



FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

PROTOTIPO DE UN SISTEMA DE CONTROL PARA OPTIMIZAR EL
CONSUMO ELÉCTRICO DE UNA BOMBA Y BLOWERS DE UNA PLANTA
DE TRATAMIENTO DE AGUAS RESIDUALES

AUTOR

Byron Steven Garcés Toro

Jonathan Patricio Ramos Sotomayor

AÑO

2020



FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

PROTOTIPO DE UN SISTEMA DE CONTROL PARA OPTIMIZAR EL
CONSUMO ELÉCTRICO DE UNA BOMBA Y BLOWERS DE UNA PLANTA
DE TRATAMIENTO DE AGUAS RESIDUALES.

Trabajo de Titulación presentado en conformidad con los requisitos
establecidos para optar por los títulos de Ingeniero en Redes y
Telecomunicaciones e Ingeniero en Electrónica y Redes de Información.

Profesor Guía

Máster José Julio Freire Cabrera

Autores

Byron Steveen Garcés Toro

Jonathan Patricio Ramos Sotomayor

Año

2020

DECLARACIÓN DEL PROFESOR GUÍA

“Declaro haber dirigido este trabajo, Prototipo de un sistema de control para optimizar el consumo eléctrico de una bomba y blowers de una planta de tratamiento de aguas residuales, a través de reuniones periódicas con los estudiantes Byron Steveen Garcés Toro y Jonathan Patricio Ramos Sotomayor, en el semestre 202020, orientando sus conocimientos y competencias para un eficiente desarrollo del tema escogido y dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación”.



José Julio Freire Cabrera
Máster en Gerencia Empresarial
CI: 1709731457

DECLARACIÓN DEL PROFESOR CORRECTOR

“Declaro haber revisado este trabajo, Prototipo de un sistema de control para optimizar el consumo eléctrico de una bomba y blowers de una planta de tratamiento de aguas residuales, Byron Steveen Garcés Toro y Jonathan Patricio Ramos Sotomayor, en el semestre 202020, dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación”.



Jean-Michel Clairand

Doctor en Ingeniería y Producción Industrial

CI: 1714736681

DECLARACIÓN DE AUTORÍA DEL ESTUDIANTE

“Declaro que este trabajo es original, de nuestra autoría, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes”.



Byron Steven Garcés Toro
C.I. 0603868415



Jonathan Patricio Ramos Sotomayor
C.I. 1717684573

AGRADECIMIENTO

Agradecemos a Dios por ser el guía en este camino de estudio donde se involucró esfuerzo físico como mental, para poder llegar a esta ansiada meta que es obtener el grado de Ingeniero, así como a cada uno de los docentes que impartieron su conocimiento durante el transcurso de la carrera, para finalizar a nuestro docente guía que con paciencia y dedicación nos pudo orientar en esta anhelada meta.

DEDICATORIA

Este trabajo de titulación se lo dedico a Dios que me brindo la inteligencia y fuerza necesaria para cumplir esta meta que es llegar a ser un profesional.

A mis padres Byron Edmundo Garces Baus y Nancy María Toro Romero quienes estuvieron en este arduo proceso brindándome su apoyo, consejos y amor desde la niñez.

DEDICATORIA

A mi mamá Teresa Sotomayor por todo el apoyo brindado durante mi crecimiento personal y profesional; además, a mis amigos por aportar con sus maravillosas ideas.

Por último, dedico el presente trabajo a todos los jóvenes soñadores que quieren mejorar la calidad de vida de los que menos tienen.

RESUMEN

El presente trabajo de titulación detalla la implementación de un prototipo para el consumo eléctrico de una bomba, blowers y a la vez permita medir la concentración de oxígeno. En la implementación del prototipo utilizamos una Raspberry Pi 4, esta permitirá tener comunicación de red sea de manera inalámbrica o punto a punto, los datos obtenidos por los sensores se almacenarán en una base de datos local, además se podrán visualizar y controlar mediante una plataforma en la nube llamada “Thingspeak”.

El capítulo I da una breve descripción de las redes de comunicación involucrando redes inalámbricas, punto a punto y los tipos de software que tenemos en el medio informático.

En el capítulo II se detalla acerca de una planta de tratamiento teniendo en cuenta sus antecedentes, procesos, los tipos de bomba que se encuentran y en especial la eficiencia eléctrica que tiene.

El capítulo III corresponde a un análisis técnico de cada una de las tecnologías a utilizar para la implementación del prototipo.

En el capítulo IV se especifica la implementación del prototipo con su respectivo diseño permitiendo corresponder al objetivo planteado.

El capítulo V explica el sistema de almacenamiento, la plataforma para la visualización de los datos que se utilizó y los resultados obtenidos en el prototipo.

ABSTRACT

This titling work details the implementation of a prototype for the electrical consumption of a pump, blowers and at the same time allow the oxygen concentration to be measured. In the implementation of the prototype we use a Raspberry Pi 4, it will have access to the communication of the Red Sea wirelessly or point to point, the data affected by the sensors will be stored in a local database, and can also be viewed and controlled by a cloud platform called "Thingspeak".

Chapter 1 gives a brief description of communication networks involving wireless, peer-to-peer networks and the types of software we have in our computing environment.

In chapter II see details of a treatment plant taking into account its background, processes, the types of pump that are found and especially the electrical efficiency it has.

Chapter III corresponds to a technical analysis of each of the technologies to be used for our prototype implementation.

Chapter IV specifies the implementation of the prototype with its respective design designed to correspond to the stated objective.

Chapter V explains the storage system, the platform for displaying the data that was detected and the results obtained in the prototype.

Finally presents the conclusions and recommendations corresponding to the completion of this degree work.

ÍNDICE

| | |
|---|----|
| 1. MARCO TEÓRICO | 1 |
| 1.1. Comunicación | 1 |
| 1.1.1. Inalámbricas | 2 |
| 1.1.1.1. Xbee | 3 |
| 1.1.1.2. Wimax..... | 4 |
| 1.1.2. Redes punto a punto | 5 |
| 1.1.2.1. Cableado UTP..... | 6 |
| 1.1.2.2. Fibra óptica | 7 |
| 1.1.3. Software | 9 |
| 1.1.3.1. Sistema | 9 |
| 1.1.3.2. Programación..... | 9 |
| 1.1.3.3. Aplicación..... | 10 |
| 1.2. IOT | 10 |
| 1.2.1. Obtención de datos..... | 10 |
| 1.2.1.1. Sensores..... | 11 |
| 1.2.2. Sistemas embebidos | 12 |
| 1.2.3. Procesamiento de Datos | 12 |
| 1.2.4. Plataformas para la visualización de datos | 13 |
| 1.2.4.1. Tableau | 13 |
| 1.2.4.2. Plotly | 14 |
| 1.2.4.3. Thingspeak..... | 15 |
| 1.3. Bombas | 15 |
| 2. ANALISIS DE LOS PROCESOS EN LAS PLANTAS DE TRATAMIENTO DE AGUAS RESIDUALES | 16 |
| 2.1. Antecedentes | 17 |
| 2.2. Procesos de las Plantas | 17 |
| 2.2.1. Procesos físicos | 18 |
| 2.2.2. Proceso Biológico | 19 |

| | |
|--|-----------|
| 2.3. Bombas de aguas..... | 21 |
| 2.4. Bombas de aire (Blower) | 22 |
| 2.5. Eficiencia eléctrica..... | 24 |
| 2.5.1. Bomba..... | 27 |
| 2.6. Problemática | 28 |
| 3. ANALISIS DE TECNOLOGÍAS DE ADQUISICIÓN DE DATOS | 28 |
| 3.1. Análisis de tecnologías..... | 28 |
| 3.1.1. Sensor Concentración de Oxígeno | 28 |
| 3.1.1.1. Gravity Analog Dissolved Oxygen Sensor SEN0237 | 28 |
| 3.1.1.2. Atlas Scientific Dissolved Oxygen Probe | 30 |
| 3.1.2. Sensor de Corriente..... | 31 |
| 3.1.2.1. ACS712 30 A Sensor de corriente..... | 31 |
| 3.1.2.2. SCT013-050 Transformador de corriente con núcleo dividido . | 33 |
| 3.1.3. Sistemas Embebidos | 34 |
| 3.1.3.1. Arduino Uno | 34 |
| 3.1.3.2. Raspberry Pi 4..... | 34 |
| 3.1.4 Comunicación..... | 35 |
| 3.1.4.1 Red Ethernet | 35 |
| 3.1.4.2. Red Wifi..... | 35 |
| 3.1.5. Base de Datos | 35 |
| 3.1.5.1. Local | 37 |
| 3.1.5.2. Nube | 37 |
| 4. IMPLEMENTACIÓN DE UN PROTOTIPO QUE PERMITA CONTROLAR Y OPTIMIZAR EL CONSUMO ELÉCTRICO DE UNA PLANTA TRATAMIENTO DE AGUAS RESIDUALES..... | 38 |
| 4.1. Diseño Físico: funcionamiento de la planta (maqueta) | 40 |
| 4.2. Diseño electrónico: materiales utilizados..... | 43 |
| 4.3. Diseño Lógico: S.O código(parte) | 46 |

| | |
|--|-----------|
| 4.3.1. Código en Arduino | 46 |
| 4.3.2. Script en Raspberry para Obtención, discriminación, envío y almacenamiento de datos obtenidos por el puerto serial..... | 48 |
| 4.3.3. Script en Raspbian para control de consumo eléctrico. | 51 |
| 5. Solución de almacenamiento, visualización y pruebas del sistema. | 53 |
| 5.1. Visualización en ThingSpeak | 56 |
| 5.2. Resultados | 59 |
| 5.3. Análisis de Errores..... | 79 |
| 6. CONCLUSIONES Y RECOMENDACIONES | 82 |
| 6.1. Conclusiones..... | 82 |
| 6.2. Recomendaciones | 83 |
| REFERENCIAS | 84 |
| ANEXOS | 88 |

1. MARCO TEÓRICO

En el presente capítulo se explica sobre los tipos de tecnologías que a lo largo del tiempo han venido evolucionando, debido a la necesidad del ser humano para comunicarse a enormes distancias. Entre las tecnologías que se mencionan en este capítulo tenemos: redes inalámbricas, redes punto a punto, IOT.

1.1. Comunicación

En la actualidad estas redes permiten la facilidad colaborativa y de transiciones de tal manera que es indispensable tener conocimiento sobre estos dos procesos y tecnologías que nos permiten llevar la información de una manera rápida, hoy en día la comunicación permite agilizar y facilitar exploraciones científicas.

A. Procedimiento de la comunicación

El ser humano a lo largo de los tiempos ha visto la manera de poder comunicarse, por tal razón hoy en la actualidad podemos tener comunicación a distancias muy grandes. La Figura 1 muestra un proceso de la comunicación ya sea a través de dispositivos o entre personas.

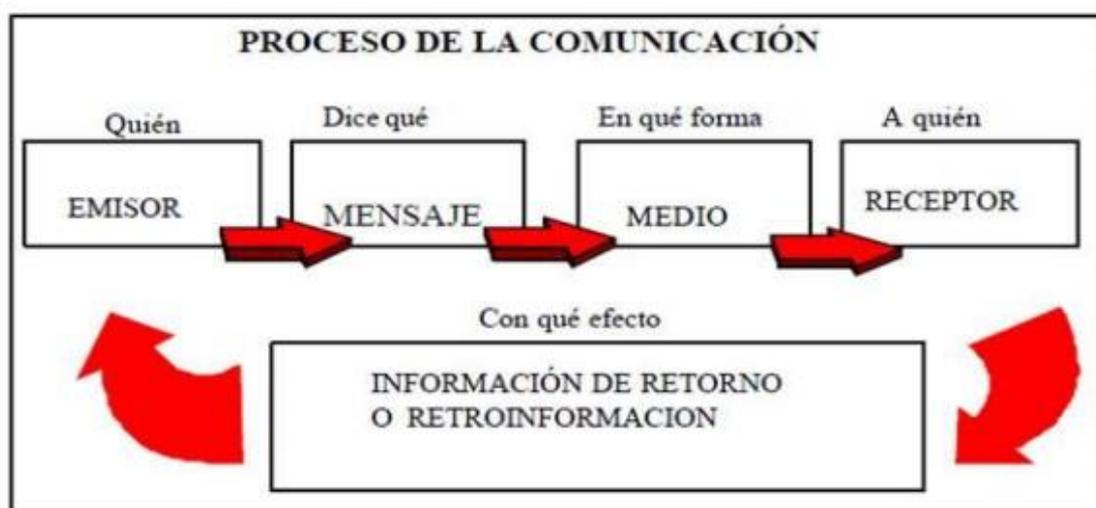


Figura 1. Proceso de comunicación

Tomado de (León-García & Widjaja, 2002)

Dentro de una red pueden funcionar como emisores y receptores los mismos dispositivos a la vez. El medio por donde se transmite la información puede ser: cable, fibra óptica. Para que exista un entendimiento tanto en el emisor como receptor se emplean protocolos como, por ejemplo: TCP/IP.

B. Red de ordenador

Es la unión de todos los dispositivos teniendo una comunicación mutua, teniendo una variedad de componentes en transmisión.

Como objetivos principales tenemos:

- Comunicación de equipos en red: En la actualidad se tienen programas que ayuda a intercambiar mensajería de una manera hablada y escrita entre ellos tenemos (Facebook, Telegram, Microsoft Teams).
- Ahorro de tiempo en el trabajo: De una manera rápida se puede acceder a la información que se necesita sin que nos tome mucho tiempo buscarla.
- Compartir recursos: Es la manera remota a la cual podemos acceder a los archivos que no tenemos en el dispositivo.

Para el acceso a la red se tiene dos topologías muy importantes que son:

- Física: En esta topología da referencia en las conexiones físicas y también se identifican las interconexiones de los terminales y los dispositivos de la infraestructura como ejemplos tenemos: los puntos de acceso inalámbricos, routers y switches.
- Lógica: Da referencia a la transferencia de las tramas de una red que van de un nodo a otro, en la capa de datos los protocolos especifican las rutas de las señales lógicas.

1.1.1. Inalámbricas

El objetivo de las redes inalámbricas es poder eliminar todo el cableado y conexión física lo que lo hace favorable al momento de reducir los costos de mantenimiento, además es una solución favorable para los lugares donde no llega el cableado.

Existen algunos tipos de redes según el alcance entre esas tenemos:

- **PAN o WPAN (Wireless Personal Area Network):** Permiten la interconexión de periféricos que se encuentran a pocos metros o centímetros de distancia del emisor y su velocidad es inferior al Mb/s, uno de los estándares que más se conoce es el bluetooth el cual permite el intercambio de archivos.
- **WLAN (Wireless Local Area Network):** Evita tener varias conexiones de cables, el que más se conoce es el llamado WiFi basado en el estándar IEEE 802.11.
- **WMAN (Wireless Metropolitan Area Network):** Es una red inalámbrica que se utiliza en áreas metropolitanas por lo general se encuentran ubicadas en barrios, urbanizaciones etc. Entre las tecnologías de esta red se encuentra WiMax que permite tener mayor cobertura y ancho de banda.
- **WWAN (Wireless Wide Area Network):** Esta red es de área amplia que se utiliza en las generaciones de los teléfonos móviles.

1.1.1.1. Xbee

Son módulos que permiten la comunicación e interconexión entre dispositivos de manera inalámbrica utilizando un estándar IEEE 802.15.4 para redes punto a punto o redes punto a multipunto.

Como características principales se tiene:

- Bajo consumo que es menor a 50 mA cuando está en funcionamiento y en modo sleep es menor a 10uA.
- Interfaz serial.
- Consta de 16 canales disponibles con 65000 direcciones para cada uno.
- Son de fácil integración.
- Su alcance es bueno para el Xbee normal llega hasta 300ft o 100 mts y para el Xbee Proo llega hasta 1 milla o 1.6 km.

De acuerdo con la Figura 2 Xbee tienes varios tipos de antenas que van a ser descritas a continuación:

Antena chip: Es un chip cerámico que actúa como antena de una manera rápida, sencilla y barata.

Antena de cable: Se reconoce por un cable pequeño que sobresale del módulo permitiendo una radiación omnidireccional.

Antena PCB: Esta antena está impresa en el circuito, el efecto es similar al de la antena de cable.

Conector U-FL: Su conector es pequeño y permite conectar tu propia antena.

Conector RPSMA: Es similar al conector U-FL solo que más voluminoso y grande con la finalidad de poder conectar nuestra propia antena.

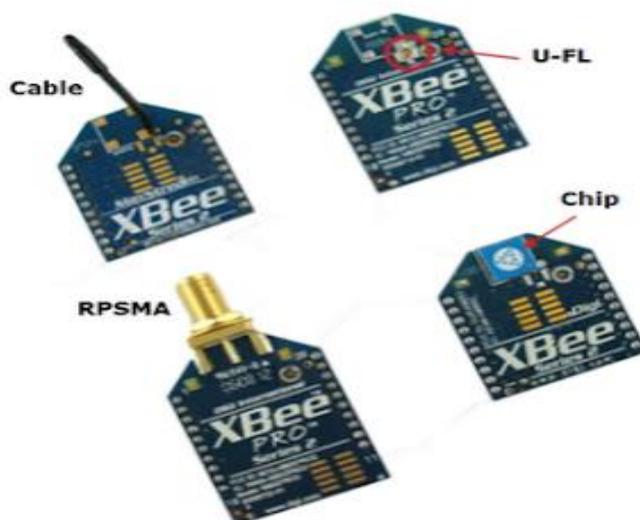


Figura 2. Módulo Xbee con sus tipos de antenas.

1.1.1.2. Wimax

Es un medio de transmisión inalámbrica que permite obtener una mayor cobertura y ancho de banda a través de ondas electromagnéticas, tiene una cobertura de hasta 80 km y en la tecnología de redes 4G el estándar que se utiliza es IEEE 802.16.

Entre las principales ventajas de Wimax tenemos:

- Es una aplicación que tiene varios servicios como: videos, Voip, datos etc.
- El nivel de seguridad es alto con respecto en la encriptación y al nivel de usuario.
- Utiliza tecnología de modulación 16 QAM y 64 QAM.
- Tasa de bits para bajada 75 Mbps y subida 25 Mbps

Como desventajas tenemos:

- Su potencia es limitada para prevenir interferencias con los demás sistemas.
- En zonas rurales escasamente pobladas el ancho de banda no se aprovecha mucho, esto implica que por usuario tenga un gran costo.
- En los dispositivos receptores el consumo de batería es alto.

1.1.2. Redes punto a punto

Es una arquitectura de red que se utiliza para comunicación única entre dos nodos. Estas redes se pueden decir que son de fácil instalación y operación, pero a medida que la red va creciendo se complica al momento de operar y coordinar. Aparte de que su eficiencia rápidamente decrece.

La clasificación de los enlaces que se interconectan con los nodos es:

- Simplex: Se da en un solo sentido la transacción.
- Half-Dúplex: Se realiza en ambos sentidos de manera aleatoria permitiendo transmitir uno a la vez la transacción.
- Full-Dúplex: Se ejecuta en ambos sentidos de manera simultánea la transacción.

Las redes punto a punto tienen las siguientes ventajas:

- Configuración es sencilla.
- Complejidad menor.

- Menor costo debido a que no es necesario utilizar servidores dedicados ni dispositivos de red.
- Como desventajas de las redes punto a punto tenemos:
- Tiene una administración no centralizada.
- Su seguridad no es muy buena.
- No permiten escalabilidad.
- Su rendimiento se reduce notoriamente.
- Los dispositivos conectados pueden actuar como servidor o cliente haciendo que su funcionamiento sea más lento.

En la figura 3 se muestra un ejemplo de red punto a punto la que tiene un máximo de dos nodos.

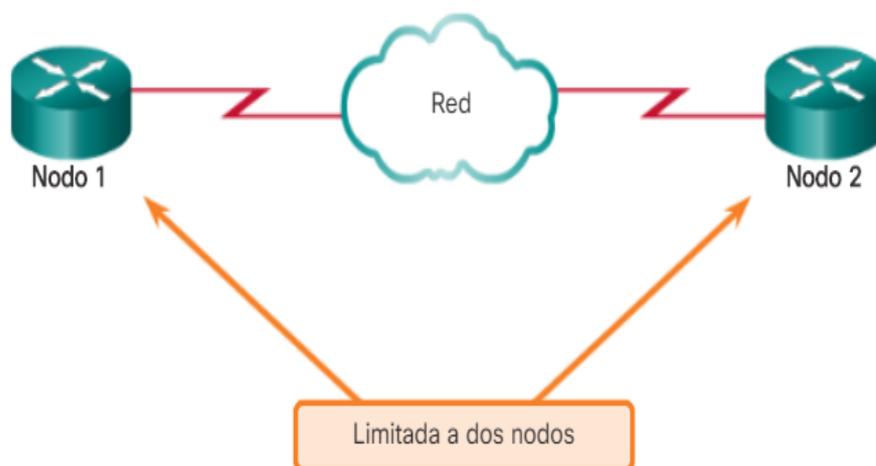


Figura 3. Topología Punto a Punto.

1.1.2.1. Cableado UTP

Es un cable de cobre que permite transportar diferentes señales o electricidad y se lo utiliza en el cableado telefónico y de computadoras para áreas locales. Este cableado tiene un efecto de anulación que se da por los hilos que se trenzan permitiendo limitación degradada de la señal proporcionando un auto blindaje.

Por lo general los cables UTP van de la mano con los conectores RJ-45, estos conectores son utilizados en algunas especificaciones de la capa física entre ellas es el Ethernet, el estándar que se utilizan para la asignación de colores en los pines es TIA/EIA.

Tipos de cables UTP:

- Cable directo de Ethernet: Es usado por lo general en la interconexión de un router con un switch y un ordenador con un switch.
- Cruzado: Se puede conectar equipos iguales como: entre dos switch, host, router.
- Consola: Es único de cisco que se conecta de un puerto serial de una CPU a un puerto de consola de un switch o router.

1.1.2.2. Fibra óptica

Es el medio y tecnología que se asocia a la transmisión de información a través de impulsos de luz a lo largo de una fibra de vidrio o de plástico, son susceptibles a las interferencias electromagnéticas.

La fibra óptica se compone de la siguiente estructura:

- Cubierta: Dióxido de Silicio, plástico.
- Revestimiento: Fibra de vidrio.
- Núcleo: Cristal (Dióxido de Silicio SiO_2 y Dióxido de Germanio GeO_2)
- Longitud de onda: Se da mediante una relación entre frecuencia y velocidad de propagación.
- Refracción: Tiene relación entre espacio libre y velocidad de propagación en el medio.
- La fibra óptica tiene las siguientes ventajas como medio de transmisión:
- Alta capacidad de transmisión (Gbps).
- Baja atenuación (0.17 dB/Km) y constantes casi con la frecuencia.
- Tamaño y peso reducidos, cables ligeros y flexibles.
- Transparentes al formato de señal o servicio, se puede evolucionar en equipos sin cambiar fibras.
- Gran anchura de banda (32 THz/Km).

