan un oxigenómetro y un equipo llamado medidor o disco de Secchi [2] con los que se verifican los parámetros de oxígeno disuelto y la turbidez, para poder asegurar que la calidad del agua sea la apropiada para las larvas de camarón se deben considerar los demás parámetros que se mencionaron en la tabla 1.1.

Las mediciones se las realizan 2 veces al día, 1 vez en la mañana y otra en la tarde para mantener un control de los parámetros del agua y poder realizar una acción en el caso de que sea necesario.

1.1 Tasa de Mortalidad elevada

El problema radica en la falta de un completo control de los parámetros de la calidad del agua, existe una alta tasa de mortalidad de las larvas de camarón, es decir ya que el personal que labora ahí realiza las mediciones de forma

manual y solamente 2 veces al día, no son suficientes para determinar las condiciones reales del agua y por ende la salud de las larvas de camarón.

El lapso entre las mediciones ocasiona el desconocimiento de las condiciones del estado del agua lo que acarrea enfermedades críticas para las larvas que tiene como resultado una mortalidad del 25% de las larvas de una siembra total de 2'000 000, que se siembran en las 6 piscinas de la camaronera José Bravo.

A esto se agrega que los datos de las mediciones son escritos en la bitácora correspondiente de cada piscina, pero no es fehaciente ni seguro ya que se puede manipular o extraviar lo que agrava el problema de la recolección de datos.

1.2 Objetivos

1.2.1 Objetivo Principal

Diseñar un sistema de sensores para medir la calidad del agua en las piscinas de la camaronera José Bravo y el monitoreo de forma remota, a través de un control autónomo para poder reducir la tasa de mortalidad de las larvas del camarón.

1.2.2 Objetivo Específico

- Tener un control preciso de los parámetros de la calidad del agua en las piscinas para mejorar los cambios necesarios si estos varían.
- Automatizar el proceso de recolección de datos referente a la calidad del agua con la ayuda de dispositivos electrónicos para tener un mejor historial de datos.
- Establecer un monitoreo no presencial del estado del agua y que esté disponible de visualizar las 24 horas del día para tener mejor referencia en que fracción de tiempo varían los parámetros del agua.
- Diseñar un dispositivo que obtenga los datos de la calidad del agua para para realizar cambios en el mantenimiento del agua en función a los valores que obtenga.

- Establecer un método de alerta para comunicar las variaciones de la calidad del agua de las piscinas para así poder realizar cambios en el mantenimiento de cada piscina.

1.2.3 Justificación

La actividad de la acuicultura específicamente el del cultivo de larvas de camarón ofrece un crecimiento económico considerable si se tiene un correcto desarrollo de preparación, siembra, alimentación y cosecha de este, todo esto se puede lograr manteniendo los parámetros óptimos de la calidad del agua para que la larva del camarón pueda desarrollarse en el ambiente apropiado.

CAPÍTULO 2

2. SISTEMA DE SENSORES DE MEDICIÓN Y VISUALIZACIÓN REMOTA

Se propone el diseño de un sistema de sensores capaz de realizar la medición de las condiciones de la calidad del agua de las piscinas, esto con el fin de poder obtener los datos necesarios para asegurar que exista un efectivo control de las piscinas y que se pueda ejecutar periódicamente de manera automática, en la figura 2.1 se visualiza el diseño general de nuestra propuesta.

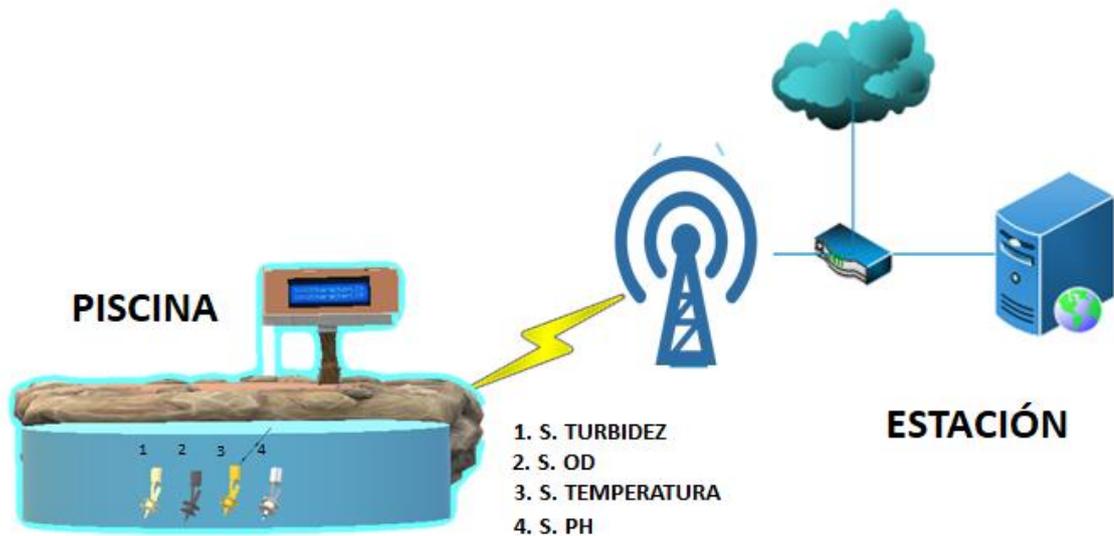


Figura 2.1. Diseño general de la solución

Se dispondrá de la colocación de dicho sistema en distintas ubicaciones de las piscinas para poder tener varias mediciones, para esto se crearán varios de este mismo sistema alcanzando el objetivo de instalarlos en todas las piscinas, tal como se lo puede observar en la Figura 2.2, con esto podremos obtener más información de la situación actual de la calidad del agua. Esta información es presentada mediante una aplicación web donde indicará el estado actual de la calidad del agua mediante los valores que presenten los sistemas de sensores ubicados en las piscinas.

Para esta solución se ha contemplado la instalación de un servidor web en la estación, el cual contendrá una página web que mostrará los valores obtenidos de los sensores para que la persona pertinente pueda revisar los resultados de manera inmediata.

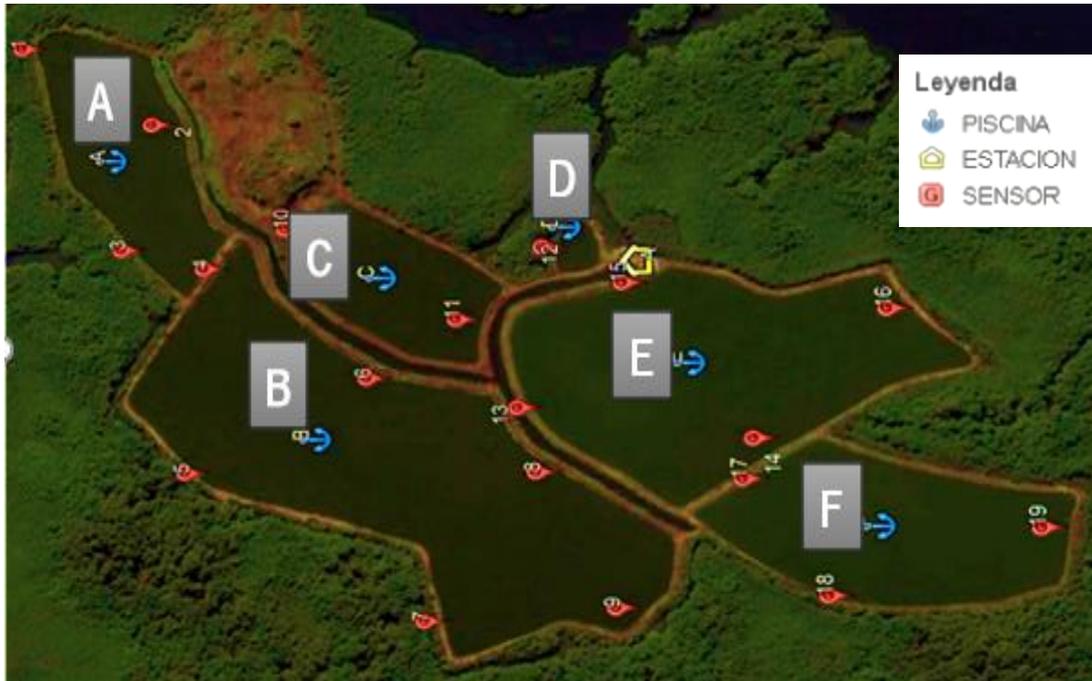


Figura 2.2. Diseño topológico sistema de sensores

La camaronera consta de 6 piscinas, por lo que se ha decidido nombrarlas por orden alfabético, además los sistemas de sensores están etiquetados por números en forma ascendente y se determina usar un total de 19 sistemas de sensores. La distribución de los sensores quedará como se muestra en la Tabla 2.1.:

Piscina	Subsistema de Monitoreo numerados
A	1,2,3
B	4,5,6,7,8,9
C	10,11
D	12
E	13,14,15,16
F	17,18,19

Tabla 2.1. Distribución de subsistema de monitoreo

Para ...tener un esquema de trabajo del contenido hemos dividido nuestra solución en 3 subsistemas que nos permitirán describir de manera óptima como funcionaria el sistema de sensores.

