



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
CARRERA DE INGENIERÍA EN ELECTRÓNICA Y REDES DE COMUNICACIÓN

“SISTEMA ELECTRÓNICO BASADO EN CRITERIOS DE APRENDIZAJE DE MÁQUINA, PARA EL RECONOCIMIENTO DE PARÁMETROS DE CALIDAD DEL AGUA, EN EL CULTIVO DE TRUCHAS ARCO IRIS DEL CRIADERO TRUCHAS AGUAS VERDES”

TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN ELECTRÓNICA Y REDES DE COMUNICACIÓN

AUTOR: EDISON JAVIER PASPUEZAN MIRA

DIRECTOR: MSC. JAIME ROBERTO MICHILENA CALDERÓN

Ibarra- Ecuador

2023



UNIVERSIDAD TÉCNICA DEL NORTE

BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

IDENTIFICACION DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DEL CONTACTO		
CEDULA DE IDENTIDAD:	0401868112	
APELLIDOS Y NOMBRES:	Paspuezán Mira Edison Javier	
DIRECCION	Ibarra	
E-MAIL	ejpaspuezan@utn.edu.ec	
TELEFONO FIJO		TELEFONO MOVIL 0968439323
DATOS DE LA OBRA		
TÍTULO:	“Sistema electrónico basado en criterios de aprendizaje de máquina, para el reconocimiento de parámetros de calidad del agua, en el cultivo de truchas arco iris del Criadero Truchas Aguas Verdes”	
AUTOR:	Paspuezán Mira Edison Javier	
FECHA:	19 de abril del 2023	
PROGRAMA:	Pregrado	
TITULO POR EL QUE OPTA:	Ingeniero en Electrónica y Redes de Comunicación	
DIRECTOR:	MSc. Jaime Roberto Michilena calderón	

CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de esta y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros

Ibarra, a los 19 días del mes abril de 2023

EL AUTOR

A handwritten signature in blue ink, appearing to read 'JAVIER PASPUEZAN MIRA EDISON', with a stylized flourish underneath.

Paspuezán Mira Edison Javier

040186811-2

CERTIFICACIÓN

MSc. JAIME ROBERTO MICHILENA CALDERÓN, DIRECTOR
DEL PRESENTE TRABAJO DE TITULACION CERTIFICA:

Que, el presente trabajo de Titulación "SISTEMA ELECTRÓNICO BASADO EN CRITERIOS DE APRENDIZAJE DE MÁQUINA, PARA EL RECONOCIMIENTO DE PARÁMETROS DE CALIDAD DEL AGUA, EN EL CULTIVO DE TRUCHAS ARCO IRIS DEL CRIADERO TRUCHAS AGUAS VERDES", ha sido desarrollado por el señor Edison Javier Paspuezán Mira, bajo mi supervisión.

Es todo en cuanto puedo certificar en honor a la verdad.



.....
MSc. Jaime Roberto Michilena Calderón

CC: 100219843-8

DIRECTOR

DEDICATORIA

A mis padres Jorge Paspuezan y Elsa Mira, por ser mi ejemplo para seguir, quienes me demostraron que con mucho esfuerzo, dedicación y sacrificio se puede llegar a obtener muchos éxitos en la vida, este logro es de ustedes.

A mis hermanos Edwin y Diego que de una u otra forma me han apoyado para que pueda ser un profesional y llegue a tener muchas metas.

A mis sobrinos Melany y Mateo les dedico este trabajo para que nunca se rindan y que con esfuerzo y dedicación lleguen hacer muchas cosas en la vida.

A mi tía Alejandrina, ya que por su cariño y amor que me brindo durante toda mi vida me motivo a seguir siempre adelante y sé que desde cualquier lugar en el cielo va a estar muy orgullosa de verme como un profesional.

Y al resto de mi familia, les dedico con mucho cariño todo mi trabajo.

Edison Javier Paspuezan Mira

AGRADECIMIENTO

A Dios, por brindarme la oportunidad de vivir y estar siempre guiando mi camino, a mis padres Jorge y Elsa, quienes creyeron en mi con su apoyo incondicional, sus consejos, enseñanzas y por la motivación que siempre me dieron hasta que finalice mi carrera como profesional.

A mis hermanos Edwin y Diego, que demostraron su preocupación y estuvieron siempre apoyándome en todo momento que necesitaba.

A ti Cindy, que a lo largo de mi carrera fuiste un apoyo incondicional en cada paso que daba, con tus consejos y motivación que siempre me guiaban a elegir lo mejor y el camino correcto para poder realizar bien las cosas.

A mis amigos y compañeros que a lo largo de toda la carrera universitaria estuvieron en los buenos y malos momentos, con todo su apoyo incondicional y los buenos consejos que perduraran de por vida.

Al Ingeniero Jaime Michilena, director de este Trabajo de Grado, por brindarme su apoyo, tiempo y conocimientos que me guiaron para poder terminar con éxito mi carrera.

A los Ingenieros Edgar Maya y Fabian Cuzme, asesores de mi proyecto de titulación, que con su tiempo y colaboración me ayudaron para finalizar de la mejor manera mi trabajo de grado.

A la Universidad Técnica del Norte juntamente con la carrera de Electrónica y Redes de comunicación por abrirme las puertas y permitirme formar ética y profesionalmente.

A todos, muchas gracias por ayudarme a cumplir este logro tan importante y significativo en mi vida.

Edison Javier Paspuezan Mira

RESUMEN

El objetivo principal de este estudio es la realización de un sistema de monitoreo de parámetros de calidad del agua. Para esto se ha realizado el proyecto en cuatro capítulos los cuales detallan todo el proceso de diseño, desarrollo e implementación. Y con ello evidenciar los beneficios que se pueden obtener en la producción de truchas con la instalación de este equipo en el criadero.

En el primer capítulo se realiza una inducción para establecer los parámetros en los que se fundamentará el proyecto, además, se detalla el alcance y una descripción de cómo se desarrollará con la implementación de componentes de electrónica básica, programación de tarjetas microcontroladoras y manejo de datos. La justificación presenta la posibilidad de una mejora en la producción de las truchas.

Dentro del siguiente capítulo se realiza la especificación de los parámetros que intervienen directamente en el cultivo de truchas tipo arcoíris, su producción y venta. La manera en cómo se va a constituir, su método de desarrollo, hardware, software y tipo de comunicación que hará a sistema funcionar.

En el tercer capítulo se presenta el proceso de diseño donde se precisan los requerimientos del usuario para el desarrollo. La selección de software y hardware con la comparativa de diferentes alternativas del mercado. Adicionalmente se presenta el diseño de placas y la interfaz gráfica que permite visualizar los datos que se obtienen con su montaje e instalación de los gabinetes en los lugares predeterminados del criadero.

En el último capítulo se desarrollan las pruebas pertinentes del sistema en base a las especificaciones y requerimientos que se establecieron en capítulos anteriores. Los resultados de cada prueba son detallados y analizados con comparativas del comportamiento del criadero antes y después de la implementación del sistema que este proyecto desarrolló.

ABSTRACT

The main objective of this study is the realization of a monitoring system for water quality parameters. For this purpose, the project has been divided into four chapters which detail the entire process of design, development, and implementation. And with this, the benefits that can be obtained in trout production with the installation of this equipment in the hatchery are made evident.

In the first chapter there is an induction to establish the parameters on which the project will be based, in addition, the scope and a description of how it will be developed with the implementation of basic electronic components, microcontroller card programming and data management. The justification presents the possibility of an improvement in trout production.

The following chapter specifies the parameters directly involved in rainbow trout farming, production, and sale. The way in which it is going to be constituted, its development method, hardware, software, and type of communication that will make the system work.

The third chapter presents the design process where the user's requirements for the development are specified. The selection of software and hardware with the comparison of different alternatives in the market. Additionally, the design of the boards and the graphic interface that allows visualizing the data obtained with the assembly and installation of the cabinets in the predetermined places of the hatchery are presented.

In the last chapter, the pertinent tests of the system are developed based on the specifications and requirements established in previous chapters. The results of each test are detailed and analyzed with comparative data on the behavior of the hatchery before and after the implementation of the system that this project developed.

CONTENIDO

IDENTIFICACION DE LA OBRA.....	1
CONSTANCIAS.....	¡Error! Marcador no definido.
CERTIFICACIÓN	2
DEDICATORIA	4
AGRADECIMIENTO	5
RESUMEN.....	6
ÍNDICE DE TABLAS	13
ÍNDICE DE FIGURAS.....	16
CAPÍTULO I: ANTECEDENTES	22
1.1. Problema.	22
1.2. Objetivos.....	24
1.2.1. Objetivo General.....	24
1.2.2. Objetivos específicos.	24
1.3. Alcance.	25
1.4. Justificación.	27
2. CAPÍTULO II: MARCO TEÓRICO	29
2.1. Acuicultura.....	29
2.1.1. Estructuras acuícolas.....	29
2.1.2. Clasificación de la acuicultura	30
2.1.2.1. Clasificación por el ambiente.....	31
2.1.2.2. Clasificación por el nivel de intensidad.	31
2.1.2.3. Clasificación por el número de especies.	34
2.1.2.4. Clasificación por el tipo de especie.....	35
2.2. Piscicultura.....	35

2.1.3. Piscicultura de producción en Ecuador.....	36
2.1.4. Trucha.....	36
2.1.4.1. Trucha arcoíris.....	37
2.1.4.2. Ciclo de vida de la trucha.....	38
2.1.5. Calidad del agua.....	39
2.1.5.1. Factores Químicos.....	39
2.1.5.2. Ejemplo de parámetros de cultivo de truchas arco iris.....	40
2.2. IoT.....	41
2.2.1. Protocolos IoT.....	41
2.2.1.1. OPC.....	42
2.2.1.2. CoAP.....	42
2.2.1.3. MQTT.....	43
2.2.1.4. Modbus.....	43
2.3. Placas de desarrollo.....	50
2.3.1. Arduino.....	50
2.3.1.1. Tipos de Arduino.....	51
2.3.1.2. Módulos / Shields.....	51
2.4. Aprendizaje de Maquina.....	54
2.4.1. Aprendizaje supervisado.....	55
2.4.2. Aprendizaje no supervisado.....	55
2.4.3. Aprendizaje por refuerzo.....	56
2.4.3.1. Algoritmos de Clasificación.....	56
2.4.3.2. Clasificación binaria.....	57
2.4.3.3. Clasificación multi clase.....	58
2.4.3.4. Clasificación en la práctica.....	58
2.4.3.5. Weka.....	59
2.5. Metodologías de Gestión de Proyectos.....	60
2.5.1. Tipos de Metodologías de Gestión de Proyectos.....	60
2.5.1.1. Metodología Cascada.....	60
2.5.1.2. Metodología de Cadena Crítica.....	62

2.5.1.3. Metodología Ágil.	63
2.5.1.4. Metodología de Híbrida.	63
3. CAPÍTULO II: DESARROLLO	64
3.1. Selección de metodología para el desarrollo	64
3.2. Metodología de Cascada: Fase 1 Análisis de Requisitos	64
3.2.1. Situación inicial	64
3.2.2. Técnicas de recolección de información.....	65
3.2.2.1. Entrevista.....	65
3.2.2.2. Resultados de la entrevista.	65
3.2.3. Propósito y ámbito del sistema	67
3.2.4. Descripción del sistema	68
3.2.4.1. Riesgos.	68
3.2.4.2. Restricciones	69
3.2.5. Determinación de Stakeholders	69
3.2.6. Definición de Acrónimos y Abreviaturas	70
3.3. Metodología de Cascada: Fase 2 Diseño	70
3.3.1. Requerimientos operacionales y de usuario.....	70
3.3.2. Requerimientos iniciales del sistema	71
3.3.3. Requerimientos iniciales de arquitectura	73
3.3.4. Selección de Software.....	74
3.3.5. Elección de Hardware	75
3.3.5.1. Tarjeta microcontroladora.	76
3.3.5.2. Sensor de temperatura.	78
3.3.5.3. Sensor de pH.	80
3.3.5.4. Interruptor de nivel.....	82
3.3.5.5. Módulo de comunicación.	83
3.3.5.6. Sensor de oxígeno disuelto.....	85
3.3.5.7. Módulo de almacenamiento.	87
3.3.5.8. Módulo de tiempo real.	89
3.3.5.9. Relé.....	91

3.3.5.10. Pantalla.	93
3.3.5.11. Alimentación.	95
3.4. Metodología de Cascada: Fase 3 Implementación.....	95
3.4.1. Diagramas de flujo.....	96
3.4.1.1. Nodo situado en la entrada de agua al criadero.....	96
3.4.1.2. Nodo situado en la piscina de crecimiento.....	98
3.4.1.3. Nodo central	100
3.4.2. Generación de la programación	102
3.4.2.1. Librerías	102
3.4.2.2. Código de programación	103
3.4.3. Esquema general de conexiones	103
3.4.3.1. Circuitos de distribución	105
3.4.3.2. Interfaz	106
3.4.4. Descripción de funcionamiento del sistema	107
3.4.4.1. Primera etapa.....	107
3.4.4.2. Segunda etapa.....	109
3.4.4.3. Tercera etapa	111
3.4.4.4. Cuarta etapa.....	112
3.4.4.5. Quinta etapa.....	113
3.4.4.6. Sexta etapa.....	114
3.4.4.7. Séptima etapa	115
3.4.4.8. Octava etapa	117
3.4.4.9. Novena etapa.....	117
3.4.5. Montaje y construcción.....	120
3.4.6. Instalación.....	129
3.4.6.1. Entornos	129
3.4.6.2. Layout de instalación	130
3.4.7. Calibración y manual de uso.....	132
3.4.7.1. Calibración de sensores	132
3.4.7.2. Calibración por software	132
3.4.7.3. Calibración de hardware.....	132

3.4.7.4. Manual de usuario	133
4. CAPÍTULO IV: PRUEBAS DE FUNCIONAMIENTO	134
4.1. Metodología de Cascada: Fase 4 Verificación.....	134
4.2. Pruebas	134
4.3. Resultados	136
4.3.1. Resultados de la Fase 1 y Fase 2.....	136
4.3.2. Resultados de la Fase 3	136
4.3.3. Resultados de la Fase 4.....	137
4.3.4. Resultados de la Fase 5	138
4.3.5. Resultados de la Fase 6.....	139
4.3.6. Resultados de la Fase 7	140
4.3.7. Resultados de la Fase 8 y Fase 9.....	141
4.3.8. Resultados de la Fase 10.....	142
4.3.9. Resultados de la Fase 11	148
4.3.9.1. Mejoras evidenciadas en base a la entrevista	148
4.3.9.2. Comparativa de resultados	149
CONCLUSIONES Y RECOMENDACIONES.....	155
Conclusiones	155
Recomendaciones.....	157
REFERENCIAS	159
ANEXOS	165

ÍNDICE DE TABLAS

Tabla 1.....	66
<i>Tabla resumen de información obtenida por la entrevista realizada al encargado.</i>	66
Tabla 2.....	67
<i>Condiciones actuales del criadero y su producción</i>	67
Tabla 3.....	69
<i>Stakeholders.</i>	69
Tabla 4.....	70
<i>Lista de Acrónimos y Abreviaturas.</i>	70
Tabla 5.....	71
<i>Requerimientos operacionales del sistema.</i>	71
Tabla 6.....	71
<i>Requerimientos de usuario del sistema.</i>	71
Tabla 7.....	72
<i>Requerimientos del sistema.</i>	72
Tabla 8.....	73
<i>Requerimientos de estructura.</i>	73
Tabla 9.....	74
<i>Tabla comparativa de las alternativas de software.</i>	74
Tabla 10.....	76
<i>Tabla comparativa de las alternativas de tarjeta micro controlada.</i>	76
Tabla 11.....	78
<i>Tabla comparativa de las alternativas de sensor de temperatura.</i>	78
Tabla 12.....	80

<i>Tabla comparativa de las alternativas de sensor de pH.</i>	80
Tabla 13.....	81
<i>Características del sensor de pH.</i>	81
Tabla 14.....	82
<i>Tabla comparativa de las alternativas para interruptor de nivel.</i>	82
Tabla 15.....	83
<i>Características del interruptor de nivel.</i>	83
Tabla 16.....	84
<i>Tabla comparativa de las alternativas para el módulo de comunicación modbus.</i>	84
Tabla 17.....	84
<i>Características del módulo Modbus RS485.</i>	84
Tabla 18.....	85
<i>Tabla comparativa de las alternativas para el sensor de oxígeno disuelto en líquidos. ..</i>	85
Tabla 19.....	86
<i>Características del sensor de oxígeno disuelto.</i>	86
Tabla 20.....	87
<i>Características del módulo de almacenamiento extraíble.</i>	87
Tabla 21.....	88
<i>Características del módulo microSD.</i>	88
Tabla 22.....	89
<i>Características del módulo de tiempo real.</i>	89
Tabla 23.....	90
<i>Características del módulo de tiempo real DS3231.</i>	90
Tabla 24.....	91

<i>Características del módulo del relé.</i>	91
Tabla 25.....	92
<i>Características del módulo relé 12V.</i>	92
Tabla 26.....	93
<i>Características de las pantallas.</i>	93
Tabla 27.....	94
<i>Características de la pantalla LCD 128x64.</i>	94
Tabla 28.....	134
<i>Descripción de las pruebas de funcionamiento para el sistema de monitoreo.</i>	134
Tabla 29.....	139
<i>Tabla de alarmas.</i>	139
Tabla 30.....	148
<i>Aspectos y parámetros de mejora en la entrevista realizada al fundador del criadero.</i>	148
Tabla 31.....	149
<i>Comparativa de valores antes y después del implementado el sistema de monitoreo y adecuaciones.</i>	149

ÍNDICE DE FIGURAS

Figura 1	32
<i>Embalse de agua idóneo para un sistema extensivo.</i>	32
Figura 2	33
<i>Embalses cubiertos semi intensivos para mayor producción.</i>	33
Figura 3	33
<i>Piscinas cubiertas y controladas en sistemas intensivos de acuicultura.</i>	33
Figura 4	37
<i>Fisiología de una trucha arcoiris.</i>	37
Figura 5	45
<i>Estructura maestro – esclavo.</i>	45
Figura 6	46
<i>Estructura de una pila de comunicación Modbus.</i>	46
Figura 7	47
<i>Estructura de una red de arquitectura Modbus.</i>	47
Figura 8	52
<i>Shield de sensores compatible con arduino.</i>	52
Figura 9	53
<i>Shields de actuadores compatibles con arduino.</i>	53
Figura 10	54
<i>Shield de comunicación compatible con arduino.</i>	54
Figura 11	57
<i>Ejemplo de clasificación binaria.</i>	57
Figura 12	58

<i>Ejemplo de clasificación de diferentes clases.</i>	58
Figura 13	61
<i>Diagrama de los pasos de la metodología de cascada.</i>	61
Figura 14	75
<i>Interfaz de la IDE de Arduino.</i>	75
Figura 15	77
<i>Arduino UNO R3.</i>	77
Figura 16	78
<i>Arduino MEGA.</i>	78
Figura 17	79
<i>DS18B20 Sensor de temperatura.</i>	79
Figura 18	81
<i>Sensor de pH con electrodo E 201-C.</i>	81
Figura 19	83
<i>Interruptor de nivel de agua.</i>	83
Figura 20	85
<i>Módulo Modbus RS485 para compatible con arduino.</i>	85
Figura 21	87
<i>Kit de sensor de oxígeno disuelto.</i>	87
Figura 22	88
<i>Módulo de almacenamiento en microSD.</i>	88
Figura 23	91
<i>Módulo RTC DS3231.</i>	91
Figura 24	93

<i>Módulo de relé 12V.</i>	93
Figura 25	94
<i>Pantalla LCD 128x64.</i>	94
Figura 26	95
<i>Adaptador de voltaje de 110V AC a 9V DC.</i>	95
Figura 27	96
<i>Diagrama de flujo del nodo ubicada en la entrada de agua.</i>	96
Figura 28	98
<i>Diagrama de flujo del nodo ubicada en la piscina de crecimiento de truchas.</i>	98
Figura 29	100
<i>Diagrama de flujo del nodo central.</i>	100
Figura 30	103
<i>Esquema de componentes del sistema de monitoreo.</i>	103
Figura 31	105
<i>Circuito impreso de distribución de voltajes, conexión del sensor de temperatura y el sensor de nivel de agua.</i>	105
Figura 32	106
<i>Circuito impreso de distribución de voltajes, conexión del sensor de temperatura.</i>	106
Figura 33	106
<i>Interfaz de las mediciones de las estaciones externas.</i>	106
Figura 34	107
<i>Matriz de bits para graficar en pantalla.</i>	107
Figura 35	108
<i>Diagrama de conexión estación de entrada de agua.</i>	108

Figura 36	109
<i>Serial de la estación de entrada de agua.</i>	109
Figura 37	110
<i>Diagrama de conexión de la estación de piscina.</i>	110
Figura 38	111
<i>Conexión de la comunicación con RS 485.</i>	111
Figura 39	112
<i>Datos recibidos por el arduino Mega mediante el Monitor Serial.</i>	112
Figura 40	113
<i>Diagrama de arduino y micro-SD.</i>	113
Figura 41	114
<i>Diagrama de arduino y RTC.</i>	114
Figura 42	115
<i>Diagrama de conexión de SIM 900 y relé con alarma.</i>	115
Figura 43	116
<i>Diagrama de arduino y LCD 128x64.</i>	116
Figura 44	117
<i>Interfaz en la pantalla del panel central.</i>	117
Figura 45	118
<i>Interfaz de la aplicación Weka.</i>	118
Figura 46	118
<i>Interfaz de tuning o selección de parámetros y filtros.</i>	118
Figura 47	119
<i>Interfaz con indicaciones de tareas.</i>	119

Figura 48	120
<i>Gráficos resultantes mediante la aplicación Weka.</i>	120
Figura 49	121
<i>Montaje de componentes de los módulos en las placas metálicas.</i>	121
Figura 50	121
<i>Colocación de las placas metálicas en los gabinetes.</i>	121
Figura 51	122
<i>Instalación de pantalla y elementos externos del gabinete.</i>	122
Figura 52	123
<i>Vista frontal de los gabinetes.</i>	123
Figura 53	123
<i>Cableado de los gabinetes.</i>	123
Figura 54	124
<i>Empotrado y encendido de los tableros.</i>	124
Figura 55	125
<i>Cableado del área de instalación.</i>	125
Figura 56	125
<i>Instalación del gabinete central.</i>	125
Figura 57	126
<i>Calibración final de los sensores y sistema.</i>	126
Figura 58	127
<i>Instalación del nodo en la entrada de agua.</i>	127
Figura 59	128
<i>Instalación del nodo de la piscina de crecimiento de truchas.</i>	128

Figura 60	129
<i>Estructura de distribución de puntos de oxigenación.</i>	129
Figura 61	131
<i>Diagrama de arduino y LCD 128x64.</i>	131
Figura 62	136
<i>Datos enviados y visualizados en el monitor serial.</i>	136
Figura 63	137
<i>Llegada de datos por comunicación Modbus.</i>	137
Figura 64	138
<i>Datos obtenidos del módulo RTC.</i>	138
Figura 65	138
<i>Programación para la creación del archivo de formato “.csv”.</i>	138
Figura 66	139
<i>Formato de datos almacenados en la tarjeta extraíble micro-SD.</i>	139
Figura 67	140
<i>Mensajes recibidos por bajo nivel de agua.</i>	140
Figura 68	140
<i>Interfaz del sistema de monitoreo.</i>	140
Figura 69	141
<i>Gabinete central montado.</i>	141
Figura 70	142
<i>Colocación de gabinetes exteriores.</i>	142
Figura 71	143
<i>Reorganización de datos obtenidos en lectura de sensores.</i>	143

Figura 72	144
<i>Datos ingresados a Weka.</i>	144
Figura 73	145
<i>Graficas de los datos obtenidos por los sensores en Weka.</i>	145
Figura 74	146
<i>Aplicación del método NaiveBayes en Weka.</i>	146
Figura 75	147
<i>Aplicación del método SMOs en Weka.</i>	147
Figura 76	150
<i>Comparación de piscinas del criadero.</i>	151
Figura 77	152
<i>Comparación de la cantidad de peces.</i>	152

CAPÍTULO I: ANTECEDENTES

El primer capítulo de este proyecto detalla el problema a investigar, los objetivos a cumplir, el alcance y finalmente la justificación del por qué se va a desarrollar este proyecto en beneficio del criadero donde se implementará el sistema.

1.1. Problema.

El criadero “Truchas Aguas Verdes”, se encuentra ubicado en la provincia del Carchi, cantón Tulcán en la parroquia de Tufiño, es un criadero dedicado a la producción de la trucha arcoíris que tiene 50000 peces distribuidos según su tamaño en 10 piscinas. Actualmente se tiene

2 personas encargadas de la crianza de los peces que realizan esta actividad de una manera artesanal, realizan la limpieza de las piscinas cada 8 días, sin tener un control adecuado de los parámetros de calidad del agua en las piscinas, pues, no se obtienen en ningún horario del día, porque no se tiene los equipos necesarios para realizar dicho procedimiento.

El no tener a la mano las mediciones de parámetros de calidad del agua causa problemas, como un menor índice en el crecimiento de los peces e incluso hasta la muerte de estos. Estos problemas en su mayoría son causados por la temperatura inadecuada del agua, la falta de oxígeno disuelto, la evolución del pH hacia valores extremos mínimos o máximos, como también por sustancias tóxicas presentes en los alimentos artificiales y residuos de pesticidas. Y en épocas de invierno también por la contaminación del agua debido a derrumbes aguas arriba del criadero lo cual trae consigo desechos de productos químicos agrícolas, y afluentes de aguas residuales. El retraso de la información puede causar inconvenientes a nivel de salud del pez o incluso la pérdida en grandes cantidades de la producción (Rivera Herrera & Yopez Aroca, 2015).

Por lo tanto, se propone un sistema electrónico de monitoreo dirigido al criadero “Truchas Aguas Verdes”, mediante el uso del protocolo Modbus para la comunicación del sistema y la utilización de la metodología en cascada para todo el desarrollo del proyecto. De esta manera el dispositivo será capaz de enviar en tiempo real todas las variaciones de los parámetros de calidad del agua, a una base de datos vinculada con un teléfono móvil, para enviar alertas cuando se generen variaciones en los rangos establecidos para cada parámetro. Con la finalidad de tener mediciones en tiempo real, con una cantidad de información suficiente para observar a gran detalle el comportamiento de los parámetros en la crianza del pez en una piscina, se plantea implementar un algoritmo para procesar automáticamente grandes datos crudos e

identificar patrones significativos y relevantes para presentarlos automáticamente al usuario lo que conlleva a una mayor facilidad para la planeación de estrategias por parte de los encargados de este sector productivo, como la limpieza de las piscinas en el momento que se genere la alerta para que se evite grandes pérdidas en la producción.

Al no tener un sistema automatizado de registro del comportamiento de los parámetros de calidad del agua, no se podrá tomar las medidas adecuadas de prevención y control para evitar inconvenientes a nivel de salud de la trucha o incluso grandes pérdidas en la producción.

1.2. Objetivos

1.2.1. Objetivo General.

Implementar un sistema electrónico de reconocimiento de parámetros de calidad del agua en el cultivo de truchas arco iris, mediante el uso del protocolo de comunicación MODBUS, para determinar que el ambiente de crianza de truchas influye en la mejora de la producción del criadero “Truchas Aguas Verdes”

1.2.2. Objetivos específicos.

- Recopilar información correspondiente al sistema, a través de revisión bibliográfica enfocada en temas del proceso de crianza de truchas arco iris, el protocolo

MODBUS y la metodología en cascada dirigida al diseño de sistemas electrónicos, para definir las bases teóricas necesarias en el desarrollo del sistema.

- Desarrollar el sistema electrónico de reconocimiento de parámetros de calidad del agua mediante una red MODBUS para la comunicación entre sensores con el nodo central, en el criadero “Truchas Aguas Verdes”, con el fin de verificar el ambiente de crianza de truchas.
- Implementar en el sistema criterios de aprendizaje de máquina mediante una elección de algoritmos que permitan la adquisición de datos para ser analizados.
- Comprobar el funcionamiento del sistema a través del prototipo, para demostrar que el ambiente de crianza de truchas influye en la mejora de la producción.

1.3. Alcance.

En el presente proyecto se desarrollará un sistema electrónico de reconocimiento de parámetros de calidad del agua, para esto se tomará en cuenta una piscina del criadero “Truchas Aguas Verdes”, que será en donde se encuentran los 10000 peces que tienen 4 meses de crecimiento, se diseñará un sistema electrónico de monitoreo que enviará una alerta a la persona encargada notificando a un móvil de posibles cambios bruscos de los parámetros de la calidad del agua, que previamente fueron registrados en la base de datos alojada dentro del criadero.

Un aspecto importante para un proyecto de truchas es el agua, pues esta debe tener ciertas condiciones de calidad y cantidad. Por lo que el proyecto iniciará con un amplio estudio relacionado al análisis de la calidad del agua durante el cultivo de truchas arco iris en una piscina.

Se realizará un análisis bibliográfico de la metodología en cascada dirigida al diseño de sistemas electrónicos y del protocolo de comunicación MODBUS. Con la metodología en cascada se podrá ver el progreso como un flujo constante hacia abajo (como una cascada) a través de las fases de implementación del software. Esto significa que cualquier fase en el proceso de desarrollo comienza solo si la fase anterior está completa. El enfoque de cascada no define el proceso para volver a la fase anterior para manejar los cambios en los requisitos. Como parte de la metodología se realizará una selección de sensores, placas y módulos electrónicos de diferentes proveedores analizando sus características, ventajas y desventajas, eligiendo los más idóneos y adaptables al desarrollo del proyecto para su implementación. Se analizará el protocolo MODBUS; un protocolo de comunicación muy extendido en dispositivos de automatización industrial y que está jugando un papel muy importante también en proyectos electrónicos dada su sencillez y fiabilidad.

Se desarrollará el sistema electrónico de reconocimiento de parámetros de calidad del agua, en el criadero “Truchas Aguas Verdes”, haciendo uso de las fases de la metodología en cascada. Mediante el diseño de una red de comunicación maestro-esclavo usando como dispositivos inteligentes la placa de desarrollo Arduino, que puedan intercambiar información sobre determinados sensores y módulos que permitan el reconocimiento de los parámetros de calidad del agua, y usando el protocolo MODBUS para la comunicación del sistema.

Se determinará las variables que se van a medir en la toma de datos de los parámetros de calidad del agua para después proponer un método de adquisición y almacenamiento de estos en una base de datos para evitar posibles errores, de igual forma se plantea implementar un algoritmo que será elegido tomando en cuenta; la precisión, el tiempo de entrenamiento, linealidad, cantidad de parámetros y características. De esta manera se podrá procesar

automáticamente grandes datos crudos e identificar parámetros significativos y relevantes para presentarlos automáticamente al usuario, lo que conlleva a una mayor facilidad para la planeación de estrategias por parte de los encargados de este sector productivo.

Una vez realizado el proyecto se realizará las pruebas de funcionamiento del sistema implementado mediante dos grupos de pruebas, el uno se lo realizará antes de implementar criterios de aprendizaje de máquina en los datos, y el otro grupo será ya implementado todo el sistema, de esta manera se evaluará que el ambiente de la crianza de truchas en la piscina seleccionada influye directamente en la mejora de la producción.

1.4. Justificación.

La crianza de peces es una actividad que ofrece desarrollo y fuente de ingresos para la sociedad. Actualmente la ciencia permite al piscicultor analizar datos que son necesarios para llevar un monitoreo y control de su producción. El análisis de los datos se enfoca en obtener la información necesaria de los factores que influyen en el crecimiento del pez y su desarrollo. Así se llevarán a estudio los resultados obtenidos los cuales son la fuente principal para la toma de decisiones (Rivera Herrera & Yopez Aroca, 2015).

Igualmente, el piscicultor debe estar preparado para enfrentar dificultades que se presentan en el desarrollo de este oficio como son: enfermedades en los peces, mala calidad del agua, deficiencias en la alimentación entre otras. En este sentido el productor está sometido a situaciones que podrían crear un estado de pérdida de motivación con el consecuente abandono de la actividad, de tal forma que cualquier perturbación sobre los criaderos de truchas es motivo

de preocupación suficiente y merece su atención debido al riesgo que esto representa para el buen funcionamiento y productividad de este campo (Padrón Maiz et al., 2013).

En la actualidad el análisis del resultado obtenido en algunos criaderos es realizado de forma manual, donde involucra la intervención de recurso humano y material. La interpretación del resultado obtenido sobre la calidad del agua le permite al profesional observar si la especie en producción está desarrollándose en el ambiente apropiado, si el sector de producción es óptimo y así brindar la asesoría necesaria para asegurar el buen uso de los recursos, en caso contrario se ejecutan las acciones pertinentes para que su labor sea de éxito (Rivera Herrera & Yopez Aroca, 2015).

Tener la información a la mano utilizando herramientas tecnológicas optimiza el tiempo que conlleva visitar cada piscina durante el día, y es de gran utilidad para tomar decisiones e incluso previsiones basadas en las estadísticas que se obtengan. Además, facilita el acceso a la información de forma independiente al lugar geográfico donde la persona interesada se encuentre, ya que no será necesario permanecer constantemente en la piscina para observar el comportamiento de esta. Mediante este proyecto se dará a conocer al público en general de la existencia de un criadero de peces con un control diario de la calidad del agua, como parámetro principal de un ambiente ideal en la crianza de truchas.

2. CAPÍTULO II: MARCO TEÓRICO

Para el presente capítulo se detallará aspectos acerca del cultivo de peces centrándose en el cultivo de la trucha arcoíris, las características y condiciones del agua y del medio donde crece la trucha arcoíris. Además, para este proyecto se tomará en cuenta el concepto del IoT, protocolos de comunicación para la obtención de datos y la posterior aplicación de aprendizaje de máquina. Con todo esto detallado se puede precisar la tecnología para estructurar el prototipo a implementarse.

2.1. Acuicultura

Se entiende por Acuicultura al proceso de producción de especies vegetales o animales criadas en cautiverio en un ambiente natural o artificial de agua dulce, salobre o salado (Gómez Gómez, 2008).

2.1.1. Estructuras acuícolas

Los productores de especies vegetales o animales acuáticos principalmente utilizan estructuras de embalse de agua como: estanques, raceways, jaulas flotantes, bateas, balsas, etc., (Lujan Monja & Cruajulca, 2014). A continuación, se detallarán algunas de estas:

- **Raceways**

Son canales lineales, se sugiere construirlos en cemento, que se instalan en serie en los cuales el agua fluye a velocidad, en estas estructuras la densidad de peces es muy alta y carece de alimento natural.

- **Tanques de geomembrana, fibra de vidrio o concreto**

Los tanques suelen ser rectangulares, redondos u ovalados donde el agua ingresa y fluye hacia su centro donde está ubicado el desagüe ayudando a la autolimpieza del tanque. La densidad de peces es regular desde los 35 kg/m³ hasta los 150 kg/m³ siempre dependiendo de la aireación del lugar.

- **Jaulas flotantes**

Las jaulas flotantes son estructuras normalmente de madera y, de cualquier forma, esta depende del propósito y lugar donde se va a instalar. Estas jaulas normalmente se usan para albergar peces en la etapa de engorde.

- **Estanques de tierra o seminaturales.**

Los estanques ya sean naturales o artificiales pueden tener cualquier forma con una profundidad no mayor a los 2 m. Estas estructuras se emplean para el desove de peces, criadero de especies animales y el engorde. Los estanques son sistemas intensivos donde los requerimientos técnicos son mínimos.

2.1.2. Clasificación de la acuicultura

Dentro de la acuicultura tener una sola clasificación puede ocasionar confusiones y tender a ser muy complejo. Es por esto que en (Nash et al., 2021) se menciona que se puede generar diferentes clasificaciones dependiendo de características como: ambientes de crianza, especie de crianza, niveles de intensificación, etc. Algunos de las clasificaciones se las detalla a continuación.

2.1.2.1. Clasificación por el ambiente.

El ambiente en el que se desarrolle la acuicultura dependerá del tipo de especie que se desee producir. De esta manera, la clasificación por el ambiente describe tres medios diferentes, (Nash et al., 2021) los describe como:

- **Acuicultura marina**

La acuicultura marina hace referencia a la reproducción, crianza y cosecha de especies vegetales y animales acuáticas en aguas con más de 30 PSU, unidades prácticas de salinidad. Esta práctica se desarrolla en los océanos para cultivos frecuentes como: conchas, ostiones, ostras, salmones, marco algas, etc.

- **Acuicultura en agua salobre**

El agua salobre es una mezcla de agua dulce y agua salada que normalmente se encuentra en las zonas costeras con una salinidad de entre 0.5 y 30 PSU. En estos lugares se debe considerar el cambio de salinidad en el tiempo debido a que se rige a la cantidad de agua dulce o salada que ingrese al lugar.

- **Acuicultura en agua dulce**

La acuicultura en agua dulce maneja rangos de salinidad menores al 0.5, esta práctica se desarrolla dentro del continente utilizando flujos de agua abundante y constante para la reproducción y la crianza de especies acuáticas.

2.1.2.2. Clasificación por el nivel de intensidad.

El nivel de intensidad hace relación a la densidad de especies en el espacio de crianza y reproducción. Tres sistemas son los más conocidos y utilizados.

- **Sistemas Extensivos**

En este caso, las actividades humanas se limitan a sembrar y cosechar una o más especies en un cuerpo dado. Suele realizarse en embalses o reservorios como se observa en la Figura 1, ya sean naturales o artificiales, sembrando organismos a baja densidad, aproximadamente 1 pez/m², y permitiéndoles sobrevivir gracias al aporte natural de alimentos que allí existe con una producción menor a los 500 kg por hectárea (Rodríguez Gómez, 2001).

Figura 1

Embalse de agua idóneo para un sistema extensivo.



Nota. Adaptado de Innovación en la producción de peces ornamentales, por Biól, 2018, Issuu. (Biól, 2018).

- **Sistemas Semi intensivos**

Aquí es donde el trabajo humano va más allá de plantar y recolectar organismos. En este sistema se comienza a utilizar alimentos balanceados para aumentar la densidad de entre 1 a 5 peces/m² implementando aireación que cubre el 10% o 15% del área como en la Figura 2. Usar una densidad de plantación más alta que la plantación extensiva ayuda a obtener mayores rendimientos (Rodríguez Gómez, 2001).

Figura 2

Embalses cubiertos semi intensivos para mayor producción.



Nota. Adaptado de Oreochromis niloticus, por Rakocy, 2009, Fao.(Rakocy, 2009).

- **Sistemas Intensivos**

Utilizar mayor densidad de población e infraestructura adecuada, Figura 3, (estanques, jaulas flotantes, etc.); requiere de mayor tecnología, como manejo de agua, sistemas de aireación, y en algunos casos oxígeno líquido, filtros biológicos, etc., para manejar densidades de 5 a 20 peces/m² con una aireación de al menos el 50% del área. A medida que se invierten más recursos, se requiere una mejor planificación, pero al mismo tiempo se obtiene una mayor producción, por lo que la rentabilidad aumenta. Básicamente, la acuicultura se lleva a cabo con fines comerciales, para lo cual se requiere tecnología de proceso (Rodríguez Gómez, 2001).

Figura 3

Piscinas cubiertas y controladas en sistemas intensivos de acuicultura.



Nota. Adaptado de 2 Sistemas de producción, por AgronoMaster, 2017, AgronoMaster. (AgronoMaster, 2017).

2.1.2.3. Clasificación por el número de especies.

En un estanque de producción de acuicultura se pueden cultivar u o varias especies a la vez dependiendo de esto se generan dos tipos de cultivo:

- **Monocultivo**

En estos cultivos se produce una sola especie. Los cultivos más frecuentes son los de tilapia y truchas.

- **Policultivo**

Los policultivos son sistemas que buscan aprovechar todo el espacio de un estanque o estructura de cultivo. Por ejemplo: un policultivo de tilapia y camarones. En esto las tilapias se desarrollan en las columnas de agua y el camarón en la base del estanque.

- **Integrados**

Un cultivo integrado es un sistema donde se aprovecha los desechos orgánicos de diferentes especies para su desarrollo en conjunto. Por ejemplo: los cerdos generan desechos con los cuales se pueden optimizar la producción de microalgas y con ello mejorar la alimentación de los peces. Es un concepto relativamente nuevo que se está desarrollando.

2.1.2.4. Clasificación por el tipo de especie.

Otra clasificación que a menudo se utiliza es la que detalla el tipo de especie que se cultiva como las siguientes:

- **Camaronicultura**

Hace referencia al cultivo de camarones, la cual es una de las actividades más importantes en el mundo donde el camarón blanco del Pacífico y el camarón tigre negro son las especies más producidas.

- **Salmonicultura**

El cultivo de salmones iniciado en Europa que luego se propagó hacia los países americanos tiene como sus productores estrella a los países de Noruega, Chile y Escocia.

- **Malacocultura**

Es el cultivo de moluscos como ostras, mejillones, conchas, etc.

- **Piscicultura**

Este término se utiliza comúnmente como sinónimo de acuicultura, pero este término se centra en el cultivo de peces en estanques o piscinas.

2.2. Piscicultura

De acuerdo con (Crovi, 2016): “La piscicultura tiene por objeto el cultivo racional de los peces, lo que comprende particularmente el control de su crecimiento y su reproducción. Se practica en estanques naturales o artificiales, en la cual podemos realizar su multiplicación, alimentación y el crecimiento de los peces, así como la puesta en funcionamiento y mantenimiento

controlado de los mismos aprovechando las condiciones hídricas que se presente en el lugar para el desenvolvimiento de la truchicultura.”

2.1.3. Piscicultura de producción en Ecuador

La piscicultura se vuelve fundamental para el desarrollo socioeconómico, pues, esta impulsa el avance tecnológico dentro del campo laboral y brinda soluciones a problemas que se presentan al momento de tener una densidad alta de producción a un bajo costo (Hidalgo Granja & Sarzosa Pruna, 2013).

Desde la conformación de la piscicultura como una actividad productiva dentro de la economía ecuatoriana se ha observado su rápida expansión a muchos sectores del país aprovechando los ecosistemas con acceso a cuerpos de agua. El potencial de producción de peces en aguas continentales según (Alvarez, 2006) enlista diferentes especies dependiendo el ambiente y la temperatura del agua, donde la trucha y el salmón son especies que se producen en aguas frías como ríos y lagos.

Dentro de la piscicultura de producción se monitorea factores como: el alimento, la oxigenación, la calidad del agua, la temperatura, etc., para mantener o aumentar la cantidad de peces que se cultivan. Es la razón por lo cual nuevas tecnologías se centran en mejorar y optimizar estos factores para potenciar cada uno de los cultivos y obtener mayor rentabilidad.