- Inmune al ruido y a interferencias electromagnéticas y a pulsos electromagnéticos.
- Seguridad (no se pueden pinchar).
- Material base muy abundante.

A continuación, se presentan las desventajas que tiene la fibra óptica como medio de transmisión:

- Atenuación y dispersión
- Costo elevado (aunque se va reduciendo) tanto la fibra como el equipo, los transmisores y receptores son caros.
- La unión de dos fibras requiere un proceso cuidadoso
- Es inevitable casi siempre la conversión electroóptica y óptico-eléctrica
- No se puede almacenar la señal como sí se hace en el dominio eléctrico con memorias. Esto es imprescindible si se quieren crear redes ópticas
- Requieren trato delicado curvaturas.

Entre la clasificación de la fibra óptica tenemos:

Por el material dieléctrico:

- Fibra óptica de silicio
- Fibra óptica de vidrio multi compuesto
- Fibra óptica plástica

Por el modo de propagación:

- Fibra óptica Multimodo (MM)
- Fibra óptica Monomodo (SM)

Por la distribución o perfil del índice de refracción:

- Fibra de índice escalonado o salto de índice (SI= Step Index).
- Fibra óptica de índice gradual (GI Graded Index).

Fibras especiales:

- Fibras de Polarización mantenida.
- Fibras de Dispersión Desplazada/plana.

1.1.3. Software

Es la unión de programas que tienen tareas ya definidas, entre ellas tenemos controladores que graban los espacios de televisión, aplicaciones que nos ayudan con el funcionamiento del celular.

1.1.3.1. Sistema

Es un conjunto de programas que interactúan con un Sistema operativo permitiendo el soporte a otros programas además garantiza un control del hardware.

El software se divide en 3 partes fundamentales las cuales son:

- **Sistema Operativos:** Permiten administrar y controlar el funcionamiento de un computador, entre sus principales funciones tenemos: administra los recursos, las tareas, archivos, brinda servicios de soporte y suministra la interfaz del usuario.
- **BIOS:** Permite que el computador tenga comunicación hacia el exterior ya que este controla la información que está en el sistema.
- **Controladores de Dispositivos:** Permiten a los programas que tienen un mayor nivel el poder interactuar con los dispositivos de hardware.
- **Comandos:** Permiten ejecutar funciones indispensables y básicas que pueden ser prioritarias.
- **Programas utilitarios:** Algunos de estos se encuentran en el sistema operativos, pero sus funciones principales se basan en poder resolver problemas en específicos, también realizan mantenimientos y algunas tareas generales.

1.1.3.2. Programación

Es el conjunto de utilidades y herramientas que facilitan a programadores el poder desplegar programas informáticos utilizando una variedad de lenguajes de programación.

- **Lenguajes de programación:** Son idiomas artificiales que se usan para desarrollar programas que permitan el control tanto lógico como físico de un ordenador brindando soluciones precisas y lógicas.
- **Editor de texto:** Permiten modificar y crear archivos de texto en código fuente o lenguaje de programación.
- **Compilador:** Permiten analizar y ejecutar otros programas.
- **Intérprete:** Su función es traducir la variedad de lenguajes de programación.

1.1.3.3. Aplicación

Son programas que no se involucran con el funcionamiento del computador más bien se instalan en él. Entre esos programas tenemos: videojuegos, herramientas de trabajo como Excel, reproductores de video o audio etc.

1.2. IOT

Es un concepto con el cual la información leída por sensores electrónicos usados en hogares, oficinas, industrias, laboratorios, teléfonos móviles, automóviles, infraestructura de ciudades y centros de datos, se transmite a un dispositivo central de monitorización. El dispositivo central está ubicado en la Internet, permitiendo la conexión digital de objetos de uso cotidiano a la misma, esto es, conectar el mundo físico con el digital por medio de computadoras y plataformas web en la nube que almacenan y procesan la información transmitida por sensores (Vega Luna et al., 2019)

1.2.1. Obtención de datos

Un proceso de obtención de datos se da mediante el mundo real (sistema analógico) donde se encuentran fenómenos físicos y se transforman en señales eléctricas. Estas señales son medidas y convertidas en formato digital para ser analizados, almacenados y procesados en una computadora. Para ello se utiliza un módulo de digitalización o tarjeta de Adquisición de Datos (DAQ) (Cabarna González et al., 2010).

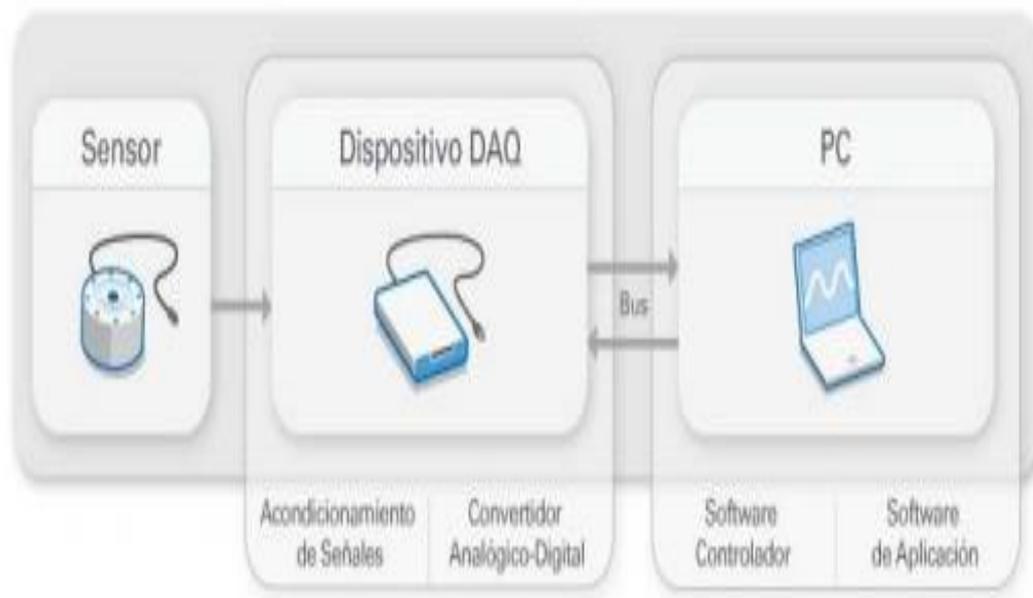


Figura 4. Sistema DAQ.

Tomado de (Cabarna González et al., 2010)

1.2.1.1. Sensores

“Un sensor es un dispositivo que, a partir de la energía del medio donde se mide, da una señal de salida transducible que es función de la variable medida”. (Areny, 2003)

Un sensor con un transductor a veces se emplea como sinónimos, con la diferencia de que el sensor es un dispositivo de entrada.

1.2.1.1.1. Sensor de oxígeno disuelto

Puede detectar el oxígeno mediante sondas, permitiendo tomar datos de una manera inmediata a parte de poder reflejar la calidad del agua en que se encuentra.

1.2.1.1.2. Sensor de corriente

También llamados transductores permiten medir en el circuito eléctrico el flujo de corriente, existen dos tipos: Corriente alterna y corriente continua.

A. Corriente Alterna o (AC): es cuando el sentido y magnitud cambian cíclicamente, esta se refiere a como la electricidad alcanza a las industrias y

hogares, incluso las señales transmitidas de radio y audio de cables eléctricos serían ejemplos de esta corriente.

B. Corriente Continua o (DC): se da cuando los electrones o cargas eléctricas en un circuito cerrado van por el mismo sentido de tal manera que se mueven del polo negativo al positivo, eso sucede en las baterías o algunas otras fuentes que generan este tipo de corriente.

En esta corriente la polaridad siempre va a ser la misma sin importar que la intensidad disminuya en cuanto al consumo de la carga, además se la suele identificar también como una corriente constante.

1.2.2. Sistemas embebidos

Los sistemas embebidos tienen sistemas operativos que se controlan por microprocesadores programados en lenguajes nativos permitiendo que la mayoría actúen en tiempo real.

Existen dos sistemas embebidos grandes (LES) y pequeños (SES) dentro de estos podemos encontrar módulos E/S, actuadores, sensores permitiendo brindar una necesidad específica en routers, sistemas satelitales, celulares etc.

1.2.3. Procesamiento de Datos

En el procesamiento de datos se acumulan y manipulan los datos con el fin de obtener información que sea más legible y se pueda utilizar.

Para el procesamiento de datos tenemos seis etapas las cuales son:

Recoger datos: Consiste en extraer los datos de fuentes disponibles entre ellas tenemos los almacenes de datos y data lakes, los datos que se encuentren disponibles deben ser fiables y bien creadas.

Se preparan los Datos: Aquí se organizan y limpian los datos permitiendo detectar errores que se encuentren y eliminarlos permitiendo crear datos de gran calidad.

Introducción de datos: Los datos limpios se insertan a su destino permitiendo que se traduzcan a un lenguaje más entendible haciendo que los datos tengan una forma utilizable.

Procesamiento: En esta etapa mediante algoritmos de machine learning se realiza el procesamiento.

Análisis de datos o Salida: Aquí finalmente los datos son utilizables para personas comunes ya que se encuentran legibles, traducidos en formas de imágenes, videos, texto simple etc.

Acumulación de datos: En esta última etapa todos los datos que fueron procesados se los almacena para su uso a futuro, si los datos se encuentran bien almacenados se podrá tener un acceso rápido y fácil.

1.2.4. Plataformas para la visualización de datos

1.2.4.1. Tableau

Es una plataforma que permite un análisis de datos de manera segura, flexible y eficaz. Es una de las mejores alternativas ya que a los usuarios les permite combinar, dar forma y limpiar los datos de una manera segura y rápida. Se puede descargar para el computador o trabajar de una manera online.

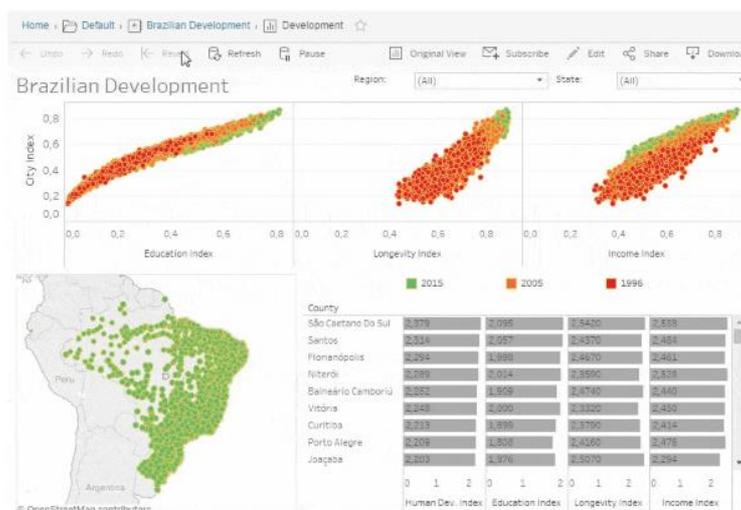


Figura 5. Visualización de datos en tableau.

Tomado de (Tableau, 2020).

1.2.4.2. Plotly

Es una plataforma donde se pueden visualizar los datos de una manera interactiva como cuadros y gráficos, usando cualquier lenguaje de programación.

Esta se encuentra en versiones gratuitas con un límite de prueba gratis y su implementación se basa en la web, nube y Saas.

Características:

- Permite que los usuarios puedan copiar, pegar, transmitir e importar datos para ser visualizados y visualizados.
- Trabaja en conjunto con Python ofreciendo un limitado entorno.
- Las gráficas se diseñan con una GUI y Python.
- Se puede compartir una URL permitiendo que otros puedan visualizar, crear una copia.

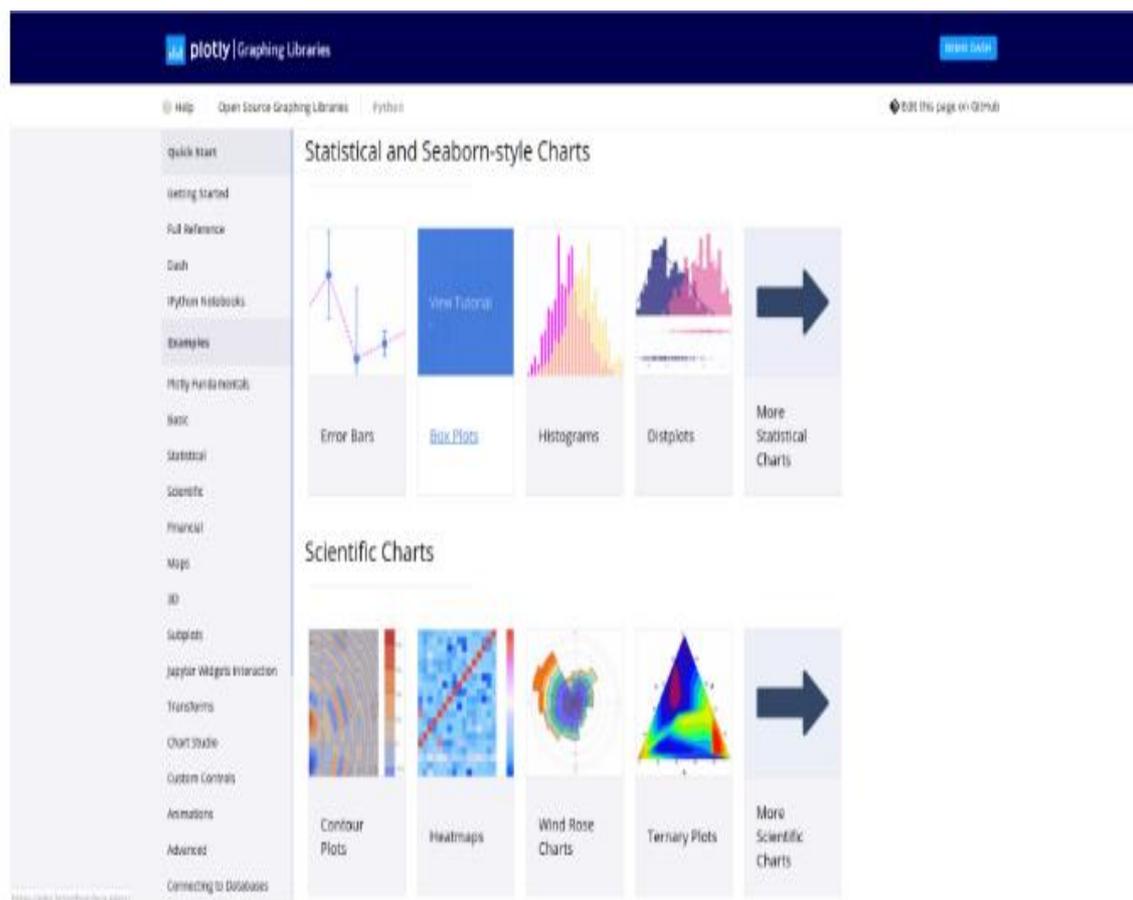


Figura 6. Plataforma Plotly.

Tomado de (Plotly, 2020)

1.2.4.3. Thingspeak

Brinda un servicio de plataforma en cuanto al análisis de IOT permitiendo analizar, agregar y visualizar un conjunto de datos en tiempo real en la nube.

Las principales características de thinkgSpeak son:

- Los datos se pueden compartir en canales públicos.
- En canales privados se pueden reunir os datos.
- Tiene alertas
- Permite visualizaciones y análisis en Matlab.
- MQTTy API RESTful.
- Capacidad de receptor datos cada en intervalos mínimos de 5 segundos.
- Licencia gratis con capacidad de 3000000 de mensajes por año.

Esta plataforma funciona con lo siguiente:

- Raspberry Pi
- Arduino
- Modulo wifi ESP8266
- Matlab

Es la exhibición de manera gráfica de los datos permitiendo comprender y de ver en una manera accesible a través de patrones de datos, tendencias y valores atípicos.

Las herramientas tecnológicas para visualizar los datos son muy importantes debido a que nos permiten analizar enormes cantidades de información para poder encaminarse a buenas decisiones en base a los datos obtenidos.

1.3. Bombas

Tienen como función el transportar el fluido desde un nivel de energía bajo hasta uno más alto, las bombas que se utilizan en las plantas de tratamiento de aguas residuales son las de tipo sumergibles las cuales constan de una integración de motor y bomba, son robustas y compactas aparte de su fácil uso y la ocupación de espacio es mínima.

Existen varios tipos de bombas que se utilizan en plantas de aguas residuales entre ellas tenemos la siguiente clasificación:

A. Desplazamiento Positivo:

- **Diafragma:** Permiten el desplazamiento de las aguas residuales en dirección específica, la funcionalidad de esta bomba es que cuando el diafragma está cerrado se detiene el agua y cuando está abierto el agua fluye.
- **Engranajes:** En esta bomba los engranajes giran dentro de la bomba para poder desplazar el agua residual, creando una succión permitiendo mover el agua residual de un lugar a otro.
- **Lóbulos:** Permiten empujar el agua residual en una sola dirección y teniendo un control sobre el flujo de agua.

B. Desplazamiento Dinámico:

- **Centrífugas:** Este tipo de bomba a través de la fuerza centrífuga construye velocidad permitiendo el desplazamiento del fluido, su componente principal son los impulsores de hilaza los cuales tienen una enorme salida y una pequeña válvula de admisión.
- **Sumergibles:** Estas bombas se colocan específicamente en los drenajes de las plantas de aguas residuales, permiten el bombeo para poder eliminar los fluidos en áreas específicas.

2. ANALISIS DE LOS PROCESOS EN LAS PLANTAS DE TRATAMIENTO DE AGUAS RESIDUALES

El presente capítulo va a tratar sobre los principales elementos eléctricos para el funcionamiento de la planta de tratamiento de aguas residuales; estos elementos en una planta tradicional están siempre en operatividad y generalmente no se tiene un registro de consumo eléctrico por semana o día

debido a que las planillas llegan al fin del mes mostrando únicamente el consumo total por el mes transcurrido.

Para ello, se va a analizar el funcionamiento de la bomba y blower que inciden con el consumo eléctrico de una planta, además de ser monitoreados los resultados durante el transcurso del día.

2.1. Antecedentes

La construcción de plantas de tratamiento de aguas residuales y sistemas de alcantarillado inició hace 150 años justamente por la presencia de organismos patógenos (C. M, Lopez-Vazquez, 2017).

El requerimiento energético de una planta de tratamiento de aguas residuales depende de la capacidad de la planta, de la tecnología de tratamiento, del tipo de tratamiento que se le dé al lodo generado y de otros factores locales específicos, (Nolasco, 2010).

Las plantas de tratamiento de aguas residuales eliminan los contaminantes que se encuentran en el agua fluente del uso humano. Por lo general estas aguas residuales se generan: residencias, industrias y locales comerciales, llegando a estas plantas para darles su debido tratamiento.

Los Blowers permiten la oxigenación de las aguas residuales en la planta de tratamiento. En otros casos proporciona aire en filtros multimedio como medio de fregado y limpieza.

Este tema es planteado debido a que, las plantas tradicionales utilizan personas que tienen que medir y registrar manualmente y no exponer a las personas que lo realizan a una contaminación que afecte la salud; para ello se ha visto la necesidad de calcular el consumo de estos dos elementos ya que son parte importante de una planta de tratamiento de aguas residuales.

2.2. Procesos de las Plantas

Una planta de tratamiento de aguas residuales tiene dos procesos básicos que se utilizan para descontaminar el agua y pasando por un proceso de

regeneración poder reutilizar dando otros usos como: urbanos, agrícolas o industriales.

Los escasos de agua dulce van aumentando sobre todo en los sectores urbanos más poblados debido, a que algunos de sus acuíferos se han ido secando o extinguiendo por el crecimiento poblacional no planificado.

2.2.1. Procesos físicos

El primer proceso de filtración de la materia orgánica donde se trata de remover el material en suspensión utilizando: rejillas finas y gruesas, triturador o desarenador, sedimentador en la Figura 7 se muestra dicho proceso.

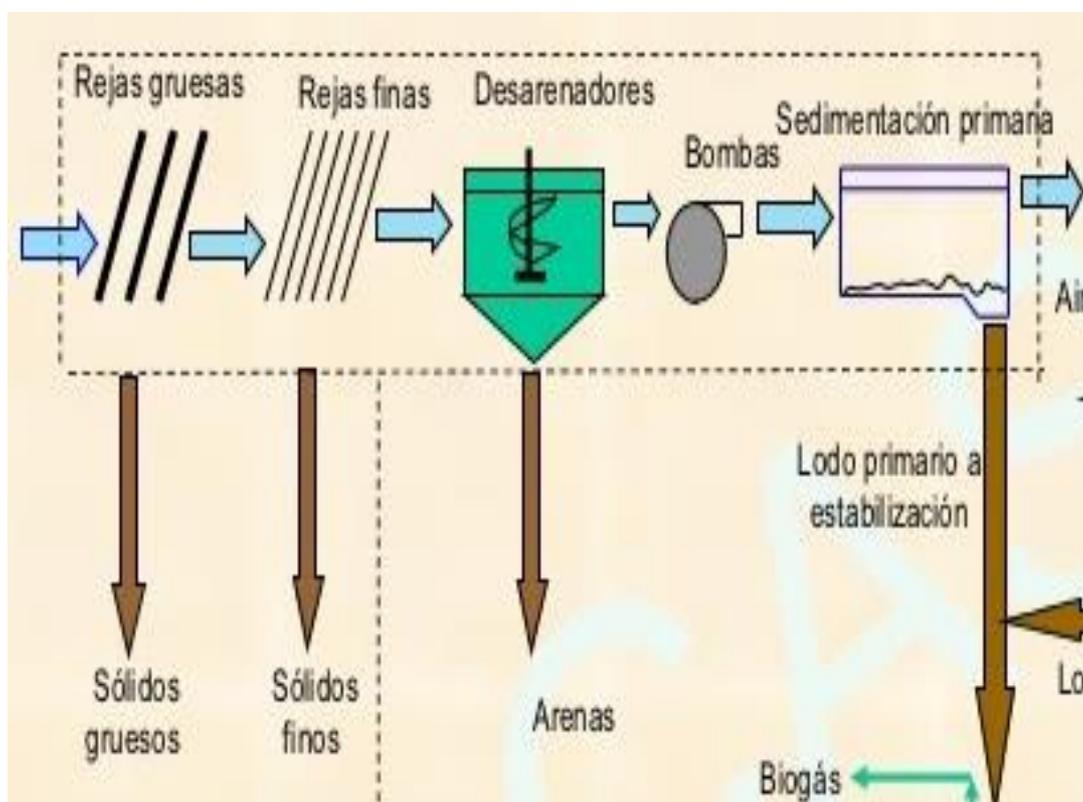


Figura 7. Filtración física.