Nuestra solución consta de 3 subsistemas como lo son:

- Subsistema de monitoreo.
- Subsistema de aplicación web.
- Subsistema de provisión de energía.

2.1. Subsistema de monitoreo

Cada uno de los 19 sub-sistemas de monitoreo constará de cuatro sensores interconectados a un dispositivo que será el Arduino UNO, los sensores medirán los parámetros del agua: temperatura, PH, oxígeno disuelto y turbidez que son necesarios para poder hacer las mediciones dispositivos que se detallarán en los siguientes sub-títulos.

2.1.1. Sensor de Temperatura DS18B20

El dispositivo para la medición del parámetro de la temperatura del agua, consta de 2 partes, una es un transistor y otra es el sensor que obtendrá los datos del agua al sumergirlo, el sensor tiene las características que presentamos en la tabla 2.2.

CARACTERÍSTICAS TÉCNICAS	VALOR DE REFERENCIA
Voltaje de alimentación (V)	3.00 a 5.50
Corriente (A)	0.50
Potencia (W)	0.30
Rango de temperatura (°C)	-55.00 a 125.00
Protocolo de comunicación	Onewire
Tamaño de cable (m)	1.00
Costo (\$)	9.00

Tabla 2.2. Características del sensor de temperatura

2.1.2. Sensor de PH

El dispositivo que se usará para la medición del PH consta de la tarjeta controladora y del dispositivo que se introduce en el agua de la piscina, las características del sensor se muestran en la tabla 2.3.

CARACTERÍSTICAS TÉCNICAS	VALOR DE REFERENCIA
Voltaje de alimentación (V)	5.00
Corriente (A)	0.01
Potencia (W)	0.31
Rango de medición (PH)	0 – 14.00
Conector	BNC
Tamaño de cable (m)	1.00
Costo (\$)	60.00

Tabla 2.3. Característica del sensor de PH

2.1.3. Sensor de Oxígeno disuelto

Este sensor está compuesto por una tarjeta controladora y un dispositivo que se introduce en el agua, el sensor mencionado tiene las características que se muestran en la tabla 2.4.

CARACTERÍSTICAS TÉCNICAS	VALOR DE REFERENCIA
Voltaje de alimentación (V)	5.00
Corriente (A)	0.02
Potencia (W)	0.08
Rango (mg/L)	0 – 20
Protocolo de comunicación	Onewire
Tamaño de cable (m)	1.00
Costo (\$)	67.95

Tabla 2.4. Característica de sensor oxígeno disuelto

2.1.4. Sensor de Turbidez SEN0189

El sensor de turbidez está compuesto del dispositivo que se introduce en el agua de la piscina y la tarjeta controladora que tiene las características que se muestran en la tabla 2.5.

CARACTERÍSTICAS TÉCNICAS	VALOR DE REFERENCIA
Voltaje de alimentación (V)	5.00
Corriente (A)	0.04
Potencia (W)	0.20
Voltaje de Salida (V)	0 – 4.50
Digital Output	High/low level signal
Tiempo de respuesta (ms)	Menor de 500
Costo (\$)	15.00

Tabla 2.5. Característica de sensor de turbidez

Todos estos sensores estarán conectados a un protoboard ^[3] que posteriormente se reemplazará por una placa de circuito impreso el cual servirá para interconectar los elementos con el Arduino Uno, éste dispositivo tiene las siguientes características:

CARACTERÍSTICAS TÉCNICAS	VALOR DE REFERENCIA
Voltaje de alimentación (V)	5.00
Corriente (A)	0.001
Potencia (W)	0.005
Largo (mm)	68.60
Ancho (mm)	53.40
Digital I/O Pin	6.00
Analog Input Pin	6.00
Costo (\$)	10.00

Tabla 2.6. Característica de Arduino UNO

Además de los sensores, nuestro sub-sistema de monitoreo tendrá implementado una pantalla Led de 2x16, con las siguientes características:

CARACTERÍSTICAS TÉCNICAS	VALOR DE REFERENCIA
Voltaje de alimentación (V)	5.00
Corriente (A)	0.28
Potencia (W)	1.40
Resolución	2 líneas x 16 caracteres
Dimensiones (mm)	65.00 x 16.00
Temperatura de operación (°C)	-20.00 – 70.00
Peso (g)	35.00
Costo (\$)	10.00

Tabla 2.7. Característica de pantalla LED

Los datos obtenidos por nuestros sensores serán enviados al punto que hemos nombrado ESTACIÓN (figura 2.1), por medio de una conexión inalámbrica, esto se logra gracias a un módulo Wireless que se conectará al Arduino UNO. Las características de dicho módulo se presentan en el siguiente subcapítulo.

2.1.5. Módulo Wireless ESP01- ESP8266

CARACTERÍSTICAS TÉCNICAS	VALOR DE REFERENCIA
Estándar	802.11 b/g/n
Voltaje de alimentación (V)	3.30
Corriente (A)	0.21
Peso (g)	20.00
Sensibilidad de recepción (dBm)	-91.00

Potencia de transmisión (dBi)	+18.50
Frecuencia (GHz)	2.40
Costo (\$)	5.00

Tabla 2.8. Características módulo Wireless ESP8266

Este módulo nos permite conectar nuestro Arduino a nuestra red Wireless, inicialmente no había mucha documentación sobre este módulo, pero actualmente el uso de este chip ha sido muy explotado y ya podemos encontrar cualquier cantidad de información sobre él.

2.1.6. Direccionamiento IP

Para que los subsistemas de sensores de medición se puedan comunicar con el subsistema de aplicación web deben tener asignado una dirección IP en la configuración de cada subsistema de monitoreo y la Raspberry Pi, por ello a cada sensor se le asignará un nombre y una dirección IP única, teniendo la red de dirección IP 192.168.100.0/27 y la dirección IP de la puerta de enlace 192.168.100.30, como se lo puede observar en la Tabla 2.9.

NOMBRE	DIRECCION IP	MASCARA
Sub-Sist. Monitoreo 1	192.168.100.1	255.255.255.224
Sub-Sist. Monitoreo 2	192.168.100.2	255.255.255.224
Sub-Sist. Monitoreo 3	192.168.100.3	255.255.255.224
Sub-Sist. Monitoreo 4	192.168.100.4	255.255.255.224
Sub-Sist. Monitoreo 5	192.168.100.5	255.255.255.224
Sub-Sist. Monitoreo 6	192.168.100.6	255.255.255.224
Sub-Sist. Monitoreo 7	192.168.100.7	255.255.255.224
Sub-Sist. Monitoreo 8	192.168.100.8	255.255.255.224
Sub-Sist. Monitoreo 9	192.168.100.9	255.255.255.224
Sub-Sist. Monitoreo 10	192.168.100.10	255.255.255.224
Sub-Sist. Monitoreo 11	192.168.100.11	255.255.255.224
Sub-Sist. Monitoreo 12	192.168.100.12	255.255.255.224
Sub-Sist. Monitoreo 13	192.168.100.13	255.255.255.224

Sub-Sist. Monitoreo 14	192.168.100.14	255.255.255.224
Sub-Sist. Monitoreo 15	192.168.100.15	255.255.255.224
Sub-Sist. Monitoreo 16	192.168.100.16	255.255.255.224
Sub-Sist. Monitoreo 17	192.168.100.17	255.255.255.224
Sub-Sist. Monitoreo 18	192.168.100.18	255.255.255.224
Sub-Sist. Monitoreo 19	192.168.100.19	255.255.255.224
Router Tp-link	192.168.100.20	255.255.255.224
Raspberry Pi3	192.168.100.21	255.255.255.224

Tabla 2.9. Direccionamiento de IP para sensores y servidor web

2.2. Subsistema de aplicación web

Los valores medidos por nuestros sistemas de sensores se mostrarán hacia el administrador de la camaronera, por medio de una página web, en la cual estará toda la información de los parámetros las piscinas, un ejemplo de esto se puede visualizar en la Figura 2.3.

Piscina D				
	PH	Temperatura °C	OD	Turbidez
Sensor1	10	27	4	22
Sensor2	7.9	28	3	23
Sensor3	8.3	27	5	22
Sensor4	9.4	30	2	25
Sensor5	7.3	29	3	27

Figura 2.3. Cuadro de información de piscina

2.2.1. Comunicación de los sensores

Arduino nos provee de un software donde podremos configurar los sensores, este software lleva por nombre ARDUINO IDE, éste a la fecha de publicación de este proyecto va por la versión 1.8, el ambiente de desarrollo de este software es el lenguaje de programación Java, Arduino UNO tiene una licencia de software libre.

Los datos enviados por el módulo Wireless conectado al Arduino UNO son recibidos en el punto llamado ESTACIÓN (figura 2.1), por un

dispositivo que procesará la información, la organizará y presentará de manera ordenada por medio de una página web, además de enviar alertas vía correo electrónico, de incidentes ocurridos con los parámetros obtenidos de los sensores, este dispositivo estará conectado a un router y a su vez a un ISP el cual le dispondrá de acceso a internet para poder realizar dicha tarea de presentación de la información. La página web será accesible permanentemente y de esta manera se resolvería una problemática de no poder contar con los datos en cualquier momento que se lo requiera.