2.1.4. Trucha

La trucha es un pez que se produce en estanques naturales o artificiales aprovechando el agua de ríos en temperaturas bajas. Su alta rentabilidad y adaptabilidad dependiendo la especie hace que la producción de trucha este en aumento.

2.1.4.1. Trucha arcoíris.

La trucha arco iris (*Ochorynchus mykiss*), como se observa en la Figura 4, es una especie nativa de las cuencas de América del Norte que desemboca en el Pacífico. Desde principios de 1874, se ha introducido en todos los continentes del mundo, excepto en la Antártida, con fines recreativos (pesca deportiva). y cultivo. Según datos de la Organización de las Naciones Unidas para la Agricultura y la Alimentación, con el desarrollo de alimentos en pellets, la producción de trucha aumentó en 1950, y este método de cultivo se utiliza ahora en las cuencas de las tierras altas de muchos países tropicales y subtropicales de Asia, África Oriental y África. Sudamérica. Su alta adaptabilidad hace que se desarrolle en toda Sudamérica como en Argentina, Brasil, Bolivia, Chile, Colombia, Ecuador, Perú y Venezuela (Vargas Conde, 2018).

Figura 4

Fisiología de una trucha arcoiris.



Nota. Adaptado de Trucha Arco Iris, por Conxemar, 2018, Conxemar. (Conxemar, 2018).

2.1.4.2. Ciclo de vida de la trucha.

Según el ministerio de pesquería, el desarrollo biológico de la trucha es de 4 etapas las cuales son:

- **Ovas**

Son los huevos fecundados que permanecen durante 30 días promedio, el tiempo de incubación dependerá de la temperatura del agua, el rango óptimo está entre los 8 a 12°C, finalmente se desprende del nido realizado por la hembra y se convierten en larvas (Hernández & Aquino, 2008).

- **Alevinos**

Son peces pequeños que miden de 3cm. a 10cm, su peso oscila entre 1.5g a 20g, en esta etapa se empiezan a formar los órganos, dura aproximadamente 14 y 20 días. Estas truchas en esta etapa ya están listas para ser vendidas (Hernández & Aquino, 2008).

- **Juveniles**

Son peces que miden de 10 cm a 15 cm, cuyo peso es generalmente de 20g a 100g, en esta etapa los organismos están completamente formados a diferencia de los adultos éstos aún no han madurado sexualmente. Las medidas para los estanques más utilizados para su manejo son 20 de largo x ancho x 1,2 m de profundidad (Hernández & Aquino, 2008).

- **Comercial.**

Han transcurrido de 15 a 18 meses, presentan un cambio en la coloración y van ganado peso para ser comercializados, estos miden 18 cm a 27 cm con un peso de 100g a 200g, viven a una temperatura promedio de 15 °C (Hernández & Aquino, 2008).

2.1.5. Calidad del agua

Las condiciones de la calidad de agua dependen de los requisitos de cada especie o de la necesidad que se desea suplir, se deben tener en cuenta características de tipo físicas, químicas y biológicas. Para la producción de truchas, este parámetro es muy importante ya que se debe considerar parámetros como el pH, el oxígeno disuelto, la temperatura y demás, para garantizar la preservación tanto de la flora y fauna; en este caso influyen directamente en el proceso de crianza de los mismos (Lamingo LLumiyinga & Moreno Garzón, 2014).

2.1.5.1. Factores Químicos.

A continuación, se realiza una descripción de los factores químicos más relevantes dentro de la producción de truchas que anhelan altas densidades de peces. Entre los factores más necesarios se puede mencionar a:

- ***Potencial de Hidrógeno (pH).***

El pH sirve para determinar si la solución es ácida o básica. La escala de pH varía de 0 a 14. En caso de ser inferiores a 7 y cercanos a 0 indican acidez; los valores próximos a 14 indican un aumento de basicidad y cuando el valor es 7 indica neutralidad (Rodríguez Gómez, 2001).

- ***Oxígeno Disuelto (DO).***

Como se conoce existe mayor oxígeno en el agua fría que en el agua caliente, por lo que los niveles de oxígeno disuelto varían entre 7 y 12 partes por millón (ppm o mg/l). en algunas ocasiones se las expresa en términos de Porcentaje de Saturación. Existe poco DO en áreas donde el material orgánico (vertidos de depuradoras, granjas, plantas muertas y

materia animal) está en descomposición, pues las bacterias necesitan consumir oxígeno para realizar dicho proceso. El Oxígeno Disuelto (DO) indica la cantidad de oxígeno disuelto en el agua. Permite determinar el nivel de contaminación que tiene el agua o si es adecuada para dar vida animal y vegetal. La calidad de agua mejora cuando los niveles de oxígeno disuelto son altos, en caso de disminuir el nivel algunos peces u organismos no pueden sobrevivir (Lamingo LLumiyinga & Moreno Garzón, 2014).

2.1.5.2. Ejemplo de parámetros de cultivo de truchas arco iris.

Después de una inducción a los factores químicos que especifican la calidad del agua óptima para la piscicultura, ahora, se describe los valores específicos de los factores químicos para las establecer las mejores condiciones dentro del cultivo de truchas:

- ***Temperatura.***

Las temperaturas que soporta esta entre 25°C y cercanos a la congelación, sin embargo, la temperatura óptima para la etapa de crecimiento está entre 10 y 17°C, se obtiene mejores resultados en rangos de 15 a 16°C indicando que valores inferiores suelen disminuir el crecimiento y valores superiores incrementan el riesgo de propagación de enfermedades.

- ***Oxígeno disuelto.***

De este factor depende mucho la supervivencia de los peces, más para salmónidos, debido a la exigencia que estas especies tienen en cuanto a oxígeno, dándose un crecimiento óptimo a una concentración de oxígeno que va entre 4.5 a 5.9 mg/L.

- ***pH.***

Se refiere a las características de acidez o basicidad que posee el agua, la importancia se debe a que actúa como un regulador del metabolismo. Ambientes acuáticos con pH ligeramente alcalino son ideales para la crianza de truchas, siendo el pH óptimo de 7.0 a 8.0. Ambientes con pH superiores a 9.0 o inferiores a 6.0 no son adecuados para la truchicultura. Al respecto se ha probado que bajos niveles de pH pueden producir hemorragias en las branquias desencadenando elevadas mortalidades. Además de inducir a que el pez entre en un estado de estrés.

El pH óptimo para el desarrollo de truchas en etapas tempranas es diferente cuando son mayores, de 6.0 a 8.5 cuando son juveniles y adultos y 6.5 a 8.0 cuando son embriones y alevines.

2.2. IoT

El internet de las cosas (IoT) se puede describir como la agrupación e interconexión de dispositivos y objetos a través de la red, donde todos pueden ser visibles e interactuar entre ellos. La implementación de las IoT en los sistemas de producción son una fuerte estrategia la cual busca potenciar las materias primas y obtener mejores beneficios con la menor cantidad de pérdidas. Para esto se adentrará en el tema de los protocolos de IoT para conocer las normativas a seguir para interconectar cada uno de los componentes de un sistema integrado (Zhi-hong & Yi-jun, 2012).

2.2.1. Protocolos IoT

Dentro de la informática y las telecomunicaciones los protocolos IoT se las conoce como un conjunto de normativas y reglas que da la capacidad a dos entidades de poder entenderse e intercambiar información, haciendo más fácil la comunicación Machine2Machine (M2M). De la misma manera en que dos personas deben hablar en un mismo idioma para lograr entenderse; los

dispositivos deben utilizar los mismos protocolos para intercambiar información. Los protocolos en redes IoT de datos emergentes cuentan con varios niveles (Semle & eFalcom, 2016):

- Aplicación: interfaz que comunica humano y máquina.
- Red: potenciador de la comunicación del enrutador con los dispositivos conectados a la misma red.
- Transporte: comunicación entre los diferentes niveles y seguridad.
- Físico: red física de comunicación.
- Vínculo de datos: transporte de datos dentro del sistema con detección y corrección de problemas.

2.2.1.1. OPC.

Significa Open Platform Communications, es un conjunto de especificaciones de automatización industrial. UA significa Arquitectura Unificada y el protocolo se utiliza para la recopilación y control de datos. Es un nuevo protocolo OPC con mejores funciones que su predecesor, OPC Classic. OPC Classic se compone de tres especificaciones independientes: OPC Data Access, OPC Alarms & Events y OPC Historical Data Access. Estas especificaciones se basan en la tecnología de Microsoft, por lo que están asociadas a los sistemas Windows. Aunque el nuevo UA es independiente de la plataforma, permite su uso en varias plataformas de hardware y sistemas operativos. OPC UA se ejecuta en las capas 5-7 del modelo (Dey & Sen, 2020).

2.2.1.2. CoAP.

Significa Protocolo de aplicación restringida. Está diseñado para su uso en dispositivos restringidos. Es decir, dispositivos con recursos limitados, por ejemplo, duración limitada

de la batería. CoAP usa UDP como protocolo de capa de transporte. UDP es un protocolo sin conexión, que es un método para transferir datos sin garantizar que el destinatario esté disponible o listo para recibir datos. Como CoAP usa UDP, es compatible con la difusión UDP y la multidifusión. CoAP se basa en la relación cliente-servidor, donde el cliente envía una solicitud al servidor y el servidor envía una respuesta al cliente. Un nodo sensor en CoAP es a menudo un servidor y nota un cliente. Entonces, el sensor tiene recursos para indicar el estado o alterar el estado al que puede acceder el cliente. CoAP está operando en la capa 7 en el modelo OSI (Bhattacharya et al., 2020).

2.2.1.3. MQTT.

Significa Message Queue Telemetry Transport, es un protocolo de mensajería diseñado para la comunicación ligera de máquina a máquina (M2M). Utiliza TCP y consta de un servidor conocido como intermediario y uno o varios sensores conocidos como clientes que están conectados al intermediario. En MQTT, los clientes pueden suscribirse a lo que se conoce como tema. Un tema es una dirección donde se publican todos los mensajes. Un cliente puede suscribirse a múltiples temas. Cada cliente también puede registrar lo que se conoce como un mensaje de Última Voluntad y Testamento en el corredor. Si el cliente se desconecta, el intermediario enviará el mensaje a los suscriptores. MQTT está trabajando en la capa 5-7 en el modelo OSI (Albuquerque et al., 2020).

2.2.1.4. Modbus.

Es un protocolo abierto desarrollado por Modicon. Cuando se lanzó en 1979, era un protocolo de comunicaciones en serie para usar con PLC. Ahora hay tres versiones principales de Modbus; Modbus RTU, Modbus ASCII y Modbus TCP. La base es la misma

para los tres tipos y solo los detalles de la interfaz son diferentes. Modbus RTU es para comunicarse en serie utilizando una representación binaria de datos. Modbus ASCII es para comunicación en serie usando caracteres ASCII para comunicación de protocolo. Y Modbus TCP está utilizando TCP / IP para la comunicación.

Modbus RTU y ASCII necesitan la ID de esclavo al comienzo del mensaje y una suma de verificación al final. Mientras que Modbus TCP solo necesita un encabezado de aplicación Modbus (encabezado MBAP) al comienzo de cada mensaje. La parte de datos de un mensaje Modbus se denomina Unidad de datos de protocolo (PDU) y es la misma para las tres versiones de Modbus. La PDU consta de un código de función Modbus y luego datos. Todo el mensaje Modbus, donde se combinan el encabezado y la PDU, se denomina unidad de datos de aplicación (ADU) (Sen, 2021).

- **Protocolo MODBUS.**

El protocolo Modbus es un protocolo estándar de la industria que permite que los controladores programables se comuniquen a través de una red o un enlace de comunicaciones local. Define una estructura de mensaje estándar que todos los controladores compatibles con Modbus reconocen e implementan, independientemente del tipo de red (Buchanan, 2004).

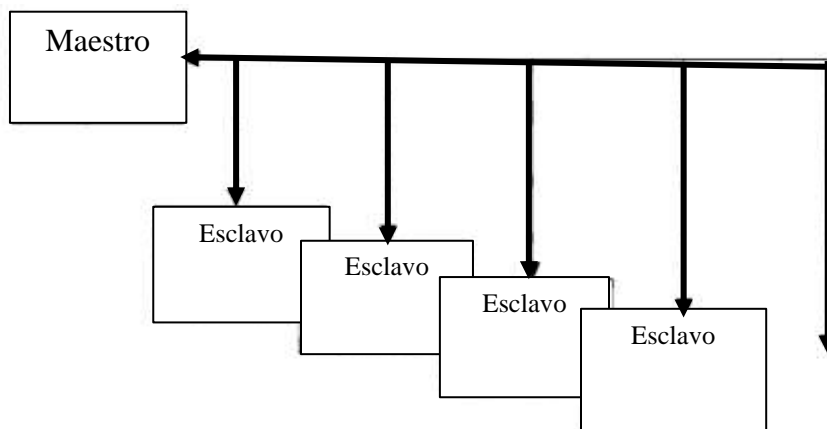
- **Transacciones en redes Modbus.**

Los controladores Modbus estándar se comunican mediante RS-232C y se pueden conectar en red o mediante un módem. Cada controlador (como un procesador host) se comunica con los dispositivos conectados (como un PLC) utilizando una técnica maestro-esclavo como se muestra la estructura de la Figura 5. El controlador (el maestro) inicia

transacciones (consultas) que se envían a los otros dispositivos (los esclavos) (Buchanan, 2004).

Figura 5

Estructura maestro – esclavo.



Nota. Adaptado de MySQL: administración y optimización, por Editions ENI, 2016, Ediciones-eni. (ENI, 2016).

El esclavo direccionado responde a la solicitud enviando datos o implementando la acción requerida. Este direccionamiento puede ser a un dispositivo individual o puede transmitirse a todos los esclavos conectados. No hay respuestas de una consulta de difusión.

- **Pila de Comunicación.**

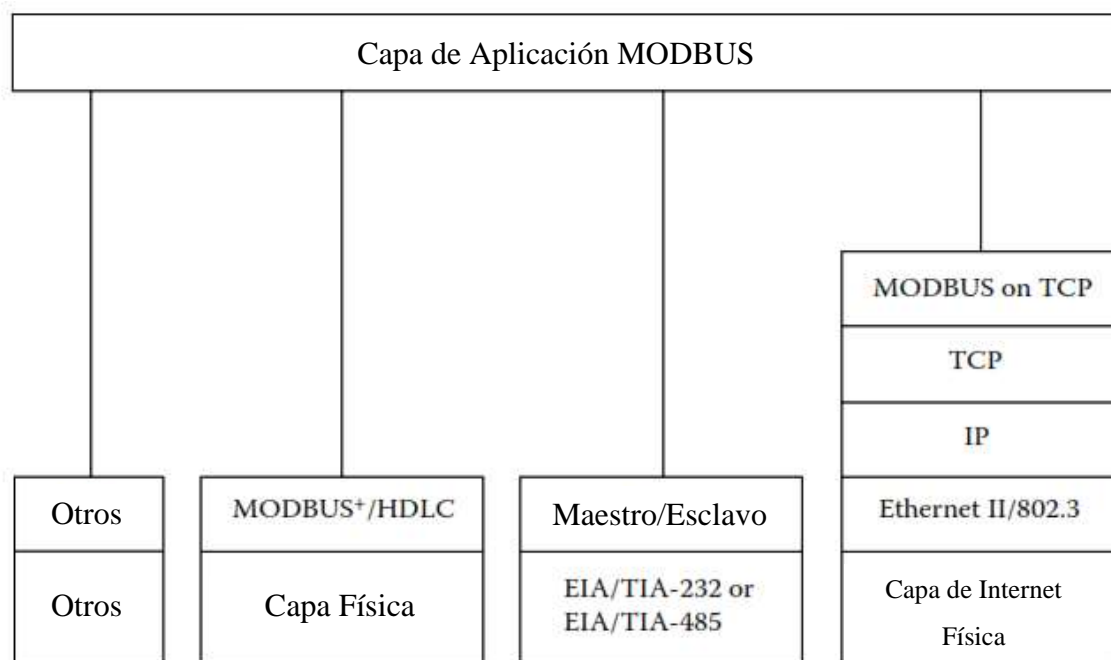
Actualmente, MODBUS se implementa utilizando lo siguiente:

- TCP / IP sobre Ethernet.
- Transmisión en serie asíncrona a través de diferentes medios, como EIA / TIA-232E, EIA-422, EIA / TIA-485-A, fibra y radio.
- MODBUS Plus: un método de paso de tokens de alta velocidad.

La Figura 6 muestra la implementación de una pila de comunicación MODBUS utilizando TCP / IP, maestro-esclavo y la capa física MODBUS Plus (Mocito et al., 2009).

Figura 6

Estructura de una pila de comunicación Modbus.



Nota. Adaptado de Modbus Application Protocol Specification, por Modbus-IDA, 2006, Modbus-IDA.

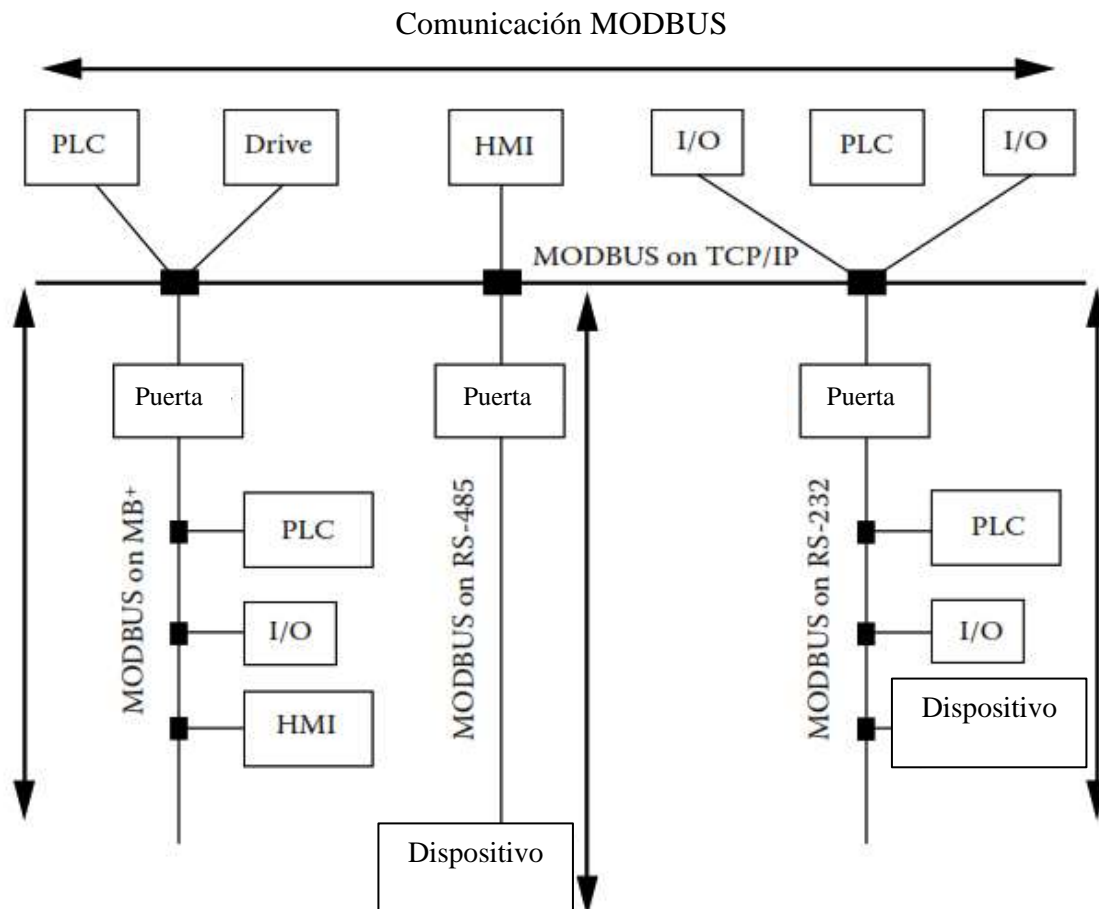
(Modbus-IDA, 2006).

- **Red de arquitectura.**

La Figura 7 muestra un esquema de arquitectura de red MODBUS. Los diferentes dispositivos como PLC, HMI y E / S se pueden conectar al MODBUS TCP / IP a través de vías individuales (Mocito et al., 2009).

Figura 7

Estructura de una red de arquitectura Modbus.



Nota. Adaptado de Modbus Application Protocol Specification, por Modbus-IDA, 2006, Modbus-IDA.

(Modbus-IDA, 2006).

Los diferentes protocolos MODBUS, a saber, MODBUS en MB +, MODBUS en RS-232 y MODBUS en RS-485, inician la comunicación remota mediante TCP / IP.

- **Comunicación.**

Se envía un mensaje de error, denominado respuesta de excepción, al maestro cuando el esclavo no puede llevar a cabo las acciones requeridas solicitadas por el maestro. La respuesta de excepción al maestro contiene lo siguiente:

- La dirección del esclavo que responde
- La acción que se solicitó al esclavo que realizará
- Una indicación de por qué no se pudo llevar a cabo la acción.

Un esclavo ignora un mensaje si contiene algún error. En tales casos, el maestro reenvía la consulta al esclavo ya que no pudo recibir una respuesta del esclavo.

- **Modos de transmisión.**

Dos tipos de modos de transmisión son posibles para transmitir datos en serial sobre una red Modbus, estos son RTU y ASCII.

Los dos modos se diferencian de varias formas: la forma en que se empaqueta la información en el campo del mensaje, la forma en que el contenido de bits del se interpretan los mensajes, la forma en que se decodifica el mensaje y la velocidad de funcionamiento a una velocidad de transmisión determinada.

Los dos modos no se pueden usar juntos y el usuario tiene la opción de seleccionar el modo particular para una determinada aplicación. El modo RTU es más rápido y robusto que el modo ASCII. Por lo tanto, encuentra más aplicaciones que el formulario ASCII. para la transmisión de mensajes (Mocito et al., 2009).

Modo de transmisión RTU.

En esto, cada byte de 8 bits se envía como dos caracteres hexadecimales de 4 bits. El formato para cada byte en modo ASCII se da a continuación:

- Codificación: binario de 8 bits, hexadecimal de 0 a 9, de la “A” a la “F”. Se componen de dos caracteres hexadecimales
- Cada campo de 8 bits del mensaje.
- Bits por byte: 1 bit de inicio, 8 bits de datos, LSB enviado primero, 1 bit para paridad par o impar, no se usa bit para no paridad, 1 bit de parada si se usa paridad, 2 bits se usan para no paridad.
- Campo de verificación de errores: CRC. Modo de transmisión ASCII

Modo de transmisión ASCII.

En esto, cada byte de 8 bits se envía como dos caracteres ASCII. El formato de cada byte en modo ASCII se da a continuación:

- Codificación: hexadecimal, caracteres ASCII del 0 al 9, de la “A” a la “F”. Un hexadecimal carácter comprende cada carácter ASCII del mensaje.
- Bits por byte: 1 bit de inicio, 7 bits de datos, LSB enviado primero, 1 bit para impares o paridad par, no se usa bit para no paridad, 1 bit de parada si se usa paridad, 2 bits utilizados sin paridad.
- Campo de verificación de errores: LRC.

- **Comprobación de errores**

El campo de verificación de errores permite al maestro confirmar la integridad del mensaje recibido del esclavo.

El método de verificación de errores empleado depende del modo de transmisión seleccionado. Puede ser una verificación de redundancia cíclica (CRC) para el modo de

unidad terminal remota (RTU) o comprobación de redundancia longitudinal (LRC) para el modo de transmisión ASCII.

Al recibir el mensaje completo, el dispositivo receptor calcula el valor de verificación de error y lo compara con el valor de verificación de error en el mensaje recibido. Si los dos están de acuerdo, no se ha producido ningún error y se toman las medidas correspondientes.

El mensaje recibido se rechaza si los dos valores difieren (Mocito et al., 2009).

2.3. Placas de desarrollo

Las placas de desarrollo hoy en día se las considera como dispositivos que cuentan con un microcontrolador programable, para la cual se puede generar y ejecutar diferentes instrucciones para un propósito con diferentes elementos (periféricos). Por defecto, estas tarjetas cuentan con puertos digitales y analógicos dentro de la electrónica digital.

Para este tipo de placas no existe un tipo de lenguaje de programación específico, es por eso, que ha existido un aumento en la utilización de estos elementos en proyectos de IoT; su facilidad de uso, poder, capacidad y gran cantidad de periféricos hace que se prefieran las placas de: Arduino, Raspberry y Onion como las más conocidas y baratas (PdD, 2020).

2.3.1. *Arduino*

La placa Arduino es una placa de circuito impreso (PCB) diseñada para usar un chip de microcontrolador, así como otras entradas y salidas digitales o analógicas. La placa tiene muchos otros componentes electrónicos necesarios para que el microcontrolador funcione o amplíe su capacidad.

Un microcontrolador es una pequeña computadora contenida en un solo circuito integrado o Chip de computadora. Los microcontroladores son una excelente manera de programar y controlar electrónica. Las placas de microcontrolador tienen un chip de microcontrolador y otros útiles conectores y componentes que permiten al usuario conectar entradas y salidas (Valdés & Pallás, 2017).

Arduino se caracteriza por encontrarse de manera libre en hardware y software en todo el mundo, su adaptabilidad, bajo costo y variedad hace que utilizar una placa arduino se una de las mejores opciones.

2.3.1.1. Tipos de Arduino

Arduino es una de las primeras plataformas microcontroladoras open source en el mundo, por lo cual es de esperarse varias versiones de este, desde sus principios con el Arduino uno, no obstante, a medida que se avanza el nivel de programación y electrónica, los requerimientos de este son más demandantes como es el procesamiento y número de pines de entrada y salida, por ende, hay diferentes tipos dependiendo del uso en que se lo aplique como: Arduino Uno, Mega, Leonardo, etc., (Moreno, 2018).

2.3.1.2. Módulos / Shields

En el ecosistema de arduino se encuentra una versatilidad enorme al manejar shields. Las shields son placas de circuitos modulares que se acoplan unas a otras para otorgar más funcionalidades extras al Arduino. Los shields pueden ser de diferentes tipos donde para este proyecto se puede destacar: Sensores, Actuadores y Comunicación (Moreno, 2018).

- *Shields de sensores.*

Los shields de arduino en la parte de sensores, como se observa en la Figura 8, cuenta con una amplia variedad los cuales pueden contar con una tarjeta adicional que hace de transductor de cualquier sensor en específico o también existe el sensor solo que con una configuración específica el arduino puede obtener la información directamente. Estos sensores utilizan diferentes tipos de comunicación como son: I2C, SPI, Serial (Moreno, 2018).

Figura 8

Shield de sensores compatible con arduino.



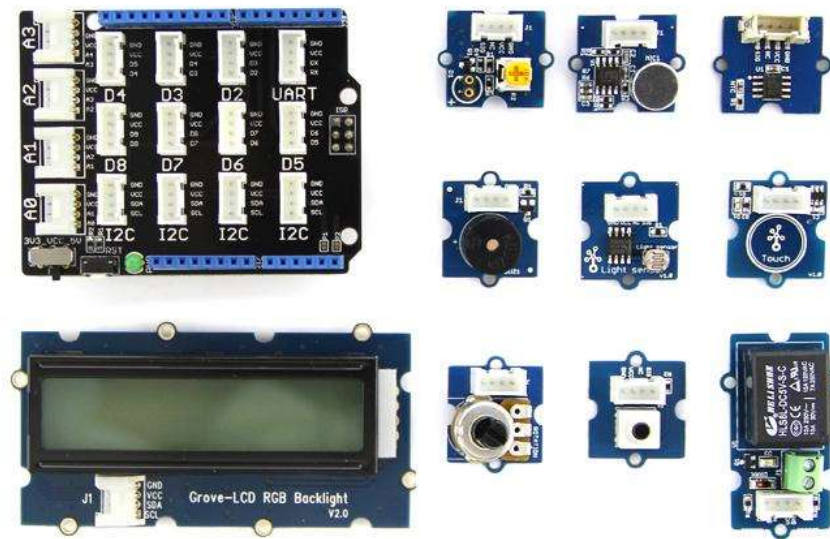
Nota. Adaptado de Shields Multi – propósito, de Electrónica y Robótica Julpin, 2018, Julpin. (Julpin, 2018).

- ***Shields de actuadores.***

Los shields enfocados en la parte de actuadores empiezan desde pequeños módulos como micrófonos, luces hasta pantallas LCD y demás, como en la Figura 9. Estos módulos al igual que los sensores utilizan diferentes tipos de comunicación y en su defecto la mayoría de control mediante los puertos digitales de la tarjeta controladora Arduino (Moreno, 2018).

Figura 9

Shields de actuadores compatibles con arduino.



Nota. Adaptado de Shields Arduino, por Aprendiendo Arduino, 2017, Aprendiendoarduino.wordpress.

(Aprendiendo Arduino, 2017).

- ***Shields de comunicación.***

En los shields de comunicación se los puede separar en dos grupos grandes: comunicación alámbrica e inalámbrica. Dentro de la comunicación inalámbrica se puede identificar a los módulos Bluetooth, WIFI y GSM como sus mayores representantes. Por otro lado, en la comunicación alámbrica se tiene al Serial, SPI, ETHERNET, I2C, MODBUS como algunos de los representantes de este tipo de módulos, como se observa en la Figura 10 (Moreno, 2018).

Figura 10

Shield de comunicación compatible con arduino.



Nota. Adaptado de Conectar Arduino a Internet, por Luis Llamas, 2021, Twitter. (Llamas, 2021).

2.4. Aprendizaje de Máquina

El aprendizaje de máquina es un campo de la inteligencia artificial que propone a los sistemas la capacidad de aprender y mejorar de manera automática, busca patrones en datos empíricos sin forzar modelos en los datos, es decir, el enfoque se basa en datos y no en modelos (Oshurko, 2016).

Desde la época de los 40, una serie de científicos se basaron en el teorema de Bayes (1812) para entender que la probabilidad de que un evento suceda depende de las condiciones previas relacionadas con el evento. He aquí que el machine Learning puede ser definido como “el maestro del reconocimiento de patrones”. Esta perspectiva y la llegada del internet hace que cantidades masivas de información entrenen los modelos para conocer las condiciones previas a cada evento que se necesite (BBVA, 2019).

Los tipos de aprendizaje de máquina se rigen según el enfoque de aprendizaje, los tipos de datos que ingresan y producen, y el tipo de problema que resuelven, existen categorías principales de algoritmos de aprendizaje automático: aprendizaje supervisado, no supervisado y por refuerzo, los cuales según (García, Cristina Gómez, 2006) son descritos a continuación:

2.4.1. Aprendizaje supervisado

El aprendizaje supervisado se aplica cuando los datos están en forma de variables de entrada y valores objetivo de salida. El algoritmo aprende la función de mapeo de la entrada a la salida. La disponibilidad de muestras de datos etiquetadas a gran escala lo convierte en un enfoque costoso para tareas en las que los datos son escasos.

Mientras quien opera el sistema conozca las respuestas correctas al problema que se soluciona, el algoritmo identifica patrones, aprende de las observaciones realizadas haciendo que se generen predicciones. Finalmente, el algoritmo genera sus predicciones las cuales son corregidas por el operador para ser pulido hasta tener un alto nivel de precisión y rendimiento (Soto & Jiménez, 2011).

2.4.2. Aprendizaje no supervisado

El aprendizaje no supervisado se aplica cuando los datos están disponibles sólo en forma de entrada y no hay una variable de salida correspondiente. Dichos algoritmos modelan los patrones subyacentes en los datos para aprender más sobre sus características. Uno de los principales tipos de algoritmos no supervisados es la agrupación en clústeres. En esta técnica, los grupos inherentes a los datos se descubren y luego se utilizan para predecir la salida de entradas invisibles.

En los procesos de aprendizaje no supervisado el algoritmo de aprendizaje automático interpreta gran cantidad de conjuntos de datos para intentar organizar y generar una estructura con los datos. Es importante mencionar que mientras más datos evalúa el algoritmo aumenta su capacidad de tomar decisiones con una mejora gradual (Cáceres, 2002).

2.4.3. Aprendizaje por refuerzo

El aprendizaje por refuerzo se aplica cuando la tarea en cuestión es tomar una secuencia de decisiones hacia una recompensa final. Durante el proceso de aprendizaje, un agente artificial obtiene recompensas o penalizaciones por las acciones que realiza. Su objetivo es maximizar la recompensa total.

Una vez definidas las reglas el algoritmo de aprendizaje automático explora diferentes opciones y posibilidades, este algoritmo evalúa cada resultado para obtener el óptimo. El sistema de enseñanza es a través de ensayo y error (Fernández, 2013). Dentro del aprendizaje por refuerzo se puede utilizar diferentes tipos de algoritmos los cuales se introducen a continuación.

2.4.3.1. Algoritmos de Clasificación.

Los algoritmos de clasificación son esenciales para predecir valores de clase de las nuevas instancias, observaciones o registros. Estos cumplen las funciones de predecir instancias sin etiquetar y describir las características que rigen en instancias de diferentes clases. Estos se basan en la minería de datos y aprendizaje de máquina (Ochoa, 2019).

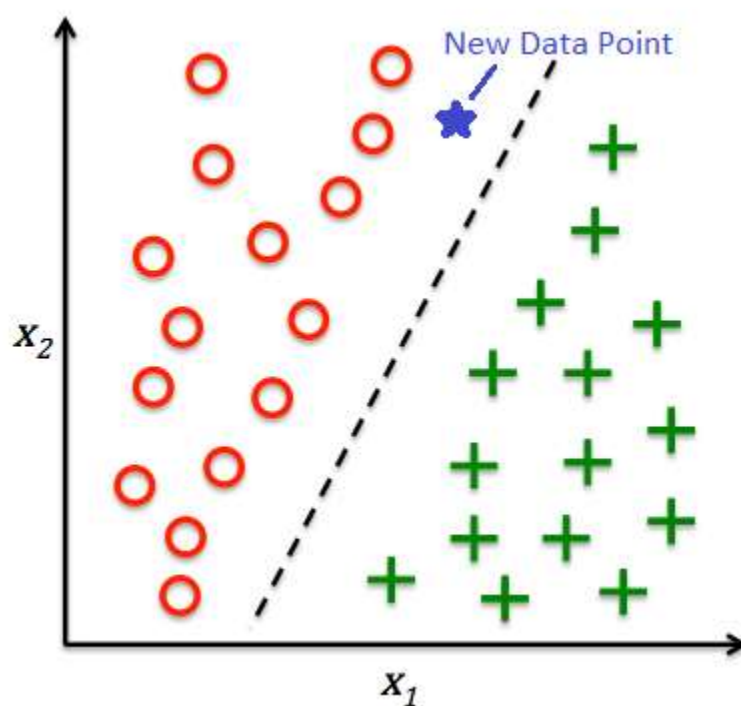
La clasificación dentro del aprendizaje supervisado es una subcategoría que tiene como objetivo predecir las etiquetas de las clases categóricas, basándose en las observaciones anteriores (Román, 2019a). Existen dos tipos principales de clasificaciones:

2.4.3.2. Clasificación binaria.

Es la clasificación donde sólo se puede otorgar dos valores diferentes, 0 o 1. En caso de ser verdadera la asignación se le otorga 1 y en caso contrario se le otorga 0. Para un ejemplo práctico se tiene dos datos (círculos y cruces) y dos características (X_1 y X_2) los cuales para observar su clasificación se los separa por medio de una línea segmentada. Una vez que se ingrese un nuevo dato este será clasificado según los datos y características ya analizadas como se observa en la Figura 11.

Figura 11

Ejemplo de clasificación binaria.



Nota. Adaptado de Aprendizaje Supervisado, por Victor Román, 2019, Medium. (Román, 2019b).

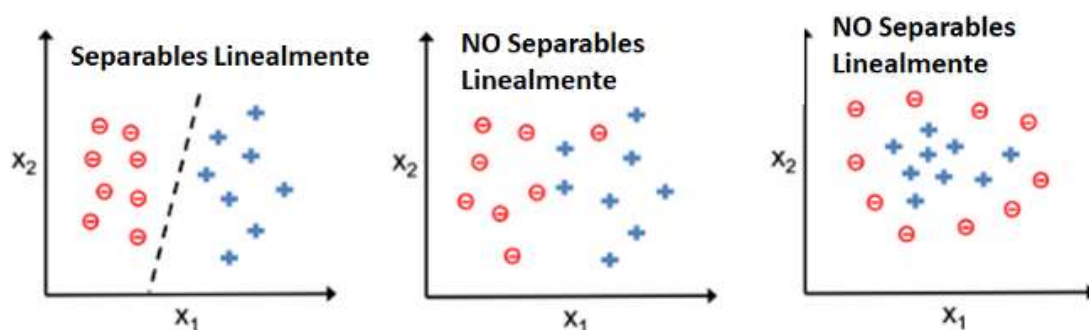
2.4.3.3. Clasificación multi clase.

En esta clasificación se le puede otorgar múltiples asignaciones como el reconocimiento de caracteres de números, del 0 al 9.

No todos los modelos de clasificación serán adecuados para las diferentes clases de conjuntos de datos. Todos convergen a una frontera de decisión lineal y en la multi clase algunos de los casos más típicos se los puede observar en la Figura 12.

Figura 12

Ejemplo de clasificación de diferentes clases.



Nota. Adaptado de Aprendizaje Supervisado, por Victor Román, 2019, Medium. (Román, 2019b).

2.4.3.4. Clasificación en la práctica.

Al implementar el Machine Learning es recomendable probar y comprobar más de un algoritmo. Este comportamiento está influenciado por los datos disponibles, número de características, clases y si son linealmente separables o no (Román, 2019a).

Existen 6 pasos principales para desarrollar un modelo de Machine Learning:

- Recopilar datos.
- Elegir una medida o indicador de éxito.
- Establecer un protocolo de evaluación.

- Preparar datos.
- Desarrollar un punto de referencia del modelo.
- Desarrollar un buen modelo y ajuste fino de sus parámetros.

2.4.3.5. Weka

Weka es una colección de algoritmos de aprendizaje automático ideal para resolver problemas en el mundo real se define como Waikato Environment for Knowledge Analysis, está escrito en java y se encuentra disponible para cualquier plataforma. Es una plataforma ideal ya que permite añadir implementar diferentes algoritmos de modelado (Lorena, 2019).

Esta plataforma gráfica se encuentra destinada al aprendizaje automático y la minería de datos. En resumen, Weka se define actualmente como un conjunto de librerías que pueden ser llamadas desde la interfaz de la aplicación. Weka contiene muchas herramientas las cuales realizan tareas básicas como:

- Reprocesó: alberga múltiples herramientas para el preprocesamiento de datos.
- Clasificar: posee algoritmos de clasificación distribuidos en paquetes.
- Racimos: posee diferentes algoritmos de segmentación como simple k-means.
- Asociado: posee algoritmos para encontrar relaciones de asociación entre diferentes variables.
- Selección de atributos: posee la capacidad de cargar datos y mediante diferentes variables genera modelos en base a los datos añadidos.
- Visualización: posee herramientas de visualización que se grafican en diferentes planos para ser analizados.

2.5. Metodologías de Gestión de Proyectos

Una metodología de gestión de proyectos es una sistematización del proceso de desarrollo. Es el conjunto de pasos organizados por los cuales se regirá la ejecución de proyectos (P. González, 2007). Utilizar una metodología de gestión de proyectos permite:

- Intercambiar experiencias o mejoras prácticas.
- Mejorar la comunicación.
- Estandarizar las actividades.
- Asegurar un proceso visible.
- Aportar herramientas.
- Mejorar el rendimiento.
- Motivar la ejecución.

2.5.1. Tipos de Metodologías de Gestión de Proyectos

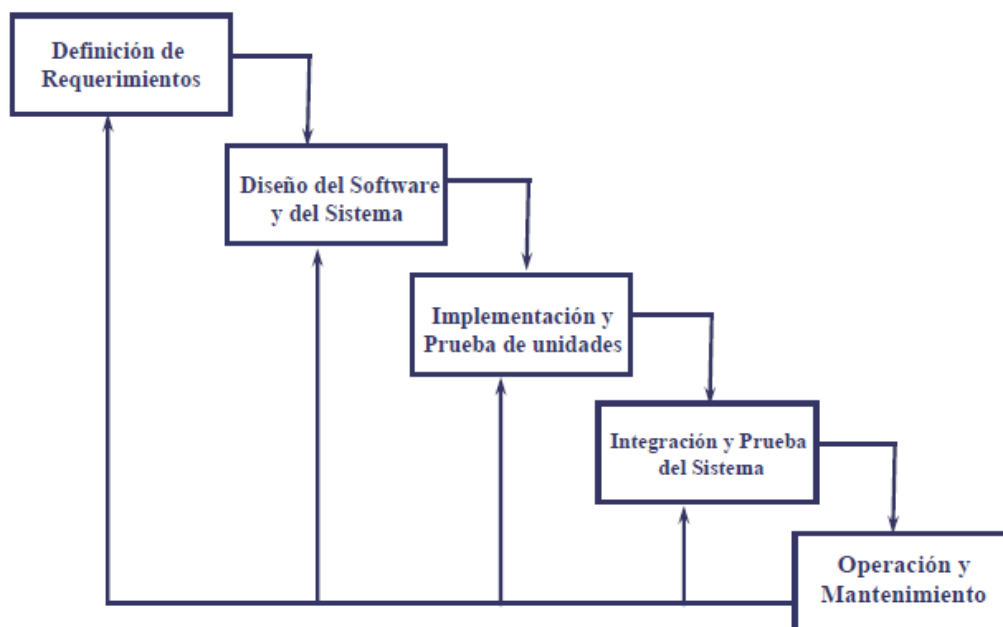
Existen diferentes metodologías para el desarrollo de proyectos, cada una tiene sus ventajas y desventajas al momento de compararlas. Entre las más utilizadas se pueden describir cuatro metodologías:

2.5.1.1. Metodología Cascada.

Es una metodología que ordena de manera rigurosa cada uno de las etapas de desarrollo de un proceso y que además cada una de las etapas debe finalizarse antes de continuar con la siguiente etapa (F. González & Calero, 2019). El proceso lineal y secuencial que sigue esta metodología se describe en las siguientes fases: Fase de análisis de requisitos, Fase de diseño, Fase de implementación, Fase de verificación y Fase de mantenimiento. A continuación, se detallan cada una de las fases enlistadas.

Figura 13

Diagrama de los pasos de la metodología de cascada.



Nota. Adaptado de Modelo Cascada Realimentado, 2020, 4tesosite. (4tesosite, 2020).

- ***Fase de análisis de requisitos.***

En esta fase se analizan las necesidades de los usuarios finales en base, aquí se obtiene la especificación de lo que tiene que hacer el sistema sin entrar en detalles técnicos.