Tomada de (Ciencias y Tecnología Benedikta, 2017).

Es importante contar con la filtración física para que toda la materia fecal se descomponga en partículas más pequeñas para que el siguiente proceso sea más eficiente.

2.2.2. Proceso Biológico

La Figura 8 se visualiza el proceso que realizan los microorganismos en cuanto a la restauración del agua, en donde se encuentra el nitrógeno, fósforo y gases para su eliminación.

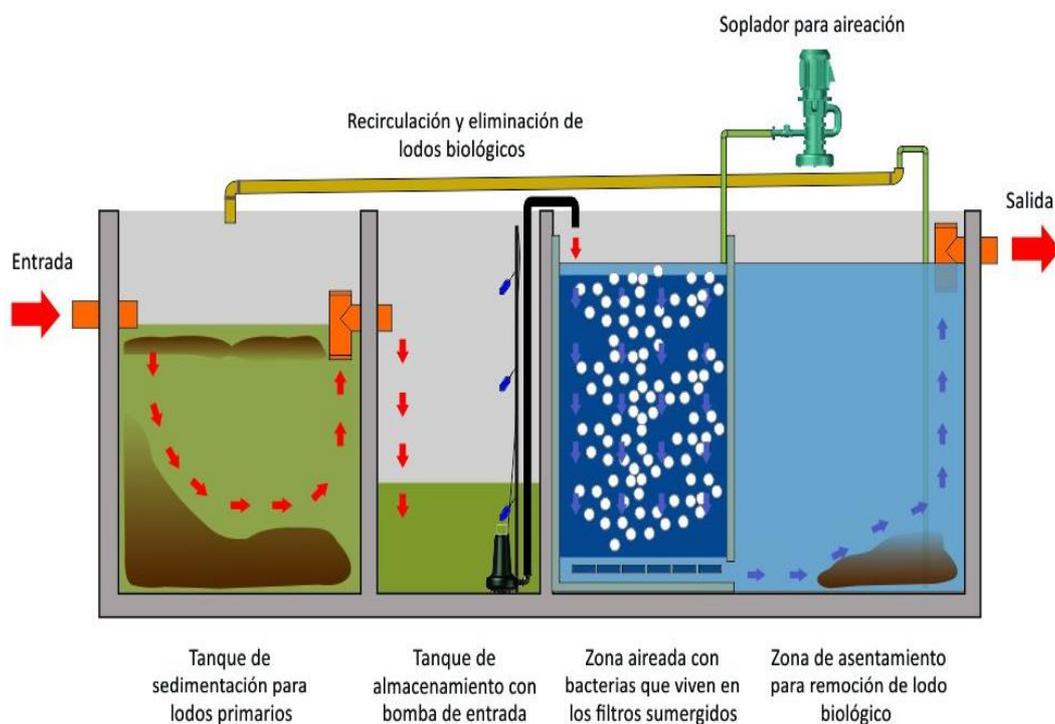


Figura 8. Planta de tratamiento de aguas residuales proceso biológico.

Tomado de (Rhino Tech Intl SA, 2020).

Para la filtración biológica se utilizan microorganismos que consumen el oxígeno del agua para poder oxidar los contaminantes y poder clarificar el agua a ser tratada. En otras palabras, se puede decir que a mayor cantidad de heces fecales (materia orgánica) se necesita más cantidad de oxígeno que utilizan los microorganismos para poder degradar la materia orgánica y pierdan contaminantes. Los microorganismos para oxidar necesitan una cantidad de oxígeno necesaria llamada DBO (Demanda Bioquímica de Oxígeno). Como se observa en la Figura 9.

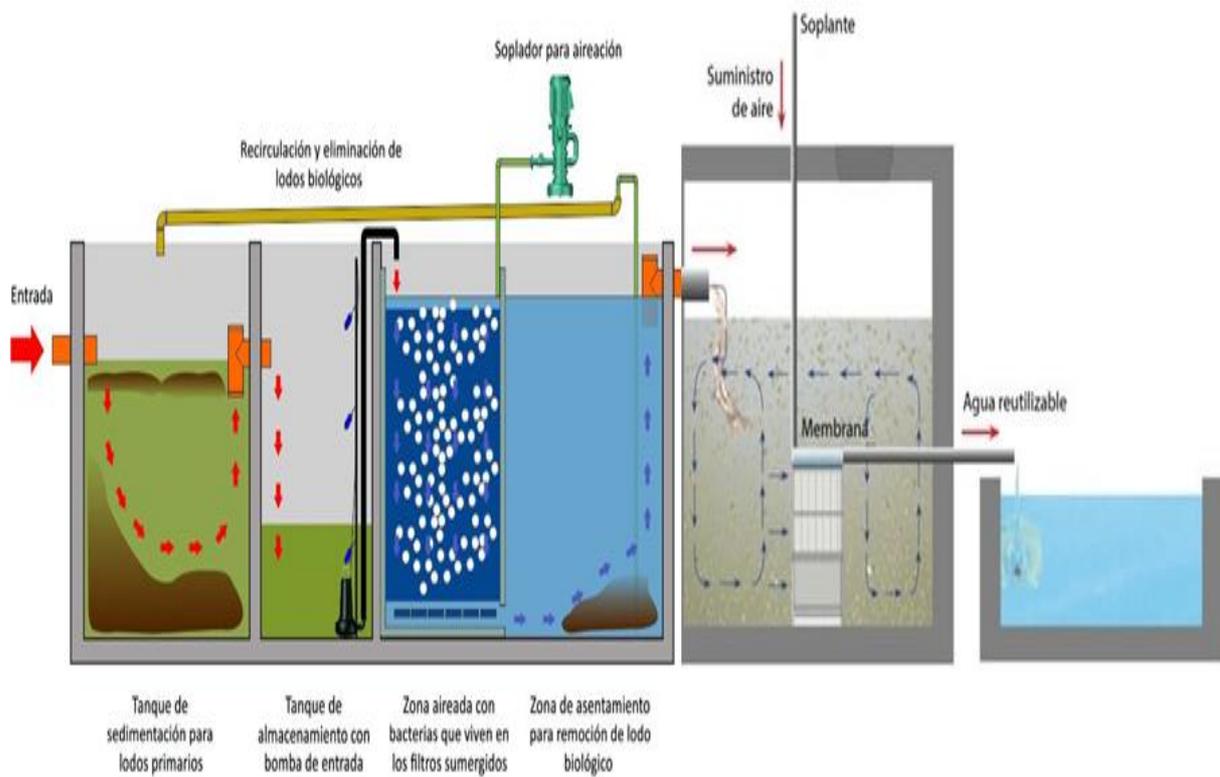


Figura 9. Proceso filtración mecánica y biológica.

Tomado de (Rhino Tech Intl SA, 2020).

Demanda bioquímica de oxígeno se la usa como una medida utilizada para saber cuánta cantidad de oxígeno es requerido para la oxidación de la materia orgánica y para determinar la DBO 5 en días, se refiere al consumo de oxígeno en los microorganismos.

| Criterio (mg/L) | Clasificación | Color |
|-----------------------|---|-----------------|
| $DBO_5 \leq 3$ | EXCELENTE NO CONTAMINADA | AZUL |
| $3 < DBO_5 \leq 6$ | BUENA CALIDAD AGUAS SUPERFICIALES CON BAJO CONTENIDO DE MATERIA ORGÁNICA BIODEGRADABLE | VERDE |
| $6 < DBO_5 \leq 30$ | ACEPTABLE CON INDICIO DE CONTAMINACIÓN. AGUAS SUPERFICIALES CON CAPACIDAD DE AUTODEPURACIÓN O CON DESCARGAS DE AGUAS RESIDUALES TRATADAS BIOLÓGICAMENTE | AMARILLO |
| $30 < DBO_5 \leq 120$ | CONTAMINADA AGUAS SUPERFICIALES CON DESCARGAS DE AGUAS RESIDUALES CRUDAS, PRINCIPALMENTE DE ORIGEN MUNICIPAL | NARANJA |
| $DBO_5 > 120$ | FUERTEMENTE CONTAMINADA AGUAS SUPERFICIALES CON FUERTE IMPACTO DE DESCARGAS DE AGUAS RESIDUALES CRUDAS MUNICIPALES Y NO MUNICIPALES | ROJO |

Figura 10. Rango de niveles de demanda bioquímica de oxígeno en días Tomado de (Conagua, 2016).

Después, el agua saliente de haber pasado por los tanques de oxigenación se la llama agua residual la cual aún no está apta para riego ni para consumo humano, se aplica otro proceso el cual se dosifican sustancias químicas los cuales pueden ser nocivas a la salud de los operarios de la planta para que esa agua residual pase a ser un agua reutilizable que es apta para el riego y no recomendable para el consumo humano.

2.3. Bombas de aguas

Estas plantas de tratamiento deben tener en cuenta algunos elementos que están presentes durante el proceso de tratamiento del agua como las materias fecales compuestas de residuos de comida no procesada, bacterias, moco. Para mover el agua residual se utilizan bombas centrifugadoras porque son capaces de mover líquidos con partículas en suspensión, pero sino son muy viscosos como por ejemplo la miel.

El principio de funcionamiento de una bomba de agua centrifugadora es el motor eléctrico que produce energía impulsora para ser transferido a los rotores o paletas en forma de energía cinética al fluido, como se observa en la figura 11. La energía cinética es cuando el giro del rotor provocado por el motor eléctrico hace que los fluidos cambien su estado de reposo a movimiento.

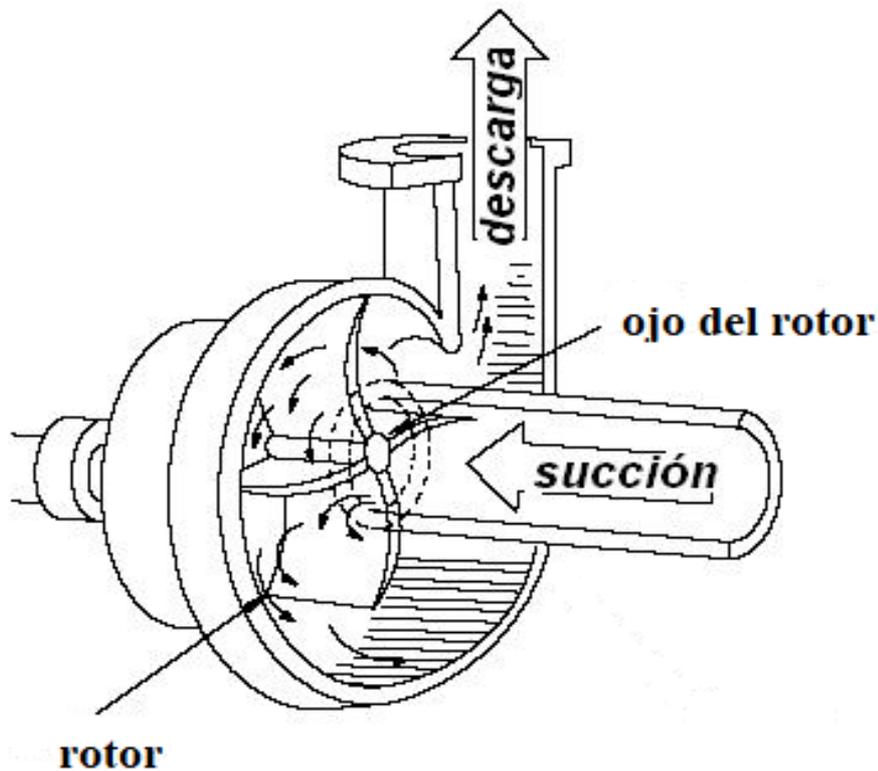


Figura 11. Partes de una bomba centrífuga.

Tomado de (Fullmecanica, 2014).

Dichas bombas utilizan motores eléctricos que superan los 2 kW*h y pasan en constante funcionamiento las 24 horas del día obteniendo un consumo de: 48 kW*h al día, 336 KW*h a la semana y 1440 kW*h al mes.

2.4. Bombas de aire (Blower)

El blower es una bomba que atrae al oxígeno con un rotor para luego hacerle circular por tubos de aire hacia los tanques de oxigenación mediante burbujas que pueden ser pequeñas, medianas o grandes; proporcionada por los difusores de oxígeno como se observa en la Figura 12.



Figura 12. Difusor de Oxígeno.

Tomado de (Manantial, 2017).

El mecanismo de funcionamiento de una bomba blower es: el motor eléctrico, el rotor diseñado dinámicamente para atraer el aire y ser empujado hacia un conducto y las entradas y salidas de aire como se observa en la Figura 13. En el Anexo 6 se muestra una bomba blower utilizada en la Planta de tratamiento de aguas residuales UDLA Park.

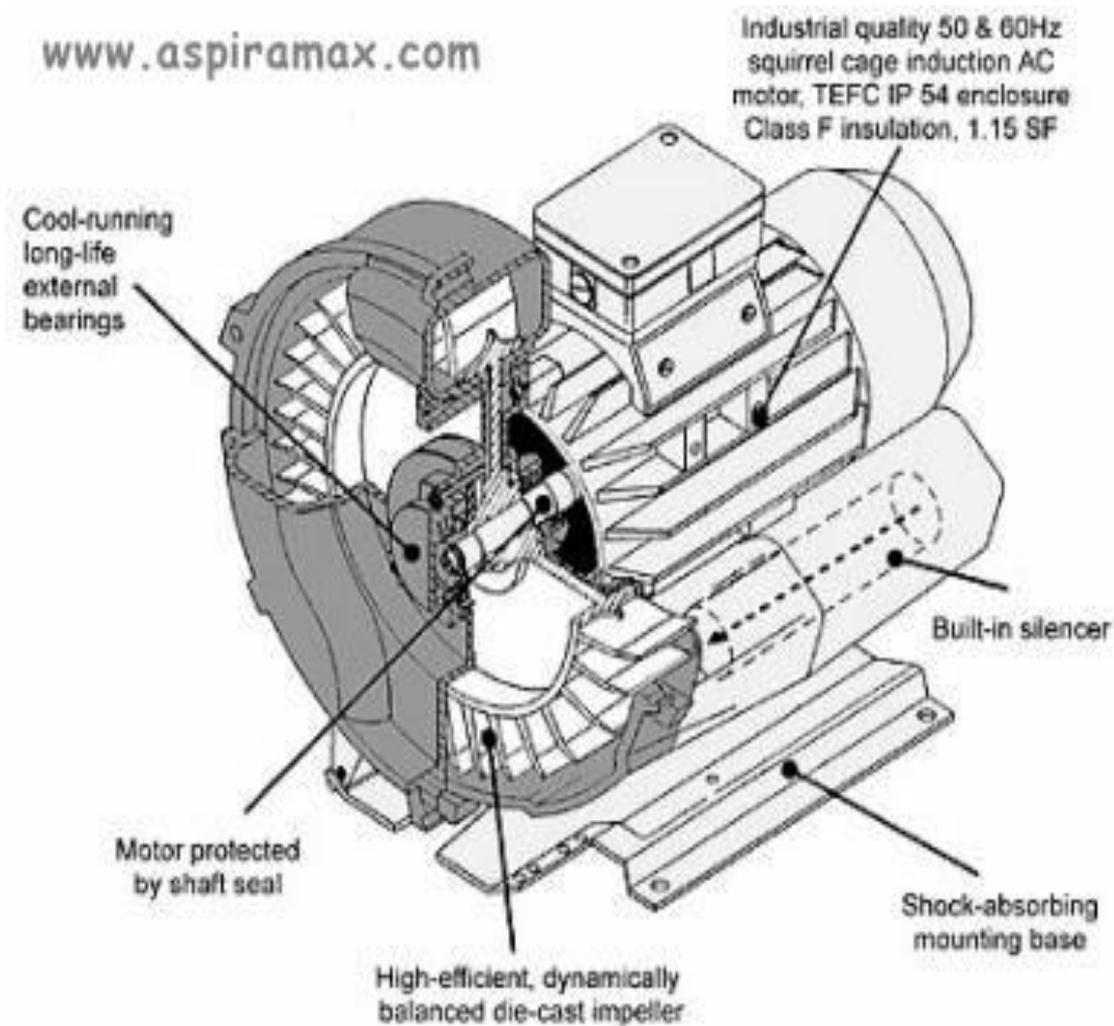


Figura 13. Partes de una bomba Blower.

Tomado de (Arteom, 2016).

2.5. Eficiencia eléctrica

La eficiencia energética es el hecho de minimizar la cantidad de energía necesaria para satisfacer la demanda sin afectar a su calidad; supone la sustitución de un equipo por otro que, con las mismas prestaciones, consuma menos electricidad. No supone, por tanto, cambios en los hábitos de consumo (el comportamiento del usuario sigue siendo el mismo), pero se consume menos energía ya que el consumo energético para llevar a cabo el mismo servicio es menor (Schallenberg et al., 2008).

La eficiencia energética es poder administrar el consumo de energía de una manera inteligente para así reducir costos y haciendo que el planeta sea sostenible.

La mayoría de los electrodomésticos que compramos no son eficientes, debido a que los productos con eficiencia suelen ser más costosos que los que no tienen.

En la actualidad lo que se busca con la eficiencia energética es la protección del medio ambiente haciendo una reducción energética y que el usuario pueda consumir lo necesario y nada más.

Para nosotros obtener eficiencia energética por lo general se opta por el uso de alguna tecnología, proceso de producción que resulte más eficiente o métodos comunes para evitar las pérdidas de energía.

Es importante la aplicación de eficiencia energética debido a que existen algunos factores como:

- Cambio climático.
- El consumo de la energía en el mundo va creciendo.
- Incremento en el costo de la energía.

Para el caso del consumo eléctrico en el prototipo, se tomó en consideración el uso de la fórmula de la eficiencia energética la cual se presentará a continuación:

$$\eta = \frac{P_u}{P_c} * 100$$

Donde:

η = Es la eficiencia eléctrica.

P_u = Potencia útil

P_c = Es la potencia consumida o también conocida como teórica.

100= Porcentaje total

Para nosotros obtener la potencia útil debemos utilizar la siguiente formula:

$$P = \frac{E}{t}$$

Donde:

P: Potencia Efectiva

E: Es la energía.

t: Es el tiempo que se emplea en realizarlo.

Y la potencia consumida (teórica) son los watts del motor, haciendo estos cálculos obtendremos el porcentaje de nuestra eficiencia.

Cuando hablamos de eficiencia, para el caso presente, el consumo de energía del blower debe ser cuando el sistema lo requiera, es decir, cuando los niveles de concentración de oxígeno disminuyan del rango ideal o si son superiores, debe activarse la bomba para mantener las condiciones ideales. A continuación, se muestra una tabla donde se indican los valores de oxígeno disuelto junto los intervalos de tiempo de funcionamiento del blower.

Tabla 1.

Revisión de parámetros fisicoquímicos como indicadores de calidad y contaminación del agua.

| [OD] mg / L | Condición | Estado del Blower |
|-------------|-----------|-----------------------------------|
| 0 | Anoxia | Encendido y censando |
| 0 - 5 | Hipoxia | Encendido y censando |
| 5 - 8 | Aceptable | Encender si OD < 7 durante 60 min |

| | | |
|--------|---------------|--|
| 8 - 12 | Buena | Apagar si OD es > 12 censar cada 60 min |
| > 12 | Sobresaturada | Apagado |

Tomado de (Carol, 2009).

2.5.1. Bomba

La bomba usada en estas plantas por lo general tiene dos fases y el neutro, por cada fase pasa un voltaje entre 230 y 400 V además a eso el consumo es de 3kW. Para el caso didáctico se utilizó bombas de peceras que consumen 2 Watts y 3 Watts.



Figura 14. Imagen de las bombas utilizadas.

2.6. Problemática

Específicamente en estas plantas, los motores utilizados por el Blower se encuentran todo el tiempo en funcionamiento sea o no necesario; debido a aquello se ha levantado una solución: censar la concentración de oxígeno disuelto que existe dentro de los tanques aireadores y comparar con las condiciones ideales de concentración de oxígeno disuelto teóricas (DO) proporcionadas por el docente especialista a cargo, los motores solo entran en funcionamiento cuando no se cumplan las condiciones ideales y como consecuencia se reduce el consumo eléctrico.

3. ANALISIS DE TECNOLOGÍAS DE ADQUISICIÓN DE DATOS

3.1. Análisis de tecnologías

Se va a detallar una comparativa de diferentes fabricantes de los elementos a utilizar para el desarrollo del prototipo.

3.1.1. Sensor Concentración de Oxígeno

Las bacterias necesitan oxígeno para realizar el trabajo de descomposición de los desechos orgánicos. Dicha concentración de oxígeno depende de algunos factores como: la aireación del agua efecto de saltos o agitaciones, la difusión del oxígeno en el entorno y para poder medir la cantidad de oxígeno que esta disuelta en el agua existen los siguientes sensores:

- Gravity Analog Dissolved Oxygen Sensor SEN0237
- Atlas Scientific Dissolved Oxygen Probe

3.1.1.1. Gravity Analog Dissolved Oxygen Sensor SEN0237

El sensor SEN0237 es compatible principalmente con el microcontrolador Arduino, el sensor ha sido diseñado para obtener el dato del (OD) que se encuentra en el agua con aplicaciones como: acuicultura, monitoreo ambiental, etc. Los datos son obtenidos de forma inmediata obteniendo una señal de salida que puede ser fácilmente acoplable a sistemas de control. (Sensors et al., 2017). En la Figura 15 se puede visualizar el sensor SEN0237



Figura 15. Sensor Oxígeno Disuelto SEN0237 del fabricante DFRobot. Tomado de (Sensors et al., 2017).

En la tabla 2 se muestran las especificaciones técnicas del sensor SEN0237.

Tabla 2.

Especificaciones técnicas sensor SEN0237

| DATASHEET | |
|---|---|
| Tipo: | Sonda Galvánica |
| Rango de detección: | 0~20 mg/L |
| Rango de Temperatura: | 0~40 °C |
| Rango de presión: | 0~50 PSI |
| Periodicidad de remplazo de la membrana: | 1~2 meses (dentro aguas sucias); 4~5 meses (dentro de aguas limpias) |
| Mantenimiento: | 1 mes |

| | |
|------------------------------|-------------|
| Tiempo Duración: | 1 año |
| Largo Cable: | 2 metros |
| Probe Connector: | BNC |
| Voltaje de Operación: | 3.3 ~ 5.5 V |
| Señal de Salida | 0 ~ 3.3 V |

Tomado de (Sensors et al., 2017).

3.1.1.2. Atlas Scientific Dissolved Oxygen Probe

Atlas Scientific ofrece soluciones para diseños de bajo costo sin reducir la operatividad del diseño; este tiene una sonda de “D.O” de membrana HDPE (polietileno de alta densidad) y un circuito encargado para la señal analógica proveniente de la sonda siendo compatible con cualquier procesador que pueda interpretar las señales analógicas, más información en el anexo 1.