El dispositivo que estará en la ESTACIÓN es un Raspberry Pi 3, gracias a su diseño se lo usará como nuestro servidor web, servidor smtp y administrador de información, esto con el fin de poder separar los datos de los distintos sensores. Las características del Raspberry se muestran a continuación:

Raspberry Pi 3.

CARACTERÍSTICAS TÉCNICAS	VALOR DE REFERENCIA
Procesador	Quad Core 1.2 Ghz
Memoria RAM	1GB
Puerto Salida Video	HDMI
Puerto USB	4
Tarjeta almacenamiento	Micro SD
Protocolo	Ethernet
Conexión inalámbrica BCM43438	Wireless LAN y Bluetooth
Costo (\$)	35.00

Tabla 2.10. Características de Raspberry Pi 3

El dispositivo Raspberry estará conectado a una antena omnidireccional marca Mikrotik modelo Groove con las especificaciones que se muestran en la tabla 2.10., ubicada en la ESTACIÓN, la antena ayudará a tener una cobertura igual o superior a la del área de la camaronera y por ende todos los sensores se podrán comunicar con la ESTACIÓN. Las características de la antena omnidireccional se muestran a continuación:

CARACTERÍSTICAS TÉCNICAS	VALOR DE REFERENCIA
Voltaje de Alimentación (V)	30.00
Potencia (W)	6.00
Alimentación de energía	PoE
Peso (g)	193.00
Sensibilidad recepción	-91 dBm a 11Mbps
Potencia de transmisión (dBi)	+27.00
Ganancia de Antena (dBi)	+6.00
Frecuencia (GHz)	2.400
Protocolo	802.11 a/b/g/n
Sistema Operativo	Router OS

Tabla 2.11. Características de antena omnidireccional

Para asegurar que exista comunicación entre algún subsistema de sensor y la denominada ESTACIÓN se deben realizar ciertos cálculos que detallamos a continuación:

- **FSL**

Para verificar la pérdida de información en el espacio libre (FSL) ^[6], ver ecuación 2.1, donde considerando el punto más lejano de la piscina se corrobora que la señal se pueda enviar y recibir sin problemas.

$$FSL = 20 \log(d) + 20 \log(f) + 32.45 \quad (2.1)$$

Donde (d) está expresado en kilómetros, (f) expresado en MHz y 32.45 es una constante.

La distancia más lejana de la camaronera a la ESTACIÓN es de 0.7 km y la frecuencia a la que se va a trabajar es de 2400 MHz, reemplazando en la ecuación 2.1, se tiene como resultado:

$$FSL = 20 \log(0.7) + 20 \log(2400) + 32.45$$

$$FSL = 97.096 \text{ dB}$$

- **Margen de Potencia**

El cálculo para determinar el margen de potencia de llegada se divide en 2 ecuaciones, la ecuación 2.2 nos determina la Potencia de recepción que es un valor necesario para determinar el margen de potencia, como se puede observar en la ecuación 2.3. Los valores se pueden verificar con las características de cada dispositivo, en el emisor verificando el valor del Tx y G_{Tx} .Antena, es decir, la potencia de salida y ganancia de la antena del emisor, la pérdida en el ambiente y las características del receptor como lo es G_{Rx} .Antena o la ganancia de la antena del receptor. Para el margen es necesario saber la potencia de llegada y la sensibilidad de señal del receptor.

$$P_{Rx} = +P_{Tx} + G_{Tx}.Antena - FSL + G_{Rx}.Antena \quad (2.2)$$

$$Margen Potencia = Sensibilidad_{Rx} - P_{Rx} \quad (2.3)$$

Para el sentido de subsistema de monitor hacia la estación:

$$P_{Rx} = +18.5 \text{ dB} + 0 - 97.096 \text{ dB} + 6 \text{ dB}$$

$$P_{Rx} = -72.596 \text{ dB}$$

$$Margen Potencia = -91 - (-72.596)$$

$$Margen Potencia = -18.404 \text{ dB}$$

Para el sentido de la estación hacia el subsistema de monitoreo:

$$P_{Rx} = +27 \text{ dB} + 6 - 97.096 \text{ dB} + 0$$

$$P_{Rx} = -64.096 \text{ dB}$$

$$Margen Potencia = -91 - (-64.096)$$

$$\text{Margen Potencia} = -26.904 \text{ dB}$$

- **Zona de Fresnel**

Se deberá colocar una torre de una altura de 20 metros e instalar la antena omnidireccional para suplir esta interrupción de visibilidad debido a las elevacion de tierra que existen en el lugar y puedan comunicar entre la antena y el módulo Wireless teniendo como distancia 700 m entre ambos puntos, como se puede ver en la Figura 2.4.

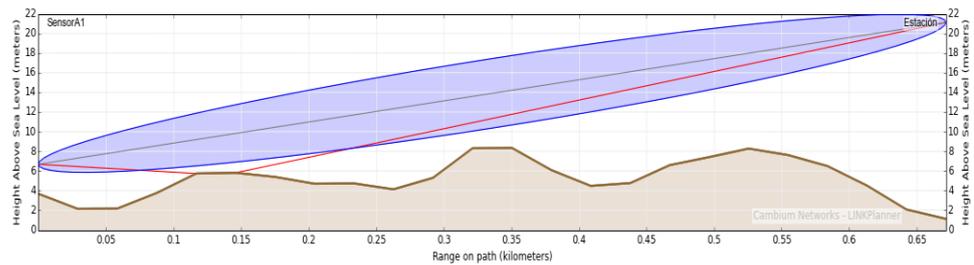


Figura 2.4. Perfil de radio enlace entre estación y Sub-sist. Monitoreo 1

Así mismo se verifica la línea de vista entre la estación y otro Sub-sistema de Monitoreo como lo es el 3 con el que se determina que hay una distancia de 500 metros para poder transmitir como está en la figura 2.5.

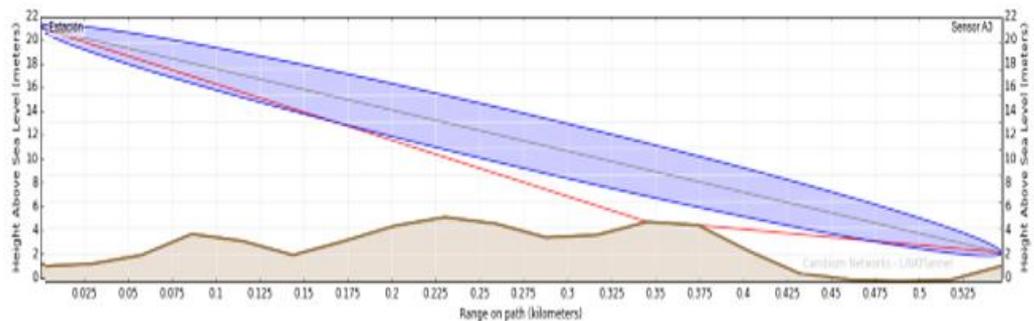


Figura 2.5. Perfil de radio enlace entre estación y Sub-sist. Monitoreo 3

Otro punto que se puede verificar es el radio enlace entre la estación y el sub-sistema de monitoreo 19 como se muestra en la Figura 2.6. que existe una elevación que es discriminante al momento de realizar la

comunicación puesto que la línea de vista entre ellos no se ve afectada ya que en el terreno no existen arboles que obstaculicen la visión.

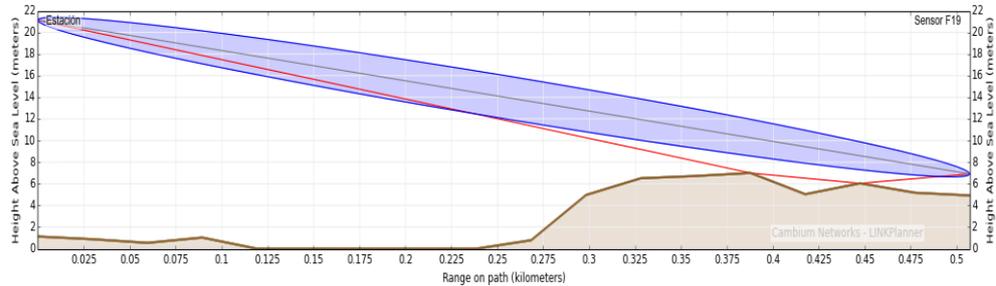


Figura 2.6. Perfil de radio enlace entre estación y Sub-sist. Monitoreo 19

Para la comunicación entre la estación y el sub-sistema de monitoreo 16 como se puede observar en la Figura 2.7, existe una elevación que no demuestra ser un problema para la línea de vista entre ellos ya que en la superficie del terreno no tiene arboles o alguna elevación adicional que obstaculice la comunicación.