- ***Fase de diseño.***

En esta fase se descompone y se organiza el sistema y sus elementos en hardware y software (dependiendo el caso). En esta parte se obtiene la descripción de la estructura de todo el sistema y las especificaciones de cada una de las partes que lo componen y su interacción entre ellas.

- ***Fase de implementación.***

En esta tercera fase se procede a la construcción, y montaje de todas las partes especificadas en un sistema global y completo.

- ***Fase de verificación.***

La cuarta fase se centra en probar, encontrar y solucionar posibles errores del sistema en cada una de sus partes y en el sistema en general.

- ***Fase de mantenimiento.***

En esta fase final se comienza la producción del sistema y la identificación de nuevos errores y adaptaciones que se deben ir aplicando para el correcto y duradero funcionamiento del sistema.

2.5.1.2. Metodología de Cadena Crítica.

Esta metodología se basa en la teoría de las restricciones, la cual está pensada para optimizar el progreso del proyecto, teniendo en cuenta que los proyectos están siempre sometidos a la incertidumbre y limitaciones. Se basa en tres principios:

- Identificar las restricciones que pueden afectar al proyecto.
- Priorizar las tareas dentro de la cadena.
- Subordinar las tareas restantes en la cadena.

2.5.1.3. Metodología Ágil.

Se fundamenta en la flexibilidad de los productos y servicios durante el proyecto. Esta metodología divide el proyecto en fases las cuales se cumplen para resultar en diferentes funcionalidades. Las fases se rigen a un esquema de la siguiente manera:

- Inicio: objetivos.
- Desarrollo: seguimiento.
- Cierre: revisión.

2.5.1.4. Metodología de Híbrida.

Esta es una de las metodologías más actuales, la cual combina los mejores rasgos de cada una de las metodologías anteriores donde el tiempo es apremiante y su desarrollo debe ser en periodos cortos y flexibles. La clave en esta metodología es la continua adaptación al cambio y la innovación.

3. CAPÍTULO II: DESARROLLO

En el presente capítulo se desarrolla el software y hardware del prototipo del sistema de monitoreo de variables de calidad de agua para el criadero de truchas Aguas Verdes en el cual se utilizará un sistema que utilizando la comunicación Modbus y estructurado en Arduino será la síntesis de lo previamente investigado.

3.1. Selección de metodología para el desarrollo

La implementación de un sistema embebido para la automatización de procesos y recursos se ha vuelto una manera óptima para aumentar la rentabilidad de muchas actividades económicas.

Escoger una metodología para desarrollar cualquier sistema embebido es una herramienta propicia para tener los mejores resultados. Para este proyecto se trabajará bajo los lineamientos de la metodología de cascada, la cual, ejecuta cada fase hasta estar completa para continuar con la siguiente, esto garantiza que cada fase cumpla con los resultados establecidos en las especificaciones preestablecidas.

3.2. Metodología de Cascada: Fase 1 Análisis de Requisitos

Esta primera fase se centra en la recolección de información la cual estructura requisitos y lineamientos por parte del cliente, tesista, directores y beneficiarios. En esta fase utiliza diferentes herramientas para obtener, analizar e interpretar la información obtenida.

3.2.1. Situación inicial

En la siguiente sección se da a conocer el estado o situación inicial del lugar o beneficiarios en donde el proyecto a desarrollarse se implementará. En base al análisis de la información

brindada se establece varios requerimientos relacionados a los componentes de hardware y software necesarios para estructurar el sistema con sus entradas y salidas.

3.2.2. Técnicas de recolección de información

Dentro de la investigación y el análisis de la información existen diferentes técnicas útiles que permiten desenvolver los datos y sintetizarlos en los requerimientos. Una de estas es la entrevista la cual es una interacción entre una o varias personas las cuales responden a debidas preguntas establecidas.

3.2.2.1. Entrevista.

En la realización de la entrevista se busca reunir la mayor cantidad de información que cubra los datos referentes a la funcionalidad del prototipo a implementarse: el uso, la comunicación, el hardware, el software, las entradas y las salidas para el diseño y el montaje del futuro sistema. Además, es recomendable estar bien empapado del tema a tratar para obtener la mayor cantidad de información que sirva a la investigación y logre cubrir hasta el mínimo detalle que pueda ser importante.

Para este proyecto la entrevista al encargado del lugar beneficiario que se encuentra en el Anexo 1 recogió información de la labor cotidiana, experiencias pasadas y expectativas para caracterizar el sistema.

3.2.2.2. Resultados de la entrevista.

Con el análisis de la información obtenida se puede generar la Tabla 1 con la recopilación de los puntos más relevantes de todo lo tratado en la entrevista del Anexo 1.

Tabla 1

Tabla resumen de información obtenida por la entrevista realizada al encargado.

REQUERIMIENTOS OBTENIDOS DE ENTREVISTA				
REQUERIMIENTOS DE USUARIO	REQUERIMIENTOS TÉCNICOS	PRIORIDAD		
		BAJA	MEDIA	ALTA
Medición de pH en el agua	Utilización de un sensor de pH con electrodo			X
Medición de oxígeno en el agua	Utilización de un sensor de oxígeno disuelto			X
Estaciones separadas	Gabinetes modulares		X	
Que se conecten entre ellas	Conexión alámbrica de media/larga distancia (Modbus)			X
Visualización de datos	HMI de visualización de datos			X
Alarmas	Activación de alarmas vía relé y SMS			X
Almacenamiento de datos	Almacenamiento en una memoria extraíble		X	
Sistemas pequeño	Gabinetes modulares no invasivos		X	
Resistencia de los equipos	Recubrimiento e impermeabilidad media/alta			X

3.2.3. Propósito y ámbito del sistema

El propósito de este proyecto es implementar un equipo electrónico de monitoreo el cual realiza un seguimiento de la calidad del agua que ingresa a las piscinas de cuidado y crecimiento de truchas mediante sensores de variables climáticas en gabinetes modulares para ser visualizada la información en una pantalla instalada en una estación central.

El objetivo de la implementación de este equipo busca mejorar la producción de truchas y lograr establecer las características óptimas de una piscina que pueda tener la mayor producción posible aprovechando el medio donde se encontrará instalada. Como punto de partida en la Tabla 2 se tiene las condiciones actuales del lugar y su producción.

Tabla 2

Condiciones actuales del criadero y su producción

CONDICIONES ACTUALES DEL CRIADERO	
ASPECTOS	ACTUALMENTE
Optimización de la calidad del agua	<i>Oxigenación de 2 por secciones</i>
Producción de truchas para la venta	<i>Salida de 100 truchas a la semana</i>
Optimización del espacio de la piscina	<i>55% del espacio de la piscina utilizado</i>
Respuesta ante alteraciones de los parámetros de calidad del agua	<i>Tiempo de respuesta de 1h</i>
Reducción de peces muertos	<i>Cantidad aproximada de 20 peces a la semana</i>
Cantidad de peces por mm ²	<i>15</i>
Tiempo de crecimiento	<i>1 - 2 meses</i>

3.2.4. Descripción del sistema

El equipo de monitoreo de la calidad del agua utiliza sensores de: temperatura, pH, oxígeno disuelto y nivel de agua para la medición del estado del agua que ingresa a las piscinas de producción de las truchas, junto a la electrónica versátil de Arduino estos sensores se enlazan a tarjetas arduino las cuales se comunican mediante el protocolo Modbus.

La información que se obtiene del agua que ingresa a las piscinas se dirige a una estación central en donde es proyectada mediante una pantalla y de manera paralela se utiliza para activar alarmas relacionadas al nivel de agua o la variación abrupta de los valores de las variables climáticas proporcionada por los sensores, adicionalmente todos los datos son almacenados en una tarjeta microSD, donde todos los datos alimentaran al algoritmo de aprendizaje de máquina.

3.2.4.1. Riesgos.

Antes de realizar el diseño se estima algunos riesgos potenciales que puedan afectar al sistema. Enlistándolos a continuación:

- En las pruebas del sistema el hardware no responda como se debería.
- Las condiciones climáticas del lugar donde se implemente dañen algún componente del hardware.
- La distancia de los nodos de comunicación afecte el recibir y enviar información.
- El tiempo de montaje tome más tiempo del esperado.
- La sensibilidad del hardware a utilizarse.
- El tipo de comunicación no cumpla con la eficiencia esperada.
- Se necesite más hardware del esperado en el transcurso del proyecto.

3.2.4.2. Restricciones

De igual manera que los posibles riesgos se puede estimar ciertas restricciones que tendrá el sistema en su montaje, ejecución o uso. A continuación, se las enlistan:

- El personal que podrán manejar el sistema.
- El sistema trabaja en una configuración esclavo - maestro.
- Software de programación libre.
- Hardware modular y adaptable adquirible en el mercado nacional.
- Hardware compatible con Arduino y Modbus.
- Interfaz gráfica, minimalista e intuitiva.

3.2.5. Determinación de Stakeholders

Los Stakeholders o partes interesadas son los sujetos involucrados con el desarrollo del equipo o dispositivo. Los Stakeholders son quienes tienen algún interés en el desarrollo, implementación o uso del proyecto y con ello definir los requerimientos que debe cumplir con cada objetivo (calidad del agua). En la Tabla 3 se observan los Stakeholders definidos para este proyecto.

Tabla 3

Stakeholders.

Orden	Stakeholders
1	Dueño del Criadero Aguas Verdes
2	Fundador del Criadero Aguas Verdes (José Domingo Paspuezán)
3	Director del proyecto (MSc. Jaime Michelena)
4	Asesor 1
5	Asesor 2
6	Autor de la Investigación (Sr. Edison Paspuezán)

3.2.6. Definición de Acrónimos y Abreviaturas

En el desarrollo de este los siguientes capítulos se necesitarán algunos acrónimos y abreviaturas con frecuencia los cuales se presentan en la Tabla 4.

Tabla 4

Lista de Acrónimos y Abreviaturas.

Abreviatura	Descripción
UTN	Universidad Técnica del Norte
CIERCOM	Carrera de Ingeniería en Electrónica y Redes de Comunicación
FICA	Facultad de Ingeniería en Ciencias Aplicadas
RSSt	Especificación de Requerimientos de los Stakeholders
RSSi	Especificación de Requerimientos del Sistema
RSEs	Especificación de Requerimientos de la Estructura

3.3. Metodología de Cascada: Fase 2 Diseño

La segunda fase dentro de la metodología de cascada a formular una solución en base a las exigencias, requerimientos o necesidades extraídas en la fase anterior. En esta fase se diseña un plan detallado de la arquitectura del proyecto haciendo énfasis en los componentes, interfaz, entorno de trabajo. A continuación, se detallan los requerimientos obtenidos.

3.3.1. Requerimientos operacionales y de usuario

En las Tabla 5 y Tabla 6 se observan los requisitos iniciales para el desarrollo del sistema de monitoreo de calidad de agua, los requisitos operacionales y de usuario, todos estos lineamientos permiten visualizar el funcionamiento y el nivel de prioridad de cada uno.

Tabla 5*Requerimientos operacionales del sistema.*

RSSt					
REQUERIMIENTOS OPERACIONALES					
NÚMERO	REQUERIMIENTOS	PRIORIDAD			RELACIÓN
		ALTA	MEDIA	BAJA	
RSSt 1	La comunicación debe ser mediante protocolo Modbus.	X			
RSSt 2	Transmisión y recepción de datos alta.	X			
RSSt 3	Mediciones de la Sensórica exactas y precisas.	X			
RSSt 4	Sistema implementado en el criadero “Aguas Verdes”.	X			

Tabla 6*Requerimientos de usuario del sistema.*

RSSt					
REQUERIMIENTOS DE USUARIO					
NÚMERO	REQUERIMIENTOS	PRIORIDAD			RELACIÓN
		ALTA	MEDIA	BAJA	
RSSt 5	Lenguaje C	X			
RSSt 6	Sistema modular		X		
RSSt 7	Visualización de la información obtenida			X	
RSSt 8	Sistema de alarmas		X		
RSSt 9	Bajo costo en Hardware		X		
RSSt 10	Software libre	X			
RSSt 11	Electrónica básica		X		
RSSt 12	Almacenamiento y análisis de datos.		X		

3.3.2. Requerimientos iniciales del sistema

Los requerimientos iniciales permiten elaborar el sistema electrónico, para esto se debe conocer el comportamiento del sistema y sus componentes para cumplir con las funciones que se desean. En la Tabla 7 se presenta algunos requerimientos iniciales que limitan al sistema que se desarrollará, los cuales son importantes en el diseño, implementación y uso del dispositivo.

Tabla 7

Requerimientos del sistema.

RSSi					
REQUERIMIENTOS DEL SISTEMA					
NÚMERO	REQUERIMIENTOS	PRIORIDAD			RELACIÓN
		ALTA	MEDIA	BAJA	
RSSi 1	Utilización de comunicación serial.	X			
RSSi 2	Fuentes de alimentación independientes.	X			
RSSi 3	Modularidad del sistema.		X		
RSSi 4	Alarmas.		X		
RSSi 5	Almacenamiento de datos.		X		
REQUERIMIENTOS DE USO					
RSSi 6	Manual de calibración.	X			
RSSi 7	Manual de mantenimiento.	X			
RSSi 8	Datasheet de sensores.		X		
RSSi 9	Extracción de datos almacenados.			X	
REQUERIMIENTOS DE CONEXIÓN EXTERNA					
RSSi 10	Comunicación mediante Modbus.	X			
RSSi 11	Conexiones de comunicación alámbrica.	X			
REQUERIMIENTOS DE CONEXIÓN INTERNA					
RSSi 12	Conexión de entradas y salidas al sistema embebido.	X			
RSSi 13	Los sensores se conectan a una tarjeta de microcontrolador en cada gabinete modular.		X		
RSSi 14	Conexiones de alimentación y protección.	X			
REQUERIMIENTOS DE PERFORMANCE					
RSSi 15	Medidas precisas y exactas en los sensores.	X			
RSSi 16	Ubicación de las estaciones modulares.	X			
RSSi 17	Interferencias en la comunicación.		X		
REQUERIMIENTOS DE INTERFAZ					
RSSi 18	Interfaz minimalista y dinámica.	X			
RSSi 19	Interfaz visible en pantalla.	X			
REQUERIMIENTOS FÍSICOS					
RSSi 20	Tendido de cable de alimentación y datos.	X			
RSSi 21	Alimentación continua.		X		
RSSi 22	Empotramiento de las estaciones modulares.	X			

3.3.3. *Requerimientos iniciales de arquitectura*

En la arquitectura que se va a diseñar para el sistema es importante describir las características como limitaciones del software y del hardware, además de, requerimientos lógicos en la parte del diseño. Estos requerimientos se detallan en la Tabla 8.

Tabla 8

Requerimientos de estructura.

RSEs					
REQUERIMIENTOS DEL SISTEMA					
NÚMERO	REQUERIMIENTOS	PRIORIDAD			RELACIÓN
		ALTA	MEDIA	BAJA	
REQUERIMIENTOS DE DISEÑO					
RSEs 1	Software libre.	X			
RSEs 2	Hardware a un bajo costo.	X			
RSEs 3	Modularidad de componentes.	X			
RSEs 4	Compatibilidad de componentes	X			
RSEs 5	Variedad de componentes.	X			
REQUERIMIENTOS DE SOFTWARE					
RSEs 6	Facilidad de programar.		X		
RSEs 7	Compatibilidad con Arduino.		X		
RSEs 8	Paquetes y librerías accesibles.		X		
RSEs 9	Variedad de librerías y paquetes.	X			
REQUERIMIENTOS DE HARDWARE					
RSEs 10	Placa con microcontrolador.	X			
RSEs 11	Calibración sencilla en sensores.	X			
RSEs 12	Conexión de equipos sencilla.	X			
REQUERIMIENTOS DE INFRAESTRUCTURA					
RSEs 13	Bases de empotramiento de componentes.	X			
RSEs 14	Recubrimientos de componentes.	X			
RSEs 15	Impermeabilidad de las estaciones.	X			
RSEs 16	Recubrimiento del tendido de cables.	X			
REQUERIMIENTOS DE ELÉCTRICOS					
RSEs 17	Alimentación eléctrica.	X			
RSEs 18	Protecciones eléctricas.	X			
RSEs 19	Distribuidores de tensión.	X			

3.3.4. Selección de Software

Una vez finalizado la obtención de los requerimientos del sistema mediante el análisis de la situación del lugar a implementarse se procede a la selección del software más idóneo para la programación de las tarjetas micro controladas para el sistema embebido.

- **Selección del software**

Mediante la Tabla 9 se realiza una comparación de diferentes softwares de la programación del sistema en base a los requerimientos establecidos con anterioridad.

Tabla 9

Tabla comparativa de las alternativas de software.

SOFTWARE	REQUERIMIENTOS						VALORACIÓN TOTAL	
	RSSt 4	RSSt 9	RSEs 1	RSEs 6	RSES 7	RSEs 8		RSEs 9
IDE Arduino	1	1	1	1	1	1	1	7
Visual Studio	1	1	1	0	1	1	0	5
AVR Studio	1	1	1	0	0	0	1	4
Cumple = 1								
No Cumple = 0								
Elección: IDE Arduino								

- **Arduino IDE**

Arduino es una plataforma de código abierto para creadores y desarrolladores muy sencilla de utilizar, además de su simplicidad en su composición al programar hace que se pueda utilizar solo con revisar conceptos básicos. Este software tiene un entorno de desarrollo, IDE Arduino como se observa en la Figura 14, en el cual se puede crear diferentes composiciones de código para diferentes utilidades aplicables en todos los controladores compatibles con esta IDE.

Figura 14

Interfaz de la IDE de Arduino.



```

Blink §
This example code is in the public domain.

http://www.arduino.cc/en/Tutorial/Blink
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}

```

Nota. Adaptado de Arduino IDE, por Aprendiendo Arduino, 2019, Aprendiendoarduino. (Aprendiendo Arduino, 2019).

3.3.5. Elección de Hardware

En la selección de hardware se debe realizar un proceso comparativo para cada componente importante dentro del sistema embebido. Como se presenta a continuación.

- Tarjeta principal
- Adaptador
- Sensor de temperatura

3.3.5.1. Tarjeta microcontroladora.

Utilizando una tabla comparativa como se observa en la Tabla 10 se utilizarán los requerimientos previamente establecidos para la selección de la tarjeta procesadora.

Tabla 10

Tabla comparativa de las alternativas de tarjeta micro controlada.

PLACA	REQUERIMIENTOS													VALORACIÓN TOTAL
	RS St 1	RS St 2	RS St 4	RS St 5	RS St 9	RS St 10	RS St 11	RS Es 1	RS Es 2	RSE s 4	RS Es 6	RS Es1 2	RS Es1 7	
Arduino UNO	1	1	1	1	1	1	1	1	0	1	0	1	1	11
Arduino MEGA	1	1	1	1	1	1	1	1	1	1	0	1	1	12
Arduino Nano	1	1	1	0	0	0	1	1	0	0	1	1	0	7
Cumple = 1														
No Cumple = 0														
Elección: Arduino UNO – Arduino MEGA														

Para el caso de esta selección la tarjeta MEGA cumple con los requerimientos necesarios y una muy cercana es la tarjeta UNO; por motivo de tamaño se necesita una tarjeta más pequeña que la MEGA por lo que la alternativa más idónea es la tarjeta Arduino UNO cumpliendo con casi el mismo funcionamiento con dimensiones inferiores.

- **Arduino UNO**

El arduino UNO, Figura 15, es una placa de desarrollo de bajo costo estructurada con el microcontrolador Atmega328P. Su alta compatibilidad la vuelve una placa idónea para cualquier proyecto que utilice sensores, actuadores y demás componentes para estructurar un proyecto. A continuación, se presentan algunas de sus características más relevantes:

Figura 15*Arduino UNO R3.***Arduino Uno R3**

Es posiblemente la placa más utilizada y robusta. Estupénda opción para iniciarse en la programación de microcontroladores. Estas son sus características técnicas:

- Microcontrolador: ATmega328P.
- Velocidad de reloj: 16 MHz.
- Voltaje de trabajo: 5V.
- Voltaje de entrada: 7,5 a 12 voltios.
- Pinout: 14 pines digitales (6 PWM) y 6 pines analógicos.
- 1 puerto serie por hardware.
- Memoria: 32 KB Flash (0,5 para bootloader), 2KB RAM y 1KB Eeprom

Nota. Adaptado de Arduino, por Blog de Tecnologías, 2018, Gobiernodecanarias. (Blog de Tecnologías, 2018).

- **Arduino MEGA**

El Arduino MEGA, de la Figura 16, es el hermano mayor del Arduino UNO el cual cuenta con mayor capacidad en su memoria y cantidad de pines lo que hace que esta tarjeta de desarrollo sea implementada en proyectos más complejos con procesos y cantidad de periféricos mayor. Algunas de sus características principales se las presenta a continuación:

Figura 16*Arduino MEGA.***Arduino Mega**

Es usada para proyectos con necesidad de mayor cantidad de pines. Estás son sus características técnicas:

- Microcontrolador: ATmega2560.
- Velocidad de reloj: 16 MHz.
- Voltaje de trabajo: 5V.
- Voltaje de entrada: 7,5 a 12 voltios.
- Pinout: 54 pines digitales (15 PWM) y 16 pines analógicos.
- 3 puertos serie por hardware.
- Memoria: 256 KB Flash (8KB para bootloader), 8KB RAM y 4KB Eeprom

Nota. Adaptado de Arduino, por Blog de Tecnologías, 2018, Gobiernodecanarias. (Blog de Tecnologías, 2018).

3.3.5.2. Sensor de temperatura.

La temperatura es una variable relevante en el proyecto a realizarse por lo que se necesita un sensor con un recubrimiento apto para los líquidos y con una precisión y exactitud semejante a un sensor de altas especificaciones para esto se realiza una tabla comparativa como la Tabla 11.

Tabla 11

Tabla comparativa de las alternativas de sensor de temperatura.

Sensor Temperatura	REQUERIMIENTOS													VALORACIÓN TOTAL
	RS St 3	RS St 4	RS St 6	RS St 8	RS St1	RS Si	RS Si1	RS Es	RS Es	RS Es	RS Es	RS Es	RS Es1	
DTH11	0	1	1	1	1	1	1	1	1	1	0	1	0	10

PT100	1	1	1	0	1	1	1	1	0	1	1	1	1	11
LM35	0	1	1	1	1	0	1	1	1	1	0	1	0	9
DS18B20	1	1	1	0	1	1	1	1	0	1	1	1	1	11

Cumple = 1

No Cumple = 0

Elección: **DS18B20 (Recubrimiento impermeable)**

- **DS18B20**

El DS18B20 de la Figura 17 es un sensor de temperatura que proporciona la salida mediante un bus de comunicación digital que es leído por las entradas digitales de Arduino. Dispone de un rango de medición de -550C a +1250C con una precisión superior a $\pm 0.50C$ en el rango -100C de +850C.

Este tipo de sensor emplea un bus de comunicación 1-Wire, único conductor, para realizar la comunicación. Este, posibilita la conexión de diversos sensores a la vez y permite emplear cables de extensa longitud sin que se deteriore la comunicación. Además, dispone de un sistema de alarma que permite grabar en una memoria no volátil los límites inferiores y superiores del censado.

Figura 17

DS18B20 Sensor de temperatura.



Nota. Adaptado de DS18B20, por Murky Robot, 2018, Murkyrobot. (Murky Robot, 2018).

3.3.5.3. Sensor de pH.

En la calidad de agua una variedad relevante es la medición de pH en el líquido vital donde encontrar un equilibrio entre calidad y costo son dos de los principales factores a considerar en la Tabla 12 de selección.

Tabla 12

Tabla comparativa de las alternativas de sensor de pH.

Sensor PH	REQUERIMIENTOS													VALORACIÓN TOTAL
	RS St 3	RS St 4	RS St 6	RS St 8	RS St1	RS Si	RS Si1	RS Es	RS Es	RS Es	RS Es	RS Es	RS Es1	
PH E-201 C	1	1	1	1	1	1	1	1	1	1	1	1	1	13
PH BX-5	1	1	1	0	1	0	1	1	0	0	1	0	0	7
PH E-201	1	1	1	1	1	1	1	1	1	1	1	1	1	13
PH E-900	0	1	1	1	1	0	1	1	0	1	1	0	1	9
Cumple = 1														
No Cumple = 0														
Elección: PH E-201C (El modelo C posee un electrodo mejorado)														

- **PH E-201 C**

Para la conexión del sensor de la Figura 18 con cualquier microcontrolador se utiliza una comunicación SPI, por lo cual es ideal para trabajar con arduino uno, además es fácil de manejar para cualquier tipo de proyecto que requiera control de acceso donde algunas de sus características se pueden observar en la Tabla 13.

El módulo sensor PH-4502C es un dispositivo que permite la medición del PH y la temperatura de la solución, empleando un electrodo E201-BNC con un conector BNC.

Tabla 13*Características del sensor de pH.*

CARACTERÍSTICAS	
Alimentación	5V
Consumo	5 – 10mA
Rango de medición	0 – 14 pH
Temperatura de medición	0 – 80 0C
Precisión	±0.1 pH (25 0C)
Tiempo de respuesta	≤ 5s
Sonda de pH	Conector BNC
Controlador	pH 2.0
Indicador	LED

Figura 18*Sensor de pH con electrodo E 201-C.*

Nota. Adaptado de Sensor de PH líquido, por Unit Electronics, 2020, Uelectronics. (Unit Electronics, 2020b).

3.3.5.4. Interruptor de nivel.

El sistema necesita informar cuando el nivel de agua haya bajado cierta altura. De esta manera utilizando la Tabla 14 se procede a seleccionar un sensor de nivel apto para el sistema a implementarse.

Tabla 14

Tabla comparativa de las alternativas para interruptor de nivel.

Sensor Nivel	REQUERIMIENTOS													VALORACIÓN TOTAL
	RS St 3	RS St 4	RS St 6	RS St 8	RS St1	RS Si 8	RS Si1	RS Es	RS Es	RS Es	RS Es	RS Es	RS Es1	
					1		3	1	2	4	6	7	1	
EF16A 0028OK	1	1	1	1	1	1	1	1	1	1	1	1	1	13
CZP 5210	1	1	1	1	1	1	1	1	1	1	1	1	1	13
Cumple = 1														
No Cumple = 0														
Elección: CZP 5210 (Por su instalación como interruptor)														

- **CZP 5210**

El sensor de nivel de líquido de la Figura 19 es un interruptor flotador, este es empleado para medir el nivel de líquido dentro de un tanque. El sensor puede activar una bomba, una alarma y otros dispositivos.

Este tipo de sensor se compone de un interruptor y un flotador magnético de láminas, el interruptor está dentro del encapsulado de la barra guía, cuando el flotador sube y baja con el nivel de líquido, sus imanes internos atraen al interruptor de lámina magnéticos, con el fin de detectar la posición de nivel de líquido como un control de nivel de líquido ON/OFF.

Información adicional se la puede visualizar en la Tabla 15.

Tabla 15*Características del interruptor de nivel.*

CARACTERÍSTICAS	
Sonda galvánica	No necesita polarización
Fuente de alimentación	3.3 - 5.5V
Salida analógica	0 a 3.0V
Interfaz de gravedad	Plug and play
Rango de detección de sonda galvánica	0 a 20 mg/L
Longitud de cable	2 metros

Figura 19*Interruptor de nivel de agua.*

Nota. Adaptado de Sensor de Nivel, por Unit Electronics, 2020, Uelectronics. (Unit Electronics, 2020a).

3.3.5.5. Módulo de comunicación.

En la Tabla 16 se realiza una comparación de varios módulos de comunicación de los cuales se realizará la selección.

Tabla 16

Tabla comparativa de las alternativas para el módulo de comunicación modbus.

Comuni cación	REQUERIMIENTOS													VALORACIÓN TOTAL
	RSS t 1	RS St 2	RS St 4	RS St 6	RS St 9	RS St1 0	RS St1 1	RS Si1 0	RS Si1 1	RS Si1 2	RS Si1 7	RS Es 7	RS Es1 2	
RTU RS 232	1	1	1	1	1	1	1	1	1	1	1	0	0	11
RS 485	1	1	1	1	1	1	1	1	1	1	1	1	1	13
RS 485- USB	1	1	1	1	1	1	1	1	1	1	1	0	0	11
Cumple = 1														
No Cumple = 0														
Elección: RS-485														

- **RS485**

El Módulo RS-485, de la Figura 20, está basado en el circuito integrado MAX485 que convierte las señales TTL en señales diferenciales compatibles con el estándar RS-485.

Además, la comunicación a través de este módulo solo puede ser half-dúplex y el modo de funcionamiento, ya sea de transmisión o receptor, se controla mediante pines destinados a este propósito. En la Tabla 17 se puede observar algunas características adicionales de este módulo compacto.

Tabla 17

Características del módulo Modbus RS485.

CARACTERÍSTICAS	
Configuración de línea	Diferencial
Modo de operación	Simplex or half dúplex
Máxima longitud de cable	4000 ft
Máxima velocidad de datos	10 kbits/s

Niveles lógicos típicos	± 1.5 to $\pm 6V$
Impedancia mínima de entrada del receptor	12Ω
Sensibilidad del receptor	± 200 mV

Figura 20

Módulo Modbus RS485 para compatible con arduino.



Nota. Adaptado de Módulo Conversor TTL a RS485, por AV Electronics, 2019, Avelectronics. (AV Electronics, 2019).

3.3.5.6. Sensor de oxígeno disuelto.

Los sensores de oxígeno disuelto normalmente utilizan un electrodo que por medio de una reacción química generan un voltaje leíble mediante el transductor del sensor y entendible por el controlador, el proceso de selección se realizará mediante la utilización de la Tabla 18 donde comparan varios sensores.

Tabla 18

Tabla comparativa de las alternativas para el sensor de oxígeno disuelto en líquidos.

Sensor Oxígeno	REQUERIMIENTOS													VALORACION TOTAL
	RS St 3	RS St 4	RS St 6	RS St 8	RS St1	RS Si	RS Si1	RS Es	RS Es	RS Es	RS Es	RS Es	RS Es1	
SEN0237-A	1	1	1	1	1	1	1	1	0	0	1	1	1	11

ENV- 40_DOX	1	1	1	1	1	1	1	1	1	0	0	0	1	1	10
Cumple = 1															
No Cumple = 0															
Elección: SEN0237-A															

- **SEN0237-A**

Este sensor es empleado en la medición de oxígeno disuelto en agua, Figura 21, con la cual se refleja la calidad de agua. Para este tipo de sensor no es importante considerar una polarización para la adquisición de las mediciones, además la solución presente en el cabezal de la membrana puede reemplazarse posibilitando la disminución del costo de mantenimiento del sensor y la señal de salida obtenida por el sensor es fácilmente acoplable a sistemas de control o de detección, características que se extrajeron de la Tabla 19.

Tabla 19

Características del sensor de oxígeno disuelto.

CARACTERÍSTICAS	
Sonda	No necesita tiempo de polarización
Voltaje de alimentación	3.3 – 5V
Salida analógica	0 – 3V
Interfaz	Plug and Play
Rango de presión máxima de trabajo	50PSI
Vida útil del electrodo	1 año en uso normal
Periodo de reemplazo de la solución en el sensor	una vez por mes
Periodo de reemplazo de la membrana	1 – 2 meses
Longitud del cable	2 metros

Conector de la sonda

tipo BNC

Figura 21*Kit de sensor de oxígeno disuelto.*

Nota. Adaptado de Sensor de oxígeno disuelto, por DFROBOT, 2018, Amazon. (DFROBOT, 2018).

3.3.5.7. Módulo de almacenamiento.

En la Tabla 20 se emplea una tabla para seleccionar el módulo que ayudará con el almacenamiento de datos en una micro-SD extraíble.

Tabla 20*Características del módulo de almacenamiento extraíble.*

Almacena miento	REQUERIMIENTOS												VALORACI ÓN TOTAL	
	RS St 4	RS St 6	RS St 9	RS St1 0	RS St1 1	RS Si 5 1	RS Si 9 1	RS Si 17	RS Es 2	RS Es 4	RS Es 6	RS Es 7		RS Es1 2
Micro -SD	1	1	1	1	1	1	1	1	1	1	1	1	0	12
SD	1	1	1	1	1	1	0	1	1	0	1	1	0	10
Cumple = 1														
No Cumple = 0														
Elección: Micro-SD														

- **MicroSD**

Este módulo de tarjeta MicroSD que se observa en la Figura 22 permite la conexión

de un microcontrolador una tarjeta de memoria, de forma que se pueden almacenar datos obtenidos de algún tipo de sensor. Además, está diseñado para acceder a la memoria MicroSD mediante la comunicación SPI, como se observa en la Tabla 21, por lo que las señales de control se etiquetan con los nombres de las señales en dicho bus de comunicaciones.

Tabla 21

Características del módulo microSD.

CARACTERÍSTICAS	
Voltaje de operación	4.5V – 5.5V
Voltaje de interfaz SPI	3.3V – 5V
Corriente de operación	200 uA – 200mA
Soporta memoria SD hasta	32 GB
Tipo de comunicación	SPI
Dimensiones	42 mm x 24 mm

Figura 22

Módulo de almacenamiento en microSD.



Nota. Adaptado de Módulo Lector de Memoria Micro SD Card, por Naylamp Mechatronics, 2020.

(Naylamp Mechatronics, 2020).

3.3.5.8. Módulo de tiempo real.

En la Tabla 22 se observan los requerimientos o necesidades previamente establecidos para la selección del módulo de tiempo real que servirá para mantener un registro de las lecturas de los sensores con fecha y hora.

Tabla 22

Características del módulo de tiempo real.

Tiempo Real	REQUERIMIENTOS												VALORACIÓN TOTAL
	RSS t 3	RS St 4	RS St 9	RS St 10	RS St 11	RS Si 3	RS Si 8	RS Si 15	RS Es 1	RS Es 2	RS Es 6	RS Es 7	
RTC DS 1307	1	1	1	1	1	0	1	0	1	1	1	1	11
RTC DS 1302	1	1	1	0	1	1	1	0	1	1	0	1	10
RTC DS 3231	1	1	0	1	1	1	1	1	1	0	1	1	11
Cumple = 1													
No Cumple = 0													
Elección: RTC DS 3231 (Menor des calibración en el periodo de un año)													

- **RTC DS3231**

El DS3231 es un módulo de reloj de tiempo real de alta precisión, Figura 23, con un oscilador integrado con compensación de temperatura de cristal (TCXO). El módulo tiene incorporado una batería que cuando se desconecta la fuente de alimentación mantiene el cronometraje. Además, integra un oscilador que mejora la precisión a largo plazo y reduce el número de componentes en la línea de producción. Este módulo mantiene su medida en segundos, minutos, horas, días, fecha, mes, año y cualquier tipo de información deseada.

Este tipo de módulo cuenta con una compensación de temperatura de precisión de referencia de voltaje y circuito comparador el cual supervisa el estado de VCC para detectar fallas de energía, proporcionar una salida de reset, y si es necesario, cambiar automáticamente a la fuente de alimentación de reserva, algunas características adicionales se observan en la Tabla 23.

Tabla 23

Características del módulo de tiempo real DS3231.

CARACTERÍSTICAS	
Tamaño	38 mm (longitud) x 22mm (base)
Peso	8g
Voltaje de funcionamiento	3.3 -5.5V
Chip de reloj	DS3231
Precisión de Reloj	0-40 ⁰ C, el 2ppm precisión, el error de 1 min.
Salida de onda	Cuadrada programable
Tiempo real	Segundo, minutos y horas
Precisión de sensor de temperatura	±3 ⁰ C
Chip de memoria	AR24C32
Interfaz de bus	11.IIC
Velocidad de transmisión	400 KHz

Figura 23

Módulo RTC DS3231.



Nota. Adaptado de Módulo reloj de tiempo real RTC DS3231, por El Octavo bit, 2020, Elocavobit. (El Octavo bit, 2020).

3.3.5.9. Relé.

Para la activación de periféricos con un accionamiento ON/OFF se utiliza un relé el cual mediante la Tabla 24 se procede a la selección de este componente.

Tabla 24

Características del módulo del relé.

Relé	REQUERIMIENTOS												VALORACIÓN TOTAL
	RSS t 3	RS St 4	RS St 9	RS St 10	RS St 11	RS Si 3	RS Si 8	RS Si 15	RS Es 1	RS Es 2	RS Es 6	RS Es 7	
Relé 12V	1	1	1	1	1	1	1	1	1	1	1	1	13
Relé 5V	1	1	1	1	1	1	1	1	1	1	1	1	13
Contac tor 12V	1	1	0	1	1	1	1	1	1	1	1	1	12
Cumple = 1													
No Cumple = 0													
Elección: Relé 12V (El voltaje del periférico es de 12V)													

- **Relé 12V**

Este tipo de módulo es ideal para controlar cargas AC o DC que pueden ser controladas por un microcontrolador MCU o con placas Arduino como se observa en la Figura 24. El módulo cuenta con un Jumper Switch para configurar la activación de los relés ya sea con un pulso alto o bajo. Este funciona como un interruptor controlado por un circuito eléctrico en el que se acciona una serie de contactos que permiten abrir o cerrar otros circuitos eléctricos independientes, algunos datos adicionales se observan en la Tabla 25.

Tabla 25

Características del módulo relé 12V.

CARACTERÍSTICAS	
Voltaje de alimentación	12V DC
Señal de control	TTL (3.3 o 5V)
Para activar salida NO	0V
Nº de relés (canales)	2 CH
Código relés	Single SRD-12VDC-SL-C
Capacidad máx.	10A/250VAC, 10A/30VDC
Corriente máx.	10A (NO), 5A (NC)
Tiempo de acción	10ms/ 5ms
Entradas	Opto acopladas

Figura 24

Módulo de relé 12V.



Nota. Adaptado de Módulo Relay 2CH 5VDC, por Naylamp, 2020, Naylampmechatronics. (Naylamp, 2020).

3.3.5.10. Pantalla.

La pantalla es un periférico elemental en el proyecto por el cual se visualiza la información que se obtiene desde los sensores de cada módulo. El proceso de selección se ayuda de la Tabla 26.

Tabla 26

Características de las pantallas.

Pantalla	REQUERIMIENTOS													VALORACIÓN TOTAL
	RSS t 2	RS St 4	RS St 5	RS St 6	RS St 7	RS St 9	RS St 10	RS Si 18	RS Si 19	RS Es 1	RS Es 6	RS Es 7	RS Es1 2	
LCD 16X2	1	1	1	1	1	1	1	1	0	1	1	1	1	12
LCD 128x64	1	1	1	1	1	0	1	1	1	1	0	1	1	11
Touch TFT 7"	1	1	0	1	1	0	1	1	1	1	0	1	0	9
Cumple = 1														
No Cumple = 0														
Elección: LCD 128x64 (Mayor dimensiones que la LCD 16x2)														

- **LCD 128x64**

Estos tipos de pantallas LCD, algunas características se observan en la Tabla 26, son diseñadas para mostrar información en forma de grafica. Posibilitando la visibilidad tanto

de gráficos como de textos. Estas pantallas vienen unidas a una placa de circuito que cuentan con pines de entrada/salida de datos. Siendo necesario el uso de tarjetas controladoras, como Arduino, para desplegar información en ella como se visualiza en la Figura 27.

Tabla 27

Características de la pantalla LCD 128x64.

CARACTERÍSTICAS	
Modelo	YB12864ZB
Chip controlador	ST7920
Modelo placa	12864B V2.0
Voltaje de operación	5V DC
Consumo corriente Back light	360mA
Soporta interfaz	Paralelo (4 u 8 bits) y Serial SPI
Filas	64
Columnas	128
Dimensiones	93x70x13.5 mm

Figura 25

Pantalla LCD 128x64.



Nota. Adaptado de GLCD LCD 128x64, por Electro Store, 2020, Grupoelectrostore. (Electro Store, 2020b).

3.3.5.11. Alimentación.

Cada uno de los gabinetes tiene una conexión directa a los 110V de la red eléctrica mediante adaptadores de voltaje de 110V AC que la convierten a 9V DC a 2 A, como se observa en Figura 26, los cuales se conectan y suministran energía a las tarjetas arduino.

Figura 26

Adaptador de voltaje de 110V AC a 9V DC.



Nota. Adaptado de Adaptador Cargador 110V/9V 2ª PLUG DC005, por Electro Store, 2020, Grupoelectrostore. (Electro Store, 2020a).

3.4. Metodología de Cascada: Fase 3 Implementación

Finalizada la fase de diseño se procede a la tercera fase la cual detalla todo el proceso de composición del software, hardware de todo el sistema para ser probado e implementado. Además,

se observa una sinergia entre la fase de selección de los componentes y como estos se relacionan para tener un funcionamiento adecuado.

3.4.1. Diagramas de flujo

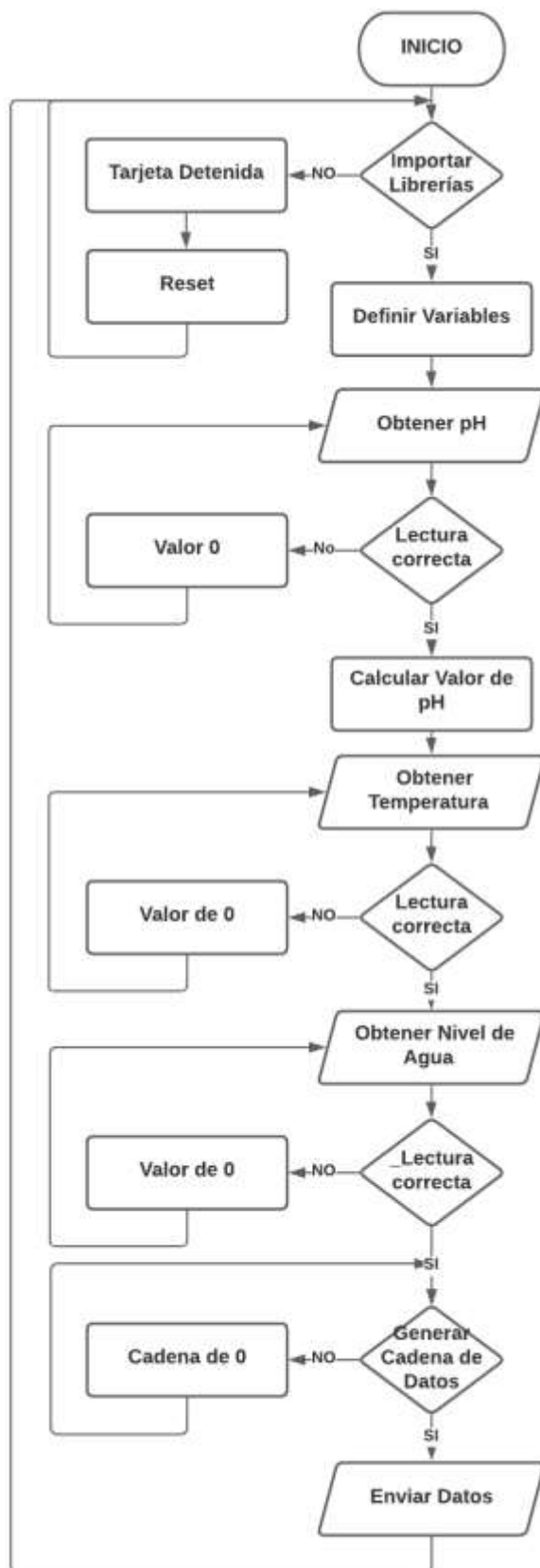
Los diagramas de flujos expresan de manera clara y simplificada los pasos o procesos continuos del sistema en funcionamiento. El sistema se fracciona en tres nodos los cuales dos son nodos de obtención de datos (sensores) y un nodo central donde se almacena, visualiza y generan el manejo de estos datos. De esta manera, los dos primeros diagramas de flujo indican los procesos de los gabinetes independientes de sensores y el último el diagrama de funcionamiento del panel general.