Figura 16. Sensor de Oxígeno Disuelto del fabricante AtlasScientific

La tabla 3.

Indica las especificaciones técnicas del sensor D.O.

| DATASHEET | |
|---|------------------------|
| Tipo: | Sonda Galvánica |
| Rango de detección: | 0~100 mg/L |
| Rango de Temperatura: | 0 ~ 60 °C |
| Rango de presión: | 0~500 PSI |
| Periodicidad de remplazo de la membrana: | ~1 Año |
| Mantenimiento: | ~18 Meses |
| Tiempo Duración: | 5 años o mas |
| Largo Cable: | 300 Metros |
| Probe Connector: | BNC Femele SMA Male |
| Voltaje de Operación: | 3.3 ~ 5 V |
| Señal de Salida | 0 mV ~ 3.3 V |

Tomado de (Deacutis, 2016).

3.1.2. Sensor de Corriente

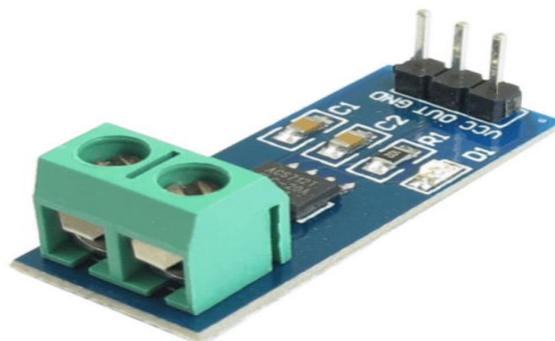
Permite registrar la corriente que pasa a través de un conductor.

Se realiza un análisis de algunos tipos de sensores con funciones similares.

3.1.2.1. ACS712 30 A Sensor de corriente

El sensor está plasmado en base a un efecto llamado hall, realiza la medición para: AC y DC. La señal que sale es una señal analógica que tiene una variación en relación con la energía eléctrica que pasa a través del sensor. Cuenta con un

diseño de fábrica que permite ser compatible con cualquier procesador o microcontrolador. (Electro Crea, 2020)



ElectroCrea.com

Figura 17. Imagen referencial del sensor.

Tomado de (Electro Crea, 2020).

La tabla 4.

Indica las especificaciones técnicas del sensor ACS712

| DATASHEET | |
|-------------------------------------|-------------------|
| Entrada Nominal (rms): | 30 A |
| Máxima Entrada: | 35 A |
| Salida Nominal: | 66 – 185 mv/A |
| Precisión: | ±4 % |
| Temperatura Operación: | -40°C hasta +85°C |
| Voltaje máximo de operación: | 2.1 Kv rms |
| Rango de Frecuencia: | 50 Hz |

Tomado de (Allegro, 2020).

3.1.2.2. SCT013-050 Transformador de corriente con núcleo dividido

Los TC o transformadores de corriente, son sensores utilizados para medir la corriente alterna y medición de consumo eléctrico. Este tipo de sensor de núcleo dividido tiene varias ventajas debido a que se puede enganchar a cualquier cable de alta tensión de sin ser invasivo al cable.



Figura 18. Sensor de corriente no invasivo.

Tabla 5.

Indica las especificaciones técnicas del sensor PZCT-02

| DATASHEET | |
|-------------------------------------|-------------------|
| Resistencia de Aislamiento: | > 1000 M Ω |
| Rango de medición: | 0 – 100 A |
| Fuerza Eléctrica max: | 380 V AC |
| Resistencia Interna: | 10 Ω |
| Temperatura Operación: | -25°C hasta +70°C |
| Voltaje máximo de operación: | 660 V |
| Rango de Frecuencia: | 50 Hz – 60Hz |

Tomado de (Alldatasheet, 2020)

3.1.3. Sistemas Embebidos

3.1.3.1. Arduino Uno

Es un dispositivo que consta de un microcontrolador que utiliza un lenguaje de programación sencillo para poder realizar proyectos para poder controlar dispositivos electrónicos.

3.1.3.2. Raspberry Pi 4

Es un ordenador de bajo costo y un tamaño reducido, está constituido por una placa base en la que se monta el procesador, una memoria RAM y un chip gráfico. La diferencia que tiene con un ordenador normal es que no tiene disco duro, pero en este caso se usa una tarjeta SD. El sistema operativo que utilizan las raspberry pueden ser varios, la mayoría de estos son basados en el kernel de Linux, hoy en día los más conocidos son Firefox OS, Android, OpenWebOS, Raspbian o Unix (Rodríguez, 2014).

Tabla 6.

Comparativa entre Arduino y Raspberry Pi

| DataSheet | Arduino Uno | Raspberry Pi 4 |
|-------------------------|-------------------------|--------------------------------------|
| Precio | \$25 | \$70 |
| Memoria RAM | 2 KB | 2 GB |
| S.O | No dispone | Android, Raspbian, IoT Core, Windows |
| Entradas I/O | 19 pines GPO | 40 pines |
| Velocidad | 16 MHz | 1.5GHz |
| Entrada Internet | No incluido costo extra | Incluye puerto ethernet y wifi |

3.1.4 Comunicación

La comunicación es la transmisión de datos entre ordenadores que se encuentran en una misma conexión para ello existen diversos protocolos que permiten que la comunicación sea de manera eficiente.

3.1.4.1 Red Ethernet

El Arduino uno no incluye la interfaz ethernet como se muestra en la Figura 19, esta interfaz hay que adquirir aparte con un costo extra para habilitar la conexión hacia la Internet.

La Raspberry Pi 4 incluye dentro de su placa la capacidad de conectarse a Internet mediante un conector RJ45.

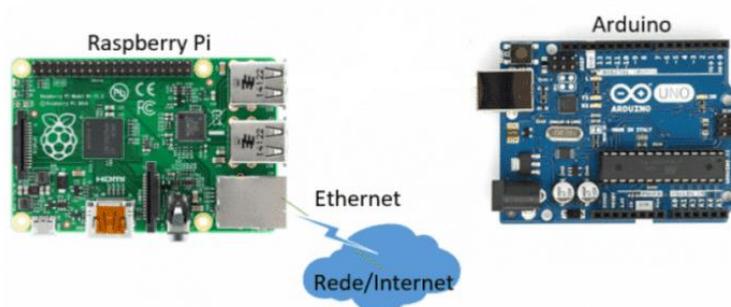


Figura 19. Interfaz ethernet.

3.1.4.2. Red Wifi

El Arduino uno no incluye el interfaz Wireless. Se debe adquirir aparte con un costo extra para habilitar la conexión hacia la Internet.

La Raspberry Pi 4 incluye dentro de su placa la capacidad de conectarse a Internet porque tiene la interfaz Wireless.

3.1.5. Base de Datos

Es el conjunto de datos los cuales están relacionados y organizados entre sí, con el fin de proporcionar de una manera más fácil al usuario la información que

necesita, consta de estructuras lógicas llamadas tablas que en su interior se encuentran columnas y registros.

Entre las principales características de una base de datos tenemos:

- Auditoría y seguridad en el acceso.
- Integridad de los datos.
- Varios usuarios pueden tener acceso concurrente.
- Optimización de consultas complejas.
- Mínima redundancia.
- Recuperación y respaldo.
- Los datos van a tener independencia física y lógica.

Las bases de datos constan de un sistema de gestión de base de datos (SGBD), que tiene como objetivo primordial el proporcionar entornos eficientes y convenientes a la misma vez permitiendo almacenar y extraer información de la base de datos, este debe mantener la información segura en cuanto a intentos de accesos no autorizados o caídas del sistema.

Para la realización de una base de datos tenemos que tomar en cuenta dos diseños los cuales son:

- Diseño conceptual: En este diseño recopilamos toda la información de lo que el cliente necesita almacenar en su base de datos, para ello se realiza un modelo entidad/relación que entre sus elementos que lo conforman tenemos:
 - Entidades: Es el tipo de objeto sobre el que se recoge información, por ejemplo: Persona, Ciudad.
 - Relación entre entidades: Es la asociación entre una o más entidades.
 - Atributos: Son las diferentes características que tiene una entidad, por ejemplo: nombre, apellido, cedula.
- Diseño Lógico: En este diseño se construye un esquema de la información que se obtuvo en el modelo conceptual, para esto se debe tomar en cuenta

la normalización que es una técnica que se utiliza para asegurar las relaciones que se adquirieron entre tablas no tengan datos redundantes.

Para este diseño se debe tomar en cuenta el tipo de clave que va a tener nuestras tablas, entre ellas tenemos:

- Clave Primaria (Primary Key): Es el identificador de la entidad, se representa como una llave o pk, por ejemplo: Pelicula(entidad) y su clave primaria podría ser id_Pelicula.
- Clave Foránea (foreign key): Se la utiliza para unir tablas, esta es la clave primaria de una tabla que migra a otra tabla.

3.1.5.1. Local

Es local, cuando sólo un usuario en un equipo la puede manipular permitiendo tener uniformidad en los términos de seguridad para cada uno de los usuarios en cuanto al acceso, además se evita la redundancia de la información.

3.1.5.2. Nube

Para gestionar y almacenar datos en la nube no se requiere una infraestructura local grande (memoria de almacenamiento y RAM) debido a que el proveedor de servicio en la nube se encarga de cubrir todas esas necesidades además de brindar el entorno, disponibilidad y mantenimiento.

Como principales ventajas de una base de datos en la nube se tiene: una instantánea escalabilidad, el proveedor puede brindar una especializada experiencia a los clientes, La infraestructura física se elimina y un ahorro notable en costos debido a la competencia en el mercado.

En la nube existen 3 capas las cuales son:

- Software como servicios.
- Plataforma como servicio.
- Infraestructura como servicio.

4. IMPLEMENTACIÓN DE UN PROTOTIPO QUE PERMITA CONTROLAR Y OPTIMIZAR EL CONSUMO ELÉCTRICO DE UNA PLANTA TRATAMIENTO DE AGUAS RESIDUALES.

El capítulo se basa en realizar la implementación del prototipo para ello se realizó como parte primordial el respectivo diseño con la finalidad de poder demostrar cómo es la infraestructura y funcionamiento de una planta de tratamiento aparte se indicará donde estarían ubicados los sensores los cuales son: concentración de oxígeno y de consumo eléctrico.

A. Diagrama de flujo del Arduino.

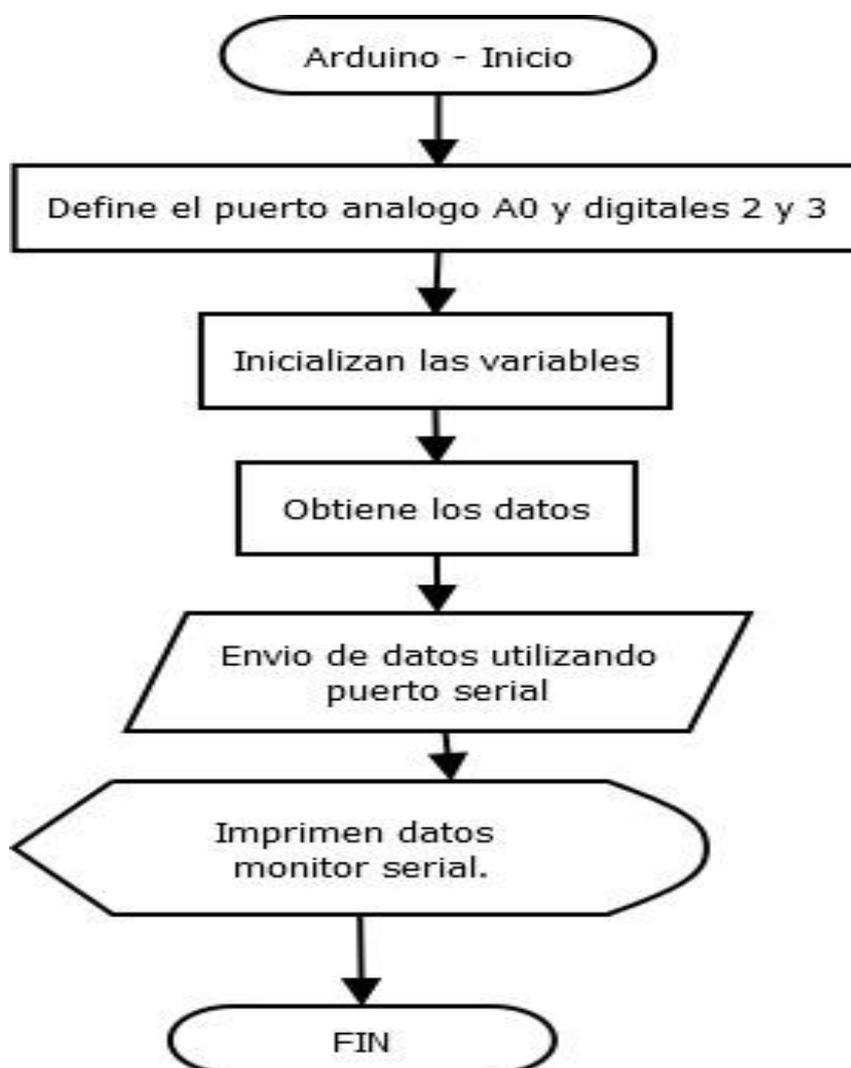


Figura 20. Recolección de datos sensores

B. Diagramas de flujo de la Raspberry Pi 4.

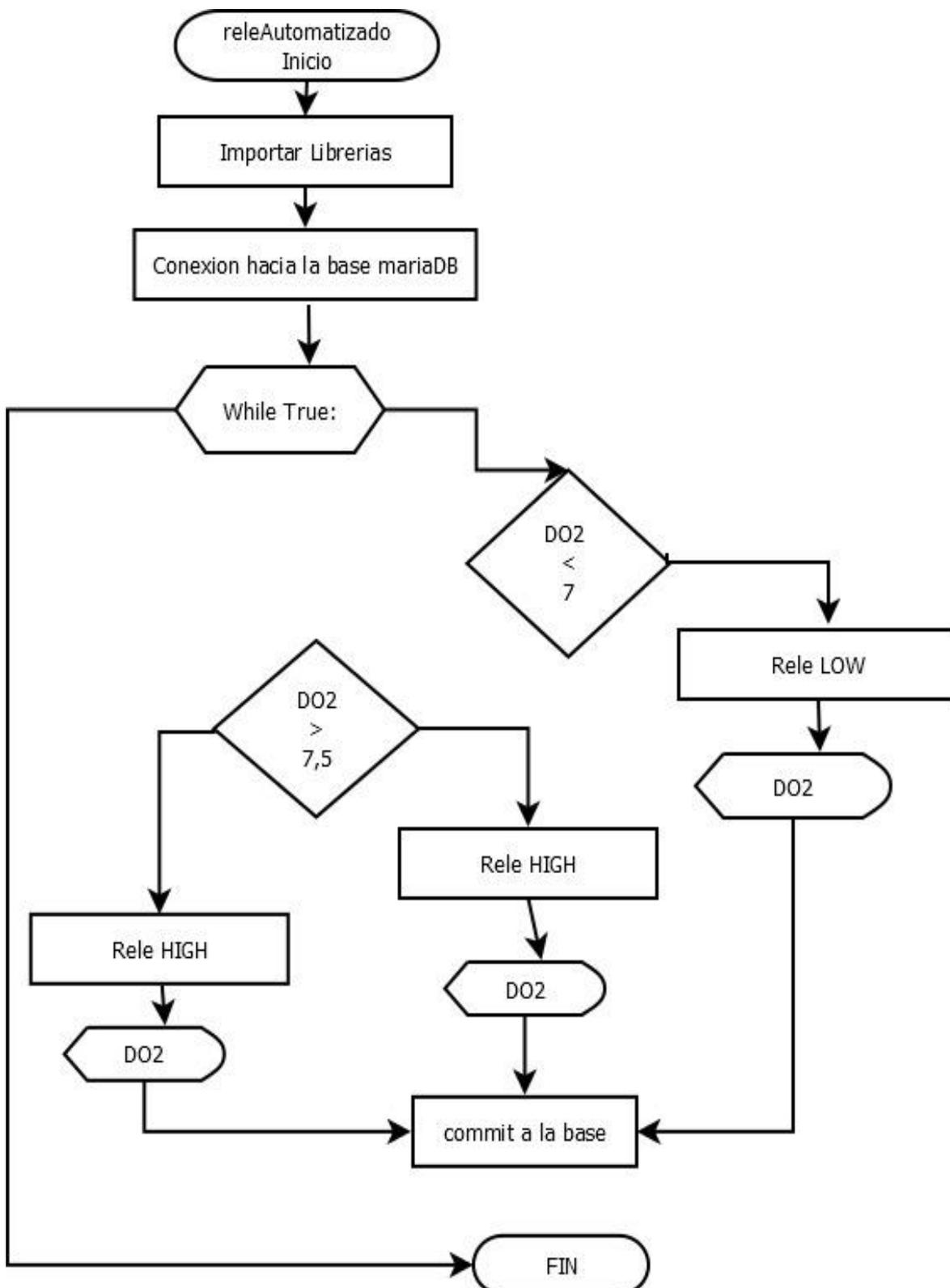


Figura 21. Diagrama de flujo consumo eléctrico

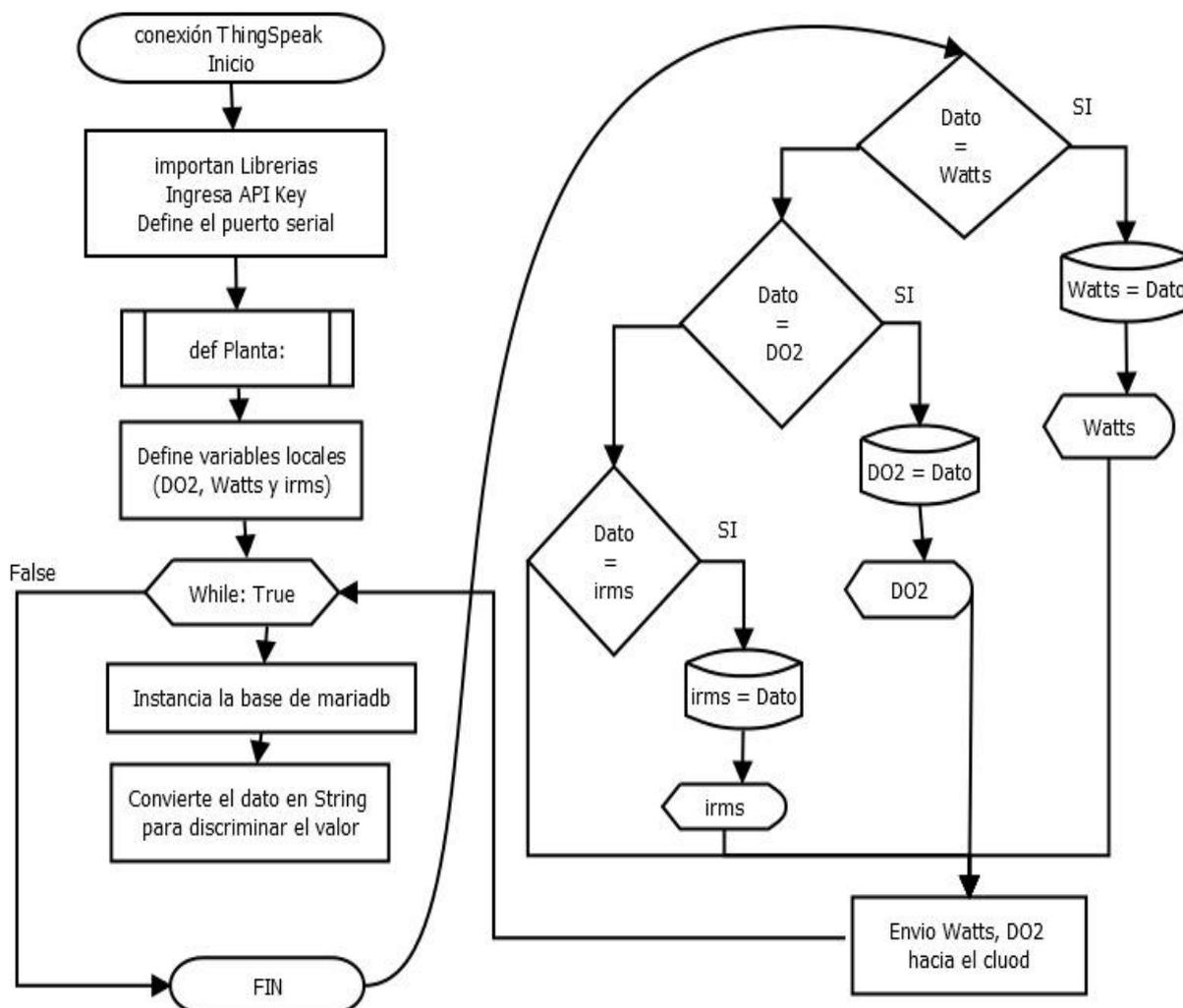


Figura 22. Diagrama de flujo publicación y almacenamiento de datos.

4.1. Diseño Físico: funcionamiento de la planta (maqueta)

La maqueta va a constar de 3 tanques funcionales los cuales van a cumplir con el funcionamiento normal de una planta de tratamiento cada tanque tendrá su respectiva funcionalidad para así nosotros poder tomar los datos correspondientes con los sensores.

El primer tanque de la Figura 23, realiza un proceso de filtrado del agua residuales, se utiliza este tipo de agua ya que esta es la que no tiene ningún tratamiento además por lo general se encuentra en alcantarillas, dentro de este tanque se encontrará una bomba de agua la cual nos brindará el impulso necesario para que el agua que se encuentra ya filtrado pase al segundo tanque a través de un tubo de plástico flexible que llegará solo hasta la parte superior.