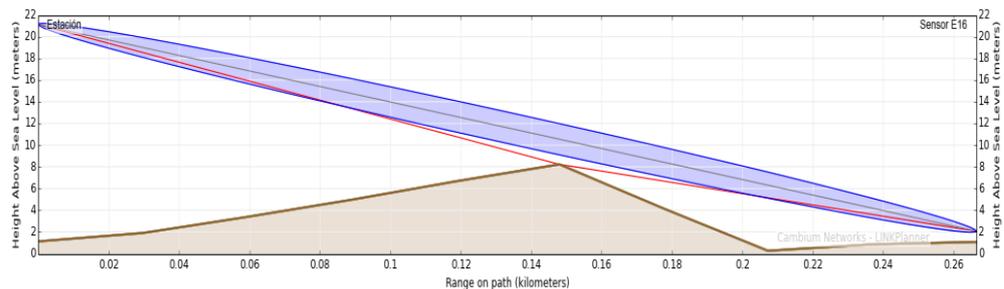


Figura 2.7. Perfil de radio enlace entre estación y Sub-sist. Monitoreo 15

2.3. Subsistema de provisión de energía

Para la alimentación de energía del subsistema de monitoreo se considerará colocar un panel solar para la alimentación de energía de los dispositivos y poder aprovechar la energía solar durante el día y en caso de no tener las condiciones suficientes se utilizará una batería de respaldo para alimentar de energía cuando no haya la energía solar mínima.

Para esto se deben tener las consideraciones de cuanta corriente necesita cada uno de los dispositivos conectados al arduino UNO.

SUB-SIST. MONITOREO MILI-AMPERIOS (mA)

Sensor turbidez	40
LCD	280
Sensor PH	10
Arduino UNO	50
Sensor Temperatura	1
Sensor O.D	20
Modulo Wifi	80

Tabla 2.12. Consumo de corriente de cada sub-sistema de monitoreo

Ya teniendo los datos de la Tabla 2.12, se puede determinar que nuestro sub-sistema de monitoreo necesita de un total de 481 mA, por motivo que nuestro sub-sistema de monitoreo usará energía solar, es necesario un panel solar que otorgue 481 mA y un voltaje de 7V.

$$P = V \cdot A \quad (2.4)$$

$$P = 7 * 0.481 = 3.37 \text{ Watts}$$

$$3.37 \text{ W} * 24 \text{ horas} = 80.8 \text{ Watts/día}$$

Con el resultado obtenido de 3.37 W de la ecuación (2.4), se conoce que el subsistema de monitoreo consume en 1 día el valor de 80.8 Watts/día.

Módulo regulador solar XL4015

Para poder regular un posible pico de corriente, en el panel solar es necesario adicionar un modulo regulador solar el cual va a regular la carga si el panel solar otorga mas de la corriente que se requiere. Las características del modulo regulador solar se presentan en la Tabla 2.13.

CARACTERÍSTICAS TÉCNICAS	VALOR DE REFERENCIA
Voltaje entrada (V)	4.0– 38.0
Voltaje ajustable salida (V)	1.25 – 36.0
Corriente de salida máx.(A)	5.0

Tabla 2.13. Características de regulador de corriente

CAPÍTULO 3

3. IMPLEMENTACIÓN DEL SISTEMA

Se puede agrupar los dispositivos de medición y comunicación para crear la implementación del sistema de monitoreo como lo pueden observar en la figura 3.1., en la que está representado cada dispositivo y como se conectaría hacia el Arduino UNO, para que puedan funcionar simultáneamente sin que llegue a existir algún inconveniente.

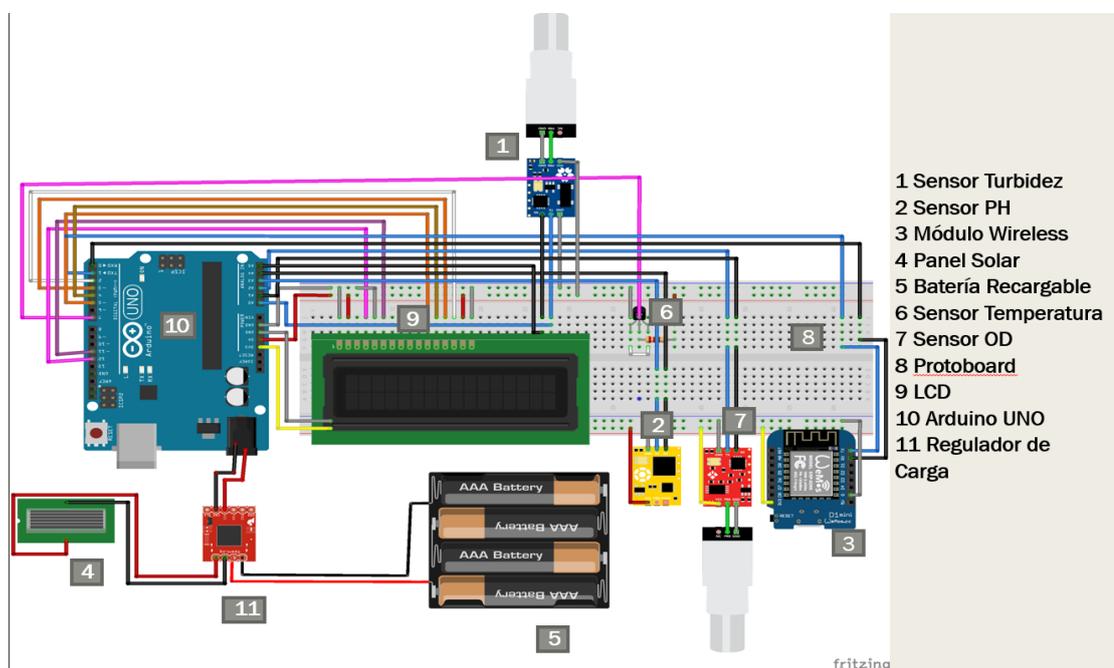


Figura 3.1. Diseño subsistema de monitoreo

Con la ayuda del circuito impreso hacemos las conexiones entre los pines del Arduino UNO y los dispositivos que actuarán de sensores, respetando los pines donde estarán conectados. Como podemos ver en la Figura 3.1, en el diseño se coloca la batería y el panel solar, esto con la finalidad de indicar que se requiere de alimentación para poder funcionar tanto los sensores como el módulo para poder comunicarnos.

3.1. Conexión de sensores

Por motivo de que nuestro proyecto es un diseño y el alcance del mismo no se contempla implementación ni ejecución del mismo, no contiene todo el código fuente de las configuraciones de cada dispositivo, se ha agregado librerías y codificación referencial para dar pautas si se desea implementar.

3.1.1. Sensor de Temperatura

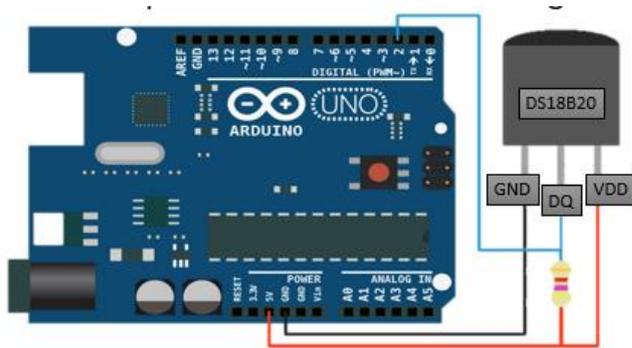


Figura 3.2. Conexión sensor hacia Arduino UNO

Este sensor de temperatura es el más indicado para poder obtener datos dentro del agua o en ambientes húmedos gracias a que la sonda de este sensor es impermeable y nos permite ingresar la misma dentro del agua, mientras que muchos sensores de temperatura no lo permiten.

Este sensor utiliza el protocolo de comunicación 1-Wire.

Pin Configurations

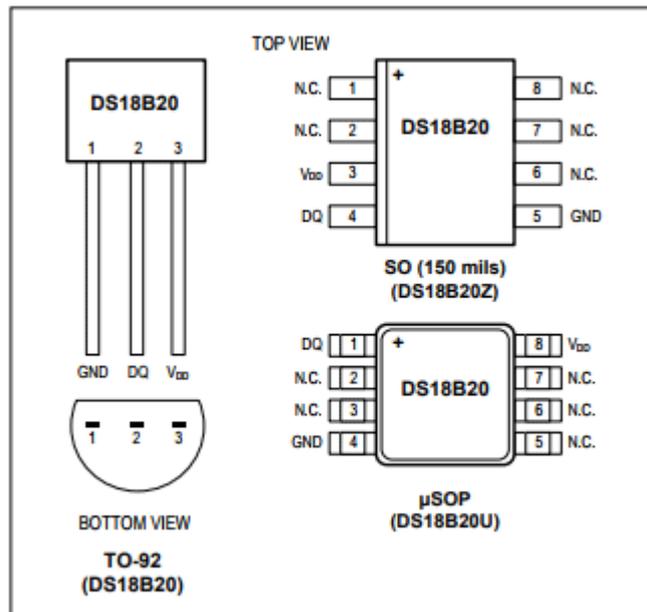


Figura 3.3. Esquemático del sensor DS18B20

Como vemos en la Figura 3.3., este sensor dispone de 3 pines, los cuales son los siguientes:

VDD: Por medio de este pin obtenemos el voltaje necesario para que el sensor funcione adecuadamente, este se alimenta entre 3V a 5.5V

GND: En este pin se conecta la toma a tierra

DQ: Tenemos este pin por el cual viajarán todos los datos por medio del protocolo 1-Wire.

El sensor tiene un rango de error para la medición de las temperaturas entre -10°C y 85°C , éste se encuentra entre $\pm 0,5^{\circ}\text{C}$, para las demás temperaturas que no se encuentren entre el rango anterior, se obtiene un margen de error de $\pm 2^{\circ}\text{C}$.