3.4.1.1. Nodo situado en la entrada de agua al criadero

El primer nodo obtendrá los datos de pH, temperatura y nivel de agua los cuales son enviados a posterior hacia el gabinete central. Este proceso se observa en el diagrama de la Figura 27.

Figura 27

Diagrama de flujo del nodo ubicada en la entrada de agua.

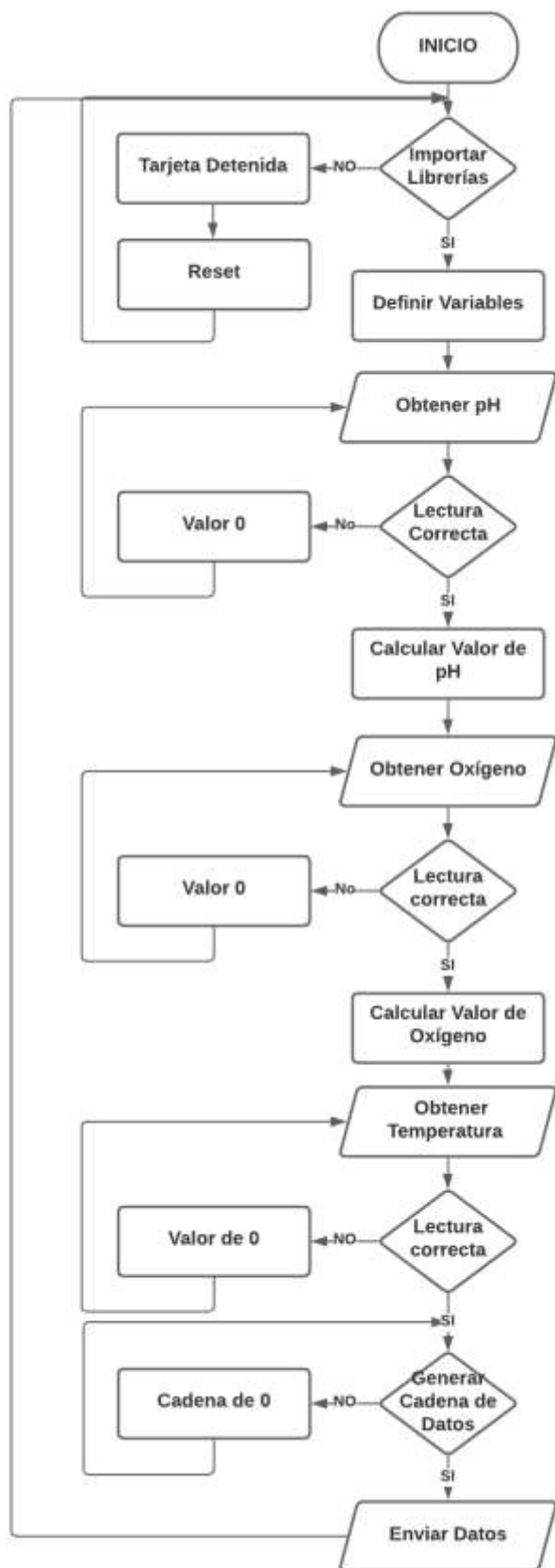


3.4.1.2. Nodo situado en la piscina de crecimiento

El segundo nodo obtiene los mismos datos a excepción del nivel de agua que se reemplazó con un sensor de oxígeno disuelto que envía los datos hacia el nodo central como se observa en el diagrama de la Figura 28.

Figura 28

Diagrama de flujo del nodo ubicada en la piscina de crecimiento de truchas.

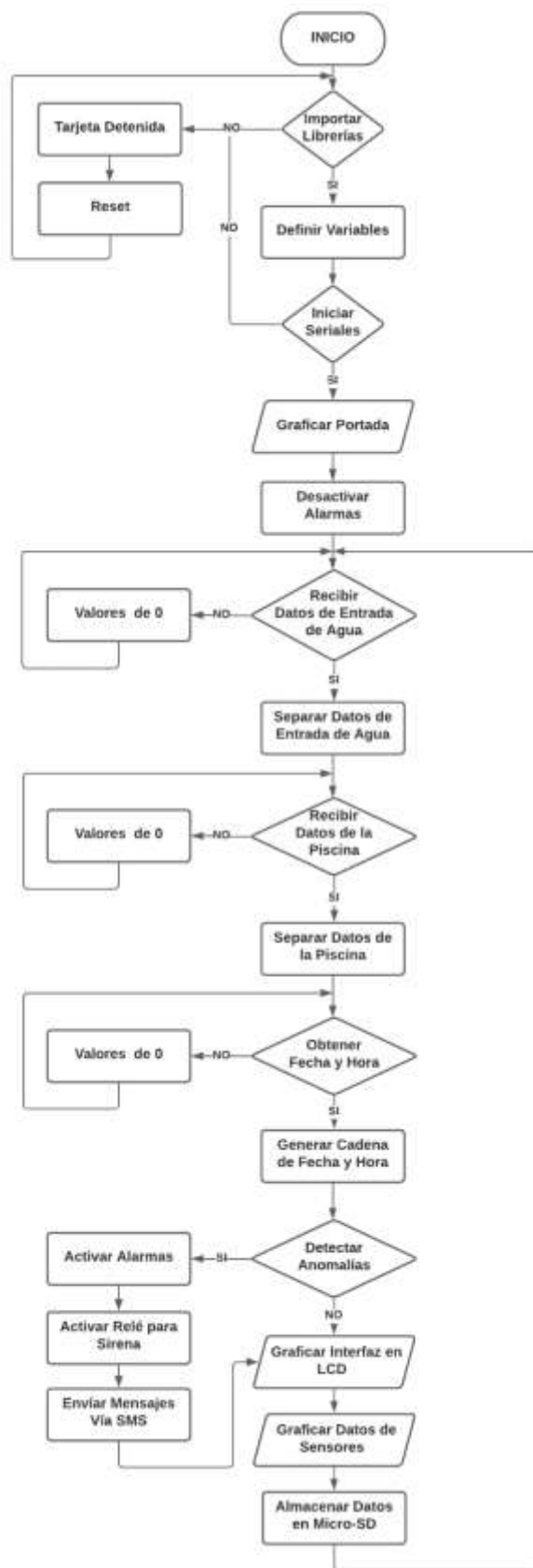


3.4.1.3. Nodo central

El nodo central es el encargado de obtener los datos obtenidos y mostrarlos en pantalla; además, almacena los datos y acciona diferente tipo de alarma. El proceso se observa en el diagrama de la Figura 29.

Figura 29

Diagrama de flujo del nodo central.



3.4.2. *Generación de la programación*

La generación del código que hace funcionar al sistema donde hace posible la interacción de todos los componentes físicos detalla dos aspectos esenciales los cuales son las líneas de código en lenguaje C y las librerías necesarias para que la tarjeta microcontroladora entienda que tipo de componentes tiene y los haga funcionar de la manera correcta.

3.4.2.1. **Librerías**

La programación del sistema de monitoreo se fragmento en tres partes en base a cada gabinete montado. Dentro de cada código independiente se utilizó librerías específicas que facilita la compilación de todos los elementos para que el sistema pueda cumplir con su funcionamiento. Los códigos de programación se encuentran en los Anexos 2, Anexo 3 y Anexo 4. A continuación, se detallan las librerías que se utilización en todas las estaciones:

- `#include <OneWire.h>`: Esta librería tiene la funcionalidad de designar un pin digital para recibir y enviar datos.
- `#include <DallasTemperature.h>`: Esta librería obtiene los datos de temperatura desde el sensor DS18B20.
- `#include <SoftwareSerial.h>`: Esta librería propia del IDE Arduino permite asignar diferentes pines digitales para comunicación serial con los pines TX y RX.
- `#include<Separador.h>`: Esta librería externa al IDE Arduino facilita la separación de caracteres de los datos recibidos mediante de la comunicación serial.
- `#include<U82glib.h>`: Esta librería permite utilizar las pantallas LCD asignando los pines para su funcionamiento.

- `#include<RTClib.h>`: Esta librería permite utilizar un módulo RTC para obtener datos de hora y fecha en tiempo real.
- `#include<SD.h>`: Esta librería permite utilizar el módulo Micro-SD para el almacenamiento de información.
- `#include<SPI.h>`: Esta librería es complementaria al almacenamiento de información debido a que en esta se asignan los pines de comunicación SPI.

3.4.2.2. Código de programación

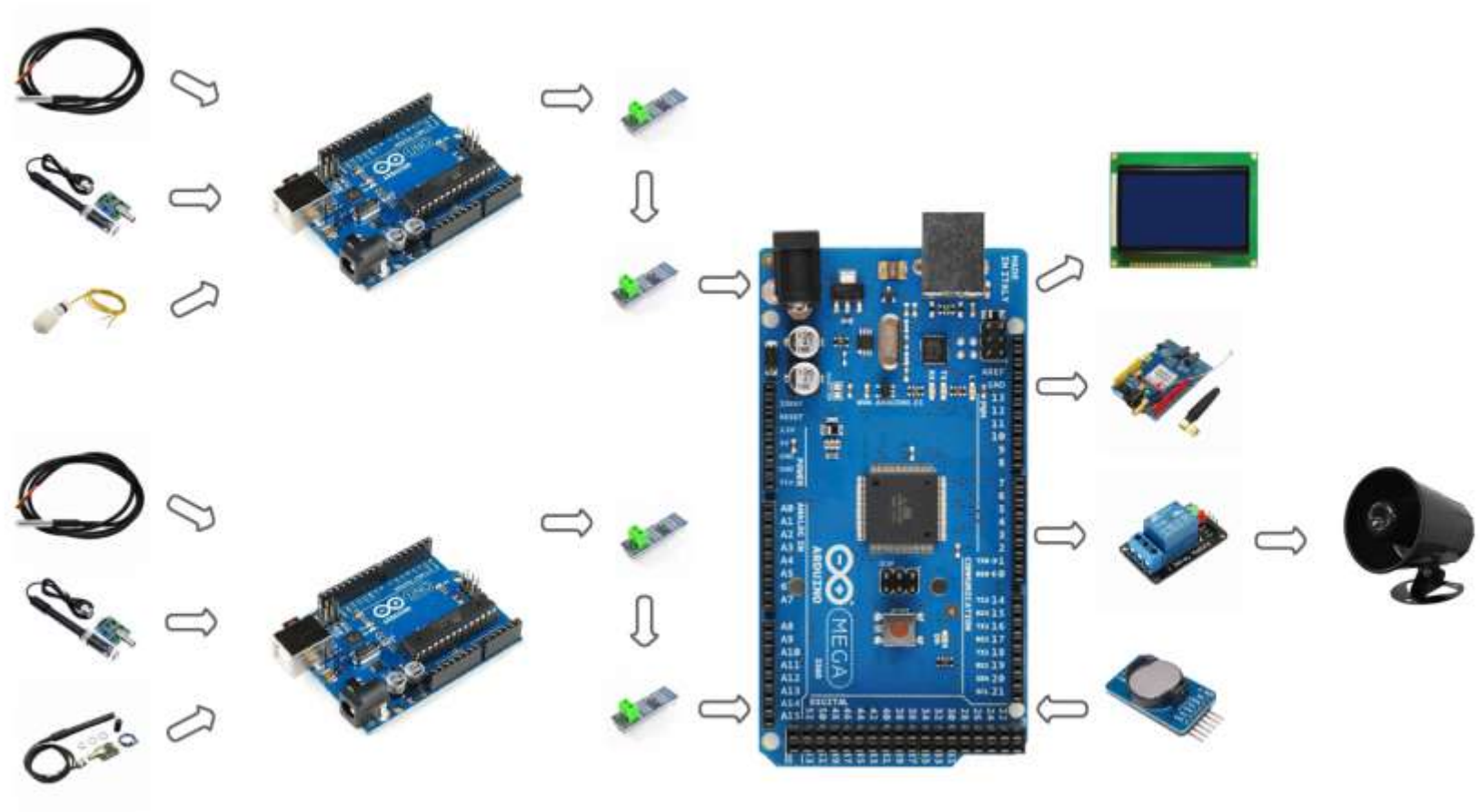
Para el código de programación existe un código propio para cada nodo los cuales se presentan en el Anexo 2, Anexo 3 y Anexo 4. Adicionalmente, en cada código independiente se encuentra una pequeña descripción de las líneas para ser comprendidas.

3.4.3. Esquema general de conexiones

En la Figura 30 se puede observar la composición de las dos estaciones exteriores, sin la conexión a la red de energía, controladas por un Arduino UNO los cuales obtienen los datos de las mediciones de los sensores y mediante un módulo Modbus envía los datos hacia una estación central el cual los recibe los organiza y los presenta mediante un periférico (LCD 128x64) mediante una interfaz amigable. Adicionalmente, el sistema cuenta con dos alarmas: la primera se controla mediante un relé que alimenta una sirena como indicador sonoro de que existe una anomalía en la alimentación de agua; la segunda es el envío de mensajes de texto de alerta hacia los celulares de los dueños del cultivo de truchas.

Figura 30

Esquema de componentes del sistema de monitoreo.



3.4.3.1. Circuitos de distribución

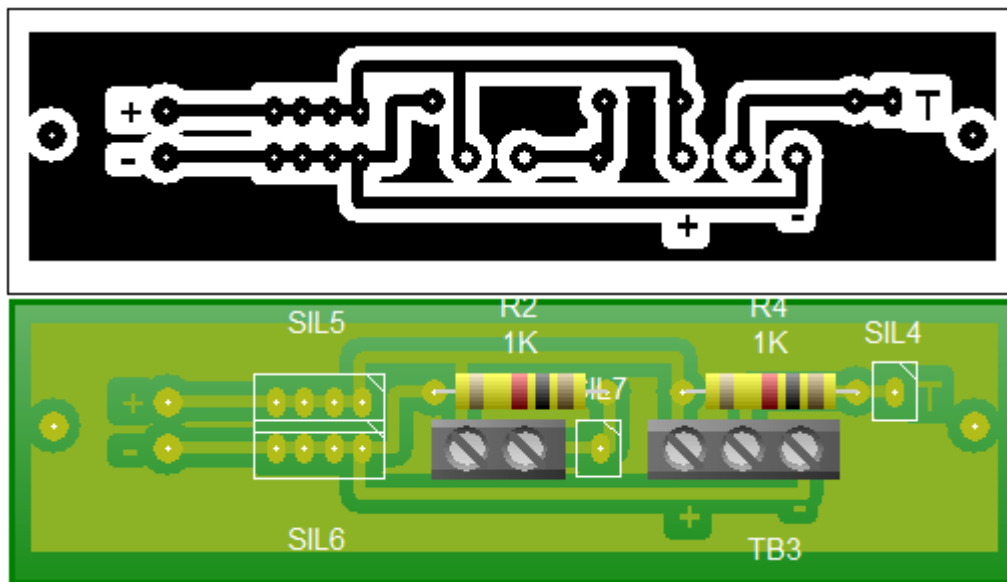
Se realizó dos placas en baquelita para la distribución de voltajes de alimentación y para la conexión de los sensores resistivos los cuales necesitan de una configuración de divisor de voltaje.

- *Circuito de distribución de voltajes, sensor de temperatura y sensor de nivel*

El diseño de la placa de voltajes tiene una forma compacta y sencilla de utilizar como se observa en la Figura 31.

Figura 31

Circuito impreso de distribución de voltajes, conexión del sensor de temperatura y el sensor de nivel de agua.

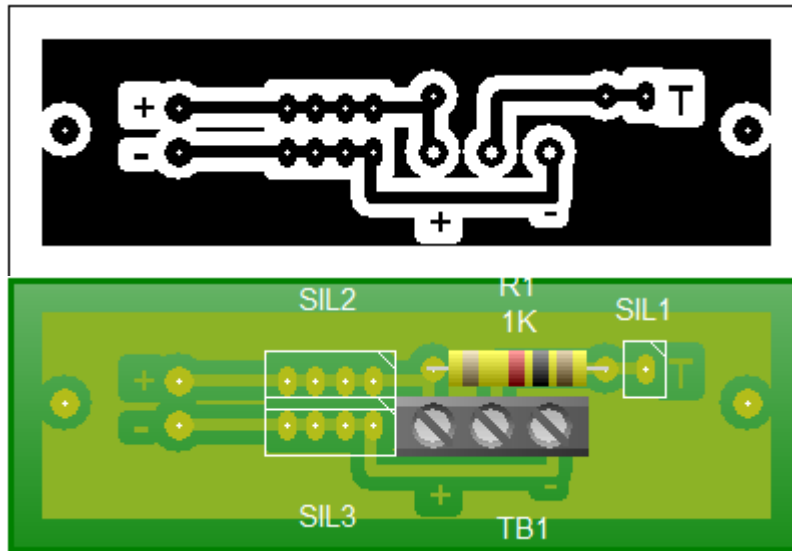


- *Circuito de distribución de voltajes, sensor de temperatura*

La segunda placa es un diseño más compacto debido a que se utilizan menor sensores dentro del circuito, esto se puede evidenciar en la Figura 32.

Figura 32

Circuito impreso de distribución de voltajes, conexión del sensor de temperatura.

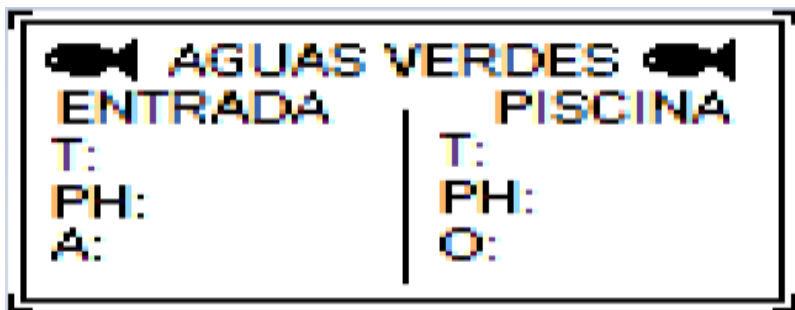


3.4.3.2. Interfaz

Para el diseño de una interfaz que se pueda presentar en la LCD 128x64 existen diferentes maneras de hacerlo. Para este proyecto se realizó el diseño de manera manual a obteniendo un archivo con extensión BMP con se observa en la Figura 33.

Figura 33

Interfaz de las mediciones de las estaciones externas.



Posterior a esto se utiliza algún programa de la diversidad que existe para convertir el diseño gráfico a una matriz de bits, en este proyecto se utilizó GIMP, que se graficarán en los pixeles de la pantalla como se observa en la Figura 34.

Figura 34

Matriz de bits para graficar en pantalla.

```
static unsigned char P5_bits[] = {
    0x0f, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0xf0, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x80, 0xfd, 0xff, 0xff, 0xff,
    0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xbf,
    0x05, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x20, 0x04, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x20, 0x04, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x20,
    0x04, 0x00, 0x10, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x20, 0x20, 0x04, 0x7f, 0x18, 0x20, 0x1c, 0x11, 0xc2, 0x21,
    0xf4, 0x3c, 0x1e, 0x8f, 0x03, 0xfe, 0x30, 0x20, 0x84, 0xff, 0x1d, 0x30,
    0x22, 0x11, 0x23, 0x42, 0x12, 0x44, 0x22, 0x41, 0x04, 0xff, 0x3b, 0x20,
    0xc4, 0xff, 0x1f, 0x50, 0x02, 0x11, 0x25, 0x40, 0x12, 0x44, 0x42, 0x41,
```

3.4.4. Descripción de funcionamiento del sistema

Para generar un sistema que cumpla con todos los requerimientos establecidos se utiliza la segmentación de procesos para comprender cada fase y las actividades que se deben realizar en el proceso de monitoreo. Todo el sistema se describe en nueve fases secuenciales las cuales se describen a detalle a continuación.

3.4.4.1. Primera etapa

La primera etapa empieza con la distribución e implementación de sensores para una de las estaciones que será instalada en la entrada de agua hacia las piscinas de las truchas. Esta estación obtiene las mediciones, las organiza y las envía hacia una estación central. Para esto se utilizan los siguientes componentes:

- Arduino UNO
- Sensor de pH
- Sensor de temperatura
- Sensor de nivel de agua
- Modbus RS485

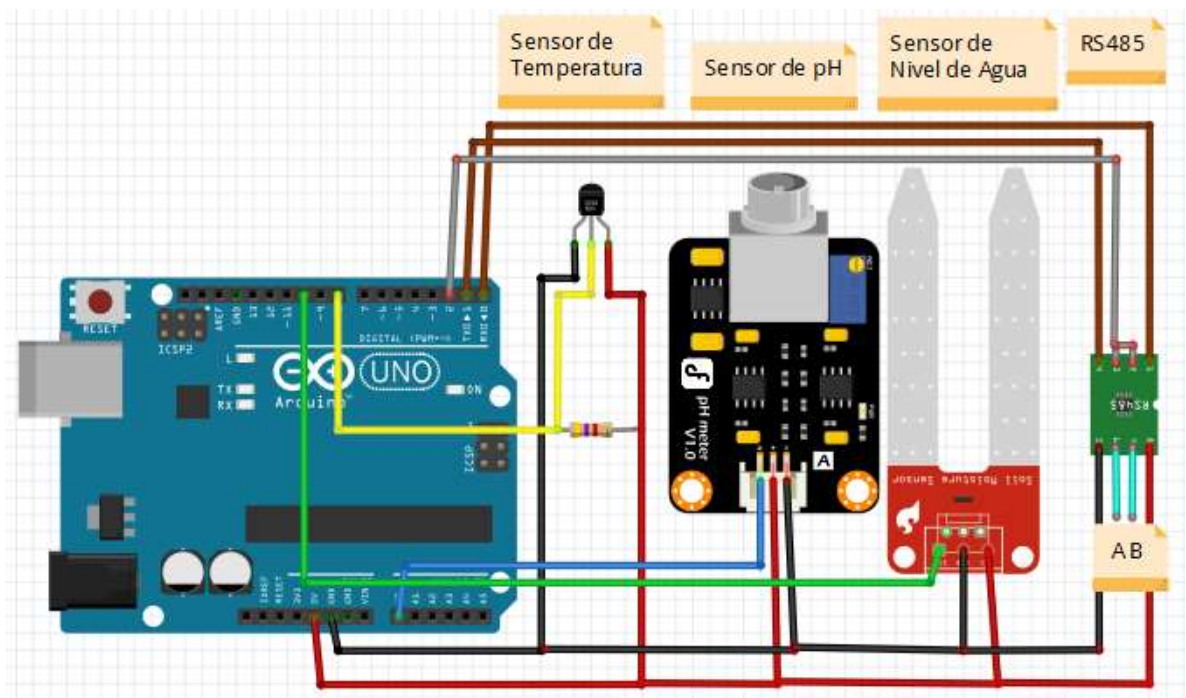
En la primera estación los sensores realizan sus mediciones, en la tarjeta procesadora se obtienen estos datos y los organiza para posteriormente enviar los datos a la central.

Para todos los valores de los sensores se utilizan decimales a excepción del sensor de nivel de agua donde solo maneja valores booleanos, 0 y 1.

Para la configuración de los componentes de la primera estación se presenta el diagrama de conexiones en la Figura 35.

Figura 35

Diagrama de conexión estación de entrada de agua.



Montado todos los componentes de la estación de la entrada de agua y mediante el IDE de arduino utilizando su código de programación que se puede visualizar en el Anexo 1, adicionalmente, en el monitor serial se puede visualizar la información que se ha organizado y que se envía mediante la comunicación Modbus, Figura 36. Todo el detalle de esta etapa corresponde al diagrama de flujo de la Figura 27.

Figura 36

Serial de la estación de entrada de agua.

```
21.45-6.5-0  
21.45-6.5-0  
21.45-6.5-0  
21.45-6.5-0  
21.45-6.5-0  
21.45-6.5-0  
21.45-6.5-0  
21.45-6.5-0  
21.45-6.5-0  
21.45-6.5-0  
21.45-6.5-0  
21.45-6.5-0  
21.45-6.5-0  
21.45-6.5-0  
21.45-6.5-0  
21.45/6.5/0  
21.45/6.5/0  
21.45/6.5/0  
21.45/6.5/0  
21.45/6.5/0  
21.45/6.5/0  
21.45/6.5/0  
21.45/6.5/0  
21.45/6.5/0  
21.45/6.5/0  
21.45/6.5/0
```

3.4.4.2. Segunda etapa

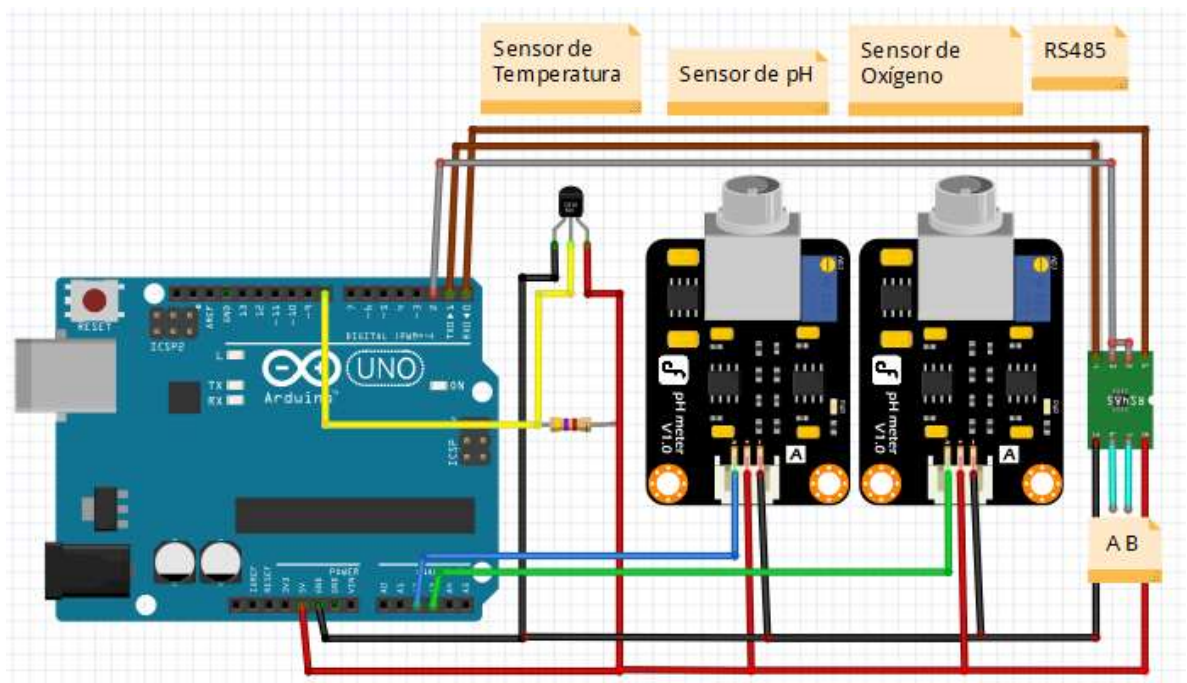
La segunda etapa consiste en la obtención de medidas de la segunda estación independiente instalada en la piscina donde se desarrollan los peces:

- Arduino UNO
- Sensor de pH
- Sensor de temperatura
- Sensor de oxígeno
- Sensor de nivel de agua
- Modbus RS485

El sistema es casi idéntico al sistema anterior con la diferencia de que se aumenta un sensor más, el sensor de oxígeno disuelto. A continuación, se presenta el diagrama de conexiones de esta estación en la Figura 37. La programación de esta estación se encuentra en el Anexo 2.

Figura 37

Diagrama de conexión de la estación de piscina.



Todo el detalle de esta etapa corresponde al diagrama de flujo de la Figura 28.

3.4.4.3. Tercera etapa

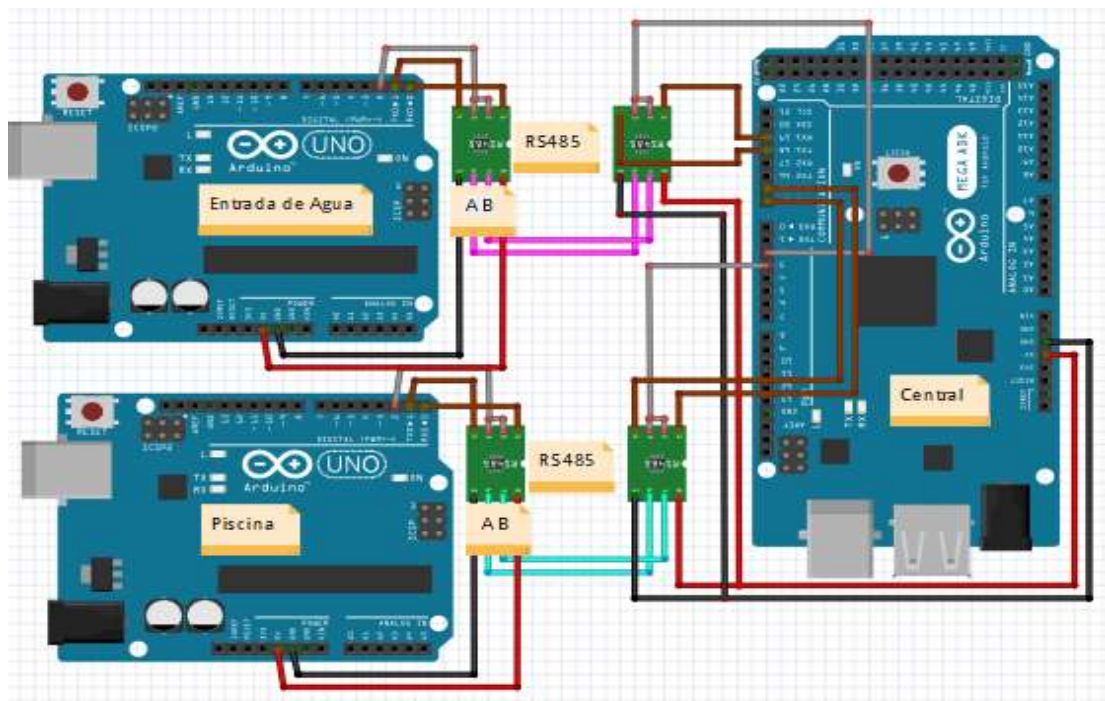
La tercera etapa consiste en la comunicación, que para este sistema se utilizan los módulos RS485 para la comunicación Modbus de dos esclavos y un maestro. Esta configuración se puede observar en la Figura 38. Los recursos que se utilizan son:

- Modbus RS485
- Arduino UNO
- Arduino Mega

Para este caso, esta comunicación utiliza los pines TX y RX de las tarjetas para enviar y recibir datos. Donde, el arduino Mega con su mayor cantidad de puertos permite obtener datos de los arduino UNO por canales separados, su programación se encuentra en la Anexo 4.

Figura 38

Conexión de la comunicación con RS 485.



Dentro del monitor serial del arduino Mega se puede visualizar uno de los canales que por los cuales se recibe los datos. Como en la Figura 39.

Figura 39

Datos recibidos por el arduino Mega mediante el Monitor Serial..

```
20.50/6.8/4
20.50/6.8/4
20.50/6.8/4
20.50/6.8/4
20.54/6.7/3
20.54/6.7/3
20.54/6.7/3
20.54/6.7/3
20.54/6.7/3
20.54/6.7/3
20.54/6.7/3
20.54/6.7/3
20.54/6.7/3
20.54/6.7/3
20.55/6.6/4
20.55/6.6/4
20.55/6.6/4
20.55/6.6/4
20.55/6.6/4
20.55/6.6/4
20.55/6.6/4
20.55/6.6/4
20.55/6.6/4
20.55/6.6/4
20.55/6.6/4
20.55/6.6/4
20.55/6.6/4
20.55/6.6/4
```

3.4.4.4. Cuarta etapa

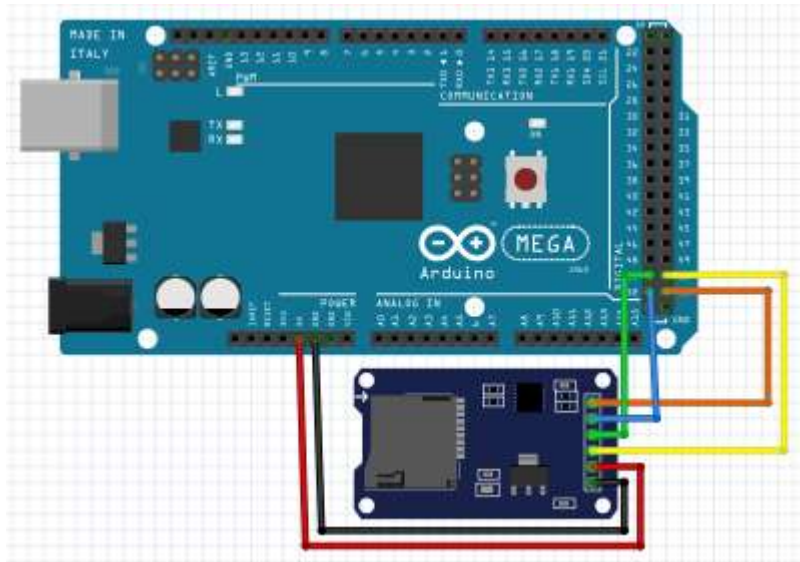
En la etapa cuatro se encuentra el almacenamiento de los datos recibidos mediante la comunicación serial de los esclavos, los componentes que intervienen dentro de este proceso son:

- Arduino Mega
- Módulo Micro SD

La programación está dentro del Anexo 4, el diagrama de conexión de esta parte del sistema se visualiza en la Figura 40. Es importante mencionar que esta etapa y posteriores describen el diagrama de flujo de la Figura 29.

Figura 40

Diagrama de arduino y micro-SD.



3.4.4.5. Quinta etapa

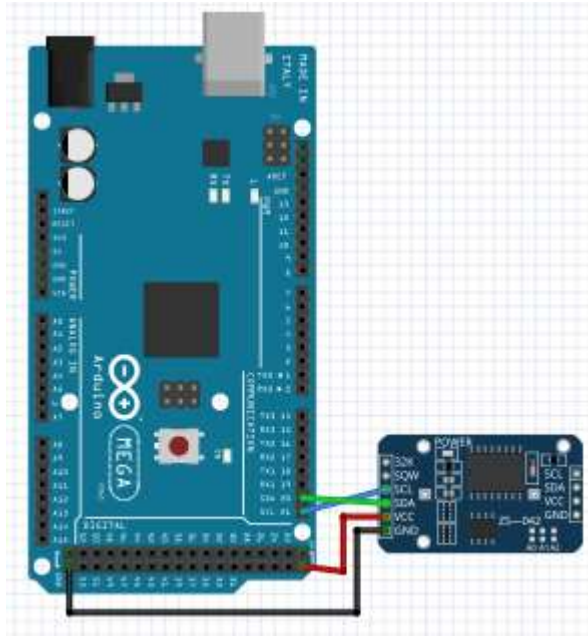
Para el almacenamiento de la información es necesario contar con la hora y la fecha para un mejor seguimiento de las medidas con respecto a periodos de tiempo, lo que se necesita para realizar este proceso es necesario utilizar los siguientes servidores:

- Arduino Mega
- RTC DS3231

La manera de configurar y obtener los datos de tiempo es sencilla como se puede ver en su programación en el Anexo 4, el diagrama de conexión se puede visualizar en la Figura 41,

Figura 41

Diagrama de arduino y RTC.



3.4.4.6. Sexta etapa

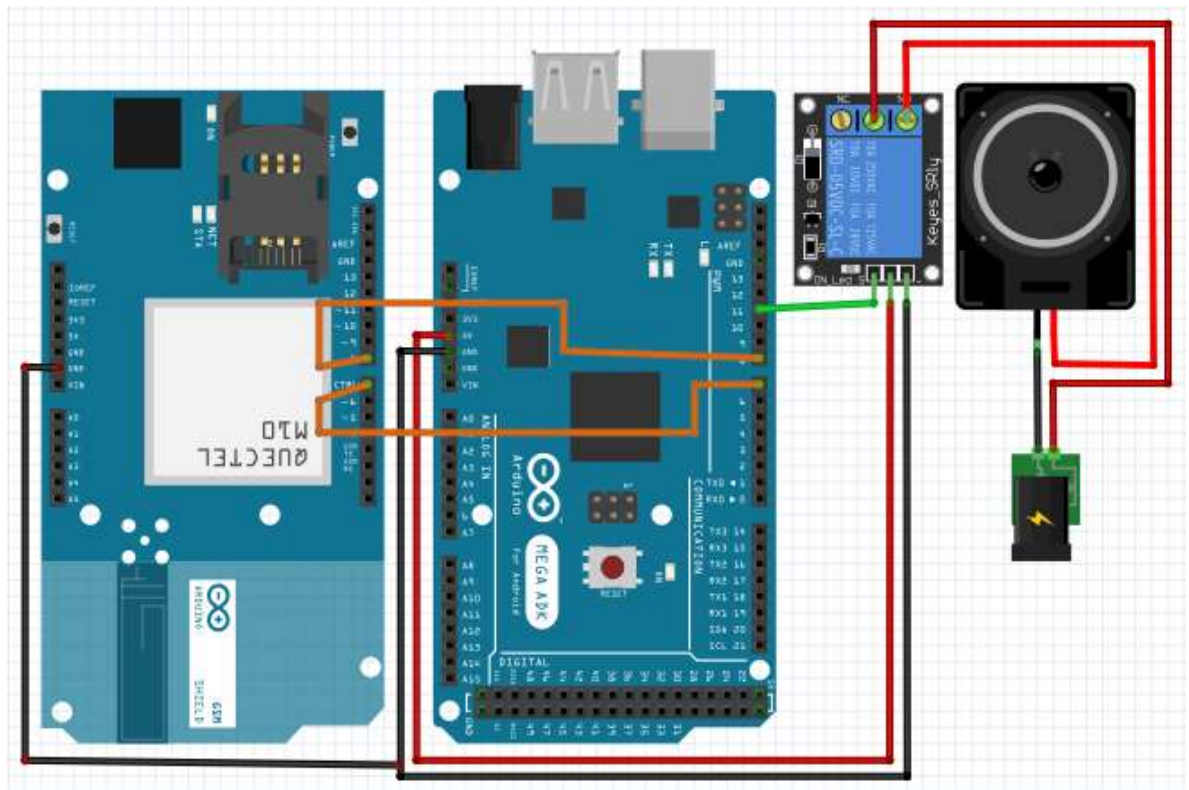
En la sexta etapa se describen las alarmas que se accionan al momento que el sensor de nivel indica un nivel bajo de agua que representa un riesgo para las truchas en las piscinas, los componentes a utilizar son:

- Arduino Mega
- Módulo relé 12 V
- Sirena 12 V
- Módulo GSM SIM900

La configuración de esta etapa se observa en la programación que se encuentra en el Anexo 4 y su diagrama de conexión se visualiza en la Figura 42.

Figura 42

Diagrama de conexión de SIM 900 y relé con alarma.



El sistema al identificar un nivel bajo de agua acciona un relé que enciende una sirena como indicador sonoro, a su vez, mediante el uso de un módulo GSM envía mensajes informativos de esta emergencia a los números registrados en la programación. Esta variación también se puede visualizar en la pantalla de la estación central.

3.4.4.7. Séptima etapa

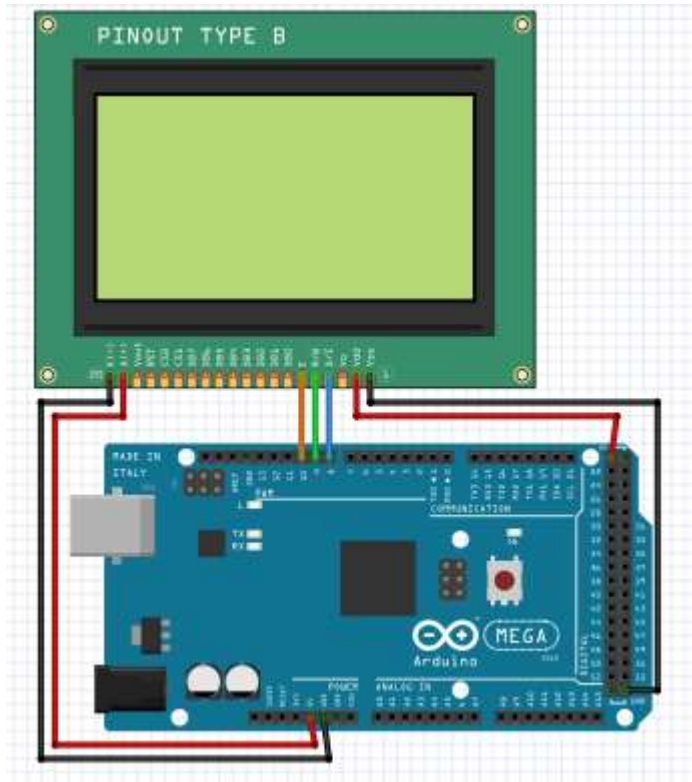
Finalmente, en la última etapa se presenta la información mediante una pantalla que por su densidad de píxeles se programa como una LCD gráfica:

- Arduino Mega
- LCD 128x64

La forma de configuración se puede observar en su programación en el Anexo 4, el diagrama de conexión se puede visualizar en la Figura 43.

Figura 43

Diagrama de arduino y LCD 128x64.