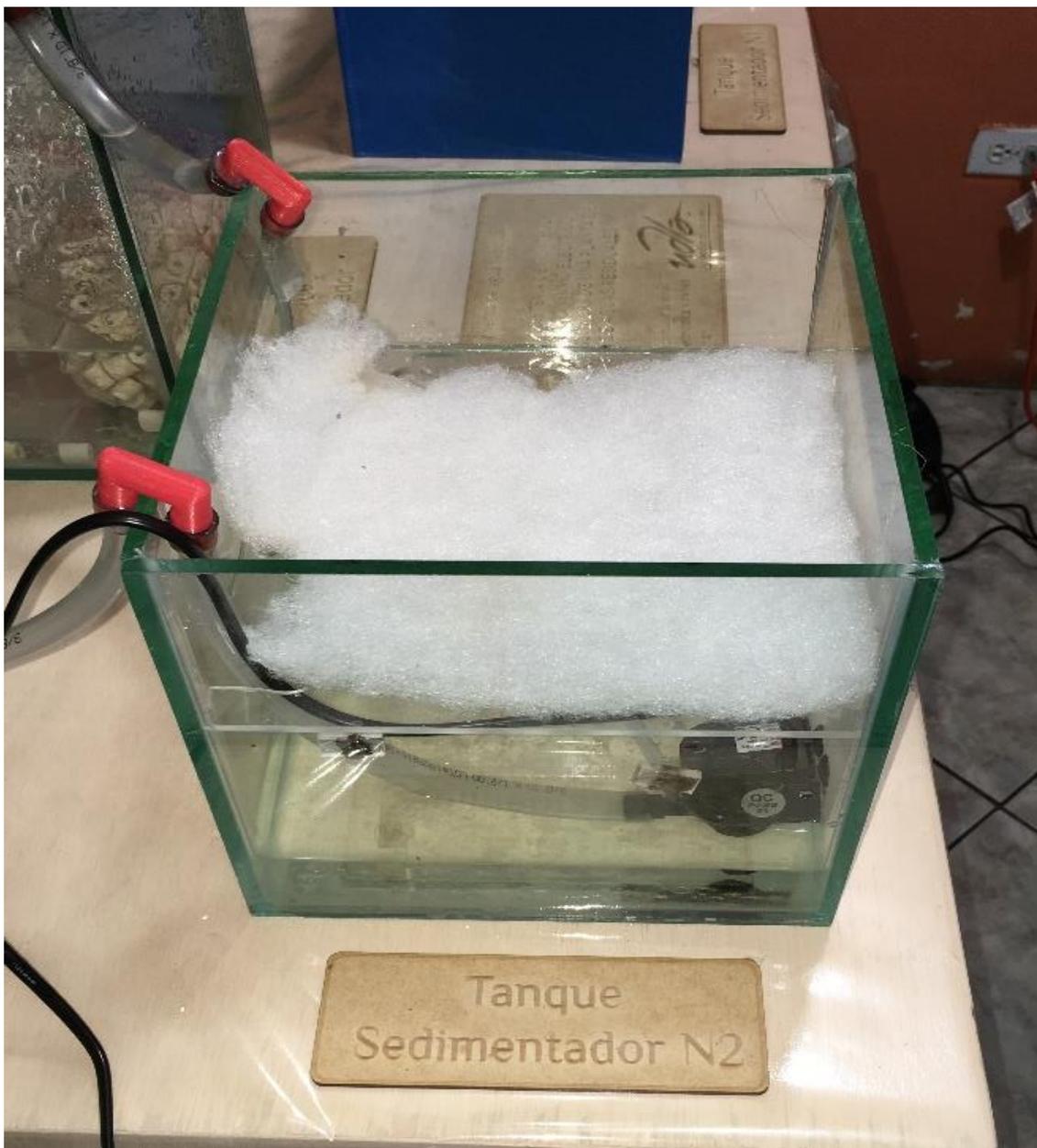


Figura 23. Tanque sedimentador funcional de la maqueta.

En el segundo tanque representada en la Figura 24, se encuentra el aireador, este va a expulsar aire en el interior del tanque generando movimiento y burbujas permitiendo oxigenar el agua, y mediante el sensor podremos adquirir los datos de concentración de oxígeno, además tendremos una bomba de agua que ayudará a trasladar el agua al tercer tanque a través de un tubo de plástico flexible que solo llegará hasta la parte superior.

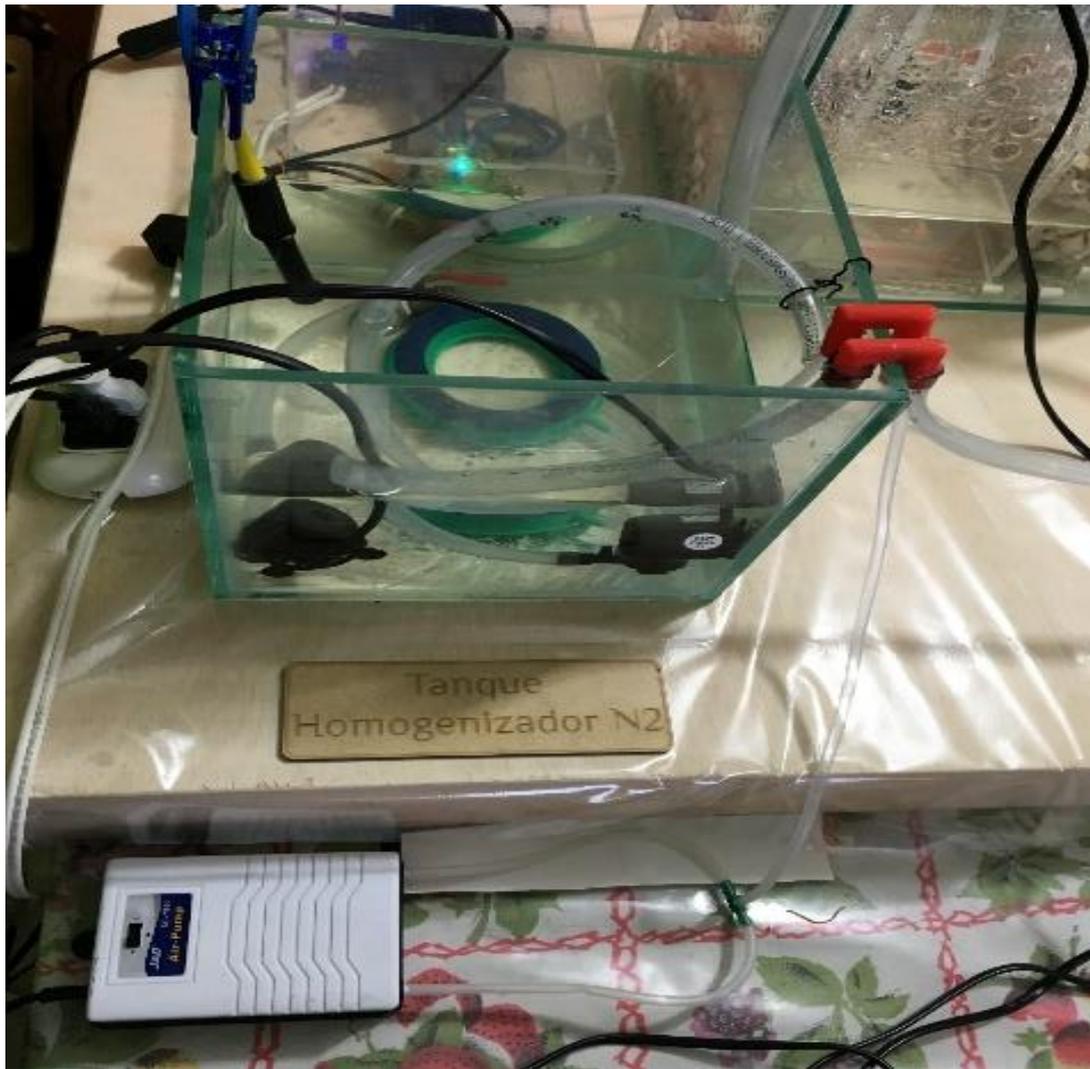


Figura 24. Tanque Homogenizador funcional de la maqueta.

El tercer tanque representado en la Figura 25, es el más importante debido a que es ahí donde estará el agua purificada y con oxigenación, mediante el proceso de filtración biológica a través de microorganismos podremos un agua reutilizable para cultivos y riegos, asimismo tendremos un tubo de plástico flexible que conectará con el primer tanque y se repetirá el ciclo.

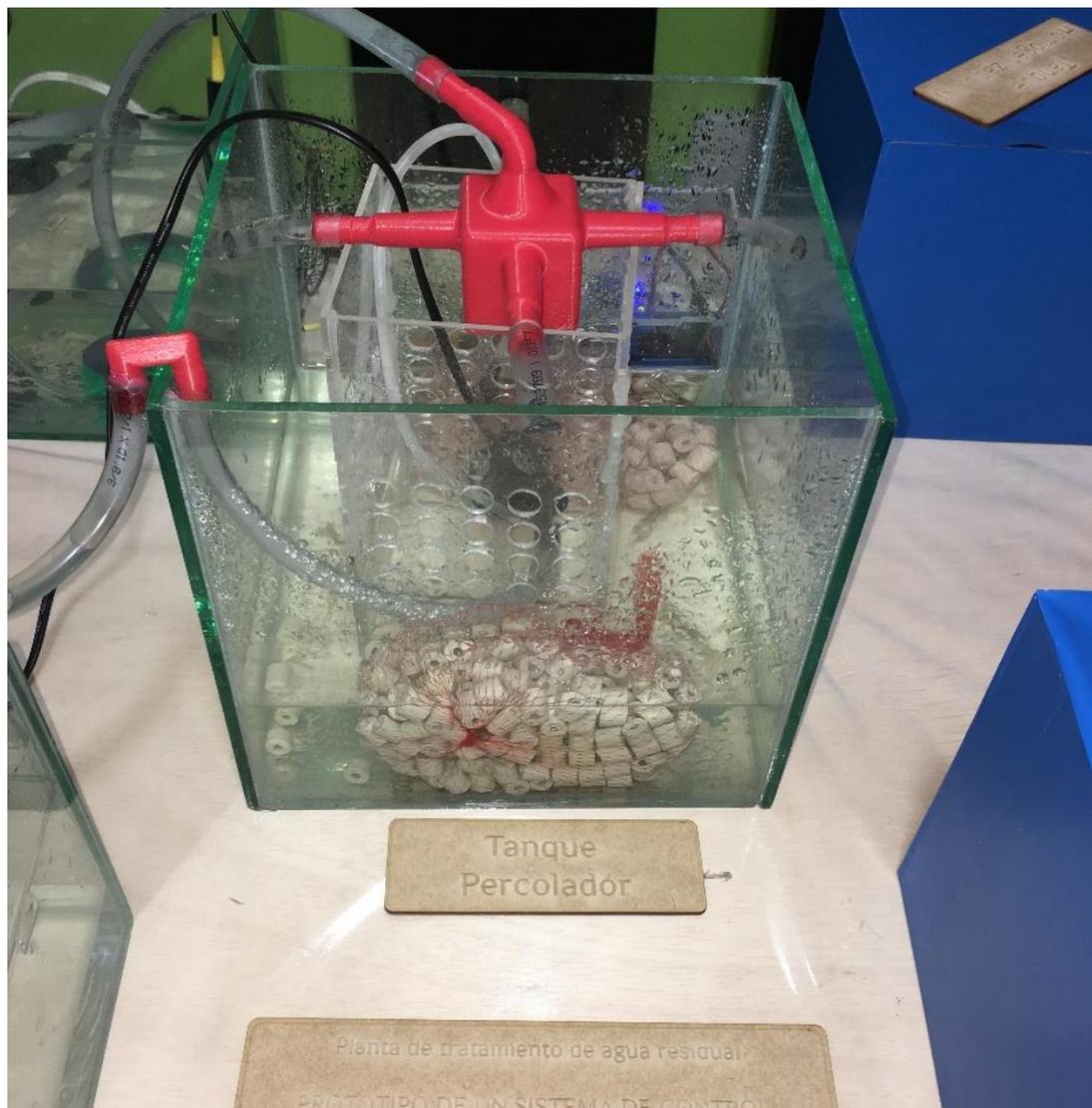


Figura 25. Tanque percolador de la maqueta.

En cuanto al consumo eléctrico de esta planta hay tomas de corriente separadas en grupos: bombas de agua, bombas de oxígeno. El sensor de consumo eléctrico se lo ubicó en el tomacorriente de la bomba de oxígeno con la finalidad de poder registrar los watts consumidos junto con un relé para su encendido y apagado.

4.2. Diseño electrónico: materiales utilizados

Para el diseño electrónico de acuerdo con el respectivo análisis antes realizado se tomaron las decisiones de trabajar con los siguientes materiales:

- Circuito electrónico del Sensor de oxígeno del fabricante AtlasScientific que se encuentra en la Figura 26, para más información visualizar el anexo 2.

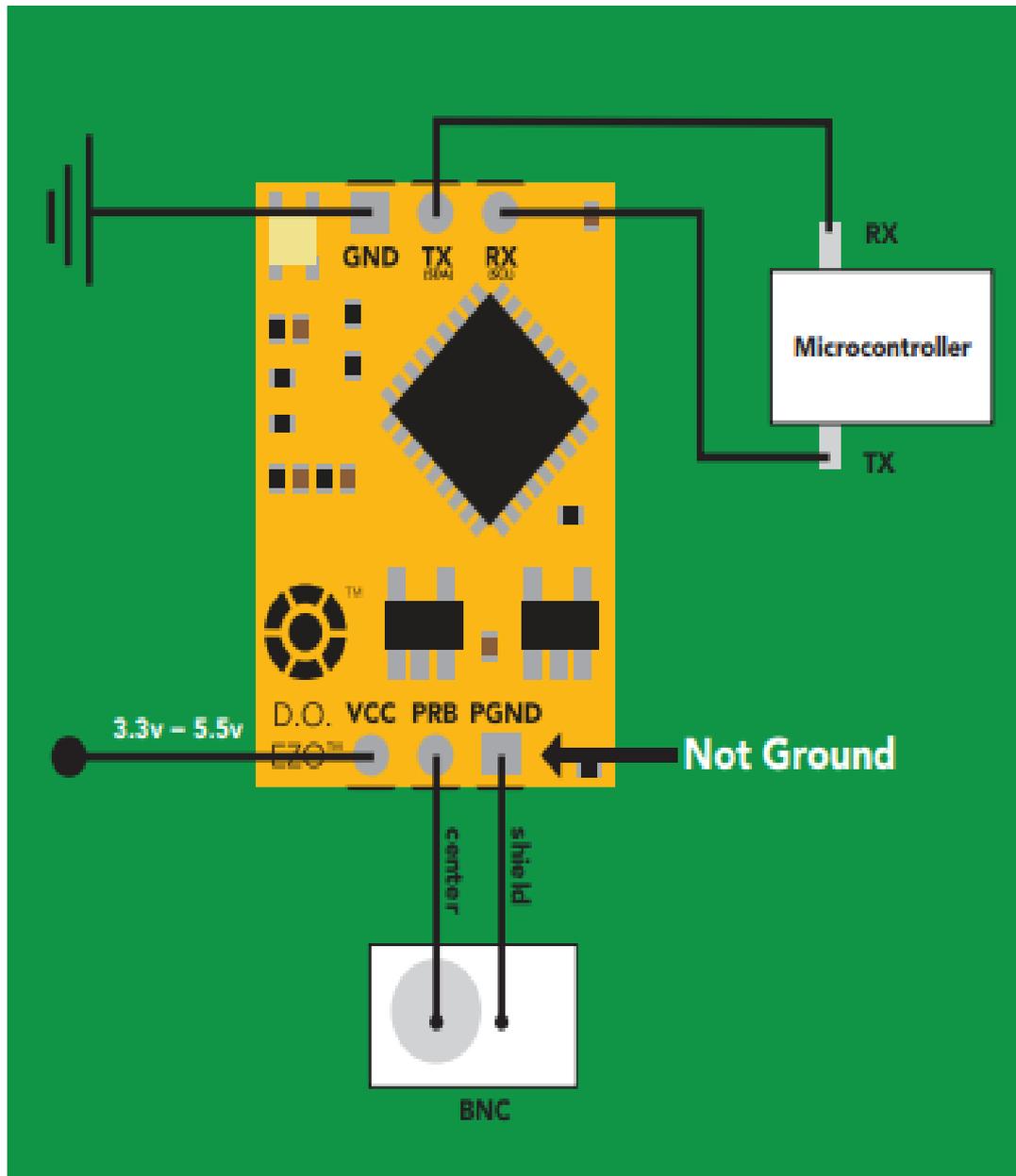


Figura 26. Circuito del Sensor de Oxígeno.

Tomado de (Deacutis, 2016).

- Circuito electrónico para la activación del Blower como se muestra en la Figura 27.

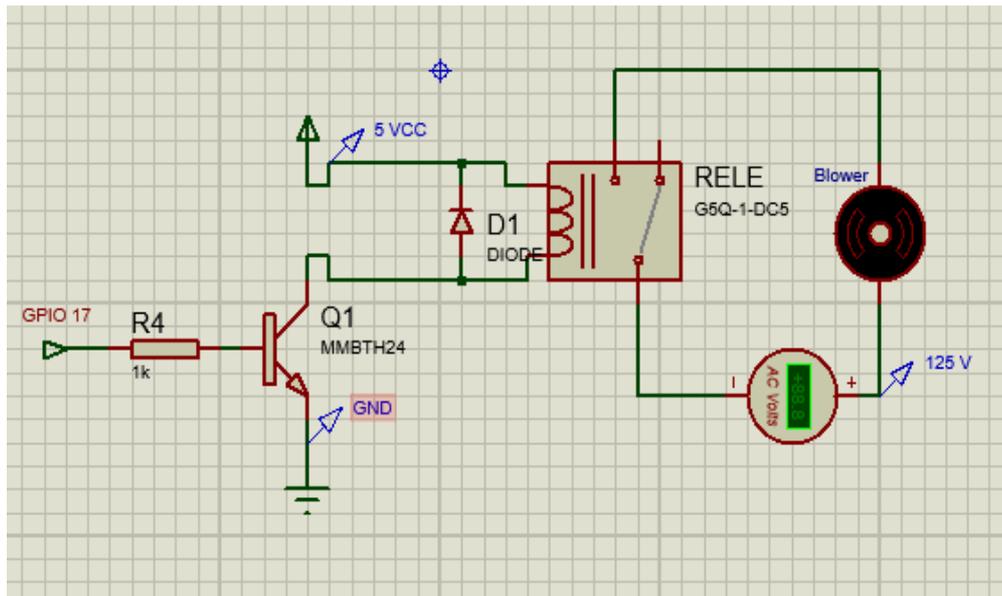


Figura 27. Diagrama realizado con el programa Proteus 8.

- La Figura 28 muestra el circuito electrónico de polarización para el sensor de corriente.

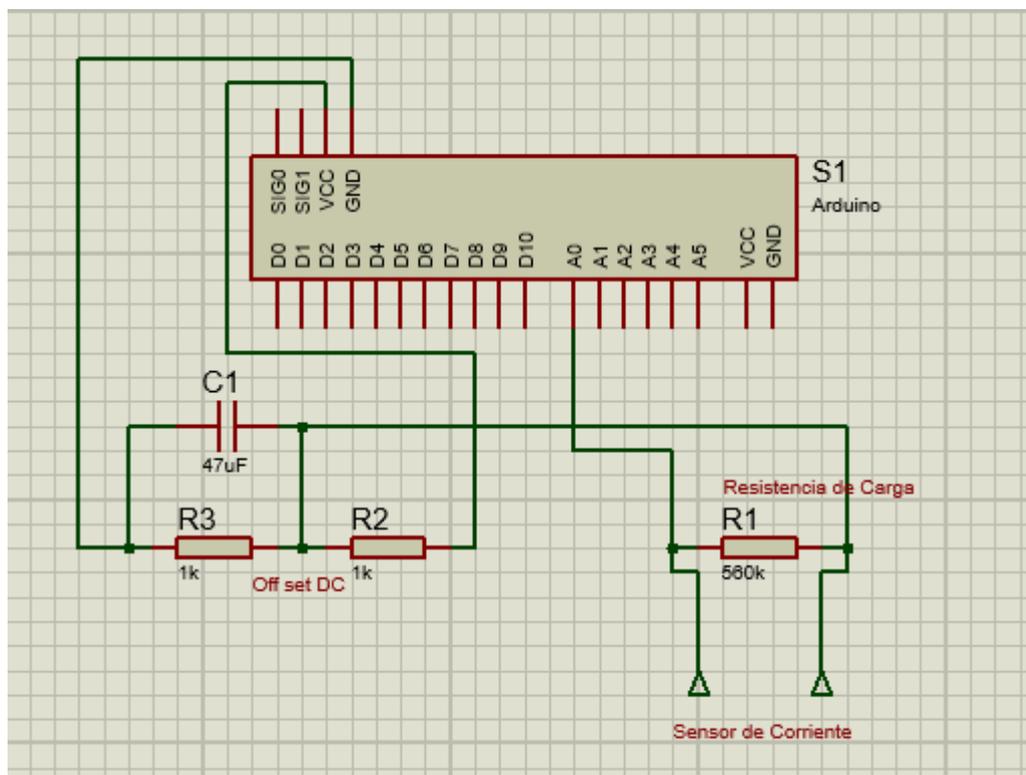


Figura 28. Realización del Diagrama en Proteus 8.

La figura 29 muestra los circuitos electrónicos anteriormente mencionados conectados respectivamente a sus puertos y en operatividad.

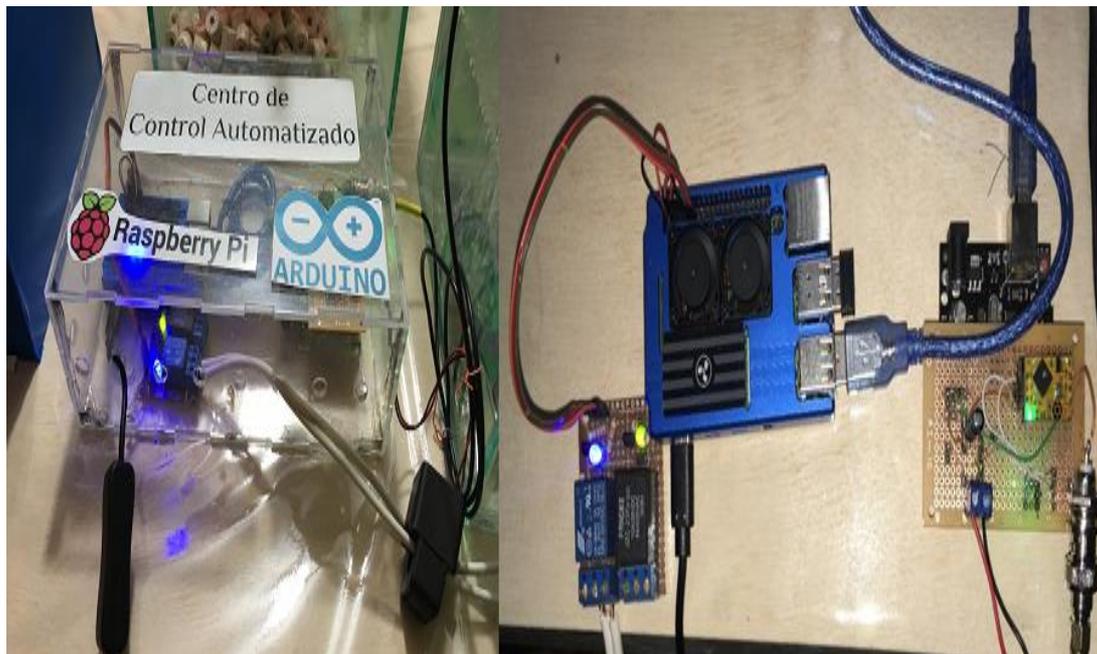


Figura 29. Elementos electrónicos del sistema.