Para conectar nuestro sensor al Arduino necesitaremos una resistencia pull-up [4], ya que para controlar el bus de comunicación se utiliza un FET de apertura que funciona como una compuerta lógica AND.

Para escoger que resistencia se utilizará, dependerá del tamaño de la longitud del cable, para nuestro caso se utilizará una resistencia de 4.7

K Ω ya que nuestro cable tiene una longitud menor a 5 metros, tal como lo indican las referencias en la tabla 3.1.

RESISTENCIA PULL-UP	LONGITUD DEL CABLE (METROS)
4.7 K Ω	de 0 a 5
3.3 K Ω	de 5 a 10
2.2 K Ω	de 10 a 20
1.1 K Ω	de 20 a 50

Tabla 3.1. Referencia de resistencia con longitud de cable

Como nos muestra la figura 3.2, el cable de color negro irá conectado a nuestro pin GND, el cable color rojo irá conectado al pin que nos otorga 5V y además se unirá a un extremo de la resistencia, y por último tenemos un cable color azul que irá conectado a un pin digital de nuestro Arduino.

Para poder convertir la señal que recibió el controlador a un dato, se debe programar el siguiente código en el arduino, lo primero que tenemos que hacer es: Abrir nuestro IDE Arduino, crear un nuevo proyecto y en nuestra primera parte del proyecto tenemos que incluir las siguientes librerías:

OneWire

DallasTemperature

Para esto solo ingresamos a nuestro gestor de librerías que contiene Arduino IDE y las buscamos y procedemos a instalarlas una vez halladas.

El código^[3] de las librerías quedaría de la siguiente manera:

Arduino

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

```

2      #include <DallasTemperature.h>
3
4      // Pin donde conectamos nuestro bus 1-Wire, en
5      // nuestro caso el numero 2
6      const int pinDatos = 2;
7
8      // Instancia a las clases OneWire y
9      DallasTemperature
      OneWire oneWireObjeto(pinDatos);
      DallasTemperature
      sensorDS18B20(&oneWireObjeto);

```

Iniciamos agregando las librerías en nuestro código para poder comprender los datos que envía el controlador, la librería DallasTemperature agrega el código para el mismo, y nuestra librería onewire implementa el protocolo 1-Wire.

Se asigna a una variable el número del pin donde colocamos el sensor. Y por último se declara dos objetos con las clases DallasTemperature y también OneWire

```

3      void setup() {
4          // Iniciamos la comunicación serial
5          Serial.begin(9600);
6          // Iniciamos el bus 1-Wire
          sensorDS18B20.begin();
7      }

```

En esta función se inicia la comunicación indicando la velocidad con la que se transmite en el puerto serial. Luego se procede a iniciar 1-Wire llamando a la siguiente función sensorDS18B20 en el que no se le ingresa ningún valor.

Arduino

```

1 void loop() {
2     // Se envía comandos al controlador
3     sensorDS18B20.requestTemperatures();
4     // Leemos y mostramos los datos de los sensores
5     DS18B20
6     Serial.print("Temperatura sensor : ");
7     Serial.print(sensorDS18B20.getTempCByIndex(0));
8     Serial.println(" C");
9     delay(1000);
}

```

Se empieza utilizando la función `sensorDS18B20.requestTemperatures` desde el IDE Arduino para solicitar un dato al controlador, el controlador comprende la petición y toma una señal que receipta el sensor, luego que toma la señal el sensor lo envía al controlador, este la convierte a un dato y se lo entrega al IDE Arduino que lo procesa para hacerlo un valor de temperatura y lo muestra por pantalla.

3.1.2. Sensor de PH

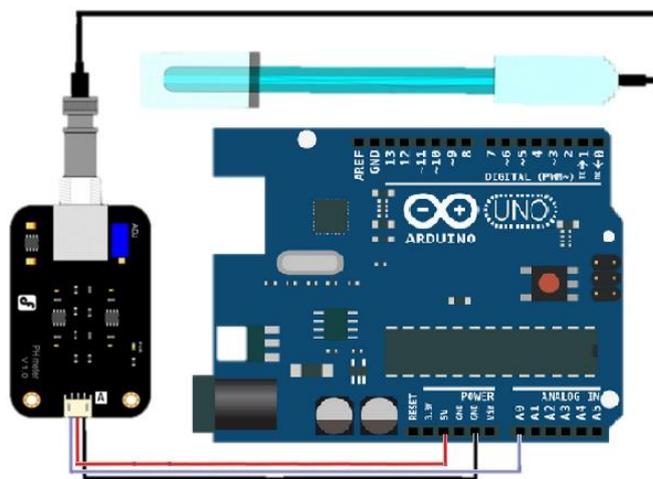


Figura 3.3. Conexión de Sensor PH hacia Arduino UNO

Para realizar la conexión tal como muestra la Figura 2.5., de nuestro sensor realizamos lo siguiente:

El cable rojo va conectado al pin de 5V, el cable negro a nuestro pin GND y por último el cable morado al pin analógico 0 de nuestro arduino.

El código por utilizar es el siguiente ^[4]:

```
// pHRead.ino
// Constants:-
const byte pHpin = A0;// conectar el pin del arduino .
// Variables:-
float Po;
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  Po = (1023 - analogRead(pHpin)) / 73.07;
  // Lee e invierte el valor de entrada analógica del sensor de pH y luego
  escala 0-14. Serial.println(Po, 2);
  // Imprime el resultado en el monitor serial.
  retraso (1000);
  // Tome 1 lectura por segundo.
  Inicializamos la constante pHpin con el numero donde hemos conectado
  nuestro sensor en el arduino.
  Declaramos la variable P0, la cual tendrá almacenado los datos leídos
  para luego proceder a presentar dichos datos.
```

3.1.3. Sensor de Oxígeno Disuelto

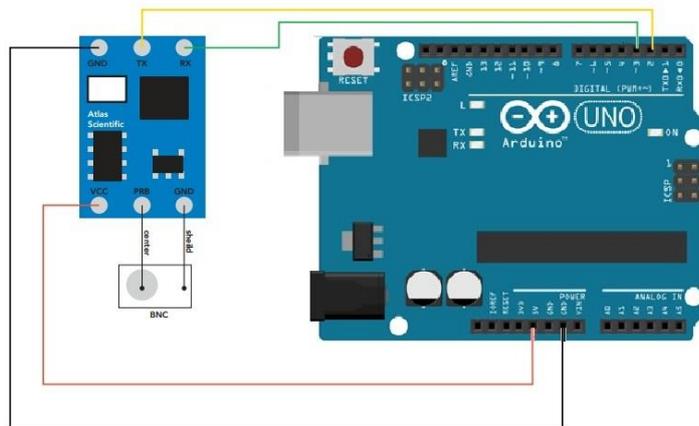


Figura 3.4. Conexión Sensor Oxígeno Disuelto hacia Arduino UNO

Nuestro sensor de oxígeno disuelto a utilizar es el modelo Atlas Scientific que nos sirve como una solución económica y fácil de usar, este sensor contiene una membrana HDPE en la sonda de oxígeno disuelto y un circuito que conecta la señal analógica que viene desde la sonda hasta el sistema que monitoreará las mediciones, en este caso sería nuestro Arduino Uno.

El contenido de esta sonda por dentro es de un tubo con una varillita de zinc metido en un electrolito

Este sensor también contiene un circuito denominado EZO que ayuda con precisión y estabilidad a la medición en el momento de realizar el su trabajo, ya que facilita al diseño evitar realizar la conversión a un valor digital de las señales analógicas que llega de la sonda.

3.1.4. Sensor de Turbidez

Para conectar nuestro sensor de turbidez será de la siguiente manera como se muestra en la figura 3.5:

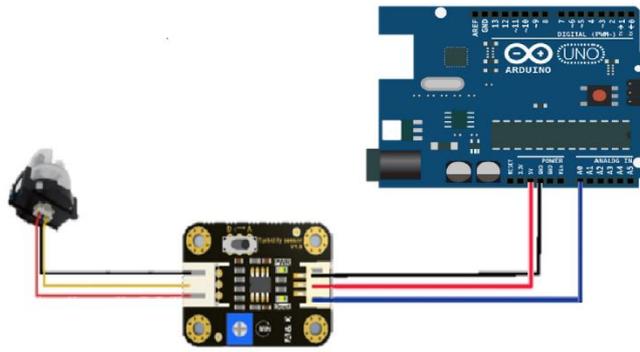


Figura 3.5. Conexión de Sensor de Turbidez hacia Arduino UNO

El color azul irá en el pin análogo 0, el cable rojo conectado al pin de 5v y por último el cable negro conectado al pin GND.

El código que se programa en el IDE Arduino para poder leer señales del sensor es relativamente corto, lo veremos a continuación ^[5]:

```
void setup() {  
  Serial.begin(9600); // Velocidad de transmisión: 9600  
}  
void loop() {  
  int sensorValue = analogRead(A0); // lea la entrada en el pin analógico  
  0:  
  float voltage = sensorValue * (5.0 / 1024.0); // Convierte la lectura  
  analógica (que va de 0 - 1023) a una tensión (0 - 5V):  
  Serial.println(voltage); // imprime el valor que lees:  
  delay(500);  
}
```

En el código anterior iniciamos en el cuerpo loop, se inicializan variables que recibirán los datos que entregue el controlador posterior de su conversión de señal analógica, se realiza un procedimiento para transformar ese dato y presentarlo en la pantalla.