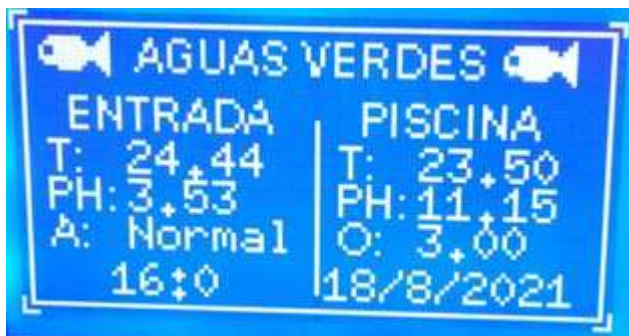
Dentro de la séptima etapa podemos encontrar el funcionamiento de Weka ya que al estar compuesta de una serie de herramientas graficas de visualización permiten que la interfaz gráfica sea ergonómica para el usuario facilitando el acceso a múltiples funcionalidades, Weka nos da la posibilidad de seleccionar entre cuatro posibles interfaces de usuario los cuales son Simple CLI, Explorer, Experimenter (Lorena, 2019)

3.4.4.8. Octava etapa

Finalmente, en la etapa final se presenta la información mediante una pantalla que por su densidad de píxeles se programa como una LCD gráfica como esta en la Figura 44:

Figura 44

Interfaz en la pantalla del panel central.



3.4.4.9. Novena etapa

La etapa adicional, después de la construcción del equipo de monitoreo se centra en la preparación de la herramienta Weka. Esta herramienta de análisis de datos con su interfaz gráfica permite ser instalada en cualquier sistema operativo. Su proceso se describe a continuación:

Partiendo de la interfaz gráfica se utiliza la opción de explorar para ubicar los datos que se desea cargar al sistema para generar las tareas que se desee como se observa en la Figura 45.

Figura 45

Interfaz de la aplicación Weka

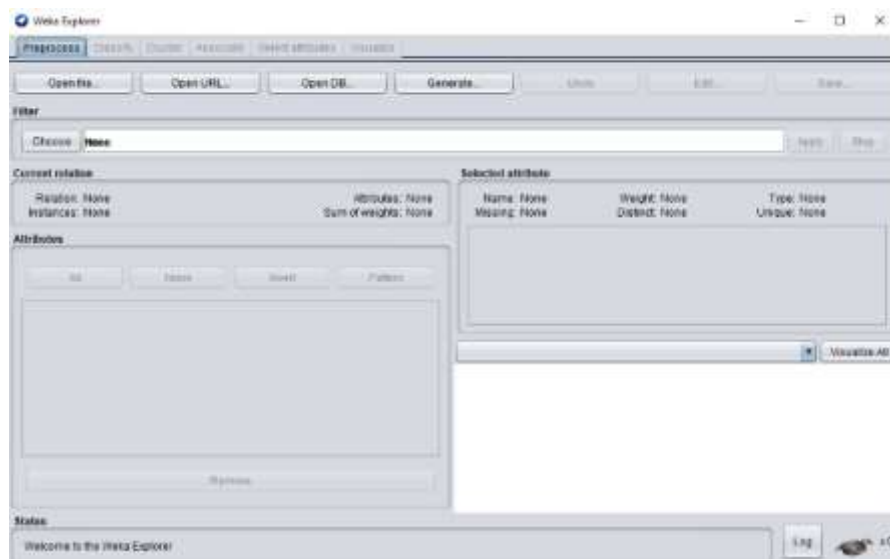


Nota. Adaptado de ¿Qué es Weka?, por Isaiaranda, 2020, Medium. (Isaiaranda, 2020).

Una vez cargados los datos se visualiza una nueva interfaz, como en la Figura 46, la cual permite empezar a realizar la selección de parámetros que se desea utilizar.

Figura 46

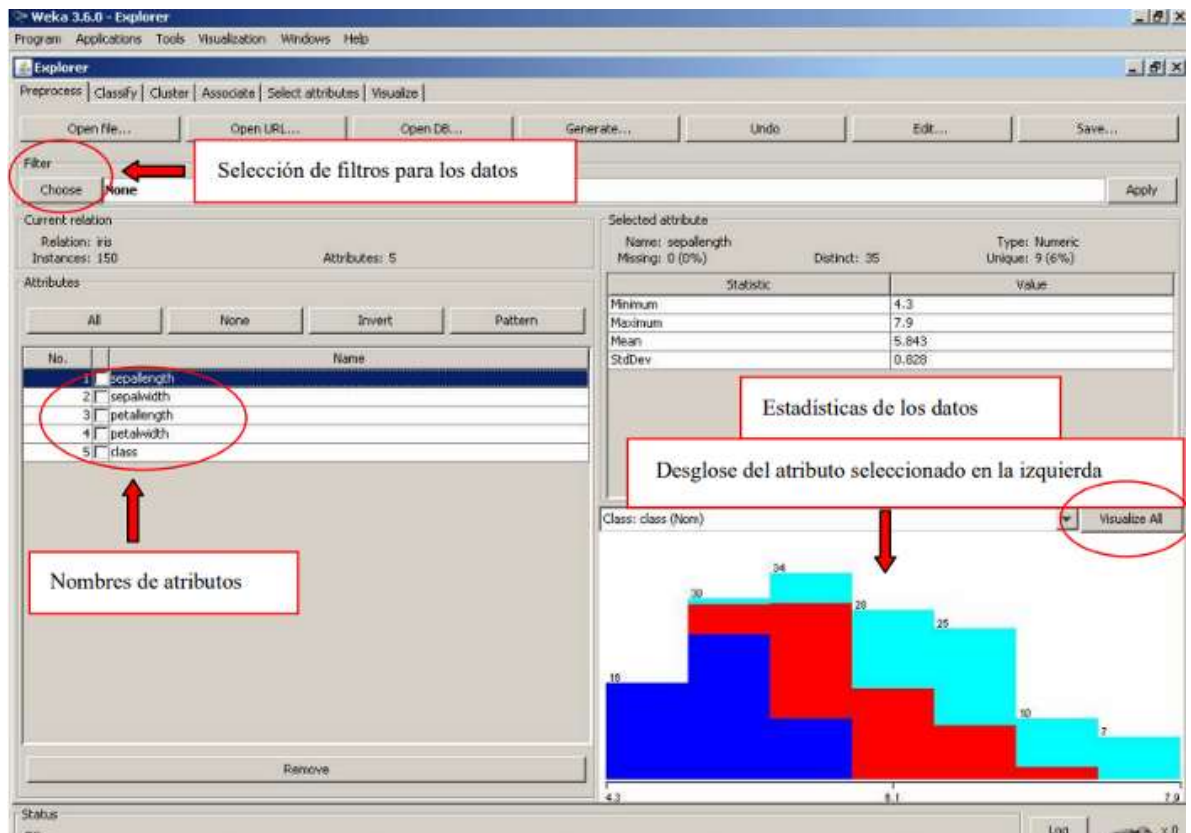
Interfaz de tuning o selección de parámetros y filtros.



En la Figura 47 se detallan mediante diferentes recuadros algunas de las selecciones que se puede realizar en la interfaz anteriormente presentada.

Figura 47

Interfaz con indicaciones de tareas.

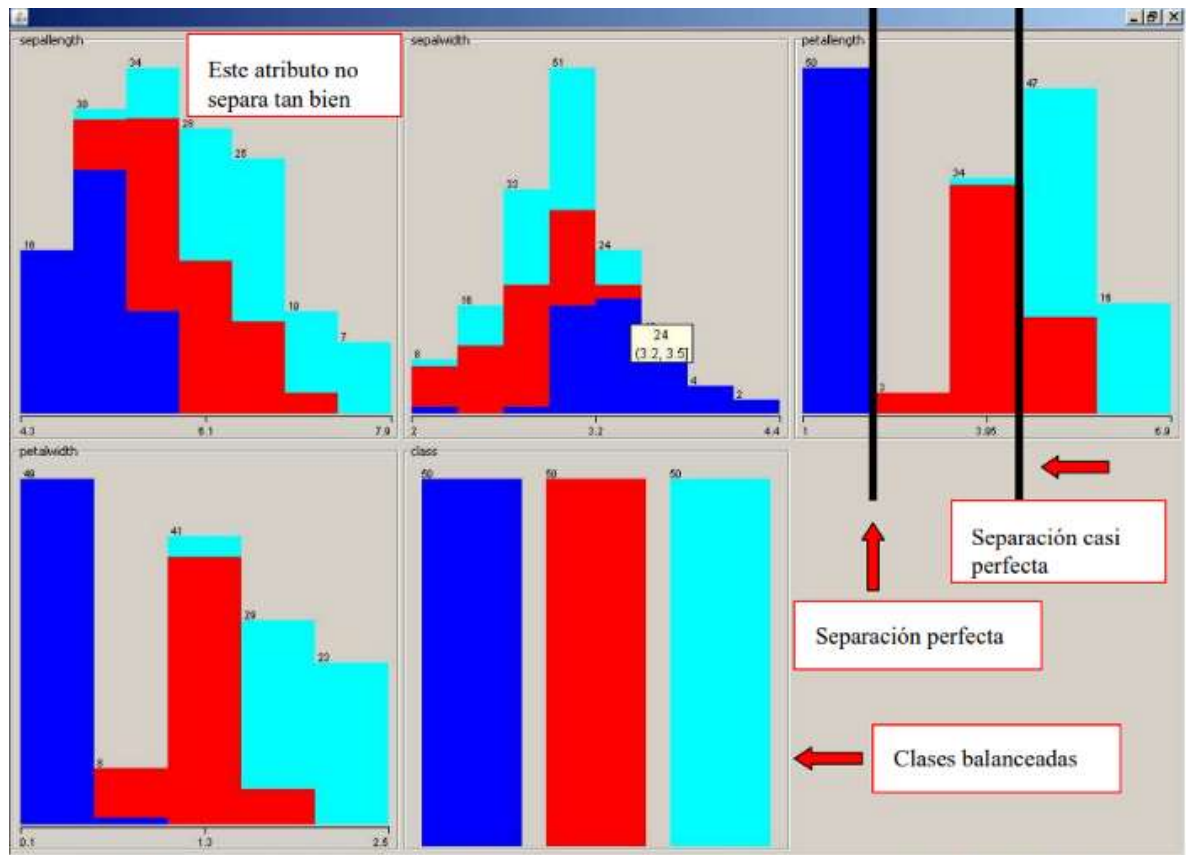


Nota. Adaptado de ¿Qué es Weka?, por Isaiaranda, 2020, Medium. (Isaiaranda, 2020).

A partir de aquí, según la selección de herramientas que se desee utilizar en Weka se analizarán los datos y se orientarán en base a los parámetros establecidos, la utilidad de esta aplicación se observa en los gráficos resultantes que la interfaz maneja, Figura 48.

Figura 48

Gráficos resultantes mediante la aplicación Weka.



Nota. Adaptado de *¿Qué es Weka?*, por Isaiaranda, 2020, Medium. (Isaiaranda, 2020).

3.4.5. Montaje y construcción

Para la construcción de los gabinetes se empotraron los componentes con su case respectivo en las placas de acero correspondientes. Cada gabinete cuenta con alimentación independiente la cual está protegida por medio de un fusible, además de un interruptor y un indicador luminoso, piloto, que permite visualizar si la estación se encuentra activa como se puede visualizar en la Figura 49.

Figura 49

Montaje de componentes de los módulos en las placas metálicas.



Posterior a eso se instalan las placas metálicas montadas dentro de cada gabinete procurando mantener siempre visible el conducto por donde los cables van a salir y entrar a cada estación, Figura 50.

Figura 50

Colocación de las placas metálicas en los gabinetes.



Se realiza el montaje de la pantalla, con el respectivo interruptor, fusible y piloto sobre la tapa del gabinete, de tal manera que la pantalla quede completamente visible y los componentes no interfieran con los componentes internos. Su montaje se puede visualizar en el Figura 51.

Figura 51

Instalación de pantalla y elementos externos del gabinete.



En la Figura 52 se puede apreciar el interruptor de encendido de gabinetes.

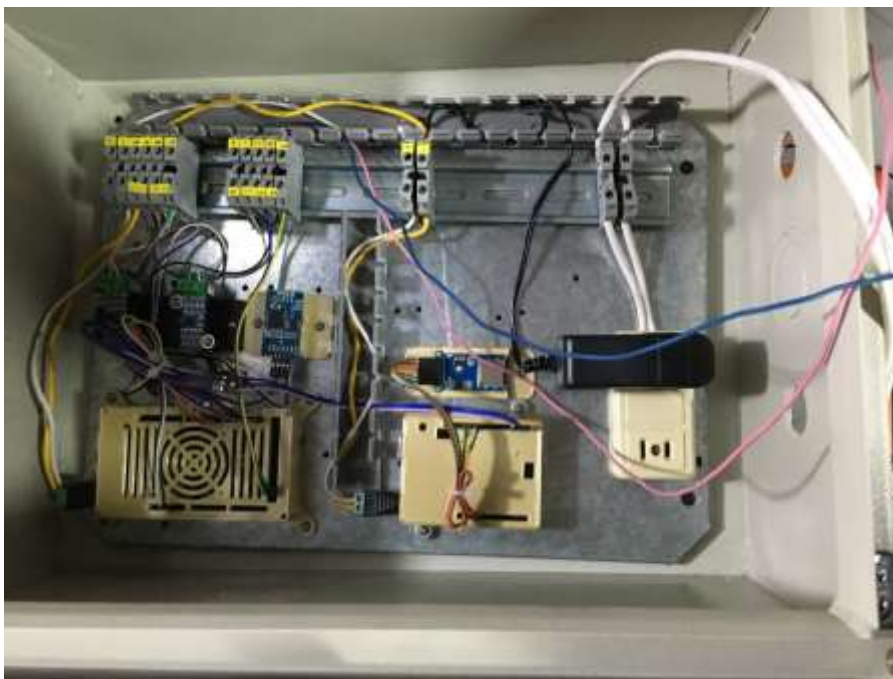
Una vez realizado el montaje, se procede al cableado de los componentes correspondientes a pantalla, piloto, interruptor y fusible, estos cables pasarán a través de una canaleta para su respectiva alimentación. Se puede visualizar en el Figura 53.

Figura 52

Vista frontal de los gabinetes.

**Figura 53**

Cableado de los gabinetes.



Seguidamente, se realiza el montaje de la placa metálica con sus componentes incluidos procurando que todos los cables se encuentren bien colocados y ordenados, para facilitar su reparación en caso de daño. En la Figura 54 se puede visualizar el montaje de la placa metálica y el encendido de los componentes.

Figura 54

Empotrado y encendido de los tableros.



En la Figura 52 es posible ver las estaciones completamente armadas y montadas con todos los componentes necesarios y su respectiva alimentación.

Con las estaciones de monitoreo listas para su instalación se procede a realizar el cableado en el lugar de aplicación. Para este cableado se hace uso de mangueras ya que el ambiente del lugar es húmedo y lluvioso, por lo que se necesita de protección adicional. Su proceso se puede visualizar en la Figura 55.

Figura 55

Cableado del área de instalación.



La colocación de la manguera se realiza siguiendo las vigas del techo para evitar cualquier interferencia. Su proceso se puede visualizar en la Figura 55.

Para el montaje del gabinete principal se desmonta la placa metálica con sus componentes y con pernos se monta sobre la pared. Posteriormente se vuelve a colocar la placa metálica con sus componentes dentro del gabinete. Como resultado tenemos la estación principal instalada y se puede visualizar en la Figura 56.

Figura 56

Instalación del gabinete central.



Antes de la instalación de la estación en la piscina de truchas se realiza una calibración de oxígeno del ambiente, para no tener errores de lectura en el lugar de instalación que está trabajando. Esta calibración final se la puede visualizar en la Figura 57.

Figura 57

Calibración final de los sensores y sistema.



Con la ayuda de un trozo de techo de policarbonato se monta la estación de ingreso de agua y además se le cubre de igual manera con techo de policarbonato para evitar daños en caso de lluvia. De esta manera queda protegido contra este tipo de eventos. Además, gracias al piloto sabemos que la estación está activa. Se puede visualizar el resultado en la Figura 58.

Figura 58

Instalación del nodo en la entrada de agua.



Al igual que con la instalación de la estación de ingreso de agua, se realiza el mismo proceso con la ayuda de techo de policarbonato para su montaje y protección contra lluvia y salpicaduras de agua. La estación se encuentra activa como se puede apreciar gracias al piloto en la Figura 59.

Figura 59

Instalación del nodo de la piscina de crecimiento de truchas.



Dentro la instalación del sistema de monitoreo y la adecuación de la piscina se estableció incluir adicionales con el fin de asegurar la mejora de los parámetros de calidad de agua. Para esto se instaló una estructura como extensión para aumentar los puntos de oxigenación de la piscina como se observa en la Figura 60. La estructura de tubos de acero y tubos PVC distribuye el agua que ingresa a lo largo de toda la piscina y cae por diferentes divisiones intercaladas con el fin de mejorar el proceso de oxigenación natural.

Figura 60

Estructura de distribución de puntos de oxigenación.



3.4.6. Instalación

Con todo el sistema ensamblado y funcionando se procede a la prueba de todo el equipo y, además, revisar algunas condiciones adicionales con respecto al ambiente donde se va a instalar, y analizar el resultado ante las condiciones adversas del medio.

3.4.6.1. Entornos

El monitoreo se realiza en Tufiño, una parroquia del cantón Tulcán en la provincia de Carchi, situada en la parte ecuatoriana del Nudo de los Pastos, al norte de la República

del Ecuador, en América del Sur. Su cabecera se encuentra a una altitud de 3200 metros sobre el nivel del mar. Su clima es frío andino debido su proximidad al volcán Chiles. La temperatura promedio fluctúa entre los 6 y 12 grados centígrados aproximadamente en la zona de pajonales y matorrales y de 4 – 6 en el sector de las Lagunas Verdes hacia el volcán Chiles.

La precipitación promedio anual varía de forma considerable en esta zona desde aproximadamente 1.000 mm en el sector de Tufiño y mayores de 1.750 mm en los alrededores del volcán Chiles. Esta alta precipitación posibilita mayormente que el ecosistema de la región soporte una alta biodiversidad de especies endémicas.

La época lluviosa abarca casi todos los meses, pero disminuye entre junio y septiembre, por lo general la humedad relativa es de más del 80% y puede exceder del 90% en las áreas forestales y de páramo. Las nubes presentes durante las horas del día pueden promediar más de un 70%, de forma particular en los meses húmedos de octubre a mayo.

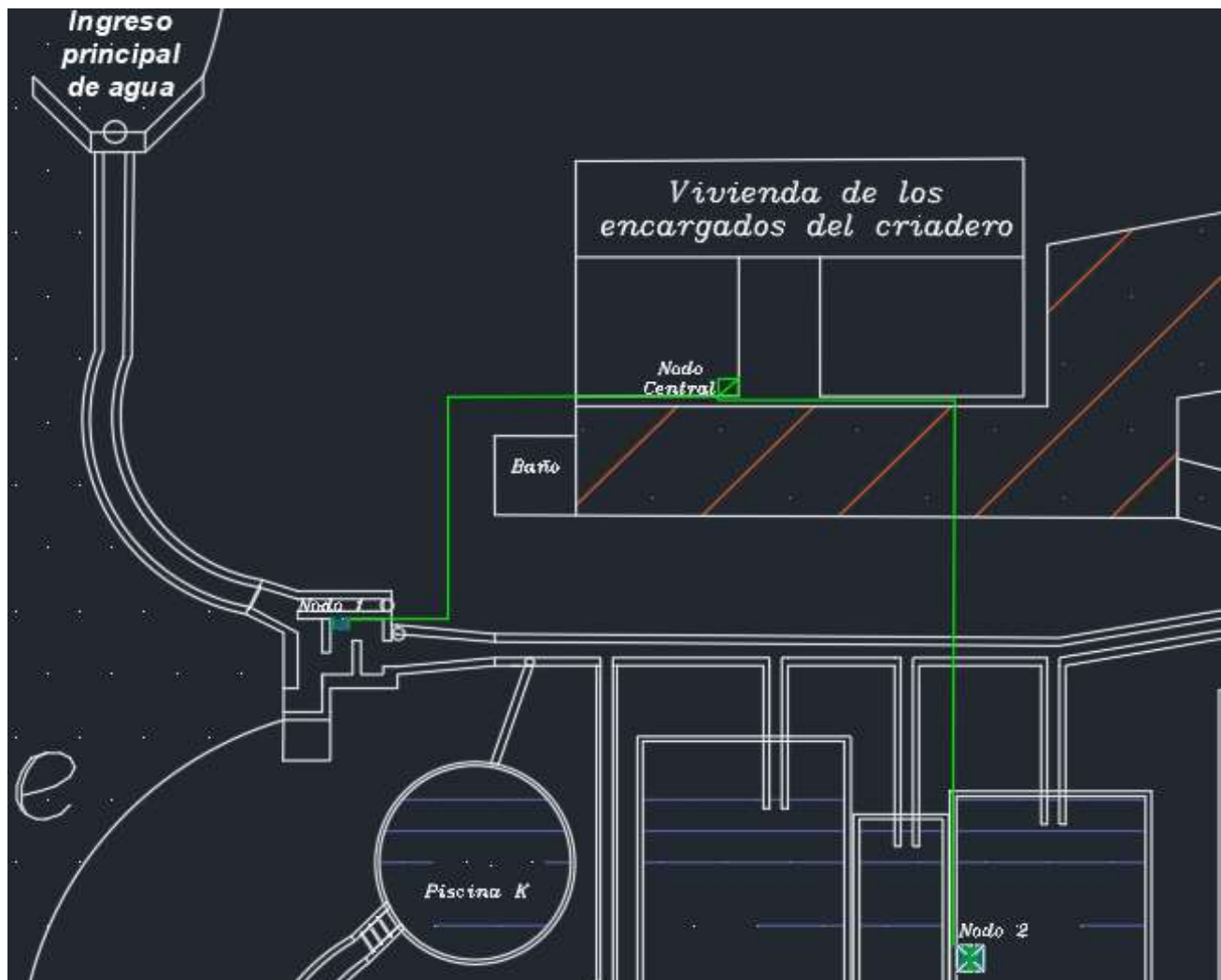
Específicamente el proyecto se realiza a 22 km de la vía Tufiño - Maldonado, a 3980 metros sobre el nivel del mar, donde se ubica un complejo de 3 lagunas llamadas Lagunas Verdes por su coloración verde y turquesa, colores originados también por el azufre que se produce en las faldas del volcán Chiles. En el Anexo 16 se puede apreciar como luce el complejo en la actualidad.

3.4.6.2. Layout de instalación

En esta parte se desarrolló el plano en 2D del Criadero de Truchas Aguas Verdes, como se puede mirar en la Figura 61.

Figura 61

Diagrama de arduino y LCD 128x64.



Como se muestra en la Figura 55 los estaciones a instalarse serán conectadas hasta la vivienda en donde se encuentran los encargados del criadero para que puedan mirar los datos de los sensores cada cierto tiempo.

3.4.7. Calibración y manual de uso

La calibración inicial y posterior a la instalación por motivo de mantenimiento es importante para asegurar la lectura exacta y precisa de cada uno de los sensores, al igual que todo el funcionamiento de los gabinetes modulares y panel central. Para esto se debe conocer y analizar ciertos aspectos como los que se detalla a continuación.

3.4.7.1. Calibración de sensores

En la tecnología de medición, la calibración del sensor se refiere a la tarea de determinar las condiciones definidas, la relación entre los valores de salida de una medición de un instrumento de medición, como un sensor o un sistema de medición, con la incertidumbre de medición y valores correspondientes de un mensaje estándar con una incertidumbre de medición correspondiente.

3.4.7.2. Calibración por software

En general, el término "software de calibración" se refiere a aplicaciones que automatizan todo el proceso de calibración o parte a través del control de la computadora. El software de calibración permite al usuario administrar sus datos de actividad y datos de calibración.

3.4.7.3. Calibración de hardware

Se refiere a la calibración del sensor de una forma manual, es decir, por medio de potenciadores o pulsadores, dependiendo del sensor que se esté calibrando puede variar su método.

3.4.7.4. Manual de usuario

Se realizó un manual de uso, en el cual se detallan las principales partes de cada una de las estaciones y sus componentes. En este documento se describe el funcionamiento de cada componente, recomendaciones y parámetros iniciales para el correcto funcionamiento de todo el sistema de monitoreo. Este manual se encuentra en el Anexo 17.

4. CAPÍTULO IV: PRUEBAS DE FUNCIONAMIENTO

A partir de las pruebas que se realizan durante el desarrollo del trabajo, se verifica el funcionamiento del dispositivo y analizar la interacción de cada componente (hardware y software) para obtener una retroalimentación.

4.1. Metodología de Cascada: Fase 4 Verificación

La cuarta fase utiliza el sistema ya ensamblado para verificar su funcionamiento segmentado en fases, las pruebas realizadas y el detalle de los resultados obtenidos.

4.2. Pruebas

Una vez construido el sistema de monitoreo e instalado en el lugar de cultivo de truchas se empiezan a realizar las pruebas de funcionamiento. En este caso las pruebas se realizan en base a once fases que resultaron en base a las etapas de funcionamiento. En la Tabla 28 se muestra la descripción de los resultados esperados.

Tabla 28

Descripción de las pruebas de funcionamiento para el sistema de monitoreo.

PRUEBAS DE FUNCIONAMIENTO		
FASE	DESCRIPCIÓN	RESULTADO ESPERADO
Fase 1: Lectura de sensores en estación de entrada de agua	Funcionamiento de la estación en la entrada de agua	Lectura correcta de cada sensor y envío de la cadena de datos por serial.

Fase 2: Lectura de sensores en estación de la piscina	Funcionamiento de la estación en la piscina de truchas	Lectura correcta de cada sensor y envío de la cadena de datos por serial.
Fase 3: Comunicación Modbus	Funcionamiento de la comunicación entre estaciones	Envío y recepción de cadenas de datos en tres estaciones exteriores y la central
Fase 4: Lectura de tiempo real	Configuración del tiempo correcto	Obtención de hora y fecha correcta
Fase 5: Almacenamiento de datos	Almacenado de información	Guardado de datos en la micro-SD en un archivo tipo .csv
Fase 6: Activación de alarmas	Funcionamiento de alarmas	Activación de la sirena por medio de un relé y envío de mensajes SMS.
Fase 7: Visualización de información	Interfaz Humano Máquina	Visualización de los datos obtenidos en tiempo real mediante una interfaz
Fase 8: Construcción de gabinetes	Funcionalidad de los gabinetes	Comprobar el montaje, fijación y recubrimiento de los gabinetes
Fase 9: Instalación de gabinetes	Pruebas de uso y funcionalidad por los usuarios del cultivo de truchas	Comprobar las funciones del sistema en el lugar.
Fase 10: Criterios de Machine Learning	Utilizar una aplicación de análisis y clasificación de datos	Visualizar un indicador de seguimientos del sistema.

Fase 11: Adecuación de piscina	Acciones de adecuación del área de crecimiento de peces	Optimización en la producción de truchas arcoíris en la piscina instalada.
---------------------------------------	---	--

4.3. Resultados

Utilizando la información de la Tabla 28 se realizan pruebas para cada fase y a continuación se describen los resultados obtenidos con respecto a las funciones que deben cumplir cada componente (hardware) y la manera de realizar el proceso (software).

4.3.1. Resultados de la Fase 1 y Fase 2

Las pruebas empiezan desde el funcionamiento de las estaciones exteriores las cuales reciben las mediciones por parte de los sensores, las agrupa y maneja para ser enviadas por medio del serial. En la programación en la IDE Arduino se utiliza la ventana serial para lograr visualizar los datos que se han recibido y se están enviando como se observa en la Figura 62.

Figura 62

Datos enviados y visualizados en el monitor serial.

Ingreso de Agua	Piscina
24.44/3.53/0	23.50/11.15/3.00
24.44/3.53/0	23.50/11.15/3.00
24.44/3.53/0	23.50/11.15/3.00

4.3.2. Resultados de la Fase 3

La interconexión del sistema mediante los módulos RS485 es verificable desde su hardware. La comunicación Modbus utiliza los pines de comunicación serial Rx y Tx en las

tarjetas Arduino, cuando existe un flujo de datos existen leds indicadores que parpadean continuamente. Adicionalmente, una vez conectados todos los gabinetes la información de las estaciones externas con los sensores debe visualizarse en la pantalla y actualizarse cada 2 segundos como se observa en la Figura 63.

Figura 63

Llegada de datos por comunicación Modbus.

Ingreso de Agua	Piscina
24.44/3.53/0	23.50/11.15/3.00
24.44/3.53/0	23.50/11.15/3.00
24.44/3.53/0	23.50/11.15/3.00



The image shows a digital display with a blue background and white text. At the top, it says 'AGUAS VERDES' flanked by two fish icons. Below this, there are two columns of data. The left column is labeled 'ENTRADA' and the right column is labeled 'PISCINA'. Each column contains three rows of data: Temperature (T), pH (PH), and a status indicator (A or O). At the bottom of each column, there is a time and date display. The left column shows '16:0' and the right column shows '18/8/2021'.

4.3.3. Resultados de la Fase 4

Su configuración se realiza dentro de la programación del gabinete central. Este módulo mantiene la hora y la fecha aún después de ser desconectado de la tarjeta controladora debido a su pila independiente que sigue alimentando el modular, algunas lecturas de hora y fecha se observan en la Figura 64.

Esta capacidad se comprueba cada vez que la estación es desconectada de la alimentación de 110V manteniendo la fecha y hora correcta. Según la datasheet este módulo pierde su configuración cada año creando un desfase de 5 mins, sin embargo, esto no se ha podido comprobar por el tiempo necesario para notar este cambio.

Figura 64

Datos obtenidos del módulo RTC.



4.3.4. Resultados de la Fase 5

El sistema de almacenamiento de información utiliza la comunicación SPI la cual utiliza un pin digital para verificar la presencia o ausencia de una tarjeta micro-SD. En caso de no existir una tarjeta extraíble en el módulo el sistema no cumple la función de almacenamiento sin interferir el monitoreo de la calidad de agua. El gabinete genera archivos de formato “.csv” como se observa en la Figura 65.

Figura 65

Programación para la creación del archivo de formato “.csv”.

```
void GUARDAR()
{
  SD.begin(53);
  dataFile = SD.open( "Aguas.csv" , FILE_WRITE);
  dataFile.print(hora);
  dataFile.print(",");
  dataFile.print( fecha);
  dataFile.print(",");
  dataFile.print("T(E)");
}
```

Cada archivo generado muestra la siguiente distribución: Hora, Fecha, Temperatura de la entrada, pH de la entrada, Nivel de agua, Temperatura de la piscina, pH de la piscina y

Oxígeno en la piscina. Esta configuración se observa en los archivos obtenidos del sistema mediante la extracción de la tarjeta como se puede observar en la Figura 66.

Figura 66

Formato de datos almacenados en la tarjeta extraíble micro-SD.

10/12/2021 T(E)	11.44	PH(E)	9.26	N(E)	NORMAL	T(P)	11.81	PH(P)	3.13	O(P)		1
10/12/2021 T(E)	11.44	PH(E)	9.16	N(E)	NORMAL	T(P)	11.75	PH(P)	2.54	O(P)		5
10/12/2021 T(E)	11.44	PH(E)	9.06	N(E)	NORMAL	T(P)	11.75	PH(P)	3.98	O(P)		2
10/12/2021 T(E)	11.44	PH(E)	9.13	N(E)	NORMAL	T(P)	11.75	PH(P)	4.07	O(P)		2
10/12/2021 T(E)	11.44	PH(E)	9.35	N(E)	NORMAL	T(P)	11.81	PH(P)	2.34	O(P)		4

4.3.5. Resultados de la Fase 6

El sistema implementado cuenta con una programación que activa ciertas alarmas vinculadas al nivel de agua, el pH y la oxigenación. En el caso del nivel de agua esta enciende una sirena de alta potencia y envía mensajes SMS, los cuales puede visualizarse en la Figura 67, a los números registrados con antelación. Lo último se repite en los casos de alteración de pH y oxígeno, para esto se puede guiar en la Tabla 29.

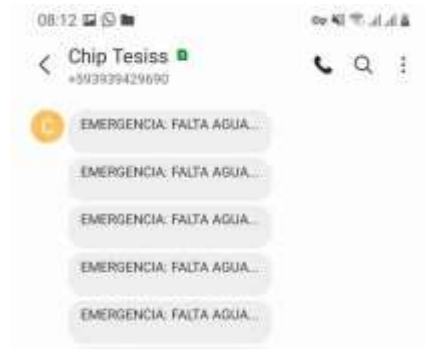
Tabla 29

Tabla de alarmas.

ALARMAS		
VARIABLES	RANGOACEPTABLE	ALARMA
Nivel de Agua	Bajo	Alto
pH	6 – 9.5	0 – 6 o 9.5 – 14
Oxígeno Disuelto	3 - 7	0 - 3 o 7 - 10

Figura 67

Mensajes recibidos por bajo nivel de agua.



4.3.6. Resultados de la Fase 7

Para la verificación de esta fase es necesario se cumplan la fase 1, fase 2 y fase 4, para con ellos visualizar toda la información necesaria dentro de la interfaz. La interfaz maneja una actualización cada 3 segundos presentando la información como en la Figura 68, facilitando el entendimiento para el usuario.

Figura 68

Interfaz del sistema de monitoreo.



4.3.7. Resultados de la Fase 8 y Fase 9

Dentro de la construcción de los gabinetes se tuvo presente un inconveniente de alta relevancia, el clima, el lugar donde se instalarán los gabinetes posee altos niveles de humedad y lluvias. Para esto se utilizó gabinetes metálicos con placas metálicas donde se instalan todos los componentes, Figura 69, adicionalmente cada gabinete contaba con un sellado de silicona para generar una impermeabilidad en los gabinetes. Cada gabinete cuenta con una esponja que absorbe la posible humedad que pueda ingresar al interior del gabinete.

Una vez instalados los gabinetes, se probó el sistema, donde la lluvia permitió confirmar la efectividad de la construcción de los paneles de control. Al abrir cada estación exterior no se podría visualizar la presencia de humedad o en el peor de los casos agua.

Figura 69

Gabinete central montado.



Los gabinetes fueron instalados en cada área correspondiente con una cubierta de Zinc y un empotramiento en la superficie próxima, la cual ayuda a inmobilizarlo como en la Figura 70. Sin embargo, a simple vista se observa que esta cubierta perderá su efectividad a la larga haciendo que se opte por implementar una caja de cemento o una estructura mejor cubierta.

Figura 70

Colocación de gabinetes exteriores.



4.3.8. Resultados de la Fase 10

La Fase 10 describe el análisis de los datos mediante graficas que se obtienen en la aplicación de Weka. Para esto se empieza con una reorganización de los datos obtenidos de las lecturas de los sensores de las estaciones, como se observa en la Figura 71.

Figura 71

Reorganización de datos obtenidos en lectura de sensores.

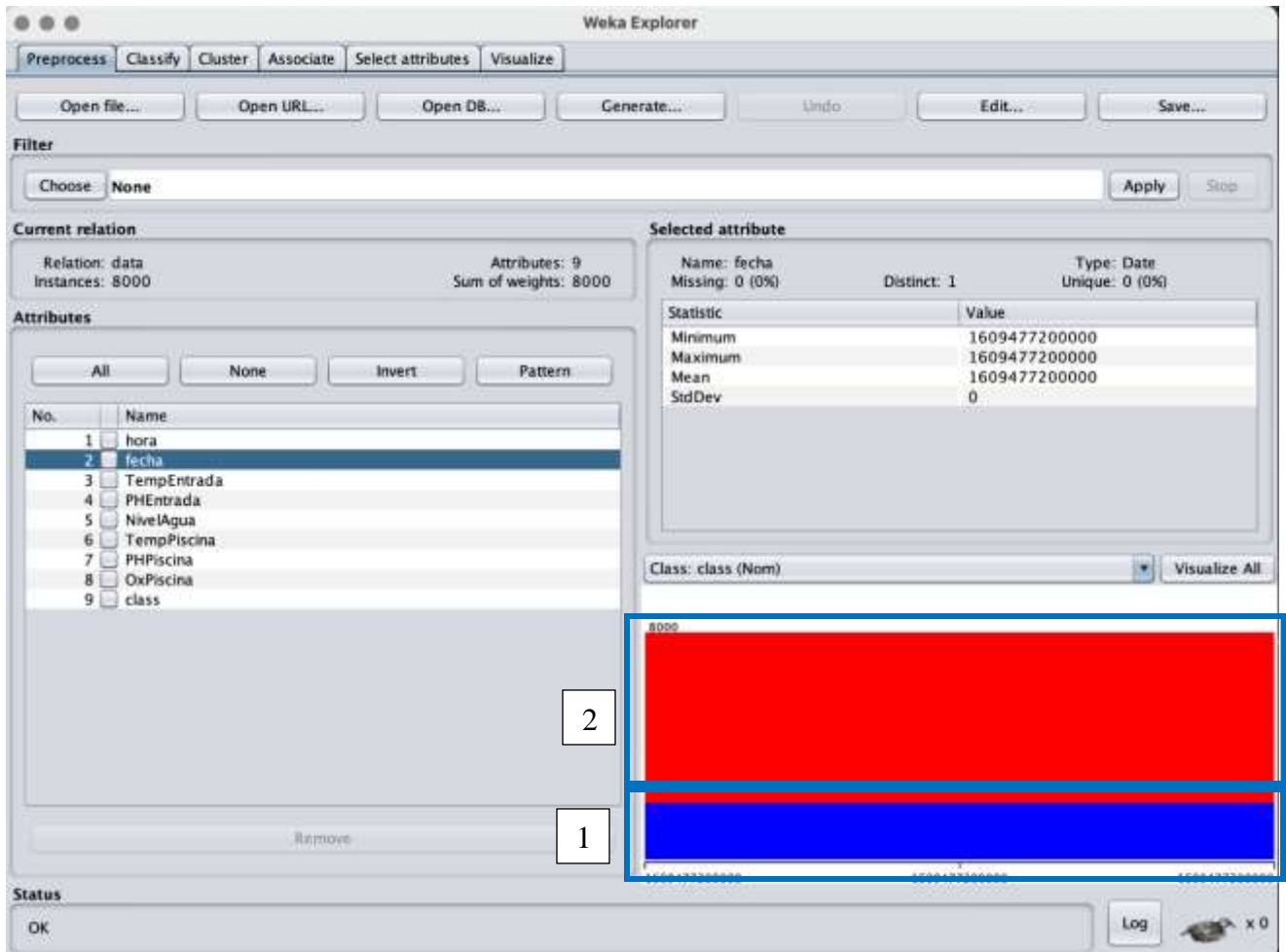
HORA	FECHA	TE	PHE	NE	TP	PHP	OP
13:17	10/12/2021	11.44	9.06	NORMAL	11.81	2.42	1
13:17	10/12/2021	11.44	9.26	NORMAL	11.81	3.13	1
13:17	10/12/2021	11.44	9.16	NORMAL	11.75	2.54	5
13:17	10/12/2021	11.44	9.06	NORMAL	11.75	3.98	2
13:18	10/12/2021	11.44	9.13	NORMAL	11.75	4.07	2
13:18	10/12/2021	11.44	9.35	NORMAL	11.81	2.34	4
13:18	10/12/2021	11.44	7.22	NORMAL	11.81	2.26	5
13:18	10/12/2021	11.44	7.55	NORMAL	11.81	3.12	2
13:18	10/12/2021	11.44	7.86	NORMAL	11.81	3.54	2
13:18	10/12/2021	11.44	8.00	NORMAL	11.81	3.40	2
13:18	10/12/2021	11.44	8.22	NORMAL	11.81	3.21	1
13:18	10/12/2021	11.44	8.33	NORMAL	11.81	0.92	3
13:18	10/12/2021	11.44	8.50	NORMAL	11.81	3.49	2
13:18	10/12/2021	11.44	8.73	NORMAL	11.81	3.30	2
13:18	10/12/2021	11.44	8.81	NORMAL	11.81	3.64	1
13:18	10/12/2021	11.44	9.00	NORMAL	11.81	3.30	5
13:18	10/12/2021	11.44	9.13	NORMAL	11.81	1.21	1
13:19	10/12/2021	11.44	9.10	NORMAL	11.88	3.26	2
13:19	10/12/2021	11.44	9.15	NORMAL	11.88	2.14	3
13:19	10/12/2021	11.44	9.23	NORMAL	11.88	2.76	3
13:19	10/12/2021	11.44	9.32	NORMAL	11.88	3.91	1

Nota. Tabla organizada de datos recolectados por las estaciones instaladas en el criadero.

Al momento de configurar el programa se ajustan parte de los límites que se establecieron en base a la investigación en capítulos anteriores. Estos son los rangos aceptables donde las mediciones de los sensores deben mantenerse. Posterior a esto se ingresan los datos donde se visualizan las primeras graficas de barras donde después que comparar 8000 datos se obtiene: de color rojo las lecturas correctas con un valor de 1 y las lecturas incorrectas de color azul con un valor de 0 como se visualiza en la Figura 72.

Figura 72

Datos ingresados a Weka.

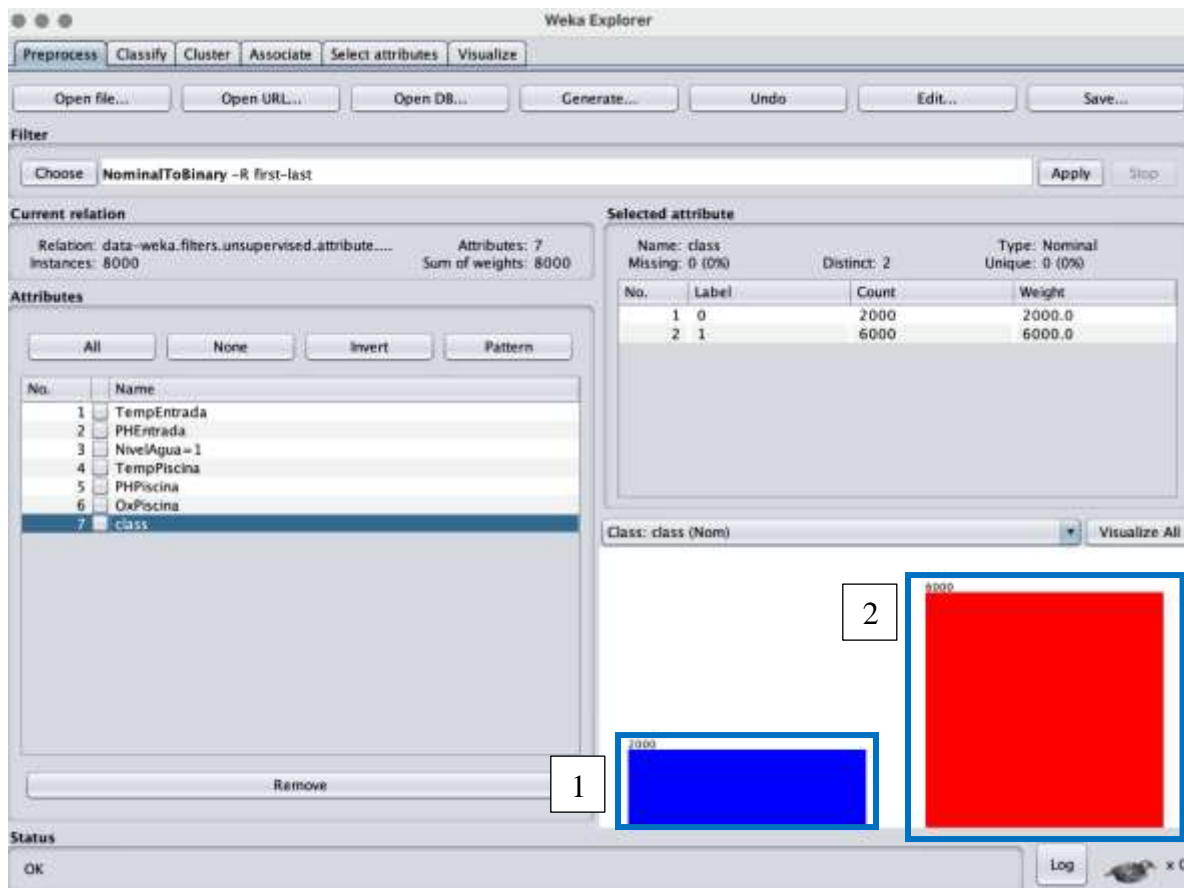


Nota. En el recuadro número 1 se observa una franja de color azul que indica una baja cantidad de datos que están fuera de los valores estimados de las variables climáticas en el criadero. En el recuadro 2 se muestra una franja de color rojo mayor a la anterior la cual indican una mayor cantidad de datos registrados que se encuentran dentro de los valores estimados del criadero.