4.3. Diseño Lógico: S.O código(parte)

Se decidió utilizar el Arduino para conectar el sensor de consumo eléctrico aprovechando la disponibilidad de los pines analógicos porque su principio de funcionamiento del sensor es devolver un valor en voltios hacia el microcontrolador del arduino; para el sensor de oxígeno disuelto se utilizó los pines digitales para imprimir los dos resultados por el puerto serial.

4.3.1. Código en Arduino

Se muestra partes importantes para la funcionalidad del script entre ellas tenemos las librerías utilizadas; el código completo se lo puede visualizar en el Anexo 3.

```
#include "EmonLib.h"
#include <SoftwareSerial.h>
#define rx 2
#define tx 3
```

```
SoftwareSerial myserial(rx, tx);
```

Se realizó un script para tener comunicación serial para ello usamos lo que es transmisión tx y recepción rx, se declararon varios tipos de variables de acuerdo con la función que van a realizar y se estableció el voltaje AC determinado.

```
SoftwareSerial myserial(rx, tx);
```

```
to work
```

```
EnergyMonitor energyMonitor;
```

```
float voltajeRed = 124.0;
```

```
String inputstring = "";
```

```
String sensorstring = "";
```

```
Scientific product
```

```
boolean input_string_complete = false;
```

```
boolean sensor_string_complete = false;
```

```
Atlas Scientific product
```

```
float DO;
```

```
is the Do
```

A continuación, se encuentra el script donde realizamos el cálculo para la obtención del número de muestras que deseamos, aparte de calcular la potencia y para finalizar para poder mostrar la información por nuestro monitor serial

```
if (sensor_string_complete == true) {
```

```
// Obtenemos el valor de la corriente eficaz
```

```
// Pasamos el número de muestras que queremos tomar
```

```
double Irms = energyMonitor.calcIrms(1484);
```

```
// Calculamos la potencia aparente
```

```

double potencia = Irms * voltajeRed;

// Mostramos la informaci3n por el monitor serie
Serial.print("Watts = ");
Serial.println(potencia);
Serial.print("Irms = ");
Serial.println(Irms);
Serial.println("DO2 = "+sensorstring);
//send that string to the PC's serial monitor

```

4.3.2. Script en Raspberry para Obtenci3n, discriminaci3n, env3o y almacenamiento de datos obtenidos por el puerto serial.

Para programar algoritmos en Raspbian se utiliza el lenguaje de programaci3n Python y se puede acceder desde la terminal o el gestor Thonny adem3s, de inicializar la recepci3n de datos por el puerto serial.

```

#Libreria para thingSpeak
import urllib3
from urllib.parse import urlencode
#Libreria para fecha y transforamr datos
from decimal import Decimal
from datetime import date
from datetime import datetime
#Libreria para la conexi3n a la base de datos MariaDB
import mysql.connector as mariadb
import time
#Lectura de datos por el puerto serial
arduino = serial.Serial('/dev/ttyACM0',9600)

```

Para tener comunicación con la plataforma ThingSpeak se utiliza una clave la cual se mostrará a continuación:

```
key = "IAA97MYBRMW3WVJJ" # API Key ThingSpeak
```

Seguidamente se levanta la conexión con la base de datos y se instancia la base para manipulación.

```
#Conexion hacia Mariadb
mariadb_connection = mariadb.connect(host='localhost', user='user1',
password='secret', database='sensor')
cursor = mariadb_connection.cursor()
#Obtencion de Datos
dato = arduino.readline()
```

Seguido, se muestra el proceso realizado para discriminar y obtener los datos enviados por el puerto serial para luego ser almacenados y visualizados local y remotamente mediante un script.

```
#convertir a string
chare = str(dato)
#Comprobando si el dato es Oxigeno
if(chare.find("DO2 =") >= 0 and chare.find("DO2 =") >= 0):
#Discriminando el valor de Oxígeno
DO2_PURGUE = chare.find("DO2 =")+5
chareb = chare.find("DO2 =")+10
#Obteniendo y almacenando el valor de Oxígeno
DO2 = chare[DO2_PURGUE:chareb]
print("DO2 =",DO2)
```

Se muestra el script de los datos recolectados que se guardan tanto en la base local como remotamente.

```
#Guardando los datos en la base local de mariadb
sql = "INSERT INTO sensor.datos(fecha,oxigeno,watts,irms) values
(%s,%s,%s,%s)"
now = datetime.now()
#array de los datos
val = (now,DO2,WATTS,IRMS)#inserción de datos en la setencia SQL
cursor.execute(sql,val)
#Guardando la sesion
mariadb_connection.commit()
#Enviando datos hacia ThingSpeak
params = urlencode({'field1': DO2,'field2': WATTS, 'key':key })
headers = {"Content-type": "application/x-www-form-urlencoded","Accept":
"text/plain"}
conn = http.client.HTTPConnection("api.thingspeak.com:80")
try:
#Abriendo al conexión hacia ThingSpeak
conn.request("POST", "/update", params, headers)
#Estado de la conxión
response = conn.getresponse()
#print (Oxigeno)
print (response.status, response.reason)
data = response.read()
#cerrando la conexión
conn.close()
except:
print ("connection failed")
break
```

Cuando se envía los datos hacia la plataforma de thingspeak; La Figura 30 muestra cómo se puede imprimir dicho estado para conocer si él envió es

exitoso. El envío de datos puede no ser exitoso por el tipo de plan contratado para thingspeak o debido al proveedor de internet contratado donde está alojado el dispositivo. El script completo "conexiónThingSpeak.py" se lo puede encontrar en el Anexo 4.

```
200 OK
DO2 = 6.91
200 OK
Watts = 1.47
200 OK
```

Figura 30. Datos obtenidos y estado del envío de datos.

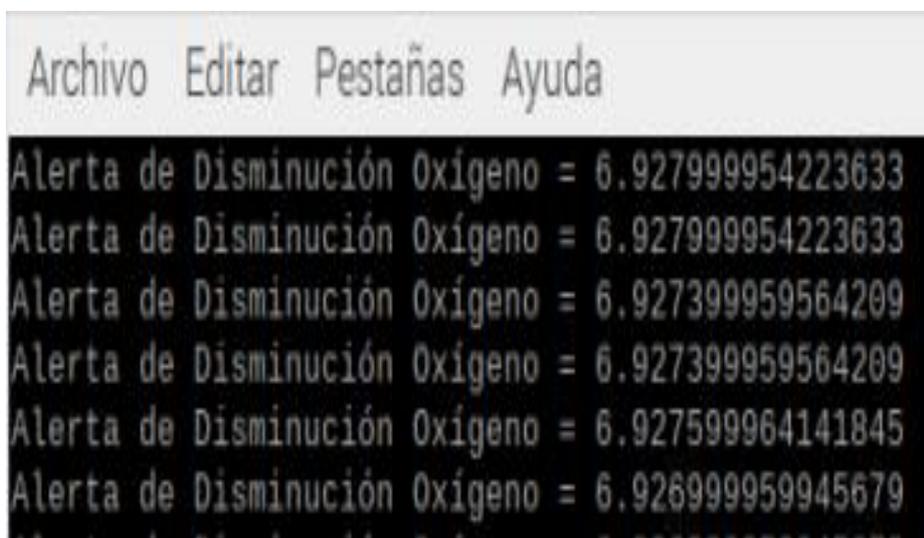
4.3.3. Script en Raspbian para control de consumo eléctrico.

Se consulta directamente hacia la base de datos los primeros 250 registros ordenados desde el más actual hacia el más antiguo para luego obtener el promedio de la concentración de oxígeno disuelto para ser evaluado con la tabla 1 de niveles de oxígeno disuelto, por si debe activar o desactivar el relé de la oxigenación durante el periodo de 1 hora.

```
cursor.execute("truncate table tmp_convert")
cursor.execute("insert into tmp_convert select cast(oxigeno as float) as DO2
from datos where oxigeno not in ('') and oxigeno not like '%s%' and oxigeno not
like '%xa%' order by fecha desc limit 100")
cursor.execute("select avg(DO2) as promDO2 from tmp_convert")
prom = cursor.fetchone()
datotr = str(prom)
dato = datotr[1:datotr.find(",")]
#print(dato)
if (float(dato) <= 7):
#Activa el rele para encender la oxigenación
```

```
print("Alerta de Disminución Oxígeno =",dato)
#GPIO.output(17,GPIO.LOW)
GPIO.output(17,GPIO.LOW)
#time.sleep(1)
elif (float(dato) > 7.5):
#Desactiva el rele de la oxigenación
print("Concentración de oxígeno ACEPTABLE =",dato)
GPIO.output(17,GPIO.HIGH)
time.sleep(3600)
else:
print("Ok",dato)
GPIO.output(17,GPIO.HIGH)
#time.sleep(15)
```

Cuando se ejecuta el script los procesos son imperceptibles gracias a las capacidades de hardware, mostrando los resultados obtenidos del cálculo en la Figura 31 en vivo paralelamente con el script de envío de datos.



```
Archivo  Editar  Pestañas  Ayuda
Alerta de Disminución Oxígeno = 6.927999954223633
Alerta de Disminución Oxígeno = 6.927999954223633
Alerta de Disminución Oxígeno = 6.927399959564209
Alerta de Disminución Oxígeno = 6.927399959564209
Alerta de Disminución Oxígeno = 6.927599964141845
Alerta de Disminución Oxígeno = 6.926999959945679
```

Figura 31. Resultados del script releAutomatizado.py.

El script completo “releAutomatizado.py” se lo puede encontrar en el Anexo 5.

5. Solución de almacenamiento, visualización y pruebas del sistema.

En el presente capítulo se detalla cómo fue el proceso de almacenar los datos de los sensores y poder visualizarlos en una plataforma web que brinda servicios de API para luego proceder a realizar las respectivas pruebas de funcionalidad completa del sistema permitiéndonos conocer cuánto tiempo se tardan los datos al momento de ser tomados, luego almacenados y para finalizar visualizados; para ello se tomaron algunos aspectos que se verán en este capítulo.

Para el almacenamiento se decidió crear una base de datos local en la Raspberry Pi 4, con la finalidad de verificar si los datos que transmitían los sensores eran los correctos. En el Arduino para la conexión y obtención de datos, el sensor de oxígeno el cual cuenta con un circuito de polarización y transformación de los datos obtenidos en milivoltios hacia valores de 1 a 2,5 V; se realizaron las siguientes conexiones: a tierra (GND), utilizamos un voltaje de 5, los pines dos y tres de la parte digital, transmisión (Tx), recepción (Rx).

Para la conexión entre el sensor de consumo eléctrico y el Arduino se utilizó la entrada analógica A0, estas tienen un rango de valores de 0 a 1024; donde el 0 no tiene ningún voltaje y 1024 equivale a 5 voltios, debido a que el sensor es susceptible al campo electromagnético de una fase de corriente; actuando como si fuera una resistencia por ende se necesita un circuito de polarización utilizando puentes de diodos, capacitores y tomando en cuenta el voltaje AC (Ecuador 120 V) para el cálculo de la resistencia de carga.

Para la obtención de la base de datos local se ingresaron varios comandos del sistema operativo Linux y se siguieron los siguientes pasos:

1. Entramos al terminal
2. Se descarga el Mariadb con sus respectivos paquetes y versión actual se puede visualizar en la Figura 32.

```

pi@raspberrypi: ~
Archivo Editar Pestañas Ayuda
pi@raspberrypi:~ $ sudo apt-get install mariadb-server

pi@raspberrypi: ~
Archivo Editar Pestañas Ayuda
pi@raspberrypi:~ $ sudo apt-get install mariadb-client

pi@raspberrypi: ~
Archivo Editar Pestañas Ayuda
pi@raspberrypi:~ $ sudo apt-get install mariadb-plugin-connect

```

Figura 32. Mariadb y paquetes.

3. Se crea una base de datos (sensor) que contiene la tabla (datos) dentro de Mariadb.
4. Dentro de la tabla creamos atributos que van a hacer referencia al sensor de oxígeno, consumo eléctrico, fecha del registro y el respectivo Id que va a ir incrementándose como se observa en la Figura 33.

```

bash: sudomariadb: orden no encontrada
pi@raspberrypi:~ $ sudo mariadb -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 155138
Server version: 10.3.22-MariaDB-0+deb10u1 Raspbian 10

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create database sensor;
MariaDB [(none)]> use sensor;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [sensor]> create table datos(id BIGINT(20) PRIMARY KEY AUTO_INCREMENT, fecha datetime, oxigeno varchar(6), watts varchar(6), irms varchar(6));

```

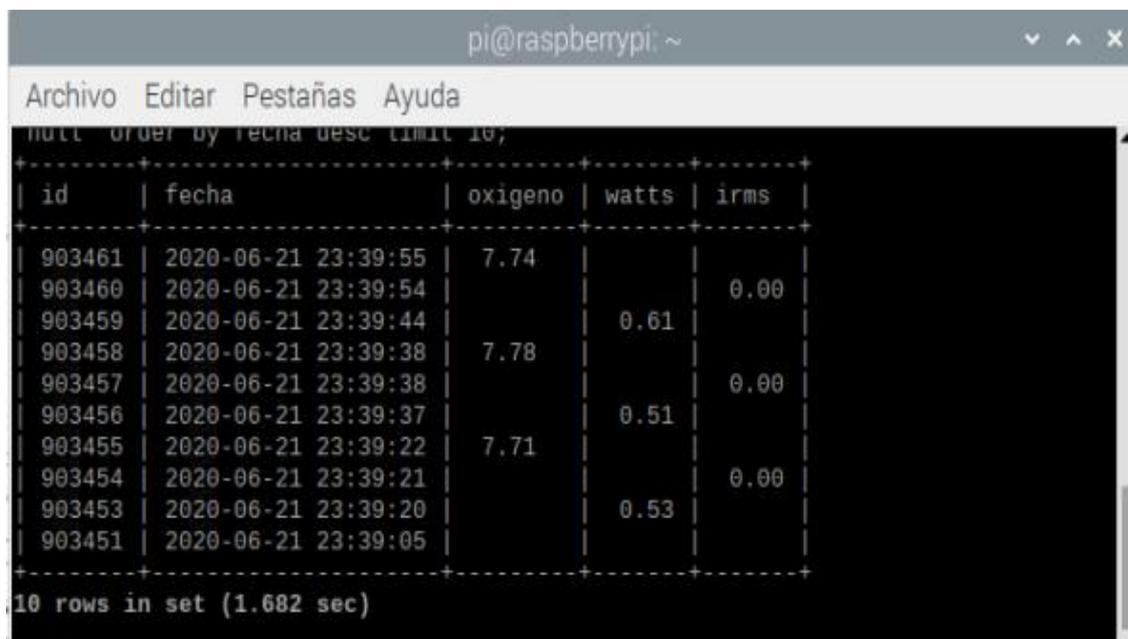
Figura 33. Sentencias SQL para crear bases y tablas.

5. Insertamos un comando para receptor los datos provenientes de los sensores a la tabla como apreciamos en la Figura 34.

```
#Guardando los datos en la base local de mariadb
sql = "INSERT INTO sensor.datos(fecha,oxigeno,watts,irms) values (%s,%s,%s,%s)"
now = datetime.now()
#array de los datos
val = (now,DO2,WATTS,IRMS)
#inserción de datos en la setencia SQL
cursor.execute(sql,val)
#Guardando la sesion
mariadb connection.commit()
```

Figura 34. Se insertan los datos en los campos correspondientes.

6. Para finalizar en la Figura 35 ingresamos un comando para que podamos visualizar los datos de manera local.



```
select * from datos order by fecha desc limit 10;
```

| id | fecha | oxigeno | watts | irms |
|--------|---------------------|---------|-------|------|
| 903461 | 2020-06-21 23:39:55 | 7.74 | | |
| 903460 | 2020-06-21 23:39:54 | | | 0.00 |
| 903459 | 2020-06-21 23:39:44 | | 0.61 | |
| 903458 | 2020-06-21 23:39:38 | 7.78 | | |
| 903457 | 2020-06-21 23:39:38 | | | 0.00 |
| 903456 | 2020-06-21 23:39:37 | | 0.51 | |
| 903455 | 2020-06-21 23:39:22 | 7.71 | | |
| 903454 | 2020-06-21 23:39:21 | | | 0.00 |
| 903453 | 2020-06-21 23:39:20 | | 0.53 | |
| 903451 | 2020-06-21 23:39:05 | | | |

10 rows in set (1.682 sec)

Figura 35. Datos almacenados localmente.

La comunicación entre el Arduino y la Raspberry se da mediante el puerto serial ya que estos dos lo tienen, pero la raspberry no tiene pines analógicos como consecuencia se optó por usar el Arduino para la recolección de datos del sensor de consumo eléctrico.

La raspberry Pi 4 va a contener todos los scripts para poder comunicarnos, almacenarlos en la base de datos y enviarlos a la plataforma en la nube, esta plataforma llamada ThingSpeak brinda las facilidades necesarias para poder visualizar los datos.

5.1. Visualización en ThingSpeak

En esta plataforma vamos a poder visualizar los datos obtenidos por nuestros sensores permitiéndonos conocer cómo es su desenvolvimiento en el transcurso de los días, en la Figura 36 se puede apreciar lo antes mencionado.



Figura 36. Gráficos para la visualización de datos.

En la plataforma en la sección de API keys se genera la llave única para el envío de datos al servicio API de thingspeak, la Figura 37 muestra lo antes mencionado.

The screenshot displays the 'API Keys' section of the Thingspeak interface. At the top, there is a navigation bar with five tabs: 'Private View', 'Public View', 'Channel Settings', 'Sharing', and 'API Keys'. The 'API Keys' tab is selected. Below the navigation bar, the heading 'Write API Key' is visible. Under this heading, there is a 'Key' label next to a text input field containing the alphanumeric string 'IAA97MYBRMW3WVJJ'. Below the input field is an orange button labeled 'Generate New Write API Key'. Further down, the heading 'Read API Keys' is visible. Under this heading, there is a 'Key' label next to a text input field containing the alphanumeric string 'E9VKOTISPZQJ7XWJ'. Below this input field is a 'Note' label next to a larger, empty text area. At the bottom of the 'Read API Keys' section, there are two buttons: a green button labeled 'Save Note' and a red button labeled 'Delete API Key'.

Figura 37. Generación del API Key.

1. Se configura la privacidad del canal en la opción sharing como se muestra en la Figura 38, para este caso va a ser público

Monitoreo de Concentración de Oxígeno

Channel ID: 1068842
 Author: mwa0000018612770
 Access: Public

Monitor de Concentración de Oxígeno en los tanques de oxigenación para optimizar el consumo eléctrico del Blower.

concentración de oxígeno, udla, telecomunicaciones

Private View Public View Channel Settings **Sharing** API Keys Data Import / Export

Channel Sharing Settings

- Keep channel view private
 Share channel view with everyone
 Share channel view only with the following users:

Email Address

| Email Address | Shared On | Delete |
|------------------------|------------|----------------------------------|
| miltiniorc@hotmail.com | 2020-05-27 | <input type="button" value="X"/> |

Help

ThingSpeak allows you to control who can view the data in your channel. Irrespective of the settings on this tab, reading data from or writing data to the fields of a channel requires the appropriate API key for the channel.

Channel Sharing Settings

- Keep channel view private: Selecting this option keeps your channel private. Only you will be able to see the channel view.
- Share channel view with everyone: Selecting this option makes the public view of your channel viewable by anyone browsing the ThingSpeak website.
- Share channel view only with the following users: Selecting this option shares the private view of your channel only with specific ThingSpeak users.

Figura 38. Configuración para que el público pueda ver los datos.

- Para finalizar los datos que se están visualizando se puede descargar en un solo archivo ingresando a DataImport/Export como se aprecia en la Figura 39, el formato más común es csv.

Private View Public View Channel Settings Sharing API Keys **Data Import / Export**

Import

Upload a CSV file to import data into this channel.

File No se eligió archivo

Time Zone ▼

Export

Download all of this Channel's feeds in CSV format.

Time Zone ▼

Help

Import

The correct format for data import is provided in this [CSV Import Template File](#). Use the field names `field1`, `field2`, and so on, instead of custom field names.

CSV Import Format

```
datetime,field1,field3,field4,field8,elevation
2019-01-01T10:11:12-05:00,11,33,44,88,10
```

Other Import and Export Options

You can also use [MATLAB](#), the [REST API](#), or the [MQTT API](#) to import and export channel data.

[Read Data](#)
[Write Data](#)

Figura 39. Privacidad del canal.

5.2. Resultados

Para la obtención de resultados se dejó funcionar la maqueta por una semana para visualizar los cambios como: consumo eléctrico y concentración de oxígeno, la Figura 40 muestra los resultados de los datos obtenidos por los sensores, en la parte izquierda se encuentra la concentración de oxígeno (DO2) indicando cómo evoluciona en el transcurso de la semana, permitiendo conocer en qué tiempos la concentración de oxígeno aumenta, disminuye o se mantiene, si nos fijamos en la hora 19:48 antes existe un nivel de oxigenación grande llegando hasta 7.28 donde se mantiene por unos minutos teniendo un descenso de oxigenación leve.

En el lado derecho se encuentran los datos tomados por el sensor de consumo eléctrico donde se puede apreciar que antes de las 19:49 existe un aumento notable del consumo eléctrico que es un poco más de 2.4 Watts, cabe recalcar que cuando existe alta concentración de oxígeno nuestra bomba va a dejar de funcionar hasta que el nivel de oxígeno disminuya permitiendo así tener una optimización del consumo eléctrico.



Figura 40. Imagen Obtenida con la aplicación de ThingSpeak donde se observa el cambio de oxígeno disuelto comparado con el consumo eléctrico.

El oxígeno proporcionado cuando el blower está activo es suficiente para que las condiciones sean favorables; dichas condiciones pueden perdurar por el periodo de 1 hora o más dando como resultado un ahorro energético y ahorro económico, la Figura 41 muestra lo antes mencionado.

Además, el blower se activa por el periodo de 1 hora y cuando transcurre la hora realiza un promedio la concentración de oxígeno disuelto con los datos actuales para así saber si activan o desactivan el blower.