3.2. Implementación de Servidor web en Raspberry Pi 3

Como ya se indicó anteriormente, nuestra Raspberry Pi tendrá implementado un servidor web en el cual contendrá una página web que nos mostrará los datos otorgados por los sensores.

Para llevar a cabo esto, debemos disponer de una memoria microSD, en nuestro caso será una de 32GB, esta capacidad es la indicada para poder almacenar datos, ya que el manejo de información solo será en texto plano no sobrecargará de información dicha memoria.

Para poder realizar la implementación del servidor web se debe de realizar los siguientes puntos:

- Descargar la imagen de Ubuntu Mate
- Grabar la imagen de Ubuntu Mate en la microSD
- Instalar Ubuntu Mate en la raspberry Pi

Descargar la imagen de Ubuntu Mate

Para poder descargar la imagen de Ubuntu Mate ^[7], debemos de dirigirnos a su página web.

Luego de acceder a la página, damos clic en la sección donde dice descarga como lo indica en la siguiente Figura 3.6.:



Figura 3.6. Descarga de Imagen Ubuntu MATE

Al acceder a la página, se nos presentará varias opciones, entre ellas un apartado que dice Raspberry PI, seleccionamos y nos llevará a una nueva página que contiene enlaces para comenzar la descarga del sistema operativo.

Grabar la imagen de Ubuntu Mate en la microSD

Para grabar la imagen en la microSD debemos descargar un programa llamado Win32 Disk Imager^[8]

Este software es muy sencillo de utilizar y nos sirve bajo Windows, nos permite instalar una imagen de un sistema operativo en una tarjeta microSD, pero hay que tener cuidado ya que, si llegamos a seleccionar una unidad equivocada, corremos el riesgo de formatear nuestro disco duro que contiene el sistema de nuestra computadora.

Una vez descargado e instalado, procedemos a ejecutar el programa, y tal como muestra la Figura 3.7, tenemos que ubicar la localización donde se encuentra nuestra imagen de Ubuntu Mate, y adicional a esto, tenemos que seleccionar la unidad a la cual vamos a formatear, luego de esto procedemos a grabar la imagen.

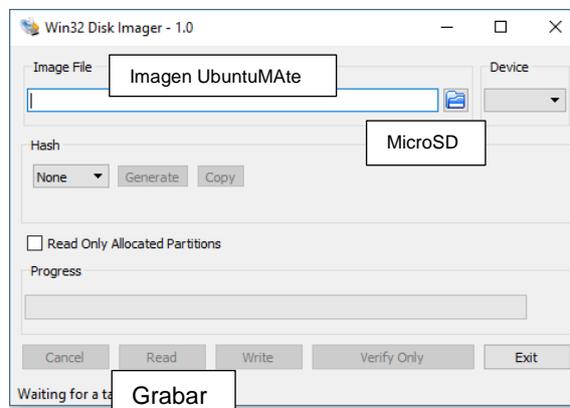


Figura 3.7. Pantalla principal Win32 Disk Imager

Instalar Ubuntu Mate en la raspberry Pi

Ya que tenemos instalada la imagen en la SD, la insertamos a nuestra Raspberry Pi y conectamos los periféricos necesarios tales como: teclado, mouse y monitor, para poder configurar nuestra raspberry.

Una vez hecho esto, encendemos nuestro raspberry y esperamos a que cargue, cuando finalice tendremos que seleccionar nuestro idioma, la

región, luego la distribución del teclado, para culminar se configura el nombre del equipo, seleccionamos en continuar, y procederá a instalar todo lo necesario y al final ya tendremos instalado nuestro sistema Ubuntu Mate como muestra la Figura 3.8:

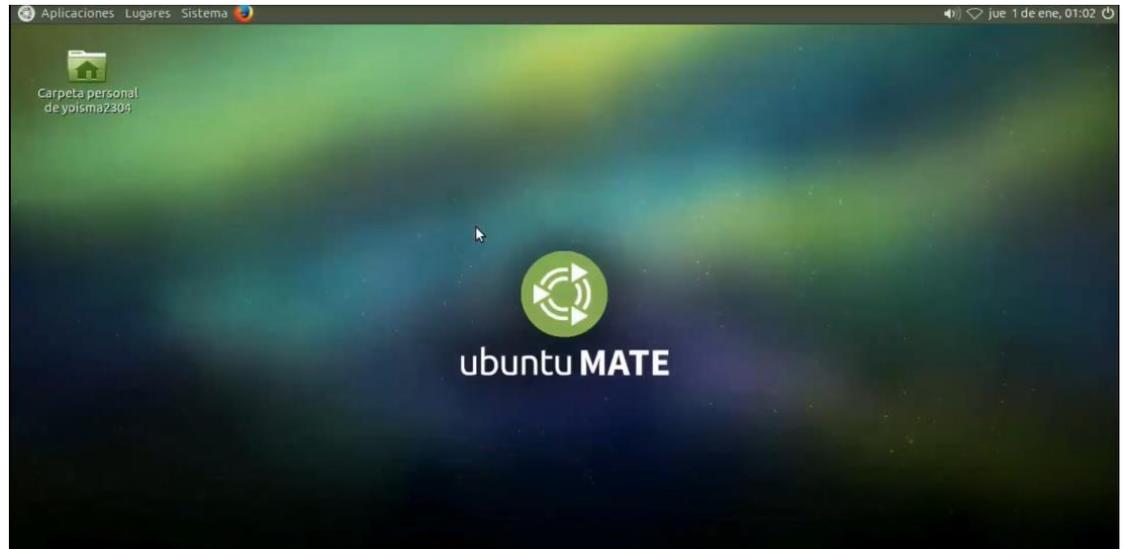


Figura 3.8. Pantalla principal Ubuntu MATE

3.3. Implementación de servidor web Apache con PHP y MySQL

Apache utiliza un grupo de usuario por defecto, al cual le daremos permisos con los siguientes comandos [9]:

- `sudo addgroup www-data`
- `sudo usermod -a -G www-data www-data`

Actualizamos repositorios e instalamos Apache y PHP

- `sudo apt-get update`
- `sudo apt-get install apache2 php5 libapache2-mod-php5`
- `sudo /etc/init.d/apache2 restart`

Continuaremos a instalar MySQL y por su puesto PHPMyAdmin, comenzaremos activando nuestra loopback, ya que si no se lo hace MySQL nos presentará un error y luego instalamos mysql y PHPMyAdmin:

- `sudo ifup lo`
- `sudo apt-get install mysql-server mysql-client php5-mysql phpmyadmin`

- Editamos el archivo: `sudo nano /etc/pHp5/apache2/pHp.ini`

Con el archivo anterior abierto, ubicamos la línea Dynamics Extensions, y anteponemos a esa línea lo siguiente:

```
extension=mysql.so
```

Guardamos y ejecutamos el siguiente comando:

- `sudo ln -s /etc/pHpmyadmin/apache.conf /etc/apache2/conf.d/pHpmyadmin.conf`
- `nano /etc/apache2/apache2.conf`

Y agregamos la siguiente línea al final del archivo:

```
Include /etc/pHpmyadmin/apache.conf
```

Reiniciamos el servicio de apache:

```
sudo /etc/init.d/apache2 reload
```

Crearemos un archivo en la ruta: `/var/www` con el nombre `testpHp.pHp` y añadimos dentro de él, la siguiente función:

```
<?pHp pHPinfo(); ?>
```

Verificamos que está correctamente, ingresando a la dirección ip de nuestro servidor en el navegador, seguido por `pHpmyadmin.pHp` y nos tiene que quedar como la Figura 3.9

`192.168.xx.xx/pHpmyadmin.pHp.`



System	Linux raspberrypi 3.6.11+ #474 PREEMPT Thu Jun 13 17:14:42 BST 2013 armv6l
Build Date	Jul 17 2013 21:59:46
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

Figura 3.9. Configuración PHP

3.4. Implementación de alertas por correo electrónico

Como se indica en uno de los objetivos del proyecto, es enviar alertas por medio de correos electrónicos cuando se suscite algún evento con los parámetros del agua de las piscinas.

Para esto vamos a implementar un cliente de correo llamado sendmail y a la vez una aplicación que nos permitirá usar nuestro mail personal, en este caso será de Gmail.