En la Figura 72 se observa que también se han ingresado los datos de hora y fecha, para eso se los elimina para el análisis teniendo como resultado la Figura 73.

Figura 73

Graficas de los datos obtenidos por los sensores en Weka.



Nota. En el recuadro número 1 se observa una barra de color azul que indica la cantidad ,2000, de datos que no se encuentran dentro los valores correctos para el criadero. En el recuadro 2 se muestra una barra de color roja que indica la cantidad, 6000, de datos que se encuentran dentro de los valores estimados del criadero.

Como primera instancia estas gráficas funcionan como indicadores visuales de “yes” y “no” donde si se cumplen los límites habrá tantos 1 posibles y en caso de no cumplirse tantos 0 posibles.

A continuación, mediante el botón Choose se selecciona el método por el cual se van a analizar los datos obtenidos. Para esto se utiliza el NaiveBayes el cual una vez aplicado muestra una ventana, Figura 74, con los siguientes resultados:

- Porcentaje de clasificación correcta

De manera numérica se muestra el porcentaje de los valores que se encuentran dentro de los rangos establecidos en base a la información obtenida en la investigación.

- Porcentaje de clasificación incorrecta

De manera numérica se muestra el porcentaje de los valores que no se encuentran dentro de los rangos establecidos en base a la información obtenida en la investigación.

- Matriz de confusión

Finalmente, en la matriz de confusión se muestran la cantidad de datos que fueron clasificados correctamente en los dos estados posibles, 0 y 1.

Figura 74

Aplicación del método NaiveBayes en Weka.

The screenshot shows the Weka Explorer interface with the NaiveBayesMultinomial classifier selected. The Classifier output section displays the following data:

Time taken to build model: 0.07 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	7213	90.1625 %
Incorrectly Classified Instances	787	9.8375 %
Kappa statistic	0.7677	
Mean absolute error	0.197	
Root mean squared error	0.3184	
Relative absolute error	52.5161 %	
Root relative squared error	73.521 %	
Total Number of Instances	8000	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0	0.997	0.130	0.719	0.997	0.835	0.789	0.903	0.560	0
1	0.870	0.003	0.999	0.870	0.938	0.789	0.903	0.974	1
Weighted Avg.	0.902	0.035	0.929	0.902	0.906	0.789	0.903	0.873	

=== Confusion Matrix ===

	a	b	Total
a	1994	6	2000
b	781	5219	6000

classified as

a = 0

b = 1

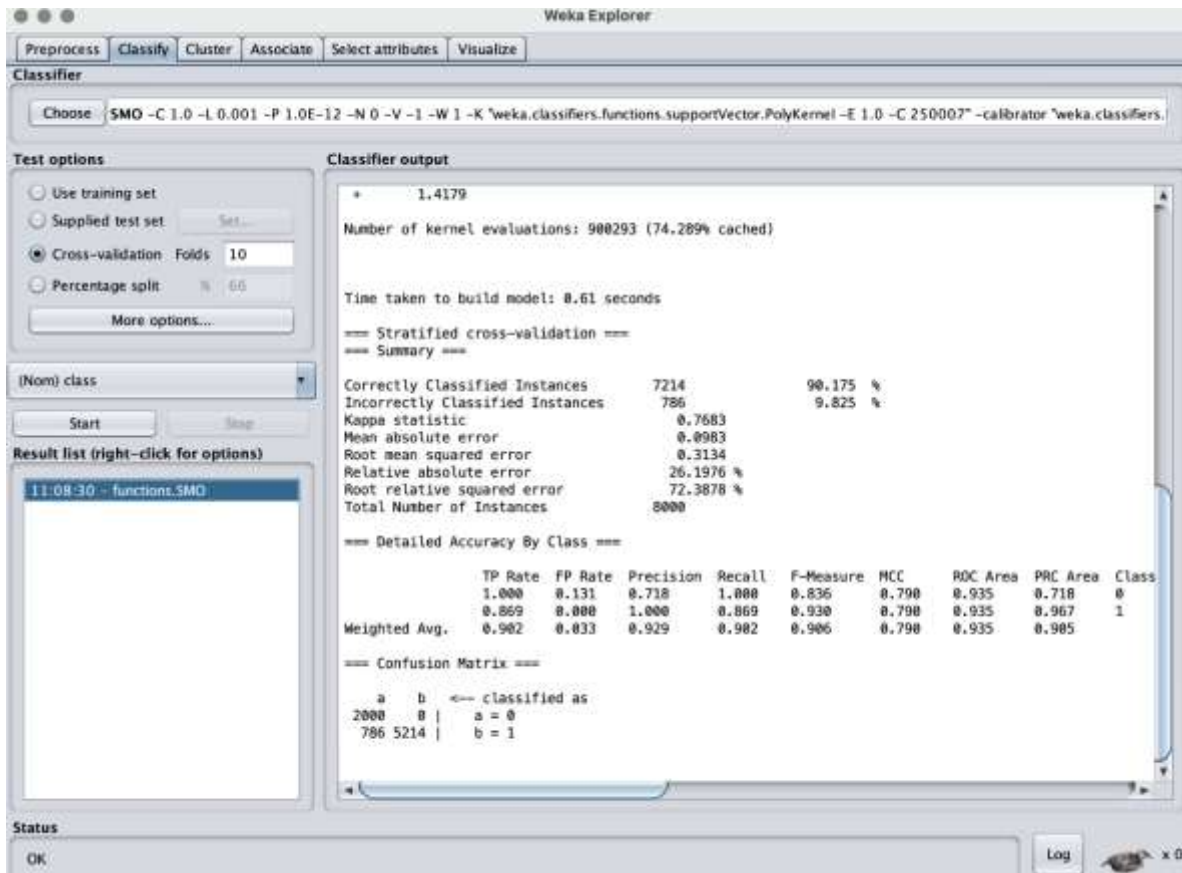
The confusion matrix is highlighted with four numbered boxes: 1 points to the 'a' column header, 2 points to the 'b' column header, 3 points to the 'a' row header, and 4 points to the 'b' row header.

Nota. El valor en el recuadro 1 muestra los negativos correctamente clasificados y el recuadro 2 los negativos incorrectamente clasificados. En el recuadro 4 se indica los positivos correctamente clasificados y en el recuadro 3 los positivos incorrectamente clasificados.

Para lograr generar una comparativa se aplica el método SMO con los mismos datos obteniendo los siguientes resultados que se pueden observar en la Figura 75.

Figura 75

Aplicación del método SMOs en Weka.



Es importante mencionar que cada sistema donde se emplee el proceso de clasificación de datos tendrá como resultado un filtro o filtros en específico que muestre los mejores resultados. Para este caso, se han presentado anteriormente los dos filtros que mejor han ejecutado el proceso de clasificación en los datos obtenidos por el sistema desarrollado.

4.3.9. Resultados de la Fase 11

Dentro de las etapas del proceso de montaje y construcción de detalle ciertas adecuaciones adicionales que se emplearon en conjunto al sistema de monitoreo desarrollado. Estas optimizaciones realizadas generaron diferentes resultados que se pudieron verificar gracias a una entrevista realizada al fundador del criadero, que se puede observar en el Anexo 18, tiempo después de poner el sistema en funcionamiento.

4.3.9.1. Mejoras evidenciadas en base a la entrevista

A continuación, en la Tabla 30 se detallan ciertos aspectos considerados como resultados de mejora y optimización de la producción de truchas arco iris en base a las respuestas obtenidas en la entrevista realizada al fundador del criadero.

Tabla 30

Aspectos y parámetros de mejora en la entrevista realizada al fundador del criadero.

ASPECTOS DE MEJORA AL IMPLEMENTAR EL SISTEMA DE MONITOREO			
ASPECTOS	MAGNITUD DE MEJORA		
	BAJA	MEDIA	ALTA
Optimización de la calidad del agua		X	
Producción de truchas para la venta		X	
Optimización del espacio de la piscina			X

Respuesta ante alteraciones de los parámetros de calidad del agua		X
Reducción de peces muertos	X	
Cantidad de peces por mm ²		X
Tiempo de crecimiento	X	

4.3.9.2. Comparativa de resultados

Una vez definidos algunos aspectos principales donde se ha podido evidenciar la mejora. Se realiza una comparativa, que se observa en la Tabla 31, con valores cuantitativos obtenidos al inicio de la implementación del sistema y valores obtenidos 4 meses después de haberse implementado.

Tabla 31

Comparativa de valores antes y después del implementado el sistema de monitoreo y adecuaciones.

COMPARATIVA DE VALORES SEGÚN ASPECTOS DE MEJORA		
ASPECTOS	ANTES	DESPUÉS
Optimización de la calidad del agua	<i>Oxigenación ingreso piscina</i>	<i>Oxigenación de 5 homogénea</i>
Producción de truchas para la venta	<i>Salida de 100 truchas a la semana</i>	<i>Salida de 152 truchas a la semana</i>
Optimización del espacio de la piscina	<i>55% del espacio de la piscina utilizado</i>	<i>90% del espacio de la piscina utilizado</i>
Respuesta ante alteraciones de los parámetros de calidad del agua	<i>Tiempo de respuesta de 1h</i>	<i>Tiempo de respuesta 10 mins</i>
Reducción de peces muertos	<i>Cantidad aproximada de 40 peces a la semana</i>	<i>Cantidad aproximada de 12 peces a la semana</i>

Cantidad de peces por m ²	15	24
Tiempo de crecimiento	1 - 2 meses	3 - 5 semanas

Dentro de estas comparativas se pueden describir de manera visual algunas de los resultados más notorios. En la Figura 76, se puede observar piscinas consecutivas en las cuales solo una posee la estructura de distribución de puntos de oxigenación donde se puede apreciar mayor presencia de burbujas y movimiento de agua sé que produce por la presencia de más caídas de agua; este efecto produce que los alevines puedan ocupar todo el espacio de la piscina y siempre tengan cerca un golpe de agua donde tomar oxígeno.

Figura 76

Comparación de piscinas del criadero.



Nota. En la Figura 76 se presenta la fotografía de tres piscinas de truchas consecutivas en la cual la piscina central donde se encuentra instalada la estructura de distribución de puntos de oxigenación cuenta con más caídas de agua y turbulencias para la mejor oxigenación del agua a diferencia de las piscinas laterales. En esta piscina central se encuentra implementado el sistema de monitoreo.

En la Figura 76 se observa dos tipos de piscina para truchas. Las piscinas laterales cuentan con una sola caída de agua la cual cumple la función de un punto de oxigenación el cual genera la inyección de oxígeno en el agua de la piscina, esta caída tiene un radio de alcance de un metro aproximadamente lo que provoca que el espacio donde el radio de alcance no llegue se vuelvan puntos bajos en oxígeno mientras más se aleja de la caída de agua resultando en la muerte de peces durante los días. Esta

hipótesis fue comprobada con la medición del oxígeno y pH en diferentes puntos de estas piscinas.

De esta manera, se planifico la idea de implementar más puntos de oxigenación. Se diseño e implemento una estructura tipo jaula, con mayores separaciones entre las patas, de tubo de acero cuadrado con bases ajustables en la altura. La estructura cuenta con la función de brindar soporte a la ampliación de tubería PVC desde la entrada de agua a la piscina hasta las diferentes caídas a una distancia de dos metros. Adicionalmente, se prefirió que las caídas tengan una salida intercalada hacia la derecha y hacia la izquierda provocando que el radio de alcance cubra todo el espacio de la piscina, reduciendo los espacios sin oxigenación hasta casi eliminarlos.

Como consecuencia de la implementación descrita anteriormente se observa la variación de la cantidad de peces en el espacio capturado debido a la existencia de oxígeno y turbulencias a lo largo de la piscina. Esto permite utilizar hasta el 90% del espacio de cada piscina aumentando la cantidad de peces por ciclo de salida y reduciendo el tiempo de crecimiento.

Figura 77

Comparación de la cantidad de peces.



Nota. En la Figura 77 se muestra dos fotografías que captura la presencia de peces en las piscinas de crecimiento. En la imagen de la derecha se observa una piscina donde no se incorporó el sistema de monitoreo, ni el sistema de distribución de puntos de oxigenación donde es mayor el espacio libre entre los alevines. En la imagen de la izquierda se observa una piscina donde si se incorporó el sistema de monitoreo y el sistema de distribución de puntos de oxigenación donde existe una mayor cantidad de peces reduciendo el espacio no aprovechado.

En el criadero Aguas Verdes se encontraban piscinas de cultivo de truchas las cuales poseen una entrada de agua para cada piscina con una extensión de 2m x 12m en las cuales existían espacios completamente vacíos y espacios muy bajos de oxigenación resultando en un bajo aprovechamiento del espacio empleado y vulnerable a cualquier anomalía en el agua que ingresa a la piscina.

Después de ser estudiado el criadero se diseñó un sistema que mejoran las falencias dentro del cultivo de truchas. De esta manera, se implementó un sistema de monitoreo de variables climáticas para el cultivo de truchas en tiempo real dividida en tres estaciones: la primera estación se instala en la entrada de agua del criadero con

sensores de temperatura, pH y nivel de agua; la segunda estación se instala en la piscina a monitorear con sensores de temperatura, pH y oxígeno; finalmente, la estación central que se encarga de mostrar los datos en tiempo real mediante una pantalla LCD con una interfaz minimalista para una interacción directa entre los usuarios y el sistema, además del almacenamiento de los datos obtenidos y la gestión de las alarmas si existe alguna anomalía en el agua que entra al cultivo.

Cada estación cuenta con sensores con un cableado y recubrimiento especializado para trabajo prolongado dentro de líquidos (agua). Cada estación modular instalada en la intemperie cuenta con una protección contra la humedad, agua y polvo. El cableado eléctrico y de comunicación se extiende a lo largo del medio dentro manguera de goma. Para entrelazar las estaciones se utiliza una comunicación Modbus cableada para evitar la pérdida de datos ante medios abruptos. Al implementar este sistema el cultivo adquiere la posibilidad de tener una respuesta inmediata ante anomalías dentro del cultivo mediante el monitoreo continuo y las alarmas implementadas, resguardando la cantidad de peces y aumentando la capacidad de producción para obtener mayores beneficios monetarios.

CONCLUSIONES Y RECOMENDACIONES

En esta sección se detallan las conclusiones y recomendaciones a las cuales se llega después del desarrollo del proyecto.

Conclusiones

Se desarrolló un sistema electrónico modular de monitoreo continuo de la calidad del agua en el cultivo de truchas “Aguas Verdes” como un equipo de control el cual permite a los propietarios almacenar y visualizar los valores de los parámetros de calidad medidos mediante sensores que intervienen directamente en el crecimiento de los alevines como: la temperatura, el pH, el nivel de agua y la cantidad de oxígeno mediante una interfaz minimalista presentada en una pantalla para una verificación constante de las piscinas y con ello dar respuesta inmediata en caso de existir alteraciones de los factores que afecten de manera crítica la producción reduciendo pérdida de peces y optimizando el ciclo de crecimiento de la trucha arco iris.

El sistema cobró vida con el software libre Arduino, el cual, debido a su accesibilidad en su programación e implementación lo hace idóneo. Además, utilizar los modulares de sensores y periféricos compatibles de arduino hace que el sistema electrónico sea versátil y admita continuas configuraciones, adaptaciones y mejoras.

El sistema de monitoreo cubre un área circular de aproximadamente 500 m² donde se encuentran: la entrada de agua desde el río que alimenta a las piscinas, la central de monitoreo y una de las piscinas de crecimiento de alevines; en este lugar se implementó el

protocolo de comunicación MODBUS mediante módulos de pequeño tamaño y de largo alcance mediante cableado perfecto para condiciones adversas como las existentes en el lugar de instalación del equipo como: lluvias intermitentes, neblina, maleza, árboles, humedad; y con ello evitar la pérdida o intermitencia de datos de las mediciones de la calidad de agua a comparación de un sistema inalámbrico.

La implementación del sistema de monitoreo desarrollado en conjunto con nueva estructura de distribución de puntos de oxigenación permitió mejorar las condiciones de la piscina donde se instaló el equipo, de esta manera, se pudo observar la optimización de los espacios muertos de la piscina pasando de un 55% a un 99% de espacio útil, el aumento de la cantidad de peces por m² de aproximadamente un 60% y la reducción del tiempo de crecimiento de los alevines de aproximadamente un 32% resultando en el cambio del tipo de piscina, pasando de una piscina de engorde a una piscina de crecimiento acelerado.

La implementación del sistema de monitoreo de parámetros de calidad de agua permitió evidenciar el aumento de la cantidad de truchas listas para la venta semanalmente, por lo cual, está generado mensualmente un incremento del 32% en las ventas de peces lo cual se traduce en una mayor cantidad de ingresos beneficiando al criadero.

Todo el sistema desarrollado puede ser implementado en otros criaderos de especies diferentes de peces para obtener una mejora en su producción y con ciertas modificaciones específicas volverse un sistema especializado para cada cultivo donde se implemente.

Recomendaciones

Un posterior reemplazo a sensores de categoría semi industrial o industrial aumentaría la capacidad de un monitoreo continuo con una vida útil mucho mayor.

Crear cámaras de monitoreo dentro de cada piscina y entradas de agua ayuda a mantener una estabilidad en las mediciones y evita que partículas intrusas puedan dañar los sensores, en especial los electrodos de los sensores de pH y oxígeno.

La inclusión de Machine Learning en este sistema de monitoreo tuvo un fin incompleto debido a que no existe un actuador o un conjunto de actuadores que solventen las anomalías registradas de manera autónoma.

Es recomendable relacionar el sistema desarrollado con otro sistema de actuadores que en respuesta al análisis de datos realizado por el Machine Learning se genere una retroalimentación y con ello una acción de respuesta de manera autónoma y automática como proyecto a futuro.

Para una mayor utilidad de la incorporación de machine learning es recomendable tener un sistema con manejo de datos en tiempo real.

Un sistema de monitoreo de la calidad de agua con diferentes tipos de alarmas incorporadas ante anomalías como: pH u oxigenación en cultivos peces hace que los usuarios puedan tener una acción inmediata para remediar el problema detectado.

Los cables de alimentación y comunicación deben ser cubiertos por una manguera plástica que asegura una mayor vida útil en medios adversos.

Los líquidos, el agua del río, que son monitoreados deben encontrarse en reposo para obtener una medición exacta y estable.

El flujo de agua moviliza diferentes partículas sólidas y aumenta la presencia de burbujas provocando que los electrodos de los sensores obtengan mediciones con muchas variaciones.

REFERENCIAS

- 4tesosite. (2020). *Modelo Cascada Realimentado*. <https://4tesosite.wordpress.com/modelo-cascada/>
- AgronoMaster. (2017). *2 Sistemas de producción*. <https://agronomaster.com/acuicultura/>.
- Albuquerque, V. H. C., Da Cruz, M. A. A., Rodrigues, J. J. P. C., Lorenz, P., & Korotaev, V. V. (2020). In.IoT - A New Middleware for Internet of Things. *IEEE Internet of Things Journal*, 8(10), 7902–7911. <https://doi.org/10.1109/JIOT.2020.3041699>
- Alvarez, M. (2006). *INFORME SOBRE EL DESARROLLO DE LA ACUICULTURA EN EL ECUADOR*.
- Aprendiendo Arduino. (2017). *Shields Arduino*. <https://aprendiendoarduino.wordpress.com/category/shields/>
- Aprendiendo Arduino. (2019). *Arduino IDE*. <https://aprendiendoarduino.wordpress.com/2016/12/11/ide-arduino/>
- AV Electronics. (2019). *Módulo Conversor TTL a RS485*.
- BBVA. (2019). *Machine Learning: ¿Qué es y como funciona?* <https://www.bbva.com/es/machine-learning-que-es-y-como-funciona/>
- Bhattacharya, P., Tanwar, S., Shah, R., & Ladha, A. (2020). Mobile Edge Computing-Enabled Blockchain Framework—A Survey. *Lecture Notes in Electrical Engineering*, 597, 797–809. https://doi.org/10.1007/978-3-030-29407-6_57
- Biól. (2018). *Innovación en la producción de peces ornamentales*. https://issuu.com/designpublications/docs/-bloq-pam_2024-1_noviembre-diciembre2018/s/11275855
- Blog de Tecnologías. (2018). *Arduino*.

- Buchanan, W. J. (2004). Modbus. *The Handbook of Data Communications and Networks*, 677–687. https://doi.org/10.1007/978-1-4020-7870-5_40
- Cáceres, J. (2002). *Reconocimiento de patrones y el aprendizaje no supervisado*.
- Conxemar. (2018). *Trucha Arco Iris*. <https://conxemar.com/es/trucha-arco-iris>
- Crovi, D. (2016). *Educación en la era de la producción* (pp. 458–461).
- Dey, C., & Sen, S. K. (2020). Industrial Automation Technologies. In *Industrial Automation Technologies*. CRC Press. <https://doi.org/10.1201/9780429299346>
- DFROBOT. (2018). *Sensor de oxígeno disuelto*.
- El Octavo bit. (2020). *Módulo reloj de tiempo real RTC DS3231*. <https://eloctavobit.com/arduino/modulo-reloj-de-tiempo-real-rtc-ds3231/>
- Electro Store. (2020a). *Adaptador Cargador 110V/9V 2ª PLUG DC005*. <https://gruoelectrostore.com/shop/baterias-y-cargadores/cargadores-dc/adaptador-cargador-110v-9v-2a-plug-dc005/>
- Electro Store. (2020b). *GLCD LCD 128x64*.
- ENI, E. (2016). *MySQL: administración y optimización*. <https://www.ediciones-eni.com/open/mediabook.aspx?idR=862b01442cece4eb14645431fd097156>
- Fernández, F. (2013). *Aprendizaje por refuerzo*.
- García, Cristina Gómez, I. (2006). *Algoritmos de aprendizaje: KNN & KMEANS*.
- Gómez Gómez, A. (2008). *Proyecto de acuicultura en aguas abiertas*. <https://repositorio.upct.es/handle/10317/35>
- González, F., & Calero, S. (2019). *Comparación de las metodologías cascada y ágil para el aumento de la productividad en el desarrollo de software*.
- González, P. (2007). *Metodologías de gestión de proyectos, alcance, impacto y tendencias*.
- Hernández, M., & Aquino, G. (2008). *MANUAL BÁSICO PARA EL CULTIVO DETRUCHA*

ARCOIRIS.

Hidalgo Granja, C. R., & Sarzosa Pruna, C. N. (2013). *Diseño y automatización de un sistema de control mediante el PLC S7-200 a los niveles de agua temperatura y oxígeno para un criadero piscícola en el sector de Saragozín.*

Isaiaranda. (2020). *¿Qué es Weka?* <https://medium.com/@isaiaranda15/que-es-weka-926c05050d44>

Julpin. (2018). *Shields Multi – propósito, de Electrónica y Robótica.* <http://www.julpin.com.co/inicio/shields-o-escudos/606-shield-o-escudo-multi-proposito-v1-con-varios-sensores-para-arduino.html>

Lamingo LLumiquinga, J. P., & Moreno Garzón, V. N. (2014). *Análisis del comportamiento del oxígeno disuelto y la demanda química de oxígeno en tres puntos (Lasso, Latacunga, Salcedo) del río Cutuchi, en la provincia de Cotopaxi, cantón Latacunga, periodo 2013 - 2014.* <http://repositorio.utc.edu.ec/handle/27000/2753>

Llamas, L. (2021). *Conectar Arduino a Internet.*

Lujan Monja, M., & Cruajulca, A. (2014). Levels of domestication in fish: Implications for the sustainable future of aquaculture. *Fish and Fisheries*, 15(2), 181–195. <https://doi.org/10.1111/FAF.12006>

Mocito, J., Rosa, L., Almeida, N., Miranda, H., Rodrigues, L., & Lopes, A. (2009). Context adaptation of the communication stack. In <https://doi.org/10.1080/17445760500356098> (Vol. 21, Issue 3). Taylor & Francis Group . <https://doi.org/10.1080/17445760500356098>

Modbus-IDA. (2006). *Modbus Application Protocol Specification.* www.modbus.org/docs/Modbus_Application_Protocol_V1_1b.pdf

Moreno, A. (2018). *Arduino: Curso práctico.*

<https://books.google.com.ec/books?id=yofDwAAQBAJ&pg=PA186&dq=tipos+de+arduino&hl=es&sa=X&ved=2ahUKEwiKspXrIOvzAhWoRTABHQqRBECQ6AF6B-AgJEAI#v=onepage&q=tipos+de+arduino&f=false>

Murky Robot. (2018). *DS18B20*.

Nash, R., Shibaev, S., & Besenyei, L. (2021). *AQUACULTURE with focus on Vietnam and Thailand*.

https://www.researchgate.net/publication/352762760_AQUACULTURE_with_focus_on_Vietnam_and_Thailand

Naylamp. (2020). *Módulo Relay 2CH 5VDC*.

Naylamp Mechatronics. (2020). *Módulo Lector de Memoria Micro SD Card*.

<https://naylampmechatronics.com/104-modulo-lector-de-memoria-micro-sd-card.html>

Ochoa, L. M. L. (2019). *An Analysis of Students' Urban Mobility using Arequipa Smart Mobility Application*.

Oshurko, I. (2016). *Quantum machine learning*.

Padrón Maiz, R. A., Valero Lacruz, L., & Gudiño Torres, M. J. (2013). IMPORTANCIA DEL REGISTRO DE VARIABLES FÍSICO-QUÍMICAS EN EL CULTIVO DE TRUCHAS EN LOS ANDES TROPICALES. *Mundo Pecuario, IX, N° 1*, 1–11.

PdD. (2020). *Todo sobre las placas de desarrollo*. <https://www.placasdedesarrollo.com/>

Rakocy. (2009). *Oreochromis niloticus*.

https://www.fao.org/fishery/docs/DOCUMENT/aquaculture/CulturedSpecies/file/es/es_niletilapia.htm

Rivera Herrera, D. I., & Yepez Aroca, E. A. (2015). *Diseño e implementación de un prototipo para la medición de calidad del agua y control de la oxigenación en forma remota orientado a la producción acuícola*. <http://dspace.ups.edu.ec/handle/123456789/10328>

- Rodríguez Gómez, H. (2001). La calidad del agua y la productividad de un estanque en acuicultura. In *Fundamentos de acuicultura continentalp.* 43-74.
http://digitool.gsl.com.mx:1801/webclient/DeliveryManager?pid=64986&custom_att_2=direct4.
- Román, V. (2019a). *Aprendizaje supervisado: Introducción a la clasificación y principales algoritmos.*
- Román, V. (2019b). *Aprendizaje Supervisado.*
- Semle, A., & eFalcom. (2016). *Protocolos IOT para considerar.*
https://editores.com.ar/sites/default/files/aa2_semle_protocolos_ilot.pdf
- Sen, S. K. (2021). Fieldbus and Networking in Process Automation. In *Fieldbus and Networking in Process Automation.* CRC Press.
<https://doi.org/10.1201/9781003149941>
- Soto, C., & Jiménez, C. (2011). *Aprendizaje supervisado para la discriminación y clasificación difusa.*
- Unit Electronics. (2020a). *Sensor de Nivel.* <https://uelectronics.com/producto/sensor-de-nivel-flotador-horizontal-para-arduino/>
- Unit Electronics. (2020b). *Sensor de PH líquido.* <https://uelectronics.com/producto/sensor-de-ph-liquido/>
- Valdés, F., & Pallás, R. (2017). *Microcontroladores Fundamentos y Aplicaciones con PIC - Fernando Valdes, Ramón Pallás Areny - Google Libros.*
<https://books.google.hn/books?id=ODenKGOHMRkC&printsec=copyright#v=onepage&q&f=false>
- Vargas Conde, M. (2018). *Trucha Arco Iris.*
<https://es.scribd.com/document/430772356/trucha-arco-iris>

Zhi-hong, Q., & Yi-jun, W. (2012). IoT Technology and Application. *ACTA ELECTONICA SINICA*, 40(5), 1023. <https://doi.org/10.3969/J.ISSN.0372-2112.2012.05.026>

ANEXOS

Anexo 1. Entrevista al fundador del criadero



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
CARRERA DE INGENIERÍA EN ELECTRÓNICA Y REDES DE
COMUNICACIÓN

ENTREVISTA EN EL CRIADERO “AGUAS VERDES”

Fecha: 24/06/2021

Lugar: Tufiño

Entrevistador: Edison Paspuezán

Entrevistado: José Domingo Paspuezán

A continuación, se presenta un escrito que resume la entrevista realizada al señor () encargado del cultivo de truchas AGUAS VERDES.

De manera introductoria se realizó una inducción acerca de los sistemas embebidos que pueden ser implementados en los criaderos haciendo énfasis en un sistema que facilite el cultivo de los peces con el monitoreo de la calidad del agua para optimizar la producción. A partir de esto se pidió información acerca de los problemas más recurrentes que existían para con ello lograr estructurar el funcionamiento del sistema a implementarse.

De esta manera se trató varias experiencias enfatizando en los siguientes temas:

El señor José Domingo Paspuezán encargado del criadero comenta que en una ocasión se perdió el cultivo de truchas donde alrededor de 5000 peces por motivo de un taponamiento en el canal de ingreso de agua hacia las piscinas eliminando el flujo de agua. Sugiriendo de esta manera alguna especie de alarma que informe si vuelve a suceder algo parecido.

En otra ocasión la pérdida fue de 2000 truchas las cuales fueron intoxicadas, ya que se había soltado químicos agrícolas en el agua del afluente que se utiliza para el criadero. La sugerencia es que se pueda visualizar cuando el pH del agua se altere.

Una condición para obtener buenos resultados en el crecimiento de los alevines es la oxigenación del agua. Es la razón por la cual se sugiere un indicador de la oxigenación y mejorar las condiciones de este factor.

También se hace énfasis con el control de calidad y sanidad para poder inspeccionar como se maneja el cultivo de las truchas en el lugar. Es un sistema que no se aplica y no se ha pensado por el momento, pero se desearía implementarlo a futuro.

Anexo 2. Código de programación arduino del nodo situado en la entrada de agua

```
#include <SoftwareSerial.h>
#include <OneWire.h>
OneWire ourWire(8);
#include <DallasTemperature.h>
DallasTemperature sensors(&ourWire);

float Temp;
float Ph;
float Voltaje;
float Calibracion = 21.30;
unsigned long int avgval;
int Promedio[10],Tempe;
int Valorph;
int Nivel;

void setup() {
  Serial.begin(9600);
  pinMode(9, INPUT);
  pinMode(2, OUTPUT);
  pinMode(2, LOW);
}

void loop() {
  PH();
  TEM();
  NL();
  ENVIAR();
}
```



```

void PH() {
  for(int i=0;i<10;i++)
  {
    Promedio[i]=analogRead(A0);
    delay(30);
  }
  for(int i=0;i<9;i++)
  {
    for(int j=i+1;j<10;j++)
    {
      if(Promedio[i]>Promedio[j])
      {
        Tempe=Promedio[i];
        Promedio[i]=Promedio[j];
        Promedio[j]=Tempe;
      }
    }
  }
  avgval=0;
  for(int i=2;i<8;i++)
  avgval+=Promedio[i];
  Voltaje=avgval*5.0/1024/6;
  Ph = -5.70 * Voltaje + Calibracion;
}

void TEM() {
  sensors.requestTemperatures(); //Se envía el comando para leer la temperatura
  Temp = sensors.getTempCByIndex(0); //Se obtiene la temperatura en °C
}

void NL() {
  if(digitalRead(10) == 1){
    Nivel = 1;
  }else{
    Nivel = 0;
  }
}

void ENVIAR() {
  digitalWrite(2,HIGH);
  Serial.print(Temp);
  Serial.print("/");
  Serial.print(Ph);
  Serial.print("/");
  Serial.println(Nivel);
  digitalWrite(2,LOW);
}

```

Anexo 3. Código de programación arduino del nodo situado en la piscina de crecimiento

```

//LIBRERÍAS
#include <OneWire.h> //Libreria de Un Cable
#include <DallasTemperature.h> //Libreria de Temperatura
#include <Arduino.h> //Libreria de Arduino

//ASIGNACIONES
OneWire ourWire(8); //Asignar un Cable Pin 8
DallasTemperature sensors(&ourWire); //Asignar Sensor

//VARIABLES
#define DO_PIN A3

#define VREF 5000 //VREF (mv)
#define ADC_RES 1024 //ADC Resolution

//Single-point calibration Mode=0
//Two-point calibration Mode=1
#define TWO_POINT_CALIBRATION 0

#define READ_TEMP (25) //Current water temperature °C, Or temperature sensor function

//Single point calibration needs to be filled CAL1_V and CAL1_T
#define CAL1_V (253) //mv
#define CAL1_T (11) //°C
//Two-point calibration needs to be filled CAL2_V and CAL2_T
//CAL1 High temperature point, CAL2 Low temperature point
#define CAL2_V (1300) //mv
#define CAL2_T (15) //°C

const uint16_t DO_Table[41] = {
  14460, 14220, 13820, 13440, 13090, 12740, 12420, 12110, 11810, 11530,
  11260, 11010, 10770, 10530, 10300, 10080, 9860, 9660, 9460, 9270,
  9080, 8900, 8730, 8570, 8410, 8250, 8110, 7960, 7820, 7690,
  7560, 7430, 7300, 7180, 7070, 6950, 6840, 6730, 6630, 6530, 6410};

uint8_t Temperature;
uint16_t ADC_Raw;
uint16_t ADC_Voltage;
uint16_t DO;

int16_t readDO(uint32_t voltage_mv, uint8_t temperature_c)
{
  if TWO_POINT_CALIBRATION == 00
    uint16_t V_saturation = (uint32_t)CAL1_V + (uint32_t)35 * temperature_c - (uint32_t)CAL1_T * 35;
    return (voltage_mv * DO_Table[temperature_c] / V_saturation);
  else
    uint16_t V_saturation = (int16_t)((int8_t)temperature_c - CAL2_T) * ((uint16_t)CAL1_V - CAL2_V) / ((uint8_t)CAL1_T - CAL2_T) + CAL2_V;
    return (voltage_mv * DO_Table[temperature_c] / V_saturation);
  endif
}

float temp; //Valor Temperatura
String oxigeno; //Valor de Oxigeno
float Ph; //Valor de Ph
float Voltaje; //Valor de Voltaje
float Calibracion = 21.45; //Valor de Calibración

```

```

unsigned long int avgval;           //Valor avg
int Promedio[10],Tempe;           //Promedio de Ph
int Valorph;                       //Valor Ph no Calibrado

void setup() {

    Serial.begin(9600);             //Iniciar Serial
    pinMode(2, OUTPUT);            //Pin de Envio Datos
    pinMode(2, LOW);                //Pin Datos Recibir
}

void loop() {
    Temperatura();                  //Obtener Temperatura
    PH();                            //Obtener PH
    Oxigeno();                       //Obtener Oxigeno
    Enviar();                         //Enviar Datos
}

void Temperatura() {
    sensors.requestTemperatures();
    temp= sensors.getTempCByIndex(0);
}

void Oxigeno(){
    Temperaturet = (uint8_t)READ_TEMP;
    ADC_Raw = analogRead(DO_PIN);
    ADC Voltage = uint32 t(VREF) * ADC Raw / ADC RES;
}

```

```

    oxigeno = (readDO(ADC_Voltage, Temperaturet))/1000;
}

void Enviar() {
    digitalWrite(2, HIGH);
    Serial.print(temp);
    Serial.print("/");
    Serial.print(Ph);
    Serial.print("/");
    Serial.println(oxigeno);
    digitalWrite(2, LOW);
}

void PH() {
    for(int i=0;i<10;i++)
    {
        Promedio[i]=analogRead(A2);
        delay(30);
    }
    for(int i=0;i<9;i++)
    {
        for(int j=i+1;j<10;j++)
        {
            if(Promedio[i]>Promedio[j])
            {
                Tempe=Promedio[i];
                Promedio[i]=Promedio[j];
            }
            {
                Tempe=Promedio[i];
                Promedio[i]=Promedio[j];
                Promedio[j]=Tempe;
            }
        }
    }
    avgval=0;
    for(int i=2;i<8;i++)
    avgval+=Promedio[i];
    Voltaje=avgval*5.0/1024/6;
    Ph = -5.70 * Voltaje + Calibracion;
}

```

Anexo 4. Código de programación arduino del nodo central de monitoreo

```

//LIBRERÍAS
#include <Separador.h> //Librería de Separación de Cadena
#include <SoftwareSerial.h> //Librería de Comunicación Serial
#include <U8g2lib.h> //Librería para Pantalla LCD 16x2
U8G2_ST7920_128X64_F_SW_SPI u8g2(U8G2_R0, 8, 9, 10, U8X8_PIN_NONE); //Designación de LCD
#include <RTCLib.h> //Librería de Reloj
#include <SD.h> //Librería para SD
#include <SPI.h> //Librería para Comunicación SPI

//ASIGNACIONES
int display = 1; //Asignar la Pantalla 1
RTC_DS3231 RTC; //Asignar RTC DS3231
const int SSpin = 53; //Asignar Activador SD

//VARIABLES
String fecha, hora, dia, mes, ano, horas, minutos; //Variables para Reloj

const int Puerta = 2; //Variables para Separador de Puerta
String puerta, p1, p2, p3;
Separador p;

const int Piscina=3; //Variables para Separador de Piscina
String piscina,a1,a2,a3;
Separador a;

File dataFile; //Archivo de Nombre
String guarda; //Variable de Datos

int Alarma = 11; //Variable de Alarma

int Alarma = 11; //Variable de Alarma
String p31;

//GRÁFICOS
const uint8_t hmi[] U8X8_PROGMEM = {
  0x0f, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0xf0, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x80, 0xfd, 0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xbf,
  0x05, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x20, 0x04, 0x00, 0x10, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x20, 0x20, 0x04, 0x7f, 0x18, 0x20,
  0x1c, 0x11, 0xc2, 0x21, 0xf4, 0x3c, 0x1e, 0x8f, 0x03, 0xfe, 0x30, 0x20,
  0x84, 0xff, 0x1c, 0x30, 0x22, 0x11, 0x23, 0x42, 0x12, 0x44, 0x22, 0x41,
  0x04, 0xff, 0x3b, 0x20, 0x44, 0xfe, 0x1f, 0x50, 0x02, 0x11, 0x25, 0x40,
  0x12, 0x44, 0x42, 0x41, 0x80, 0xfc, 0x3f, 0x20, 0x44, 0xfe, 0x1f, 0x48,
  0x32, 0x91, 0xc4, 0x41, 0xf2, 0x3c, 0x42, 0x8f, 0x83, 0xfc, 0x3f, 0x20,
  0xc4, 0xff, 0x1f, 0x78, 0x22, 0x91, 0x07, 0x81, 0x12, 0x24, 0x42, 0x01,
  0x84, 0xff, 0x3f, 0x20, 0x84, 0xff, 0x1c, 0x88, 0x22, 0x51, 0x28, 0x82,
  0x11, 0x44, 0x22, 0x41, 0x04, 0xff, 0x39, 0x20, 0x04, 0x7c, 0x18, 0x84,
  0x1c, 0x4e, 0xc8, 0x81, 0xf1, 0x44, 0x1e, 0x9f, 0x03, 0xf8, 0x30, 0x20,
  0x04, 0x00, 0x10, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x20, 0x20, 0x04, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x20, 0x04, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x20,
  0x04, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x20, 0x04, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x20, 0x04, 0xf0, 0x44, 0xdf,

```

```

//GRAFICADORES
void HMI(void) {
  u8g2.drawXBMP( 0, 0, 128, 64, hmi);
  u8g2.sendBuffer();
}

void PORTADA(void) {
  u8g2.drawXBMP( 0, 0, 128, 64, port);
  u8g2.sendBuffer();
  delay(1000);
}

void setup()
{
  Serial.begin(9600); //Inicia Serial
  Serial3.begin(9600); //Inicia Serial 3
  RTC.begin(); //Inicia RTC
  //RTC.adjust(DateTime(__DATE__, __TIME__)); //Ajustar RTC
  pinMode(Puerta, OUTPUT); //Activador de Modbus Puerta
  digitalWrite(Puerta, LOW); //Modbus Puerta Recibir
  pinMode(Piscina, OUTPUT); //Activador de Modbus Piscina
  digitalWrite(Piscina, LOW); //Modbus Piscina Recibir
  pinMode(4, OUTPUT); //Activador de Modbus Enviar
  digitalWrite(4, LOW); //Modbus Recibir
  //GLCD
  u8g2.begin(); //Iniciar LCD 128x64

  u8g2.setFontMode(1); //Seleccionar LCD Modo 1
  u8g2.setDrawColor(1); //Seleccionar LCD Color 1
  PORTADA(); //Graficar Portada
  pinMode(Alarma, OUTPUT); //Pin Alarma Salida
}

void loop()
{
  E1(); //Recibir y Separar Datos Puerta
  E2(); //Recibir y Separar Datos Piscina
  E3(); //Obtener Datos Clock
  ALARMA(); //Revisión de Alarma
  GLCD(); //Graficar Datos
  GUARDAR(); //Guardar Datos
  EGSM(); //Enviar GSM
}

void E1()
{
  puerta= Serial.readString();
  p1= p.separa(puerta, '/', 0);
  p2= p.separa(puerta, '/', 1);
  p3= p.separa(puerta, '/', 2);
}
}