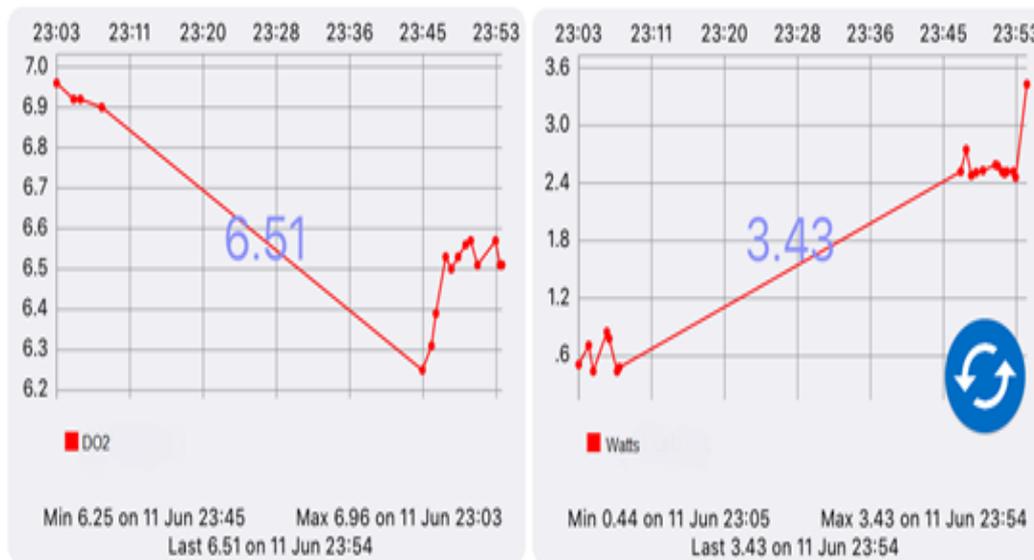


Figura 41. Cuando el blower es apagado los niveles de concentración de oxígeno comienzan a disminuir hasta niveles no favorables.

En la Figura 42 se observa el comportamiento de la concentración de oxígeno disuelto y consumo eléctrico por un periodo de 3 días donde pudimos notar que en el sensor de oxígeno cuando existen ejes altos es porque hay un aumento de oxigenación en él agua mientras, que cuando existen ejes hacia abajo quiere decir que hay una disminución en el nivel de oxígeno, en el caso del consumo eléctrico es similar cuando tiene un pico hacia arriba es porque el consumo eléctrico aumentó y cuando el pico es hacia abajo es porque hay una disminución en el consumo eléctrico, en cierto punto cuando existe un alto nivel de oxigenación nuestras bombas se apagan evitando tener un consumo eléctrico excesivo.

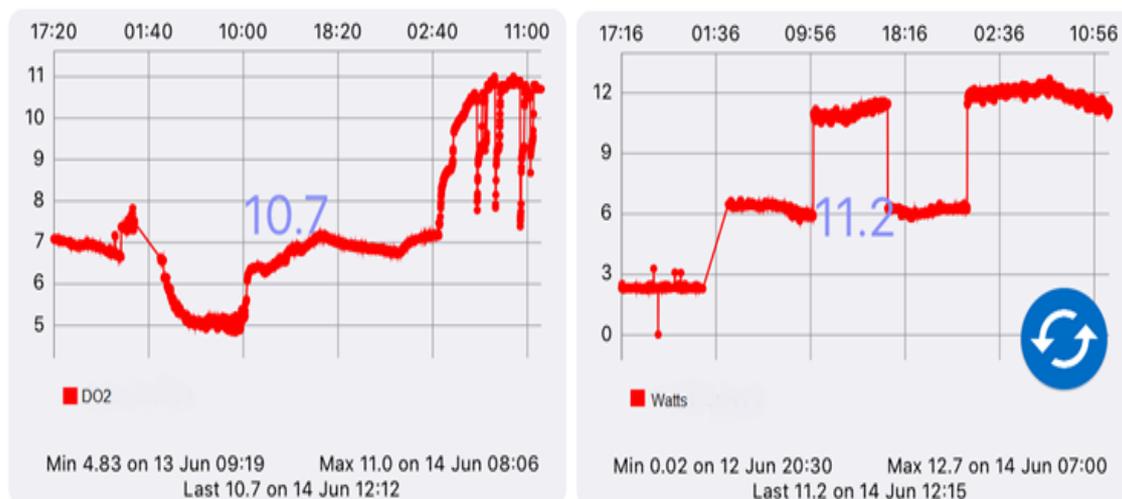


Figura 42. Evolución de los datos.

Las siguientes tablas que vienen a continuación muestran el seguimiento de los datos obtenidos por nuestros sensores en el lapso de un mes, en el cual tenemos como puntos principales: Fecha, Hora, entry_id y el dato censado. Se identificó que tenemos un total de 32687 registros en concentración de oxígeno.

Tabla 7.

Datos del sensor de concentración de oxígeno de 11 de junio del 2020

| Fecha | Hora | | entry_id | Dato |
|-----------|---------|-----|----------|------|
| 11/6/2020 | 7:55:37 | UTC | 1 | 6.12 |
| 11/6/2020 | 7:57:08 | UTC | 2 | 6.04 |
| 11/6/2020 | 7:57:54 | UTC | 3 | 6.05 |
| 11/6/2020 | 7:58:24 | UTC | 4 | 6.07 |
| 11/6/2020 | 7:58:54 | UTC | 5 | 6.04 |
| 11/6/2020 | 8:00:10 | UTC | 6 | 6.15 |

| | | | | |
|-----------|---------|-----|----|------|
| 11/6/2020 | 8:00:55 | UTC | 7 | 6.06 |
| 11/6/2020 | 8:01:25 | UTC | 8 | 6.15 |
| 11/6/2020 | 8:01:55 | UTC | 9 | 5.88 |
| 11/6/2020 | 8:02:10 | UTC | 10 | 5.89 |
| 11/6/2020 | 8:02:25 | UTC | 11 | 5.75 |
| 11/6/2020 | 8:03:26 | UTC | 12 | 5.73 |

Tabla 8.

Datos del sensor de concentración de oxígeno de 17 de junio del 2020

| Fecha | Hora | | entry_id | Dato |
|-----------|----------|-----|----------|------|
| 17/6/2020 | 23:06:13 | UTC | 10151 | 6.93 |
| 17/6/2020 | 23:07:00 | UTC | 10152 | 6.94 |
| 17/6/2020 | 23:08:14 | UTC | 10153 | 6.96 |
| 17/6/2020 | 23:09:39 | UTC | 10154 | 6.98 |
| 17/6/2020 | 23:10:00 | UTC | 10155 | 6.93 |
| 17/6/2020 | 23:10:27 | UTC | 10156 | 6.99 |
| 17/6/2020 | 23:11:33 | UTC | 10157 | 6.96 |
| 17/6/2020 | 23:11:49 | UTC | 10158 | 6.96 |
| 17/6/2020 | 23:12:10 | UTC | 10159 | 6.98 |

| | | | | |
|-----------|----------|-----|-------|------|
| 17/6/2020 | 23:12:38 | UTC | 10160 | 6.93 |
| 17/6/2020 | 23:13:31 | UTC | 10161 | 6.95 |
| 17/6/2020 | 23:13:47 | UTC | 10162 | 6.97 |

Tabla 9.

Datos del sensor de concentración de oxígeno de 30 de junio del 2020

| Fecha | Hora | | entry_id | Dato |
|-----------|----------|-----|----------|------|
| 30/6/2020 | 14:07:41 | UTC | 24455 | 4.76 |
| 30/6/2020 | 14:07:58 | UTC | 24456 | 4.85 |
| 30/6/2020 | 14:08:51 | UTC | 24457 | 4.87 |
| 30/6/2020 | 14:10:23 | UTC | 24458 | 4.86 |
| 30/6/2020 | 14:11:35 | UTC | 24459 | 4.78 |
| 30/6/2020 | 14:12:30 | UTC | 24460 | 4.84 |
| 30/6/2020 | 14:12:56 | UTC | 24461 | 4.74 |
| 30/6/2020 | 14:15:46 | UTC | 24462 | 4.81 |
| 30/6/2020 | 14:16:03 | UTC | 24463 | 4.74 |
| 30/6/2020 | 14:17:28 | UTC | 24464 | 4.73 |
| 30/6/2020 | 14:19:28 | UTC | 24465 | 4.75 |
| 30/6/2020 | 14:21:14 | UTC | 24466 | 4.68 |

Tabla 10.

Datos del sensor de concentración de oxígeno de 11 de julio del 2020

| Fecha | Hora | | entry_id | Dato |
|-----------|---------|-----|----------|------|
| 11/7/2020 | 5:04:10 | UTC | 38609 | 4.55 |
| 11/7/2020 | 5:04:41 | UTC | 38610 | 4.55 |
| 11/7/2020 | 5:05:12 | UTC | 38611 | 4.55 |
| 11/7/2020 | 5:05:32 | UTC | 38612 | 4.55 |
| 11/7/2020 | 5:07:33 | UTC | 38613 | 4.53 |
| 11/7/2020 | 5:07:49 | UTC | 38614 | 4.56 |
| 11/7/2020 | 5:09:52 | UTC | 38615 | 4.52 |
| 11/7/2020 | 5:11:09 | UTC | 38616 | 4.52 |
| 11/7/2020 | 5:11:24 | UTC | 38617 | 4.54 |
| 11/7/2020 | 5:11:55 | UTC | 38618 | 4.52 |
| 11/7/2020 | 5:12:10 | UTC | 38619 | 4.56 |
| 11/7/2020 | 5:12:25 | UTC | 38620 | 4.56 |

Seguido, en el consumo eléctrico se obtuvieron un total de 33054 registros, en las tablas que vienen a continuación se encuentra lo que es la fecha, hora, el entry_id que ocupa y el dato censado.

Tabla 11.

Datos del sensor consumo eléctrico de 11 de junio del 2020.

| Fecha | Hora | | entry_id | Dato |
|-----------|---------|-----|----------|------|
| 11/6/2020 | 7:55:52 | UTC | 1 | 2.55 |
| 11/6/2020 | 7:56:07 | UTC | 2 | 2.86 |
| 11/6/2020 | 7:56:23 | UTC | 3 | 2.60 |
| 11/6/2020 | 7:56:38 | UTC | 4 | 2.55 |
| 11/6/2020 | 7:56:53 | UTC | 5 | 2.52 |
| 11/6/2020 | 7:57:38 | UTC | 6 | 2.57 |
| 11/6/2020 | 7:59:09 | UTC | 7 | 2.78 |
| 11/6/2020 | 7:59:24 | UTC | 8 | 2.47 |
| 11/6/2020 | 7:59:55 | UTC | 9 | 2.45 |
| 11/6/2020 | 8:00:40 | UTC | 10 | 2.55 |
| 11/6/2020 | 8:01:10 | UTC | 11 | 0.59 |
| 11/6/2020 | 8:02:40 | UTC | 12 | 0.75 |

Tabla 12.

Datos del sensor consumo eléctrico de 17 de junio del 2020.

| Fecha | Hora | | entry_id | Dato |
|-----------|---------|-----|----------|------|
| 17/6/2020 | 3:29:26 | UTC | 16251 | 0.58 |

| | | | | |
|-----------|---------|-----|-------|------|
| 17/6/2020 | 3:29:53 | UTC | 16252 | 0.45 |
| 17/6/2020 | 3:30:40 | UTC | 16253 | 0.43 |
| 17/6/2020 | 3:31:59 | UTC | 16254 | 0.41 |
| 17/6/2020 | 3:32:51 | UTC | 16255 | 0.55 |
| 17/6/2020 | 3:33:49 | UTC | 16256 | 0.43 |
| 17/6/2020 | 3:35:24 | UTC | 16257 | 0.42 |
| 17/6/2020 | 3:36:12 | UTC | 16258 | 0.39 |
| 17/6/2020 | 3:36:42 | UTC | 16259 | 0.41 |
| 17/6/2020 | 3:37:34 | UTC | 16260 | 0.41 |
| 17/6/2020 | 3:38:06 | UTC | 16261 | 0.42 |
| 17/6/2020 | 3:38:33 | UTC | 16262 | 0.42 |

Tabla 13.

Datos del sensor consumo eléctrico de 30 de junio del 2020.

| Fecha | Hora | | entry_id | Dato |
|-----------|----------|-----|----------|------|
| 30/6/2020 | 20:57:00 | UTC | 25413 | 0.39 |
| 30/6/2020 | 20:57:46 | UTC | 25414 | 0.40 |
| 30/6/2020 | 20:58:07 | UTC | 25415 | 0.42 |

| | | | | |
|-----------|----------|-----|-------|------|
| 30/6/2020 | 21:04:48 | UTC | 25416 | 0.41 |
| 30/6/2020 | 21:05:34 | UTC | 25417 | 0.38 |
| 30/6/2020 | 21:06:45 | UTC | 25418 | 0.42 |
| 30/6/2020 | 21:07:06 | UTC | 25419 | 0.41 |
| 30/6/2020 | 21:13:00 | UTC | 25420 | 0.42 |
| 30/6/2020 | 21:13:18 | UTC | 25421 | 0.41 |
| 30/6/2020 | 21:14:09 | UTC | 25422 | 0.40 |
| 30/6/2020 | 21:14:46 | UTC | 25423 | 0.42 |
| 30/6/2020 | 21:17:02 | UTC | 25424 | 0.40 |

Tabla 14.

Datos del sensor consumo eléctrico de 11 de julio del 2020.

| Fecha | Hora | | entry_id | Dato |
|-----------|---------|-----|----------|------|
| 11/7/2020 | 5:38:46 | UTC | 32762 | 0.80 |
| 11/7/2020 | 5:42:02 | UTC | 32763 | 0.38 |
| 11/7/2020 | 5:42:51 | UTC | 32764 | 0.35 |
| 11/7/2020 | 5:44:18 | UTC | 32765 | 0.81 |
| 11/7/2020 | 5:45:54 | UTC | 32766 | 0.89 |
| 11/7/2020 | 5:46:39 | UTC | 32767 | 0.49 |

| | | | | |
|-----------|---------|-----|-------|------|
| 11/7/2020 | 5:46:55 | UTC | 32768 | 0.82 |
| 11/7/2020 | 5:47:10 | UTC | 32769 | 0.84 |
| 11/7/2020 | 5:47:42 | UTC | 32770 | 0.80 |
| 11/7/2020 | 5:47:57 | UTC | 32771 | 0.91 |
| 11/7/2020 | 5:48:42 | UTC | 32772 | 0.44 |
| 11/7/2020 | 5:49:30 | UTC | 32773 | 0.87 |

Como método didáctico se utilizó el software Matlab para mostrar de mejor manera los registros que nos da el Thinkspeak, tanto para el consumo eléctrico como para la concentración de oxígeno.

A continuación, se muestran las Figuras del consumo eléctrico realizadas en el software Matlab:

La Figura 43 muestra la hora exacta y los niveles de potencia (Watts) que ha alcanzado el consumo eléctrico en el prototipo, cabe recalcar que nuestro sistema recolectó datos de un mes para verificar como es el comportamiento de los ejes como se menciona en la Figura 42.

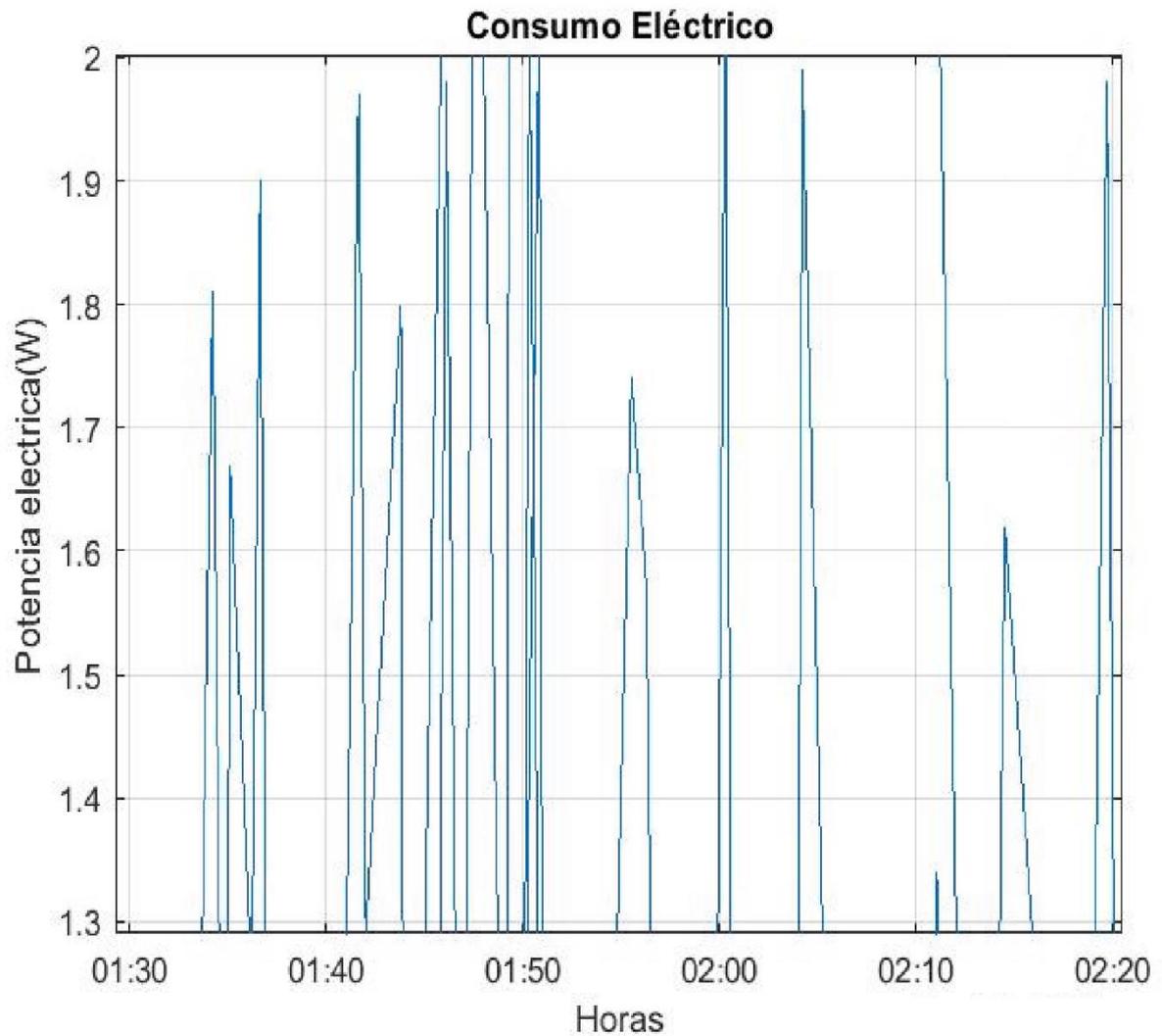


Figura 43. Comportamiento de la potencia eléctrica cuando el prototipo se encuentra en funcionamiento.

En la Figura 44, se muestra en diferentes horas los niveles de potencia eléctrica que obtuvo el sensor, permitiendo conocer si existió una disminución o elevación de consumo eléctrico, para ello tomamos una hora en específico la cual fue desde las 8:15 hasta las 9:15, determinando que en ese intervalo de tiempo existe un consumo eléctrico de 6.68 Wh, esto quiere decir que hubo una notoria disminución.

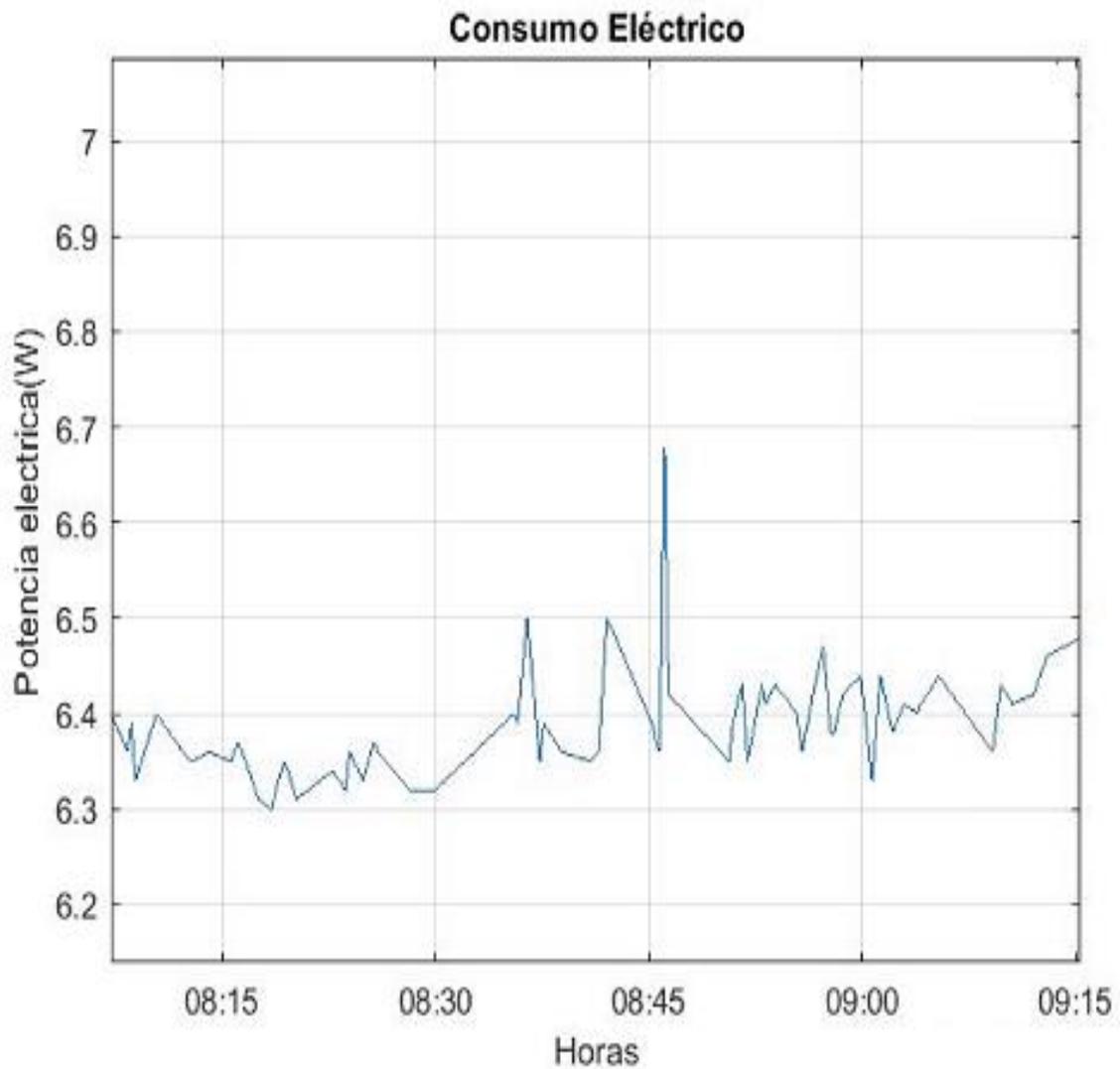


Figura 44. Potencia eléctrica en intervalos de una hora con eficiencia.

Se analizó otro rango de tiempo el cual fue desde las 14:00 hasta las 17:30, donde se apreció un elevado consumo eléctrico superando los 16.5 Wh, que a su vez disminuye con el pasar del tiempo de una manera notoria y otra vez vuelve a elevarse, todo lo antes mencionado se puede apreciar en la Figura 45.