Para esta implementación ingresaremos los siguientes comandos en la raspberry, en el orden que se muestra a continuación:

```
sudo apt-get install sendmail  
sudo apt-get install ssmtp  
sudo apt-get install mailutils
```

A continuación editaremos la configuración de ssmtp con el siguiente comando:

```
sudo nano /etc/ssmtp/ssmtp.conf
```

Y al final del archivo agregamos lo siguiente y editando con nuestros respectivos datos:

```
AuthUser=miusuario@gmail.com
AuthPass=mipassword
FromLineOverride=YES
mailhub=smtp.gmail.com:587
UseSTARTTLS=YES
```

Y por último, para poder adjuntar archivos en nuestro mail instalaremos la siguiente herramienta llamada mpack, con el siguiente comando:

```
sudo apt-get install mpack
```

Para poder adjuntar los datos de los sensores primeros tendremos que exportarlos desde mysql a un archivo .csv, y lo haremos con la siguiente línea de comando:

```
mysql -u root -p mibase -e "select * from datos" -B > sensores.csv
```

Una vez hecho esto podemos utilizar nuestro cliente de correo para enviar los datos:

```
mpack -s "Envío de datos de sensores" ./sensores.csv
correoreceptor@gmail.com
```

3.5. Implementación de seguridades en el servidor web

Debemos implementar algunas seguridades ^[10] en nuestro servidor web para así poder evitar futuros ataques, las configuraciones que se establecerían serían las siguientes:

3.5.1. Cambio de directorio por defecto

Apache trae por defecto un directorio en el cual se colocan la página web, este directorio por defecto es el siguiente: “/var/www/html”, se

recomienda cambiar este directorio a uno nuevo, como lo vemos en la siguiente línea:

```
$ sudo mkdir -p /opt/www/html
```

3.5.2. Evitar búsqueda de DNS

Con esto logramos que todos los nombres de host se nos registren, para ello tenemos que realizar la siguiente configuración:

```
$ sudo nano /etc/apache2/apache2.conf → abrimos este archivo
```

```
HostnameLookups Off → editamos esta línea para que quede en Off
```

3.5.3. No mostrar versión del servidor

Para realizar esta configuración realizamos lo siguiente:

```
$ sudo nano /etc/apache2/conf-enabled/security.conf
```

Se realiza las siguientes modificaciones:

```
#Deshabilitar la firma de páginas web por el servicio Apache
```

```
ServerSignature Off
```

```
#No mostrar todos los datos del verbose mode en apache
```

```
ServerTokens Prod
```

3.5.4. Petición HTTP de seguimiento

En este caso desactivaremos estas peticiones de la siguiente manera:

```
$ sudo nano /etc/apache2/conf-enabled/security.conf
```

Realizamos estos cambios:

```
# Set to one of: On | Off | extended
```

```
TraceEnable Off
```

3.5.5. Inhabilitar las Etags

Tenemos que editar el archivo de seguridad de apache:

```
$ sudo nano /etc/apache2/conf-enabled/security.conf
```

Realizamos los siguientes cambios:

```
#Desactivar las Etags
```

```
FileETag None
```

3.5.6. Denegar el acceso al directorio raíz

Editamos lo siguiente:

```
$ sudo nano /etc/apache2/conf-enabled/security.conf
```

Y realizamos los siguientes cambios:

```
<Directory />  
Order deny,allow  
Deny from all  
Options None  
AllowOverride None  
</Directory>
```

3.5.7. Bloquear los ataques DoS HTTP

Para esto activamos “reqtimeout”:

```
$ sudo a2enmod reqtimeout
```

3.5.8. Apagar el módulo status

```
$ sudo a2dismod status
```

3.6. Configuración de modo puente en dispositivo Mikrotik

Para la configuración del router mikrotik realizaremos los siguientes pasos:

- Abrir el programa WinBox que nos permitirá administrar nuestro router Mikrotik como se muestra en la Figura 3.10:

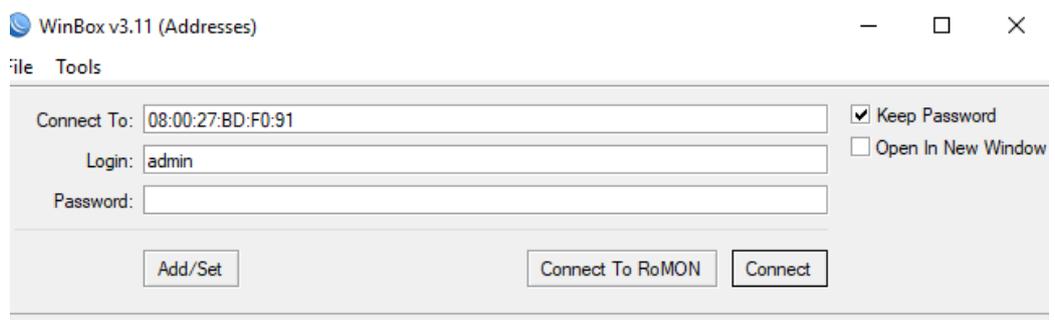


Figura 3.10. Configuración WinBox

- Debemos crear una interfaz de puente, para eso vamos a la opción Bridge y seleccionamos en Añadir como se muestra en la Figura 3.11, en esta sección dejamos todo por defecto y tan solo cambiamos el nombre de cómo se llamará el puente y luego aplicamos la configuración.

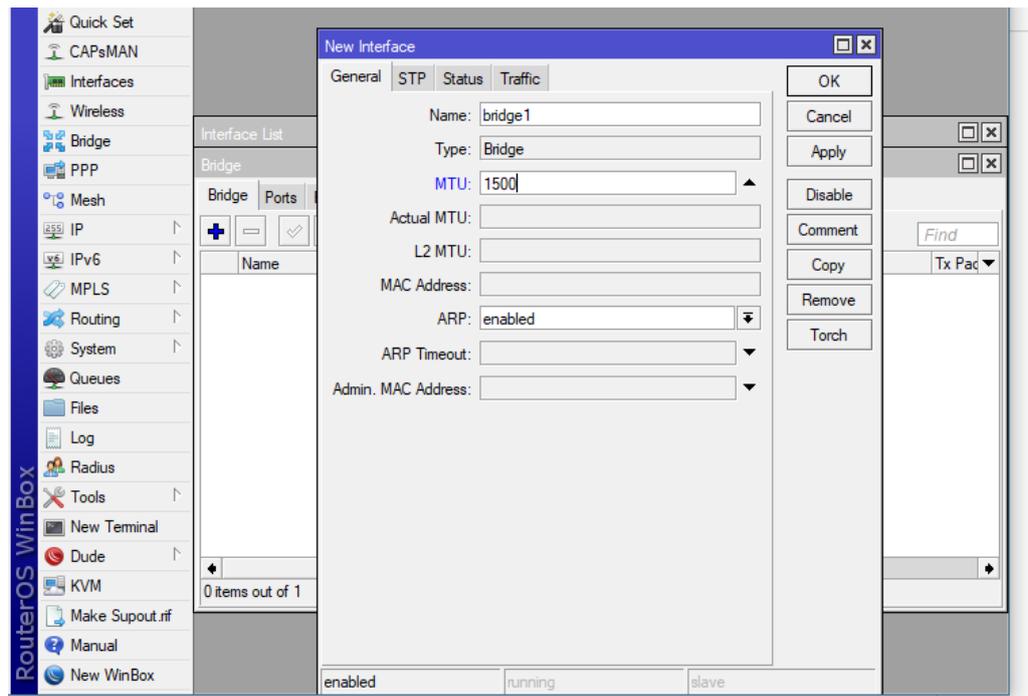


Figura 3.11. Configuración nueva interface

- Ahora configuraremos el puerto Bridge, para hacer esto ahora vamos a la pestaña Ports y seleccionamos en agregar y ahí vincularemos al bridge que hemos creado con la interfaz ethernet

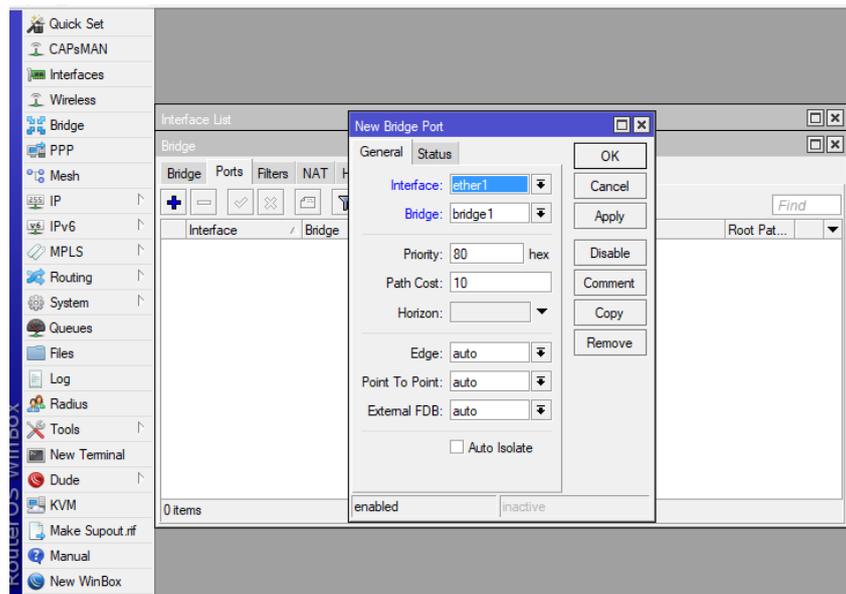


Figura 3.12. Configuración puerto bridge

3.6. Configuración de router

En la casa denominado en esta documentación como “Estación”, se implementará un router Tp-Link WR940N que se conectará a la antena que nos proveerá la empresa proveedora de internet que la detallaremos más adelante en este documento.