```

```

void E1()
{
  puerta= Serial.readString();
  p1= p.separa(puerta, '/', 0);
  p2= p.separa(puerta, '/', 1);
  p3= p.separa(puerta, '/', 2);

}

void E2()
{
  piscina= Serial3.readString();
  a1= a.separa(piscina, '/', 0);
  a2= a.separa(piscina, '/', 1);
  a3= a.separa(piscina, '/', 2);
}

void E3()
{
  DateTime now = RTC.now();
  dia = now.day();
  mes = now.month();
  ano = now.year();
  horas = now.hour();
  minutos = now.minute();
  fecha = dia+"/"+mes+"/"+ano;
  hora = horas+":"+minutos;
}

void ALARMA()
{
  if(p3.toInt() == 1)
  {
    digitalWrite(Alarma, HIGH);
    p31 = "NORMAL";
  }
  if(p3.toInt() == 0)
  {
    digitalWrite(Alarma, LOW);
    p31 = "ALARMA";
  }
}

```

```

void GLCD()
{
    HMI();
    u8g2.setFont(u8g2_font_6x10_tf);
    //Entrada
    u8g2.setCursor(24,33);
    u8g2.print(p1.toFloat());
    u8g2.setCursor(24,41);
    u8g2.print(p2.toFloat());
    u8g2.setCursor(24,49);
    if(p3.toInt() == 1){
        u8g2.print("Normal");
    }else{
        u8g2.print("Alarma");
    }
    //Piscina
    u8g2.setCursor(86,33);
    u8g2.print(a1.toFloat());
    u8g2.setCursor(86,41);
    u8g2.print(a2.toFloat());
    u8g2.setCursor(86,49);
    u8g2.print(a3.toFloat());
    //Hora
    u8g2.setCursor(20,59);
    u8g2.print(hora);
    //Fecha
    u8g2.setCursor(66,59);
    u8g2.print(fecha);

    u8g2.sendBuffer();
    delay(50);
}

void GUARDAR()
{
    SD.begin(53);
    dataFile = SD.open("Aguas.csv", FILE_WRITE);
    dataFile.print(hora);
    dataFile.print(",");
    dataFile.print(fecha);
    dataFile.print(",");
    dataFile.print("T(E)");
    dataFile.print(",");
    dataFile.print(p1);
    dataFile.print(",");
    dataFile.print("PH(E)");
    dataFile.print(",");
    dataFile.print(p2);
    dataFile.print(",");
    dataFile.print("N(E)");
    dataFile.print(",");
    dataFile.print(p3l);
    dataFile.print(",");
    dataFile.print("T(P)");
    dataFile.print(",");
    dataFile.print(al);
}

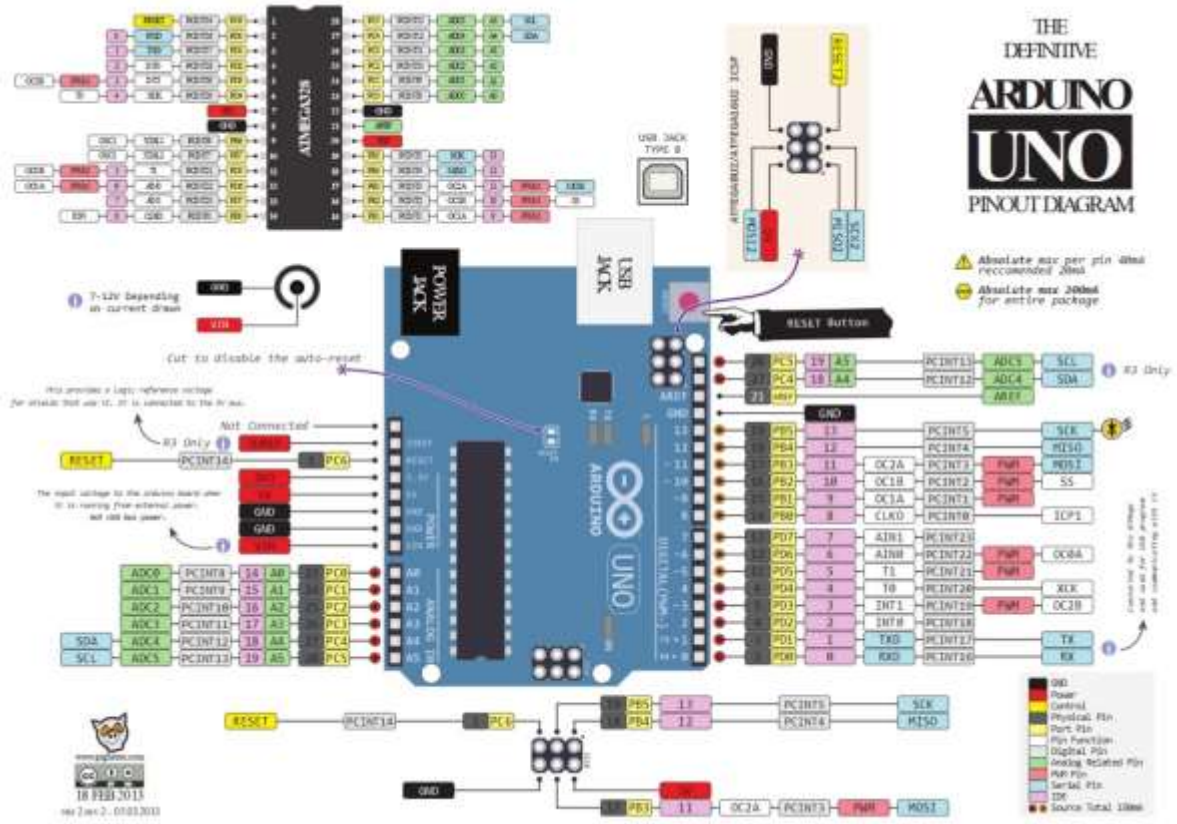
```



```
dataFile.print(",");
dataFile.print("PH(P)");
dataFile.print(",");
dataFile.print(a2);
dataFile.print(",");
dataFile.print("O(P)");
dataFile.print(",");
dataFile.println(a3);
dataFile.close();
}

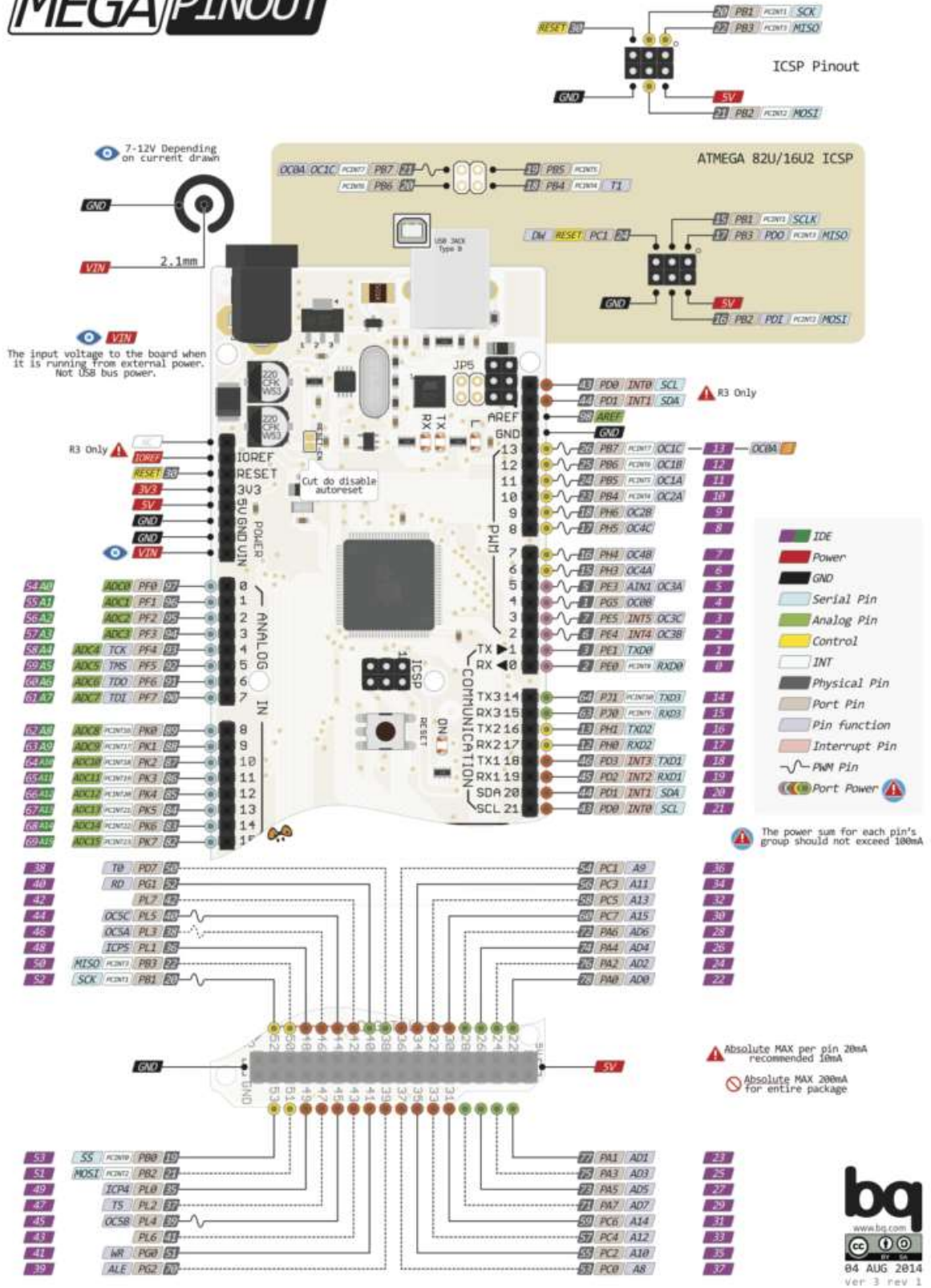
void EGSM()
//Funcion para mandar mensaje de texto
{
  String Numeros_Movil [] = { /*#1*/"+593968439323", /*#2*/"+593967077482", /*#3*/"+593983316796"};
  int Cantidad_Movil = 3;
  SIM900.print("AT+CMGF=1\r");
  delay(100);
  for (int n = 0; n < Cantidad_Movil; n++)
  {
    delay(200);
    SIM900.println("AT+CMGS=\"" + Numeros_Movil[n] + "\"");
    delay(100);
    SIM900.println("EMERGENCIA: FALTA AGUA...");
    delay(100);
    SIM900.println((char)26);
    delay(100);
    SIM900.println();
    delay(1000);
    Serial1.println("SMS sent successfully");
  }
}
```

Anexo 5. Diagrama de pines del arduino UNO



Anexo 6. Diagrama de pines del arduino Mega

MEGA PINOUT



Anexo 7. Datasheet DS18B20



DS18B20

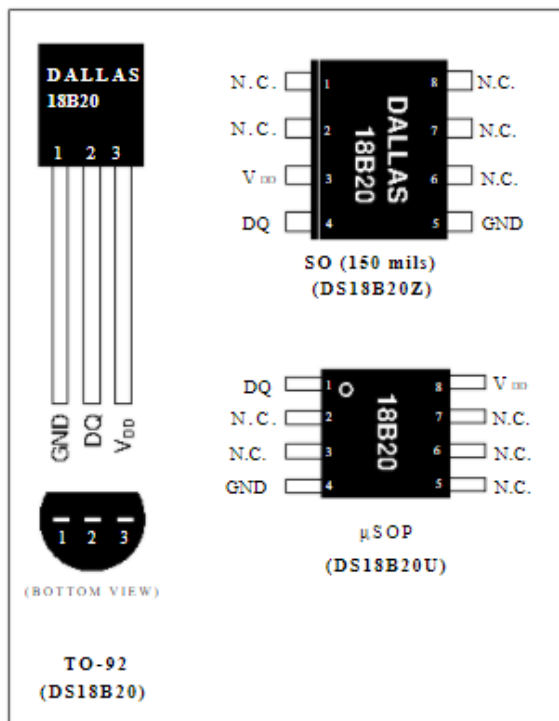
Programmable Resolution 1-Wire Digital Thermometer

www.maxim-ic.com

FEATURES

- Unique 1-Wire® Interface Requires Only One Port Pin for Communication
- Each Device has a Unique 64-Bit Serial Code Stored in an On-Board ROM
- Multidrop Capability Simplifies Distributed Temperature-Sensing Applications
- Requires No External Components
- Can Be Powered from Data Line; Power Supply Range is 3.0V to 5.5V
- Measures Temperatures from -55°C to +125°C (-67°F to +257°F)
- ±0.5°C Accuracy from -10°C to +85°C
- Thermometer Resolution is User Selectable from 9 to 12 Bits
- Converts Temperature to 12-Bit Digital Word in 750ms (Max)
- User-Definable Nonvolatile (NV) Alarm Settings
- Alarm Search Command Identifies and Addresses Devices Whose Temperature is Outside Programmed Limits (Temperature Alarm Condition)
- Available in 8-Pin SO (150 mils), 8-Pin μ SOP, and 3-Pin TO-92 Packages
- Software Compatible with the DS1822
- Applications Include Thermostatic Controls, Industrial Systems, Consumer Products, Thermometers, or Any Thermally Sensitive System

PIN CONFIGURATIONS



DESCRIPTION

The DS18B20 digital thermometer provides 9-bit to 12-bit Celsius temperature measurements and has an alarm function with nonvolatile user-programmable upper and lower trigger points. The DS18B20 communicates over a 1-Wire bus that by definition requires only one data line (and ground) for communication with a central microprocessor. It has an operating temperature range of -55°C to +125°C and is accurate to ±0.5°C over the range of -10°C to +85°C. In addition, the DS18B20 can derive power directly from the data line ("parasite power"), eliminating the need for an external power supply.

Each DS18B20 has a unique 64-bit serial code, which allows multiple DS18B20s to function on the same 1-Wire bus. Thus, it is simple to use one microprocessor to control many DS18B20s distributed over a large area. Applications that can benefit from this feature include HVAC environmental controls, temperature monitoring systems inside buildings, equipment, or machinery, and process monitoring and control systems.

Anexo 8. Datasheet PH E-201C

PH meter(SKU: SEN0161)

From Robot Wiki

Contents

- 1 Introduction
- 2 Applications
- 3 Specification
- 4 pH Electrode Size
- 5 pH Electrode Characteristics
- 6 Use the pH Meter
 - 6.1 Connecting Diagram
 - 6.2 Step to Use the pH Meter
 - 6.3 Sample Code
- 7 Precautions
- 8 Documents



Analog pH Meter Kit

Introduction

Need to measure water quality and other parameters but haven't got any low cost pH meter? Find it difficult to use with Arduino?

Here comes an analog pH meter, specially designed for Arduino controllers and has built-in simple, convenient and practical connection and features. It has an LED which works as the Power Indicator, a BNC connector and PH2.0 sensor interface. To use it, just connect the pH sensor with BNC connector, and plug the PH2.0 interface into the analog input port of any Arduino controller. If pre-programmed, you will get the pH value easily. Comes in compact plastic box with foams for better mobile storage.

Attention:In order to ensure the accuracy of the pH probe, you need to use the standard solution to calibrate it regularly. Generally, the period is about half a year. If you measure the dirty aqueous solution, you need to increase the frequency of calibration.

Applications

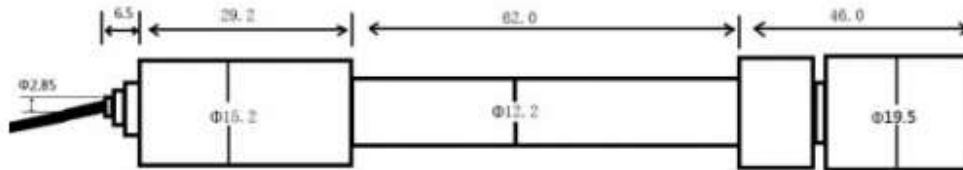
- Water quality testing
- Aquaculture

Specification

- Module Power : 5.00V
- Module Size : 43mm×32mm
- Measuring Range:0-14PH
- Measuring Temperature :0-60 °C

- Accuracy : $\pm 0.1\text{pH}$ (25 °C)
- Response Time : $\leq 1\text{min}$
- pH Sensor with BNC Connector
- PH2.0 Interface (3 foot patch)
- Gain Adjustment Potentiometer
- Power Indicator LED
- Cable Length from sensor to BNC connector:660mm

pH Electrode Size



pH Electrode Characteristics

The output of pH electrode is Millivolts, and the pH value of the relationship is shown as follows (25 °C):

VOLTAGE (mV)	pH value	VOLTAGE (mV)	pH value
414.12	0.00	-414.12	14.00
354.96	1.00	-354.96	13.00
295.80	2.00	-295.80	12.00
236.64	3.00	-236.64	11.00
177.48	4.00	-177.48	10.00
118.32	5.00	-118.32	9.00
59.16	6.00	-59.16	8.00
0.00	7.00	0.00	7.00

Anexo 9. Datasheet ZP 5210

ZP5210 Sensor de nivel de agua vertical



El sensor de nivel de líquido es un interruptor flotador, este dispositivo es utilizado para medir el nivel de líquido dentro de un tanque, el sensor puede activar una bomba, una alarma u otros dispositivos. Utiliza este sensor en una pecera, aires acondicionados, tanques de agua, y otras más aplicaciones, donde se requiera controlar el nivel de líquido.

Este sensor de nivel de líquido se compone de un interruptor y un flotador magnético de láminas, el interruptor de láminas magnético esta dentro del encapsulado de la barra guía, cuando el flotador sube y baja con el nivel de líquido, sus imanes internos atraen al interruptor de lámina magnéticos, con el fin de detectar la posición de nivel de líquido como un control de nivel de líquido.

- **Material:** Polímero y PP
- **Represión de contacto (Max):** 10 Watts
- **Voltaje del switch (Max):** 50 VDC
- **Corriente del switch (Max):** 0.5 A



ElectroCrea.com

Anexo 10. Datasheet RS-485



MAX481/MAX483/MAX485/ MAX487-MAX491/MAX1487

Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers

General Description

The MAX481, MAX483, MAX485, MAX487-MAX491, and MAX1487 are low-power transceivers for RS-485 and RS-422 communication. Each part contains one driver and one receiver. The MAX483, MAX487, MAX488, and MAX489 feature reduced slew-rate drivers that minimize EMI and reduce reflections caused by improperly terminated cables, thus allowing error-free data transmission up to 250kbps. The driver slew rates of the MAX481, MAX485, MAX490, MAX491, and MAX1487 are not limited, allowing them to transmit up to 2.5Mbps.

These transceivers draw between 120 μ A and 500 μ A of supply current when unloaded or fully loaded with disabled drivers. Additionally, the MAX481, MAX483, and MAX487 have a low-current shutdown mode in which they consume only 0.1 μ A. All parts operate from a single 5V supply.

Drivers are short-circuit current limited and are protected against excessive power dissipation by thermal shutdown circuitry that places the driver outputs into a high-impedance state. The receiver input has a fail-safe feature that guarantees a logic-high output if the input is open circuit.

The MAX487 and MAX1487 feature quarter-unit-load receiver input impedance, allowing up to 128 MAX487/MAX1487 transceivers on the bus. Full-duplex communications are obtained using the MAX488-MAX491, while the MAX481, MAX483, MAX485, MAX487, and MAX1487 are designed for half-duplex applications.

Applications

Low-Power RS-485 Transceivers
Low-Power RS-422 Transceivers
Level Translators
Transceivers for EMI-Sensitive Applications
Industrial-Control Local Area Networks

Next Generation Device Features

- ◆ For Fault-Tolerant Applications
 - MAX3430: \pm 80V Fault-Protected, Fail-Safe, 1/4 Unit Load, +3.3V, RS-485 Transceiver
 - MAX3440E-MAX3444E: \pm 15kV ESD-Protected, \pm 60V Fault-Protected, 10Mbps, Fail-Safe, RS-485/J1708 Transceivers
- ◆ For Space-Constrained Applications
 - MAX3460-MAX3464: +5V, Fail-Safe, 20Mbps, Profibus RS-485/RS-422 Transceivers
 - MAX3362: +3.3V, High-Speed, RS-485/RS-422 Transceiver in a SOT23 Package
 - MAX3280E-MAX3284E: \pm 15kV ESD-Protected, 52Mbps, +3V to +5.5V, SOT23, RS-485/RS-422, True Fail-Safe Receivers
 - MAX3293/MAX3294/MAX3295: 20Mbps, +3.3V, SOT23, RS-485/RS-422 Transmitters
- ◆ For Multiple Transceiver Applications
 - MAX3030E-MAX3033E: \pm 15kV ESD-Protected, +3.3V, Quad RS-422 Transmitters
- ◆ For Fail-Safe Applications
 - MAX3080-MAX3089: Fail-Safe, High-Speed (10Mbps), Slew-Rate-Limited RS-485/RS-422 Transceivers
- ◆ For Low-Voltage Applications
 - MAX3483E/MAX3485E/MAX3486E/MAX3488E/MAX3490E/MAX3491E: +3.3V Powered, \pm 15kV ESD-Protected, 12Mbps, Slew-Rate-Limited, True RS-485/RS-422 Transceivers

Ordering Information appears at end of data sheet.

Selection Table

PART NUMBER	HALF/FULL DUPLEX	DATA RATE (Mbps)	SLEW-RATE LIMITED	LOW-POWER SHUTDOWN	RECEIVER/DRIVER ENABLE	QUIESCENT CURRENT (μ A)	NUMBER OF RECEIVERS ON BUS	PIN COUNT
MAX481	Half	2.5	No	Yes	Yes	300	32	8
MAX483	Half	0.25	Yes	Yes	Yes	120	32	8
MAX485	Half	2.5	No	No	Yes	300	32	8
MAX487	Half	0.25	Yes	Yes	Yes	120	128	8
MAX488	Full	0.25	Yes	No	No	120	32	8
MAX489	Full	0.25	Yes	No	Yes	120	32	14
MAX490	Full	2.5	No	No	No	300	32	8
MAX491	Full	2.5	No	No	Yes	300	32	14
MAX1487	Half	2.5	No	No	Yes	230	128	8

For pricing, delivery, and ordering information, please contact Maxim Direct at 1-888-629-4642, or visit Maxim Integrated's website at www.maximintegrated.com.

19-0122; Rev 10; 9/14

MAX481/MAX483/MAX485/ MAX487-MAX491/MAX1487

Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers

ABSOLUTE MAXIMUM RATINGS

Supply Voltage (V _{CC})	12V	14-Pin SO (derate 8.33mW/°C above +70°C)	667mW
Control Input Voltage (RE, DE)	-0.5V to (V _{CC} + 0.5V)	8-Pin μMAX (derate 4.1mW/°C above +70°C)	830mW
Driver Input Voltage (DI)	-0.5V to (V _{CC} + 0.5V)	8-Pin CERDIP (derate 8.00mW/°C above +70°C)	640mW
Driver Output Voltage (A, B)	-8V to +12.5V	14-Pin CERDIP (derate 9.09mW/°C above +70°C)	727mW
Receiver Input Voltage (A, B)	-8V to +12.5V	Operating Temperature Ranges	
Receiver Output Voltage (RO)	-0.5V to (V _{CC} + 0.5V)	MAX4_ _C_ _/MAX1487C_ _A	0°C to +70°C
Continuous Power Dissipation (T _A = +70°C)		MAX4_ _E_ _/MAX1487E_ _A	-40°C to +85°C
8-Pin Plastic DIP (derate 9.09mW/°C above +70°C)	727mW	MAX4_ _M_ _/MAX1487MJA	-55°C to +125°C
14-Pin Plastic DIP (derate 10.00mW/°C above +70°C)	800mW	Storage Temperature Range	-65°C to +160°C
8-Pin SO (derate 5.88mW/°C above +70°C)	471mW	Lead Temperature (soldering, 10sec)	+300°C

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC ELECTRICAL CHARACTERISTICS

(V_{CC} = 5V ±5%, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Differential Driver Output (no load)	V _{OD1}				5	V
Differential Driver Output (with load)	V _{OD2}	R = 50Ω (RS-422)	2			V
		R = 27Ω (RS-485), Figure 4	1.5		5	V
Change in Magnitude of Driver Differential Output Voltage for Complementary Output States	ΔV _{OD}	R = 27Ω or 50Ω, Figure 4			0.2	V
Driver Common-Mode Output Voltage	V _{OC}	R = 27Ω or 50Ω, Figure 4			3	V
Change in Magnitude of Driver Common-Mode Output Voltage for Complementary Output States	ΔV _{OC}	R = 27Ω or 50Ω, Figure 4			0.2	V
Input High Voltage	V _{IH}	DE, DI, RE	2.0			V
Input Low Voltage	V _{IL}	DE, DI, RE			0.8	V
Input Current	I _{IN1}	DE, DI, RE			±2	μA
Input Current (A, B)	I _{IN2}	DE = 0V; V _{CC} = 0V or 5.25V, all devices except MAX487/MAX1487	V _{IN} = 12V		1.0	mA
			V _{IN} = -7V		-0.8	
		MAX487/MAX1487, DE = 0V, V _{CC} = 0V or 5.25V	V _{IN} = 12V		0.25	mA
			V _{IN} = -7V		-0.2	
Receiver Differential Threshold Voltage	V _{TH}	-7V ≤ V _{CM} ≤ 12V	-0.2		0.2	V
Receiver Input Hysteresis	ΔV _{TH}	V _{CM} = 0V		70		mV
Receiver Output High Voltage	V _{OH}	I _O = -4mA, V _{ID} = 200mV	3.5			V
Receiver Output Low Voltage	V _{OL}	I _O = 4mA, V _{ID} = -200mV			0.4	V
Three-State (high impedance) Output Current at Receiver	I _{OZR}	0.4V ≤ V _O ≤ 2.4V			±1	μA
Receiver Input Resistance	R _{IN}	-7V ≤ V _{CM} ≤ 12V, all devices except MAX487/MAX1487	12			kΩ
		-7V ≤ V _{CM} ≤ 12V, MAX487/MAX1487	48			kΩ

Anexo 11. Datasheet SEN0237-A

Gravity: Analog Dissolved Oxygen Sensor SKU:SEN0237

[\(http://www.dfrobot.com/\)](http://www.dfrobot.com/) [Home](https://www.dfrobot.com/) [Sensors & Modules](https://www.dfrobot.com/category-156.html) [Sensors](https://www.dfrobot.com/category-156.html) [Liquid Sensors](https://www.dfrobot.com/category-36.html) [Liquid Sensors](https://www.dfrobot.com/category-68.html)

Contents

- 1 Introduction
- 2 Specification
- 3 Board Overview
- 4 Tutorial
 - 4.1 Requirements
 - 4.2 Prepare the Probe
 - 4.3 Connection Diagram
 - 4.4 Probe Calibration
 - 4.5 Sample Code
- 5 FAQ
- 6 More Documents



<https://www.dfrobot.com/product-1628.html>
Gravity: Analog Dissolved Oxygen Sensor
SKU: SEN0237
<https://www.dfrobot.com/product-1628.html>

Introduction

There's an old saying regarding to keeping fish, "Good fish deserves good water". Good water quality is very important to the aquatic organisms. Dissolved oxygen is one of the important parameters to reflect the water quality. Low dissolved oxygen in water will lead to difficulty in breathing for aquatic organisms, which may threaten the lives of aquatic organisms their lives. We launched a new opensource dissolved oxygen sensor kit, which is compatible with Arduino. This product is used to measure the dissolved oxygen in water, to reflect the water quality. It is widely applied in many water quality applications, such as aquaculture, environment monitoring, natural science and so on. This sensor kit helps you quickly to build your own dissolved oxygen detector. The probe is a galvanic probe, no need polarization time, and available at any time. The filling solution and membrane cap is replaceable, leading to the low maintenance cost. The signal converter board is plug and play, and has the good compatibility. It can be easily integrated to any control or detecting system. This product is easy to use with high compatibility. With open-source code and detailed tutorial provided, this product is very suitable for your water projects in protecting detecting the dissolved oxygen concentration for the aquatic organisms.

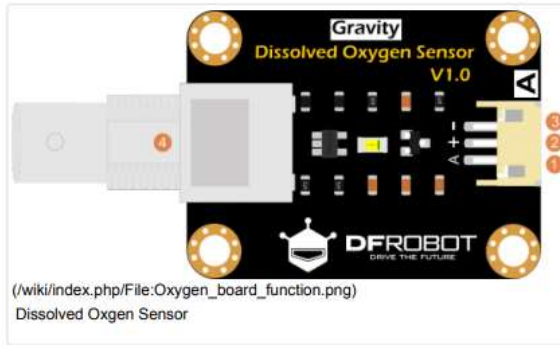


(wiki/index.php/File:Warning__yellow.png)

1. The filling solution is 0.5 mol/L NaOH solution. You need to pour it in the membrane cap before use. Please be caution with the operation because the solution is corrosive. Please wear gloves! If the solution accidentally drops onto the skin, wash your skin with plenty of water immediately.
2. The oxygen permeable membrane in the membrane cap is sensitive and fragile. Be caution when handling with it. Fingernail and other sharp objects should be avoided.
3. The DO sensor will consume a little oxygen during the measurement. Please gently stir the solution and let the oxygen to be distributed evenly in the water.

Specification

- **Dissolved Oxygen Probe**
 - Type: Galvanic Probe
 - Detection Range: 0~20 mg/L
 - Temperature Range: 0~40 °C
 - Response Time: Up to 98% full response, within 90 seconds (25°C)
 - Pressure Range: 0~50 PSI
 - Electrode Service Life: 1 year (normal use)
 - Maintenance Period:
 - Membrane Cap Replacement Period:
 - 1~2 months (in muddy water);
 - 4~5 months (in clean water)
 - Filling Solution Replacement Period: Once every month
 - Cable Length: 2 meters
 - Probe Connector: BNC
- **Signal Converter Board**
 - Supply Voltage: 3.3~5.5V
 - Output Signal: 0~3.0V
 - Cable Connector: BNC
 - Signal Connector: Gravity Analog Interface (PH2.0-3P)
 - Dimension: 42mm * 32mm/1.65 * 1.26 inches



No.	Label	Description
1	A	Analog Signal Output (0~3.0V)
2	+	VCC (3.3~5.5V)
3	-	GND
4	BNC	Probe Cable Connector

Tutorial

This tutorial will show how to use this dissolved oxygen sensor kit. The dissolved oxygen probe is a precision electrochemical sensor. Please pay attention to the usage details.



(/wiki/index.php/File:Warning_yellow.png)

- Before using the dissolved oxygen probe, 0.5 mol/L NaOH solution should be added into the membrane cap as the filling solution of the probe. As NaOH solution has strong corrosivity, protective gloves should be put on before handling the solution. If the solution accidentally drops onto the skin, wash your skin with plenty of water immediately.
- The oxygen permeable membrane in the membrane cap is sensitive and vulnerable. Be caution when handling with it. Fingernail and other sharp objects should be avoided.
- During the measuring process, the oxygen probe will consume a little oxygen. You need to gently stir the water and let the oxygen to be distributed evenly in water. On the other hand, do not stirring violently to prevent the oxygen in the air from quickly entering into the water.

Requirements

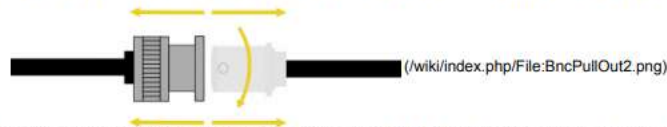
- **Hardware**
 - Dissolved Oxygen Probe (With Membrane Cap) x 1
 - 0.5mol/L NaOH Solution x 1
 - DFRduino UNO R3 (<https://www.dfrobot.com/product-838.html>) (or similar) x 1
 - Dissolved Oxygen Signal Converter Board x 1
 - Analog Cable (3Pin) x 1
- **Software**
 - Arduino IDE (Version requirements: V1.0.x, V1.6.x or V1.8.x). [Click to Download Arduino IDE from Arduino®](https://www.arduino.cc/en/Main/Software%7C) (<https://www.arduino.cc/en/Main/Software%7C>).

Prepare the Probe

For a new dissolved oxygen probe, 0.5 mol/L NaOH solution should be added into the membrane cap firstly as the filling solution. If the probe has been used for some time and the error grows greatly, it is time to change the filling solution. The following tutorial details how to fill the probe with the NaOH solution.

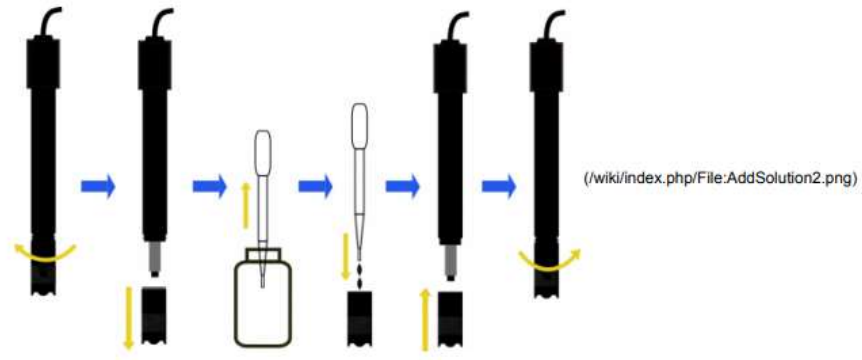
1. Pick up the probe and take off the BNC jumper

- The BNC jumper is protecting the probe by shorting the output signal so as to keep the two poles of the probe at the same voltage level when probe is not filled with NaOH solution. When the probe is filled with NaOH solution, it outputs a voltage signal, and as a result, the probe should not be shorted. In sum, before filling the probe with the NaOH solution, the BNC jumper should be taken off, or it may shorten the lifespan of the probe.



2. Unscrew the membrane cap from the probe and fill about 2/3 volume of the cap with 0.5 mol/L NaOH solution. Make sure the probe is in vertical position with respect to the horizontal plane. Carefully screw the cap back to the probe. It would be nice if a little bit solution overflows out of the cap to ensure the probe is fully filled with NaOH solution.

- When screwing the cap back to the probe, the probe should be in vertical position with respect to the horizontal plane to avoid creating bubbles in the filling solution.
- If the cap is fully filled with NaOH solution, there will be too much solution overflowing when screwing the cap back to the probe. If the filling solution is too little, bubbles may be created inside the cap. In sum, the best way is to fill about 2/3 volume of the cap, a little bit overflow when screwing the cap back to the probe.

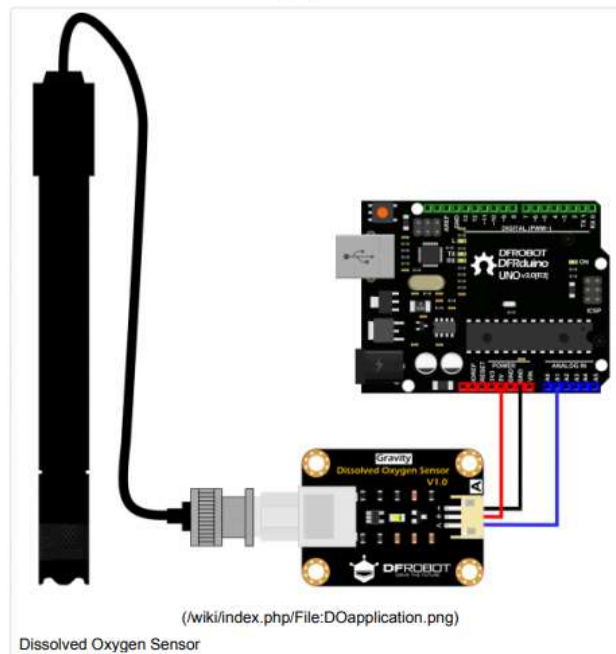


Connection Diagram

When the probe is filled with NaOH solution, it needs calibration for once. Before calibration, please connect the probe as the following diagram. Connect the probe to BNC connector of the signal converter board. Connect the board to the analog input of Arduino main-board.

Connection Diagram

When the probe is filled with NaOH solution, it needs calibration for once. Before calibration, please connect the probe as the following diagram. Connect the probe to BNC connector of the signal converter board. Connect the board to the analog input of Arduino main-board.



Dissolved Oxygen Sensor

Anexo 12. Datasheet Micro SD

Micro SD Card Micro SDHC Mini TF Card Adapter Reader Module for Arduino

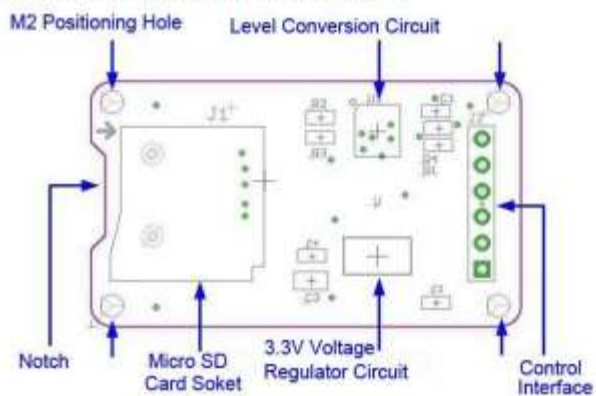


Description

- The module (MicroSD Card Adapter) is a Micro SD card reader module for reading and writing through the file system and the SPI interface driver, SCM system can be completed within a file MicroSD card
- Support Micro SD Card, Micro SDHC card (high speed card)
- Level conversion circuit board that can interface level is 5V or 3.3V
- Power supply is 4.5V ~ 5.5V, 3.3V voltage regulator circuit board
- Communications interface is a standard SPI interface
- 4 M2 screws positioning holes for easy installation
- Control Interface: A total of six pins (GND, VCC, MISO, MOSI, SCK, CS), GND to ground, VCC is the power supply, MISO, MOSI, SCK for SPI bus, CS is the chip select signal pin;
- 3.3V regulator circuit: LDO regulator output 3.3V for level conversion chip, Micro SD card supply;
- Level conversion circuit: Micro SD card to signal the direction of converts 3.3V, MicroSD card interface to control the direction of the MISO signal is also converted to 3.3V, general AVR microcontroller systems can read the signal;
- Micro SD card connector: self bomb deck, easy card insertion.
- Positioning holes: 4 M2 screws positioning holes with a diameter of 2.2mm, so the module is easy to install positioning, to achieve inter-module combination.

Interface Parameters:

Items	Min	Typical	Max	Unit
Power Voltage VCC	4.5	5	5.5	V
Current	0.2	80	200	<u>mA</u>
Interface Electrical Potential	3.3 or 5			V
Support Card Type	Micro SD Card(<=2G), <u>Mirco</u> SDHC Card(<=32G)			—
Size	42X24X12			mm
Weight	5			g

Mirco SD Card Interface Module:

Anexo 13. Datasheet RTC DS3231

DS3231

Extremely Accurate I²C-Integrated RTC/TCXO/Crystal

General Description

The DS3231 is a low-cost, extremely accurate I²C real-time clock (RTC) with an integrated temperature-compensated crystal oscillator (TCXO) and crystal. The device incorporates a battery input, and maintains accurate timekeeping when main power to the device is interrupted. The integration of the crystal resonator enhances the long-term accuracy of the device as well as reduces the piece-part count in a manufacturing line. The DS3231 is available in commercial and industrial temperature ranges, and is offered in a 16-pin, 300-mil SO package.

The RTC maintains seconds, minutes, hours, day, date, month, and year information. The date at the end of the month is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with an AM/PM indicator. Two programmable time-of-day alarms and a programmable square-wave output are provided. Address and data are transferred serially through an I²C bidirectional bus.

A precision temperature-compensated voltage reference and comparator circuit monitors the status of V_{CC} to detect power failures, to provide a reset output, and to automatically switch to the backup supply when necessary. Additionally, the RST pin is monitored as a pushbutton input for generating a μ P reset.

Benefits and Features

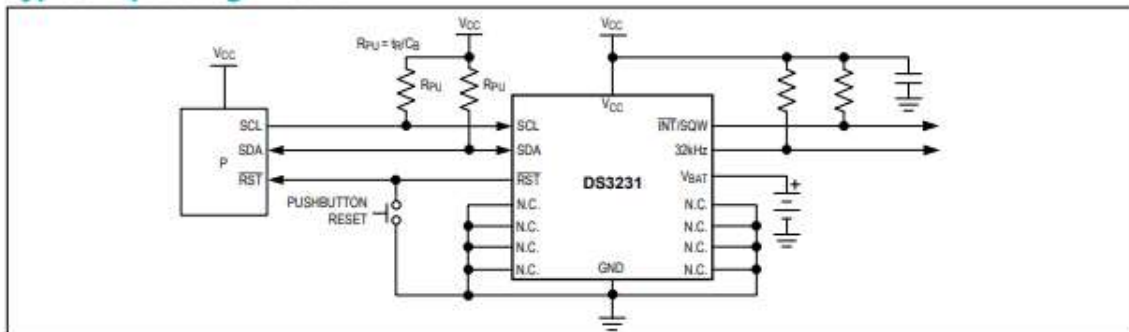
- Highly Accurate RTC Completely Manages All Timekeeping Functions
 - Real-Time Clock Counts Seconds, Minutes, Hours, Date of the Month, Month, Day of the Week, and Year, with Leap-Year Compensation Valid Up to 2100
 - Accuracy ± 2 ppm from 0°C to +40°C
 - Accuracy ± 3.5 ppm from -40°C to +85°C
 - Digital Temp Sensor Output: ± 3 °C Accuracy
 - Register for Aging Trim
 - RST Output/Pushbutton Reset Debounce Input
 - Two Time-of-Day Alarms
 - Programmable Square-Wave Output Signal
- Simple Serial Interface Connects to Most Microcontrollers
 - Fast (400kHz) I²C Interface
- Battery-Backup Input for Continuous Timekeeping
 - Low Power Operation Extends Battery-Backup Run Time
 - 3.3V Operation
- Operating Temperature Ranges: Commercial (0°C to +70°C) and Industrial (-40°C to +85°C)
- Underwriters Laboratories® (UL) Recognized

Applications

- Servers
- Telematics
- Utility Power Meters
- GPS

Ordering Information and Pin Configuration appear at end of data sheet.

Typical Operating Circuit



Underwriters Laboratories is a registered certification mark of Underwriters Laboratories Inc.