La configuración en este router será la de configurar su ip por la 192.168.100.20, la configuración de DNS se implementará los DNS de Google los cuales son: 8.8.8.8 y 8.8.4.4.

Por cuestiones de seguridad no se habilitará el servicio de DHCP y todas las IP que tendrán los dispositivos serán administradas manualmente.

3.7. Proveedor del servicio de internet

Para este proyecto se planificaría contratar un servicio de internet inalámbrico que provea de un plan de internet de al menos 5.0 MBps ya que debe transmitir datos en todo momento que se lo requiera, debe ser una conexión confiable, estable y que brinde soporte técnico.

En el mercado existen varios proveedores que tienen cobertura otorgando internet inalámbrico hacia el sector donde está ubicada la camaronera José Bravo y que brinden los requerimientos que necesitamos.

Los proveedores que tienen cobertura en el sector son:

- Univisa.
- PuntoNet.

Se analizó a las 2 empresas antes mencionadas, y verificamos que ambas nos ofrecen un soporte técnico inmediato, instalación del servicio en 3 días laborables, internet inalámbrico, y un mismo ancho de banda con la diferencia de que PuntoNet nos ofrece un menor costo por el plan de internet, reduciendo nuestro presupuesto.

El servicio de internet nos llegaría por medio de un enlace inalámbrico desde la radio base ubicada en la matriz de Punto Net hacia la antena que nos instalaría la empresa en la estación de nuestra camaronera. El plan que nos ofrece Punto Net está denominado "Plan Oro", el cual nos ofrece una navegación de 5Mbps con un soporte técnico 24 horas los 7 días de la semana.

CAPÍTULO 4

1. CRONOGRAMA DE TRABAJO Y PRESUPUESTO

4.1. Plan de trabajo

Para realizar este proyecto se plantea desarrollarlo en 5 etapas, las que estas divididas de la siguiente manera:

- Subsistema de monitoreo: En esta etapa realiza el proceso de adquisición de sensores luego de esto se procede con la configuración de los mismos.
- Subsistema de aplicación web: Se realiza la instalación del servidor web, la implementación de la página web e implementación de alertas por vía mail.
- Implementación y configuración de router Tp-Link y Mikrotik.
- Pruebas
- Documentación

Todo este plan de trabajo tomó 61 días, se puede ver más detallado en la siguiente Figura 4.1.:

Modo de	Nombre de tarea	Duración	Comienzo	Fin
	▸ Diseño de control de parámetros de la calidad del agua de una camaronera	61 días	mar 10/10/17	mar 2/1/18
	▸ Implementación subsistema de monitoreo	31 días	mar 10/10/17	mar 21/11/17
★	Analisis de cantidad de equipos a necesitar	2 días	mar 10/10/17	mié 11/10/17
★	Compra de dispositivos	14 días	jue 12/10/17	mar 31/10/17
★	Conexión de dispositivos	5 días	mié 1/11/17	mar 7/11/17
★	Programación de sensores	10 días	mié 8/11/17	mar 21/11/17
	▸ Implementación subsistema de aplicación web	10 días	mié 22/11/17	mar 5/12/17
★	Instalación y configuración de servidor web	1 día	mié 22/11/17	mié 22/11/17
★	Programación de página web	7 días	jue 23/11/17	vie 1/12/17
★	Configuración de sistema de alertas por mail	2 días	lun 4/12/17	mar 5/12/17
	▸ Implementación y configuración de routers	5 días	mié 6/12/17	mar 12/12/17
★	Configuración de router Tp-Link	2 días	mié 6/12/17	jue 7/12/17
★	Configuración de router Mikrotik	3 días	vie 8/12/17	mar 12/12/17
	▸ Pruebas	5 días	mié 13/12/17	mar 19/12/17
★	Calibracion de Sensores	3 días	mié 13/12/17	vie 15/12/17
★	Pruebas de medición con diferentes muestras	2 días	lun 18/12/17	mar 19/12/17
★	▸ Documentación	10 días	mié 20/12/17	mar 2/1/18

Figura 4.1. Plan de trabajo

4.2. Presupuesto

Para ejecutar el proyecto que solucionará la problemática de la camaronera, se deberán solventar los gastos que se presentan en el cuadro a continuación:

Cantidad	Nombre	Valor Unitario	Valor Total
19	Sensor de Temperatura	\$9,00	\$171,00
19	Arduino UNO	\$10,00	\$190,00
19	Sensor de Turbidez	\$15,00	\$285,00
19	Display LCD	\$4,00	\$76,00
19	Sensor de pH	\$60,00	\$1.140,00
19	Batería	\$80,00	\$1.520,00
19	Panel solar	\$10,00	\$190,00
19	Modulo Wireless	\$15,00	\$285,00
19	Sensor oxígeno disuelto	\$67,95	\$1.291,05
1	Antena Routerboard Mikrotik	\$69,00	\$69,00
1	Raspberry Pi	\$35,00	\$35,00
1	Router Tp-Link	\$30,00	\$30,00
1	Plan de internet 5mb anual	\$321,60	\$321,60
1	Mano de obra	\$1.000,00	\$1.000,00
	TOTAL		\$6.603,65

Tabla 4.1. Costos de sistema

Como podemos observar, el costo total a invertir en el proyecto será de \$6603.65.

CONCLUSIONES Y RECOMENDACIONES

➤ Conclusiones

Existe una gran cantidad de dispositivos, como sensores que realizan la misma acción con una diferencia de precisión que va acompañada del costo de éste, es decir, a mayor precisión para entregar datos también es mayor el costo de adquirir el dispositivo.

En internet se puede encontrar mucha codificación para poder realizar pruebas de funcionamiento con los sensores que seleccionamos y de esta manera poder realizar una aplicación personalizada de manera ágil y fácil.

Este sistema de sensores se puede adaptar a otros ambientes de acuicultura mientras estos requieran tener los mismos datos que se obtienen en este proyecto.

➤ Recomendaciones

Se recomienda cada 6 meses dar mantenimiento a los sensores realizando un cambio de membrana a los mismos, ya que estos se desgastan con el uso constante.

Se recomienda modificar la tecnología de comunicación entre los subsistemas de monitoreo y la estación, es decir, usar la tecnología GSM en el módulo de conexión inalámbrico y usar el servicio SMS de un operador de telefonía móvil para el envío de las alertas.

BIBLIOGRAFÍA

- [1] Anónimo, "Litopenaeus vannamei", Es.m.wikipedia.org, 2018. [Online]. Available: https://es.m.wikipedia.org/wiki/Litopenaeus_vannamei.
- [2] N. Rodriguez, "DISCO DE SECCHI", Clubdecienciaslmgga.blogspot.com, 2018. [Online]. Available: <http://clubdecienciaslmgga.blogspot.com/2014/08/disco-de-secchi.html?m=1>.
- [3] Luis del Valle Hernández. ¿. soy?, "DS18B20 sensor de temperatura para líquidos con Arduino", Programar fácil con Arduino, 2018. [Online]. Available: <https://programarfácil.com/blog/arduino-blog/ds18b20-sensor-temperatura-arduino/>.
- [4] Anónimo, "Cómo medir pH desde tu laptop en tiempo real con Arduino?", gidahatari, 2018. [Online]. Available: <http://gidahatari.com/ih-es/como-medir-ph-desde-tu-laptop-en-tiempo-real-con-arduino>.
- [5] F. Course, A. Tutorials and A. Shield, "Tutorial 09: Reading Analog Pins and Converting the Input to a Voltage - Programming Electronics Academy", Programming Electronics Academy, 2018. [Online]. Available: <https://programmingelectronics.com/tutorial-09-reading-analog-pins-and-converting-the-input-to-a-voltage/>.
- [6] pHoebe-click, "MICROWAVE COMMUNICATION PowerPoint Presentation, PPT - DocSlides", DocSlides.com, 2018. [Online]. Available: <http://www.docslides.com/microwave-communication>.
- [7] M. Wimpress, "Ubuntu MATE for the Raspberry Pi 2 and Raspberry Pi 3", Ubuntu MATE, 2018. [Online]. Available: <https://ubuntu-mate.org/raspberry-pi/>.
- [8] "Win32 Disk Imager", SourceForge, 2018. [Online]. Available: <http://sourceforge.net/projects/win32diskimager/>.
- [9] Alejandro Esquivá Rodríguez, "Tutorial Raspberry Pi – 15. Instalación de Apache + MySQL + PHP", Geekytheory.com, 2018. [Online]. Available: <https://geekytheory.com/tutorial-raspberry-pi-15-instalacion-de-apache-mysql-php>.
- [10] Esaú A., "8 Consejos de seguridad servidores Apache", OpenWebinars, 2018 [Online]. Available: <https://openwebinars.net/blog/consejos-seguridad-servidores-apache/>

ANEXOS

DataSheet modulo Oxígeno Disuelto.

https://www.atlas-scientific.com/files/datasheets/circuit/DO_EZO_Datasheet.pdf

DataSheet sensor DS18B20

<https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>

DataSheet Arduino Uno

<https://www.farnell.com/datasheets/1682209.pdf>

DataSheet Raspberry PI

<http://docs-europe.electrocomponents.com/webdocs/14ba/0900766b814ba5fd.pdf>