DS3231

Extremely Accurate I²C-Integrated
RTC/TCXO/Crystal

Absolute Maximum Ratings

Voltage Range on Any Pin Relative to Ground -0.3V to +6.0V
 Junction-to-Ambient Thermal Resistance (θ_{JA}) (Note 1) 73°C/W
 Junction-to-Case Thermal Resistance (θ_{JC}) (Note 1) 23°C/W
 Operating Temperature Range
 DS3231S 0°C to +70°C
 DS3231SN -40°C to +85°C

Junction Temperature +125°C
 Storage Temperature Range -40°C to +85°C
 Lead Temperature (soldering, 10s) +260°C
 Soldering Temperature (reflow, 2 times max) +260°C
 (see the *Handling, PCB Layout, and Assembly* section)

Note 1: Package thermal resistances were obtained using the method described in JEDEC specification JESD51-7, using a four-layer board. For detailed information on package thermal considerations, refer to www.maximintegrated.com/thermal-tutorial.

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Recommended Operating Conditions

($T_A = T_{MIN}$ to T_{MAX} , unless otherwise noted.) (Notes 2, 3)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Supply Voltage	V_{CC}		2.3	3.3	5.5	V
	V_{BAT}		2.3	3.0	5.5	V
Logic 1 Input SDA, SCL	V_{IH}		0.7 x V_{CC}		$V_{CC} + 0.3$	V
Logic 0 Input SDA, SCL	V_{IL}		-0.3		0.3 x V_{CC}	V

Electrical Characteristics

($V_{CC} = 2.3V$ to $5.5V$, V_{CC} = Active Supply (see Table 1), $T_A = T_{MIN}$ to T_{MAX} , unless otherwise noted.) (Typical values are at $V_{CC} = 3.3V$, $V_{BAT} = 3.0V$, and $T_A = +25^\circ C$, unless otherwise noted.) (Notes 2, 3)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Active Supply Current	I_{CCA}	(Notes 4, 5)	$V_{CC} = 3.63V$		200	μA
			$V_{CC} = 5.5V$		300	
Standby Supply Current	I_{CCS}	I ² C bus inactive, 32kHz output on, SQW output off (Note 5)	$V_{CC} = 3.63V$		110	μA
			$V_{CC} = 5.5V$		170	
Temperature Conversion Current	$I_{CCSCONV}$	I ² C bus inactive, 32kHz output on, SQW output off	$V_{CC} = 3.63V$		575	μA
			$V_{CC} = 5.5V$		650	
Power-Fail Voltage	V_{PF}		2.45	2.575	2.70	V
Logic 0 Output, 32kHz, INT/SQW, SDA	V_{OL}	$I_{OL} = 3mA$			0.4	V
Logic 0 Output, RST	V_{OL}	$I_{OL} = 1mA$			0.4	V
Output Leakage Current 32kHz, INT/SQW, SDA	I_{LO}	Output high impedance	-1	0	+1	μA
Input Leakage SCL	I_{LI}		-1		+1	μA
RST Pin I/O Leakage	I_{OL}	RST high impedance (Note 6)	-200		+10	μA
V_{BAT} Leakage Current (V_{CC} Active)	I_{BATLKG}			25	100	nA

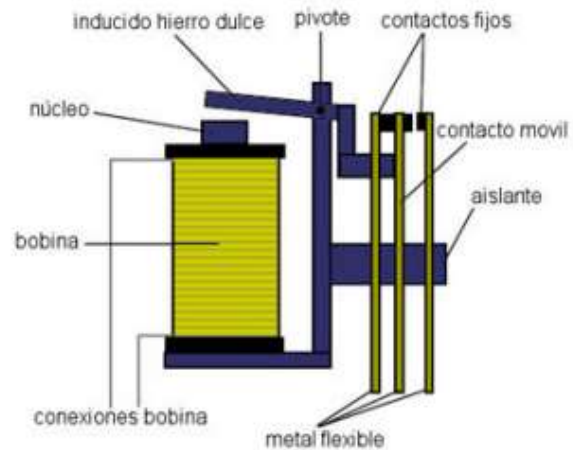
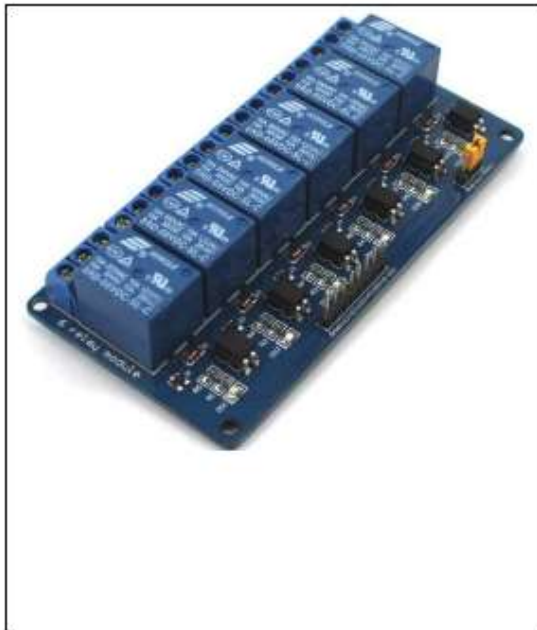
Anexo 14. Datasheet Relé 12V

MÓDULO DE RELÉS

El módulo que vamos a emplear es uno muy típico y a un precio aceptable:

Como podemos ver, en este módulo (que es el que he empleado para el tutorial) trae dos entradas para poder activar dos relés, pero podemos encontrar módulos con más relés e incluso de un solo relé, lo que necesitemos para nuestros proyectos.

(Versión 15-6-19)



Esquema de funcionamiento de un Relay

Descripción del Producto

Plataforma: Arduino 8051 AVR PIC DSP ARM MSP430 TTL. (Son distintas plataformas)

Control Dispositivos varios/cargas

Voltaje de operación 250VAC/30VDC

Voltaje de la bobina (relé) 5V

Corriente de operación 10A. *Algunos fabricantes aclaran:*

Corriente máx: 10A (NO), 5A (NC)

Tiempo de acción: 10 ms / 5 ms

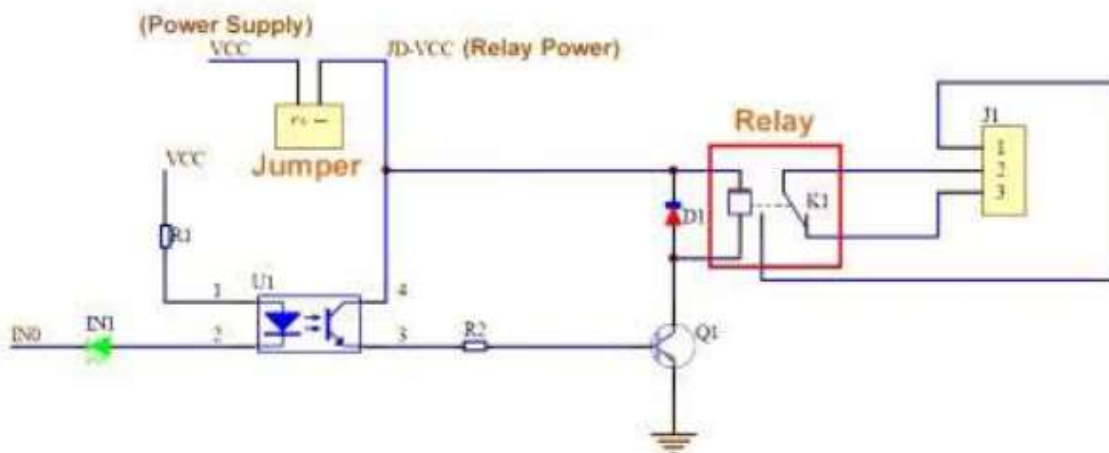
Corriente de activación por relé 15mA~20mA **(ATENCIÓN: VEA ANEXO1)**

Canales 1-2 - 4 - 6 -8 (independientes protegidos con Optoacopladores)

LED indicador Para cada canal (cuando bobina está activa)

Este módulo de relevadores (relés) es para conmutación de cargas de potencia. Los contactos de los relevadores están diseñados para conmutar cargas de hasta 10A y 250VAC (30VDC), aunque se recomienda usar niveles de tensión por debajo de estos límites. Las entradas de control se encuentran aisladas con optoacopladores para minimizar el ruido percibido por el circuito de control mientras se realiza la conmutación de la carga. La señal de control puede provenir de cualquier circuito de control TTL o CMOS como puede ser un microcontrolador. Este módulo es ideal para conmutar cargas de corriente alterna conectadas a la red eléctrica. Soporta todos los microcontroladores, aplicaciones en zonas industriales, control del PLC, entre otros. Este módulo es capaz de controlar varios equipamientos de alta corriente durante un tiempo prolongado. Puede ser controlado por muchos microcontroladores como Arduino, 8051, AVR, PIC, DSP, ARM, MSP430, TTL.

Esquema de conexión para que funcione todo correctamente:



Como podemos ver en la foto del módulo, observamos que tenemos dos tiras de pines, una que es en la que esta el jumper, que tiene tres pines y podemos seleccionar alimentar todo el módulo de forma conjunta (es decir seleccionar con el jumper jd-vcc y vcc), o bien alimentar de forma independiente tanto optoacopladores como relés (quitando el jumper), alimentar todo de manera conjunta parece la forma más sencilla de hacerlo, así que he dejado el jumper uniendo jd-vcc y vcc.

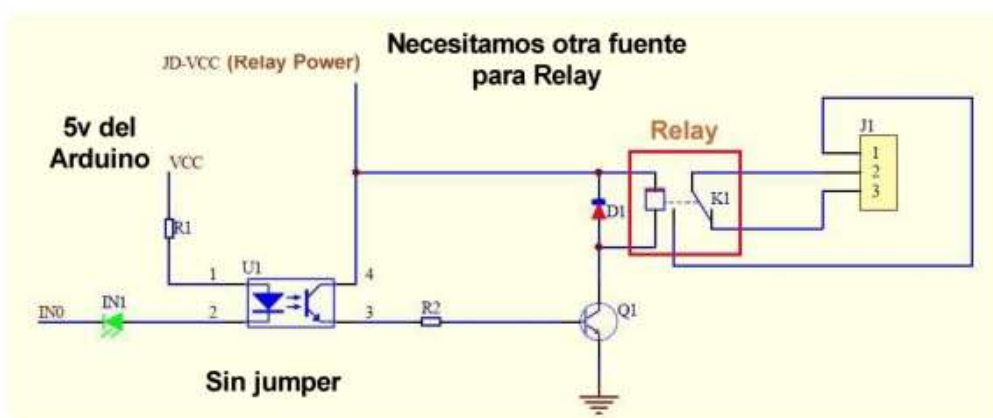
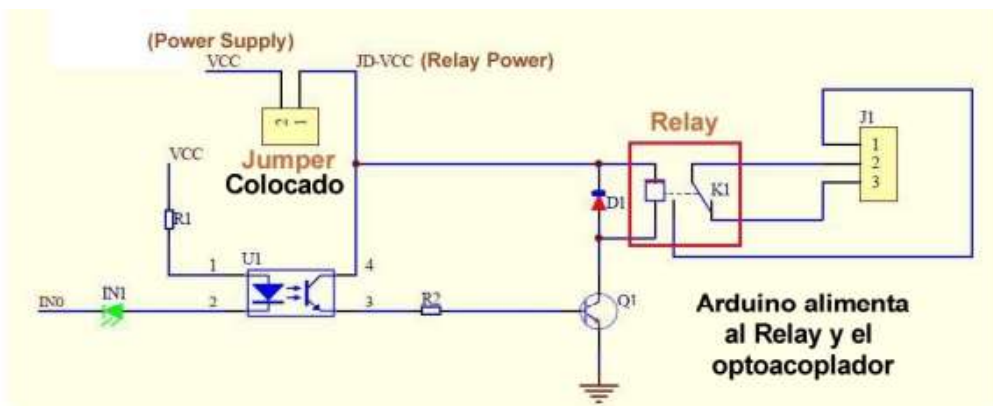
Y la otra tira de pines que es donde nos aparece en el siguiente orden:

GND IN1 IN2 VCC

Aquí es donde unimos la alimentación con nuestro arduino, es decir:

GND del módulo con GND del Arduino

VCC del módulo con 5V del Arduino



Entradas **IN1 e IN2** (se activan las bobinas de los relés cuando colocamos a **GND** dichas entradas) conectados a los pines que deseemos de nuestro Arduino, y serán los encargados de activar las bobinas de los relés correspondientes.

Anexo 15. Datasheet LCD 128X64

Sitronix

ST7920

Chinese Fonts built in LCD controller/driver

Main Features

- Operation Voltage Range:
 - 2.7V to 5.5V
- Support 8-bit, 4-bit and serial bus MPU interface
- 64 x 16-bit display RAM (DDRAM)
 - Supports 16 words x 4 lines (Max)
 - LCD display range 16 words x 2 lines
- 64 x 256-bit Graphic Display RAM (GDRAM)
- 2M-bits Character Generation ROM (CGROM):
Support 8192 Chinese words (16x16 dot matrix)
- 16K-bit half-width Character Generation ROM (HCGROM):
Supports 126 characters (16x8 dot matrix)
- 32-common x 64-segment (2 lines of character) LCD drivers
- Automatic power on reset (POR)
- External reset pin (XRESET)
- With the extension segment drivers, the display area can up to 16x2 lines
- Built-in RC oscillator:
Frequency is adjusted by an external resistor
- Low power consumption design
 - Normal mode (450uA Typ VDD=5V)
 - Standby mode (30uA Max VDD=5V)
- VLCD (V_O to V_{SS}): max 7V
- Graphic and character mixed display mode
- Multiple instructions:
 - Display Clear
 - Return Home
 - Display ON/OFF
 - Cursor ON/OFF
 - Display Character Blink
 - Cursor Shift
 - Display Shift
 - Vertical Line Scroll
 - Reverse Display (by line)
 - Standby Mode
- Built-in voltage booster (2 times)
V_{OUT}: max 7V
- 1/33 Duty (with ICON)

Function Description

ST7920 LCD controller/driver IC can display alphabets, numbers, Chinese fonts and self-defined characters. It supports 3 kinds of bus interface, namely 8-bit, 4-bit and serial. All functions, including display RAM, Character Generation ROM, LCD display drivers and control circuits are all in a one-chip solution. With a minimum system configuration, a Chinese character display system can be easily achieved.

ST7920 includes character ROM with 8192 16x16 dots Chinese fonts and 126 16x8 dots half-width alphanumerical fonts. Besides, it supports 64x256 dots graphic display area for graphic display (GDRAM). Mix-mode display with both character and graphic data is possible. ST7920 has built-in CGRAM and provide 4 sets software programmable 16x16 fonts.

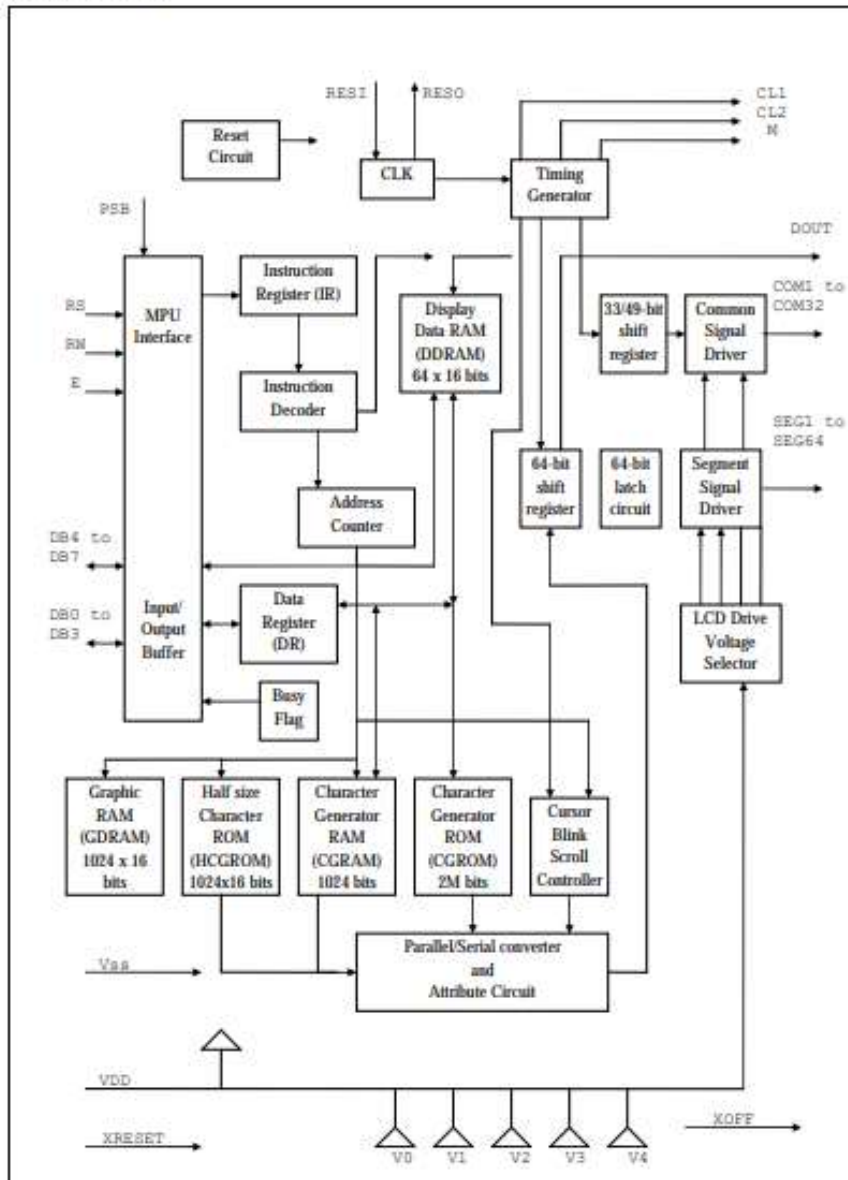
ST7920 has wide operating voltage range (2.7V to 5.5V). It also has low power consumption. So ST7920 is suitable for battery-powered portable device.

ST7920 LCD driver consists of 32-common and 64-segment. Company with the extension segment driver (ST7921) ST7920 can support up to 32-common x 256-segment display.

Part Number	Font Code
ST7920-0A	BIG-5 Code Set (Traditional Chinese)
ST7920-0B	GB Code Set (Simplified Chinese)
ST7920-0C	Chinese (Traditional/Simplified) & Japanese
ST7920-0F	Chinese (Traditional/Simplified), Japanese & Korean

ST7920 Specification Reversion History		
Version	Date	Description
C1.7	2000/12/15	<ol style="list-style-type: none"> VCC changed to VDD. VLCD changed from VCC-V5 to V0-VSS. DC characteristics input High voltage (Vih) changed to 0.7VDD. DC characteristics output High voltage (Voh) changed to 0.8VDD.
C1.8	2001/03/01	<ol style="list-style-type: none"> Chip Size changed. ICON 256 dots changed to 240 dots. XOFF normal high sleep Low changed to normal low sleep High. Added XOFF application. Modified application of ST7920: PIN 4-6 are floating. (PIN 4-6 are test pin) Modified voltage doubler CAP1P, CAP1M, CAP2M capacitors polarity
C1.9	2001/05/28	<ol style="list-style-type: none"> Icon RAM TABLE changed. (TABLE-6) Booster description modified. (PAGE-29) AC Characteristics modified. Added 2Line 16 Chinese Word (32Com X 256Seg) application circuit. Added oscillation resistor's relation to power consumption and frequency.
C2.0	2001/07/03	<ol style="list-style-type: none"> Added Register initial values. Voltage booster CAP1M CAP1P polarity changed (PAGE-30).
V2.0	2001/08/17	<ol style="list-style-type: none"> Modified Table 7 (PAGE-14). Change to English version.
V2.0c	2001/10/18	<ol style="list-style-type: none"> Modified page-38 Serial interface timing diagram.
V2.0d	2002/05/09	<ol style="list-style-type: none"> Add the standard code (Japan, GB code, BIG-5 code).
V3.0	2002/10/11	<ol style="list-style-type: none"> Delete sleep mode function.
V3.1	2003/04/11	<ol style="list-style-type: none"> Modified GDRAM Address (AC5...AC0, 00h...3Fh).
V3.2	2003/09/09	<ol style="list-style-type: none"> Add the CGROM and HCGROM test application circuit.
V3.3	2004/03/29	<ol style="list-style-type: none"> Updat the using method for ICON.
V3.4	2005/5/24	<ol style="list-style-type: none"> ICON no used.
V3.5	2005/5/24	<ol style="list-style-type: none"> Add VOUT voltage limitation. Remove IRAM related descriptions.
V3.6	2005/6/6	<ol style="list-style-type: none"> Fix the check sum count number on Page 28-30. 655360->655362, 10240->10242. Modify the description about serial interface.
V3.7	2007/7/24	<ol style="list-style-type: none"> Add CGROM/HCGROM checksum operation time.
V3.8	2007/12/20	<ol style="list-style-type: none"> Add "Clear DDRAM" step before check sum process.
V3.9	2008/3/3	<ol style="list-style-type: none"> Modify 4-bit initial sequence.
V4.0	2008/8/18	<ol style="list-style-type: none"> Add Font Code "0F" at Page 1. Modify the description of Font Code Table at Page 1.

System Block Diagram



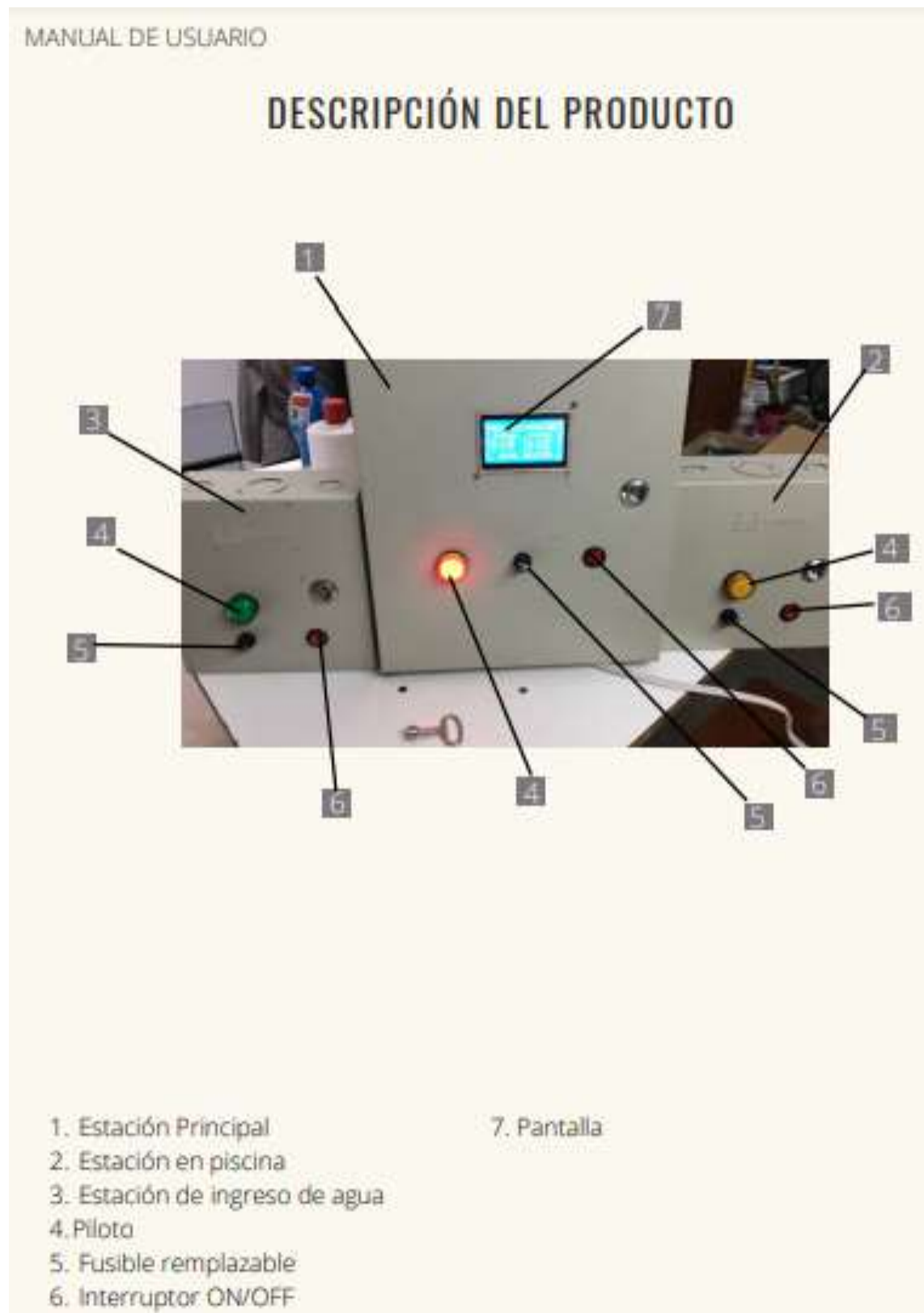
Anexo 16. Instalaciones del criadero de truchas “Aguas Verdes”



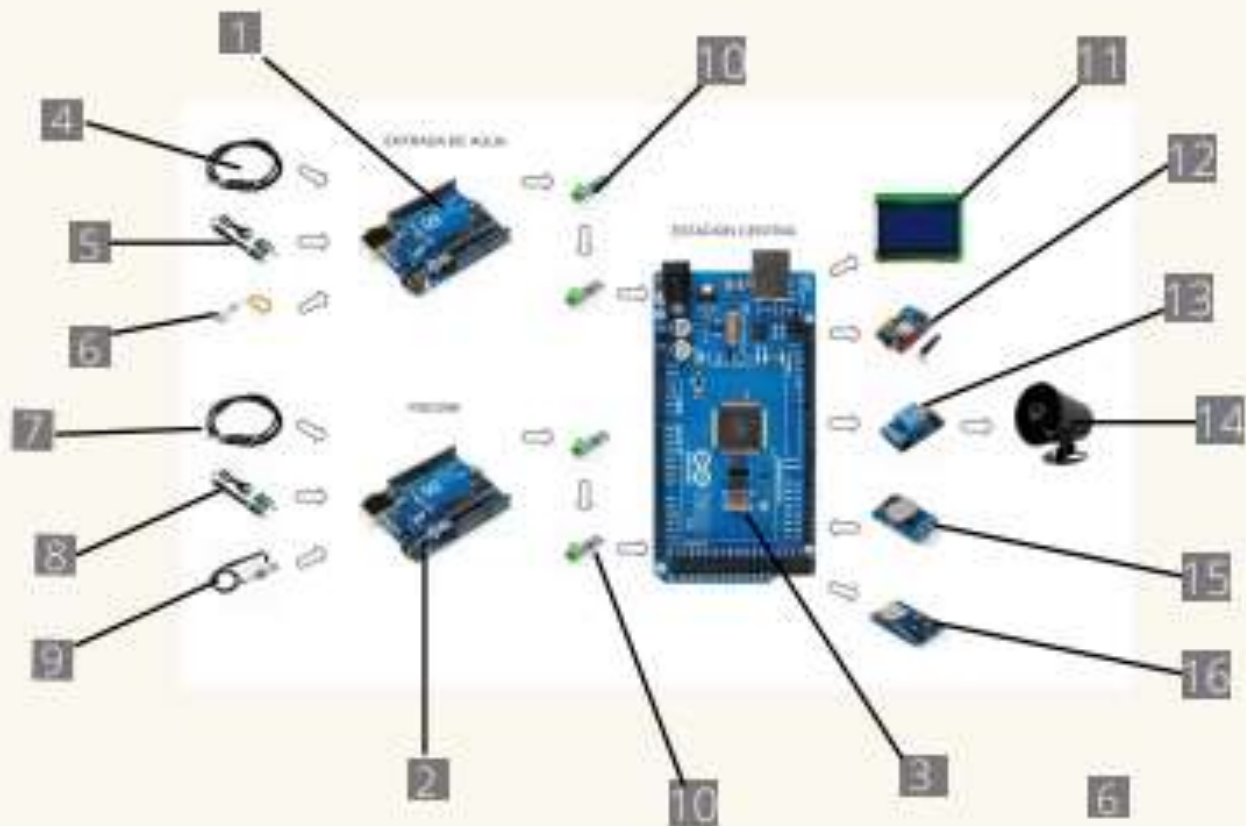
Nota. Manguera del cableado de alimentación de las estaciones.



Nota. Mangueras separadas del cableado de alimentación (1) y datos (2) de las estaciones.

Anexo 17. Manual de uso del sistema de monitoreo de calidad del agua

DESCRIPCION DE COMPONENTES



1. Arduino UNO (Entrada de agua)
2. Arduino UNO (Piscina)
3. Arduino MEGA
4. Sensor de temperatura (entrada)
5. Sensor de Ph (entrada)
6. Sensor de nivel de agua
7. Sensor de temperatura (piscina)
8. Sensor de Ph (piscina)

9. Sensor nivel de oxígeno
10. Módulos RS485
11. Pantalla
12. Módulo GSM
13. Relé
14. Sirena
15. Módulo RTC fecha y hora
16. Módulo lector SD card

USO

Paso 1: Verificación

- Los cables de alimentación y correctamente conectados.
- Pilotos encendidos.
- Tarjeta SD conectada y funcionando.
- Tarjeta SIM con saldo mínimo de 1 dólar.

Paso 2: Encendido

Se enciende en primer lugar las estaciones: Ingreso de agua y Piscina.1
Seguidamente se enciende la Sirena
Finalmente se enciende la estación principal.

Paso 3: Manejo

En la pantalla de la estación principal es posible visualizar en tiempo real los valores que están siendo medidos a través de los sensores.

Para obtener todos los valores que han sido obtenidos es necesario abrir la estación principal y retirar la tarjeta SD, para introducirla en un computador y visualizar los archivos *.csv que contienen todos los datos requeridos.



PRECAUCIÓN

En caso de existir una medida de sensor no permitida como puede ser, bajo nivel de agua o cambio en el nivel de Ph, se activarán las alarmas para que el inconveniente pueda ser atendido lo antes posible

CALIBRACIÓN

Sensor de Ph

·Para la calibración de nivel de Ph, limpie el bulbo de vidrio del electrodo sumergiéndolo en agua destilada o agua desionada, muévelo para apartar los restos de la solución del sustrato, luego saque el electrodo y déjelo secarse. Siguiendo, encienda el medidor en el modo de calibración. En muchos medidores, la pantalla va a parpadear un número, como el 4,0. Esto significa que el bulbo de vidrio necesita ser sumergido en la solución de 4,0. Debe de verter la solución de calibración 4,0 en un vaso y después sumerja el electrodo en ella. Una vez que se calibra en la solución de 4,0, el número 7,0 va a comenzar a parpadear. Enjuague el electrodo con agua destilada, déjelo secar y luego sumérjalo en la solución de calibración de 7,0. Una vez que para de parpadear, está calibrado y listo para utilizarse.

Sensor de nivel de oxígeno

La calibración de este sensor se debe realizar en las condiciones del ambiente a utilizar ya que de acuerdo a la temperatura ambiente este nivel de oxígeno puede variar.

RECOMENDACION

Se recomienda la realización de un mantenimiento preventivo en el cual se realice la calibración de los sensores mensualmente. También se lo puede realizar cada 15 días.

Anexo 18. Entrevista de resultados al fundador del criadero



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
CARRERA DE INGENIERÍA EN ELECTRÓNICA Y REDES DE
COMUNICACIÓN
ENTREVISTA EN EL CRIADERO "AGUAS VERDES"

Fecha: 22/12/2021

Lugar: Tufiño

Entrevistador: Edison Paspuezán

Entrevistado: José Domingo Paspuezán

1.- ¿Ha observado algún cambio en la producción?

Se ha notado un aumento del 20% de pascas listas para vender con un tamaño y peso promedio.

Los pascas crecen más rápido en la piscina que se robó el sistema.

2.- ¿Ha observado algún cambio en las piscinas?

Se puede colocar más pascas en las piscinas.

Ya no hay espacios vacíos.

Los pascas crecen en menos tiempo.

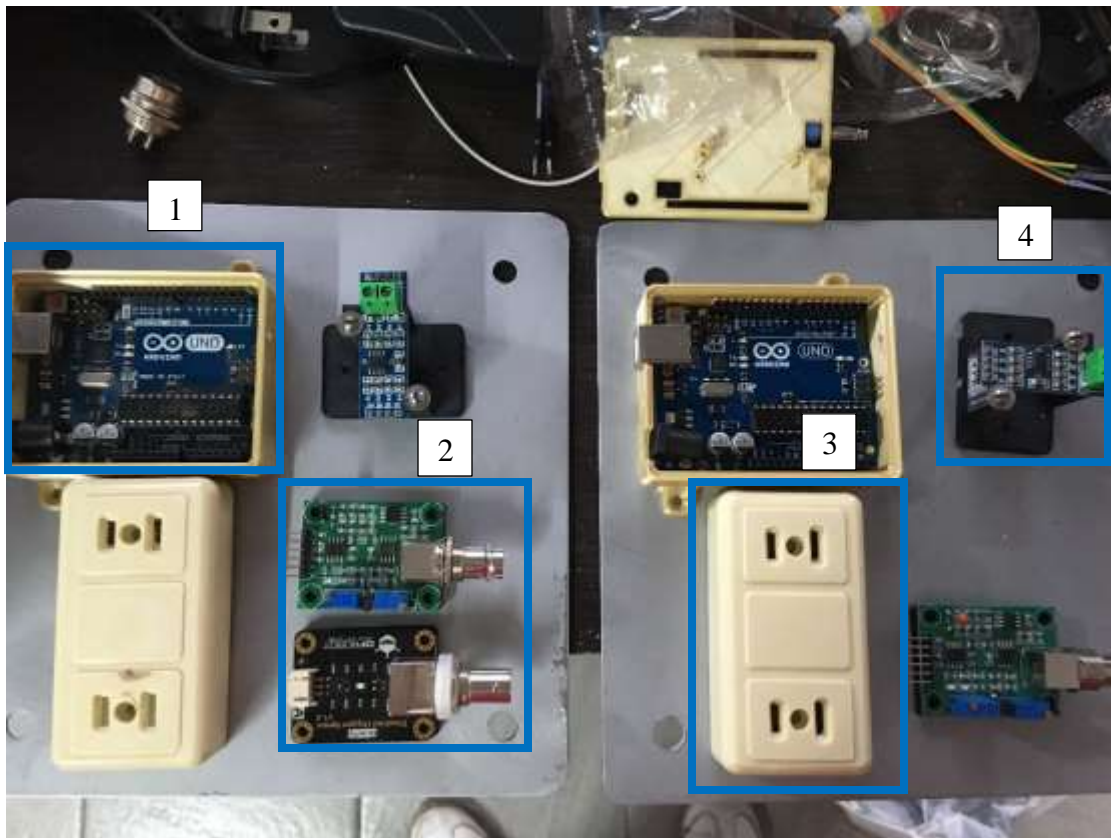
3.- ¿Qué otros cambios se han notado en el criadero con la implementación del sistema de monitoreo de calidad del agua?

La alarma de nivel de agua ayuda a reducir los desperdicios del canal lo antes posible.

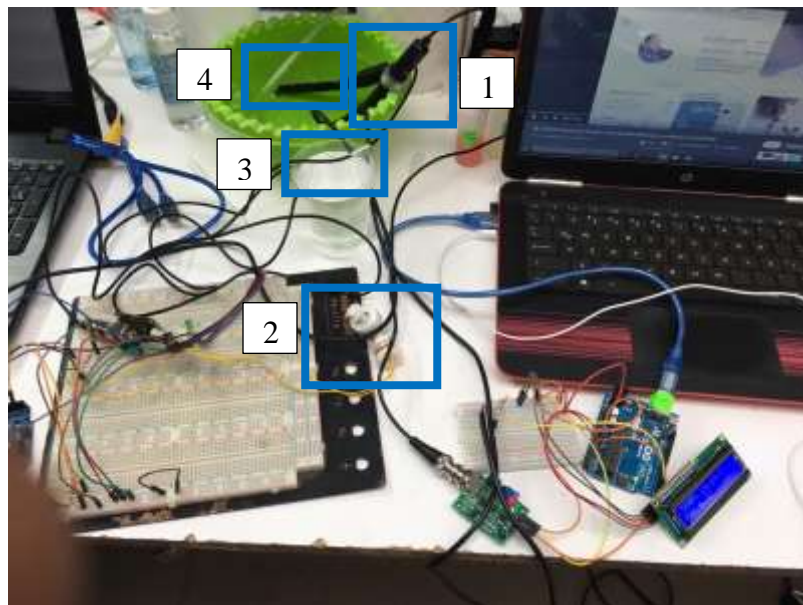
La estructura de puntos de oxigenación mejoran el uso total del espacio.

Se cambió de uso: de piscina para engordar a piscina de crecimiento.

Anexo 19. Construcción de tableros de monitoreo calidad del agua

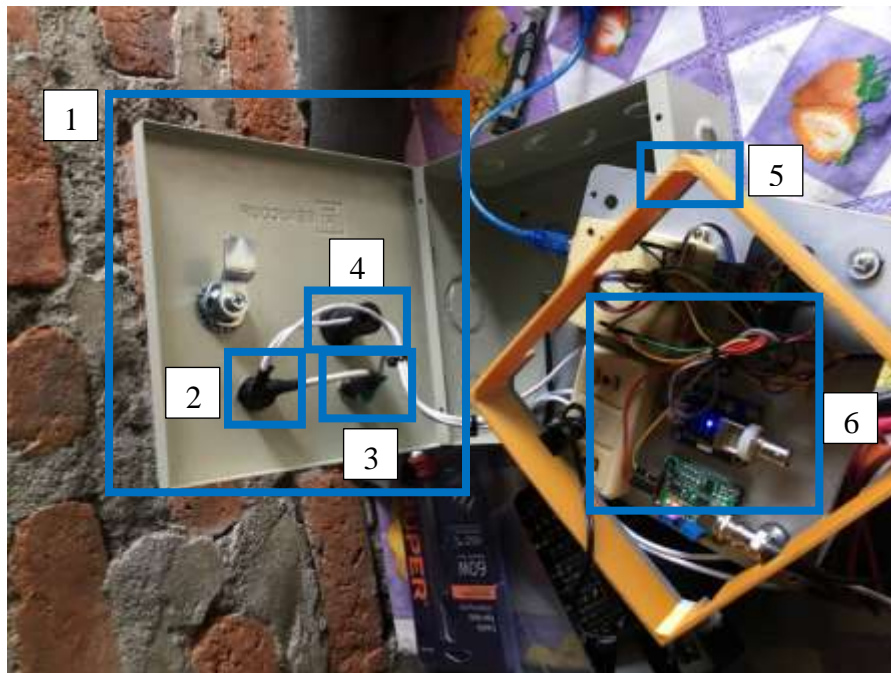


Nota. (1) Tarjeta controladora Arduino UNO, (2) Tarjetas de los sensores, (3) Tomas de corriente, (4) Módulo Modbus.

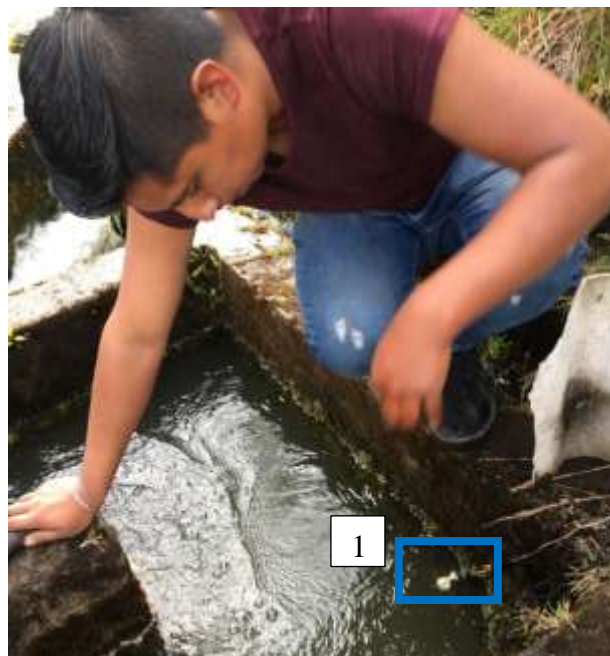


Nota. Prueba y revisión de los sensores de pH (1), nivel (2), temperatura (3) y oxígeno (4).

Anexo 20. Instalación de tableros de monitoreo calidad del agua



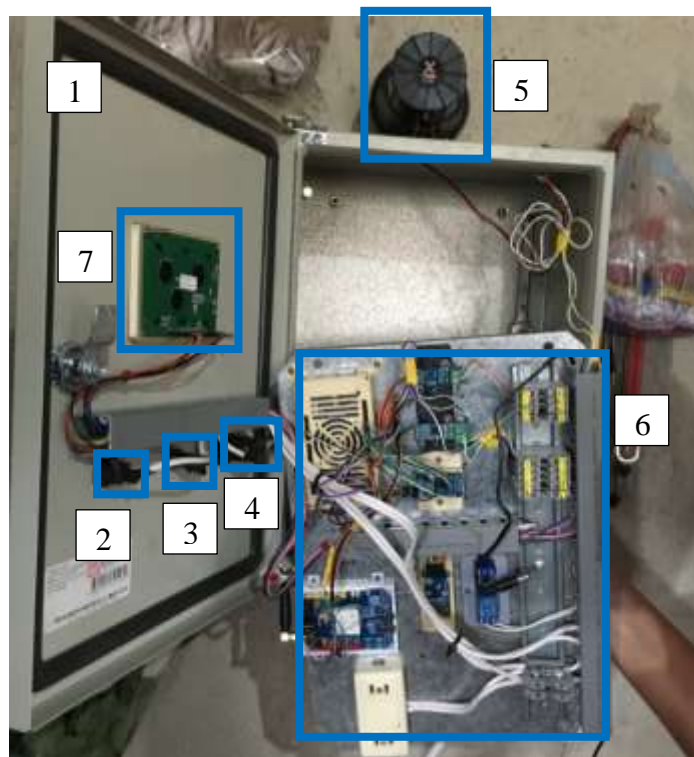
Nota. Caja protectora de la estación (1), Botón de encendido (2), Fusible (3), Indicador piloto (4), Cubierta en PLA (5) y Placa de electrónica de la estación (6).



Nota. Colocación del sensor de nivel de agua (1).



Nota. Estación de monitoreo antes de ser cerrada y encendida.



Nota. Gabinete de la estación (1), Botón de encendido (2), Fusible (3) y Indicador piloto (4), Sirena (5) y Placa de electrónica de la estación (6) y Pantalla LCD (7).

Anexo 21. Funcionamiento de los tableros de monitoreo de calidad del agua



Nota. Pantalla LCD de la estación central funcionando.



Nota. Lugar de instalación de la estación de monitoreo de la piscina de truchas (1) y la estación de monitoreo en la entrada de agua (2).