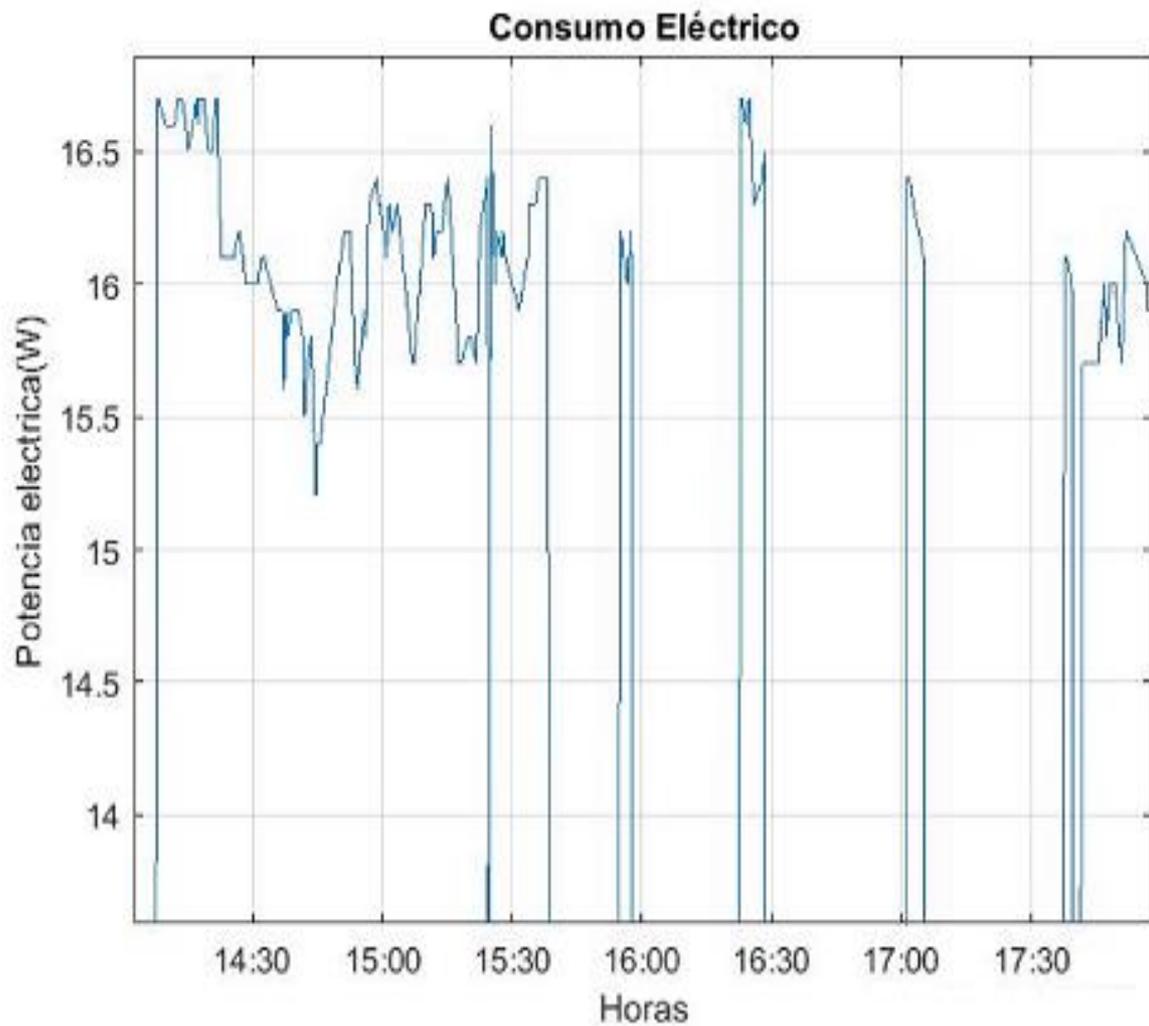


Figura 45. Análisis de intervalo de tiempo de tres horas de la potencia con eficiencia eléctrica.

La Figura 46 muestra la evolución de la potencia eléctrica con eficiencia energética, se dice eso debido a que existe un descenso de potencia eléctrica brusco que llega a 0, esto se refiere a que las bombas se apagaron ya que el nivel de oxigenación es demasiado alto a lo normal entonces existe un tiempo en el que la bomba no consume energía hasta que los niveles de oxigenación disminuyan notablemente.

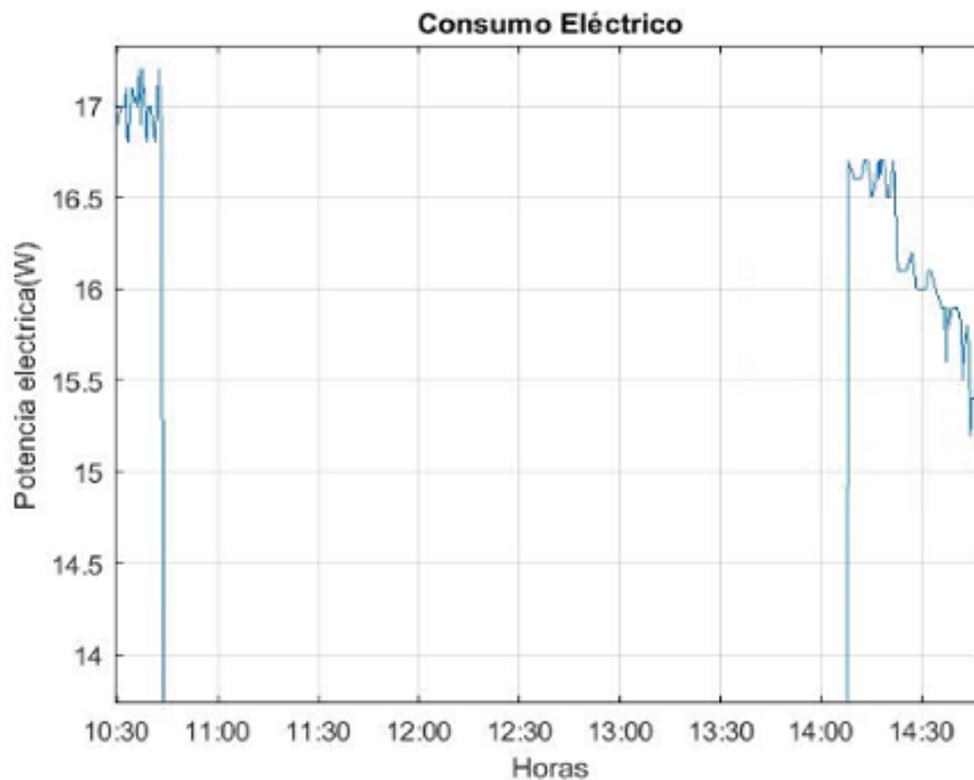


Figura 46. Evolución de la potencia con eficiencia energética.

A continuación, se muestran las Figuras de la concentración de oxígeno realizadas en el software Matlab:

En la Figura 47 se puede apreciar los diferentes datos en cuanto a lo que es la concentración de oxígeno con su respectiva hora en la que ese dato fue registrado, esta figura ha tenido diferentes comportamientos a lo largo de un mes por ejemplo podemos notar que en ciertos tiempos existe una elevada y baja concentración de oxígeno, es necesario recalcar que cuando existe demasiada oxigenación las bombas se apagan para evitar el consumo eléctrico excesivo.

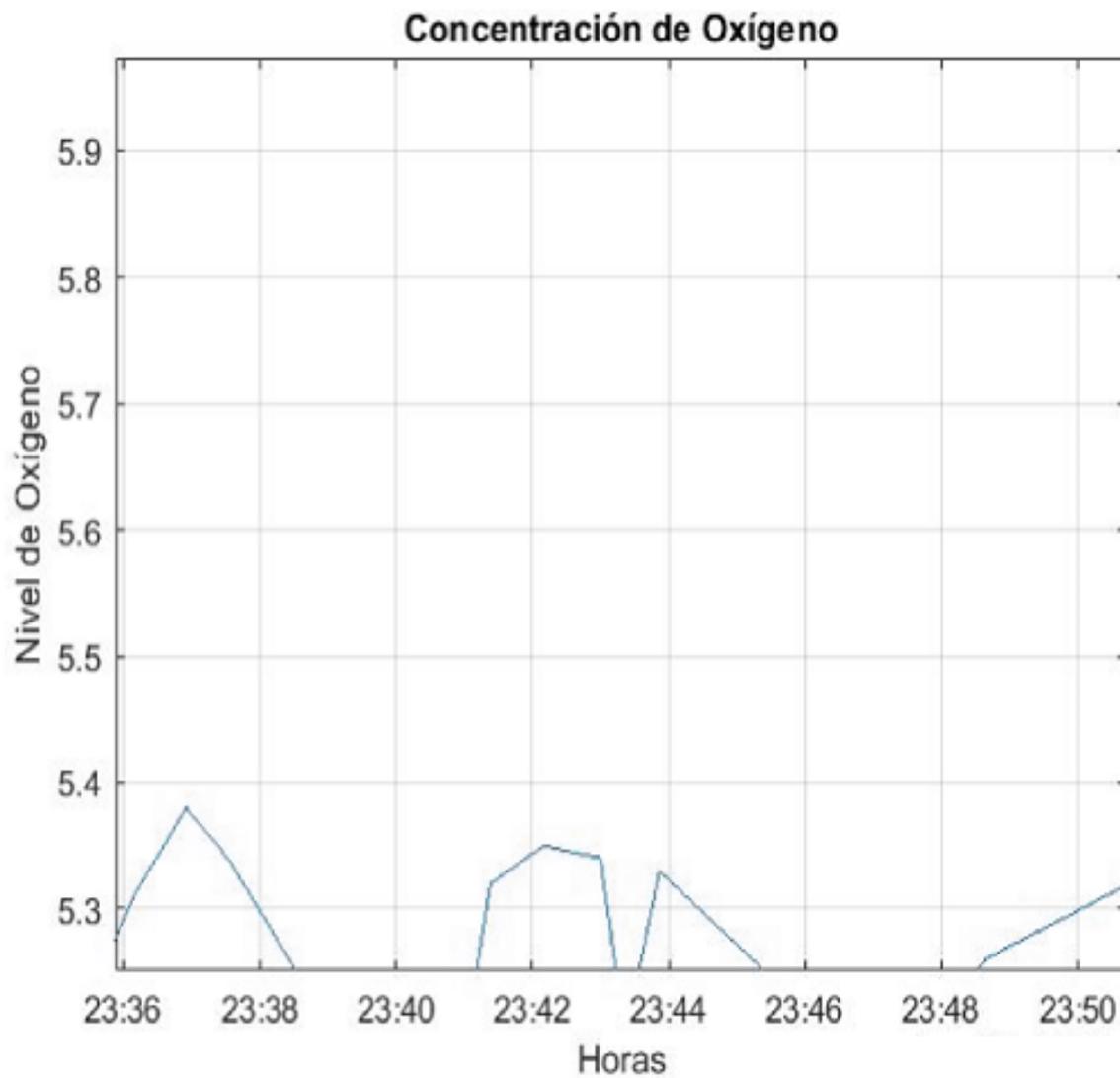


Figura 47. Comportamiento de los datos obtenidos del sensor de oxígeno.

La Figura 48 se observa que con el pasar del tiempo teniendo en cuenta que la bomba esta encendida, la concentración comienza a aumentar, cabe recalcar que al obtener niveles de oxígeno mayores a 8 la bomba deja de funcionar teniendo un lapso de tiempo para que la concentración de oxígeno disminuya y así volver a repetir el proceso.



Figura 48. Elevación del nivel de oxígeno

En la Figura 49 se visualiza un descenso en el nivel de oxígeno, esto se debe a que excedió el límite el cual es 8 por ende la bomba se apagó y comenzó a disminuir el nivel de oxígeno, así podemos tener un ahorro en el consumo eléctrico.

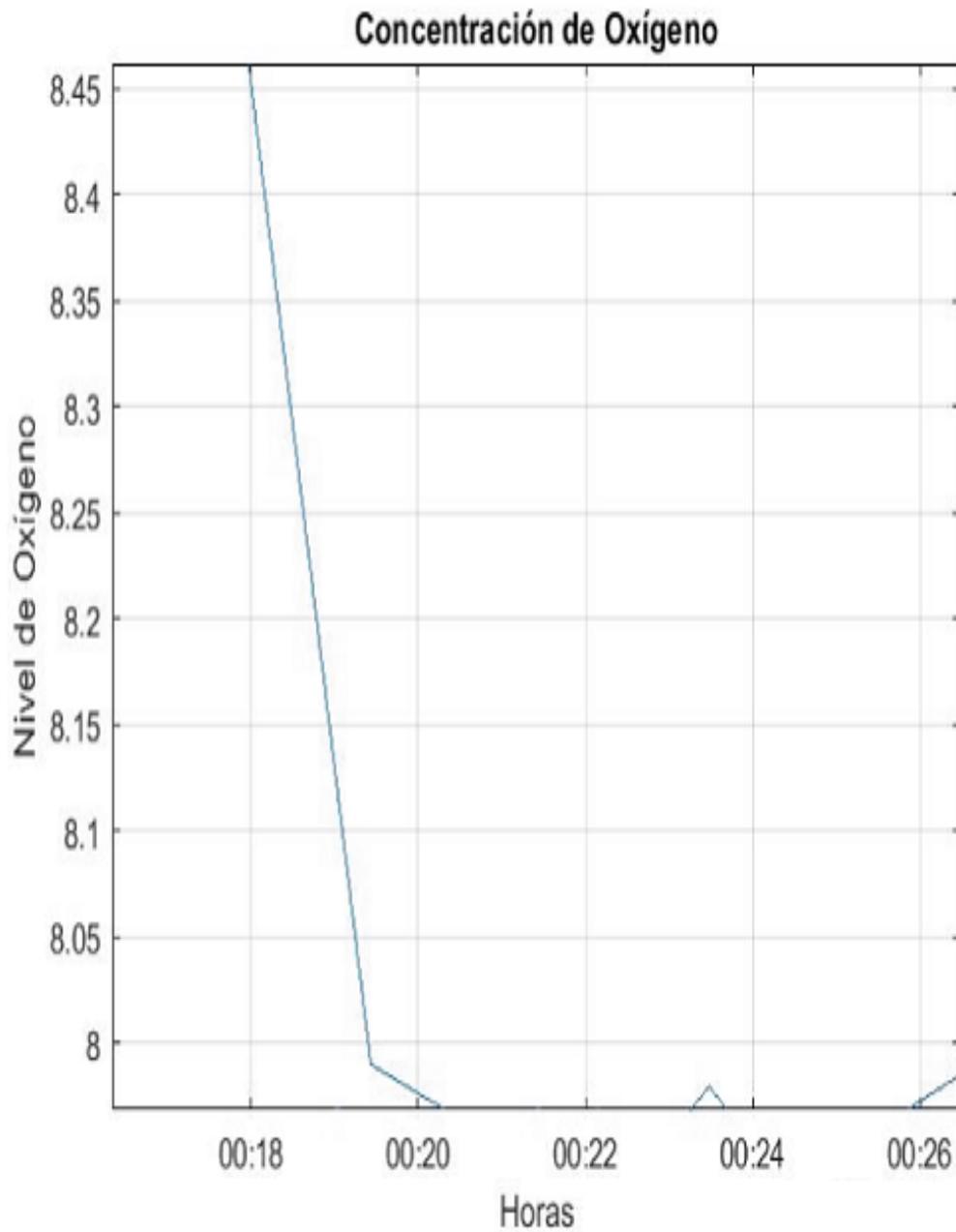


Figura 49. Disminución del nivel de oxígeno.

A continuación, se mostrarán las Figuras realizadas en la plataforma Tableau haciendo una comparativa de los registros tomados por el sensor permitiendo hacer una comparativa entre como es el comportamiento de la potencia eléctrica con nuestro sistema y sin nuestro sistema.

En la Figura 50 se tomaron registros en el tiempo de 3 horas donde claramente se puede visualizar que el consumo de energía constante que representa la línea de color gris cuando no está aplicado el sistema; la línea de color celeste es la que representa el sistema se tomó un intervalo de tiempo de 3 horas que empieza desde las 9:10 a 12:10; a partir de las 10:40 hasta las 12:10 no se registró consumo eléctrico elevado porque los niveles de oxígeno son satisfactorios.

Consumo Eléctrico 3 horas

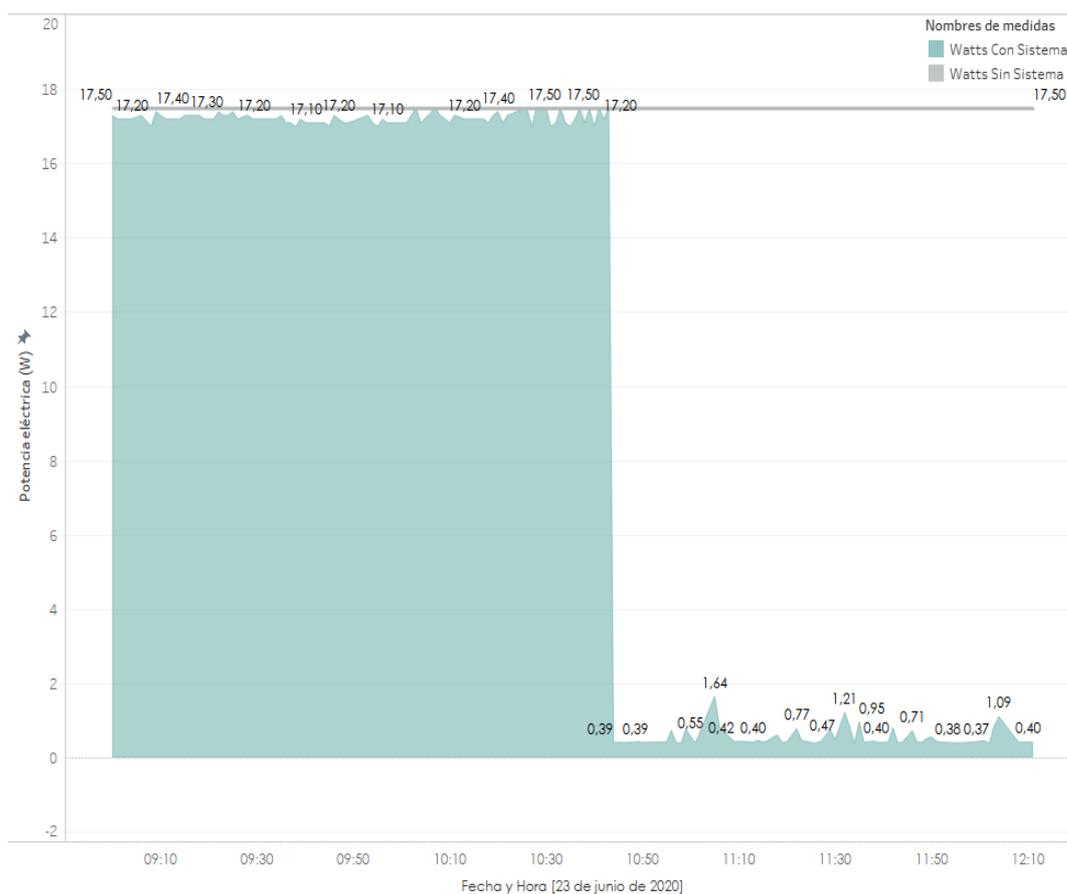


Figura 50. Comparativa del consumo eléctrico en tres horas utilizando nuestro sistema y sin nuestro sistema.

La Figura 51 se visualiza 1 día de funcionamiento, se puede ver espacios que no están sombreados, estos están representando el funcionamiento del sistema obteniendo una eficiencia energética.

Consumo Eléctrico 1 Día

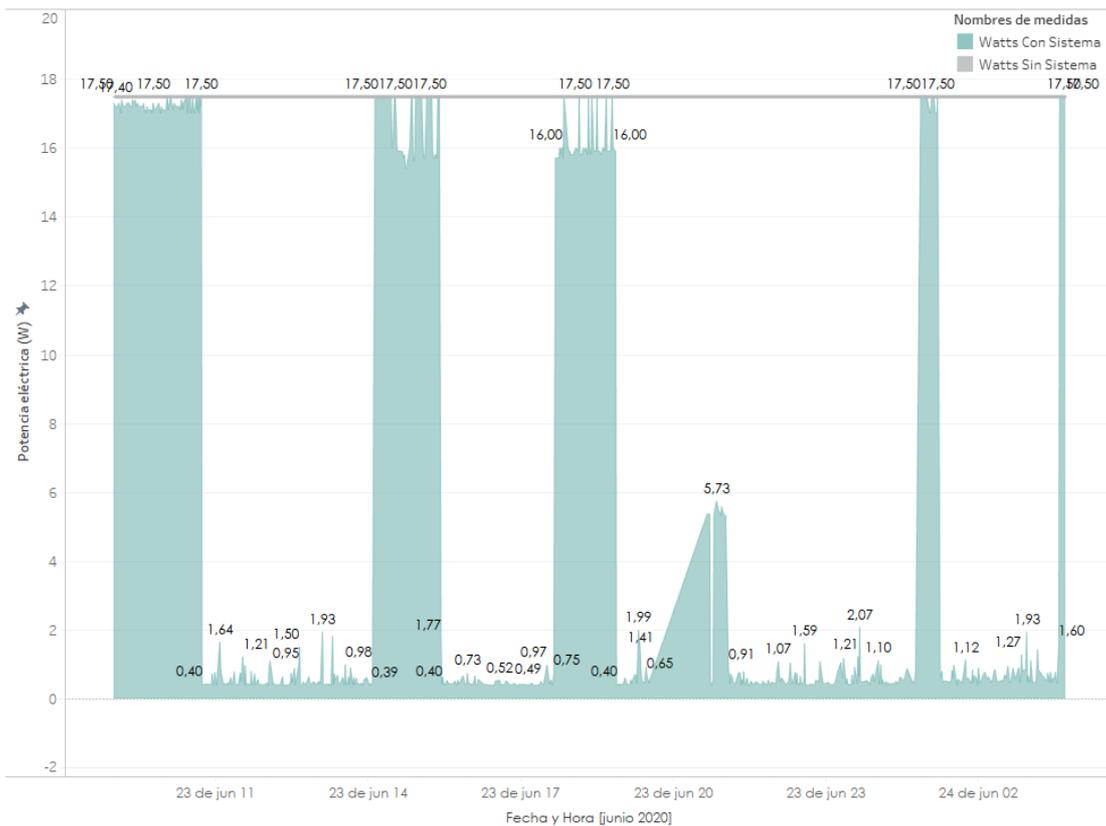


Figura 51. Diferencia entre la aplicación de nuestro sistema y cuando no se aplica para el consumo eléctrico.

Para la Figura 52 se tomaron registros de 1 mes de evolución se puede visualizar como se ha comportado el sistema, activando las bombas únicamente cuando los niveles de oxigenación disminuyen a menos de 6 de concentración de oxígeno en donde se obtuvo una eficiencia energética del 75%.

Consumo Eléctrico 1 Mes

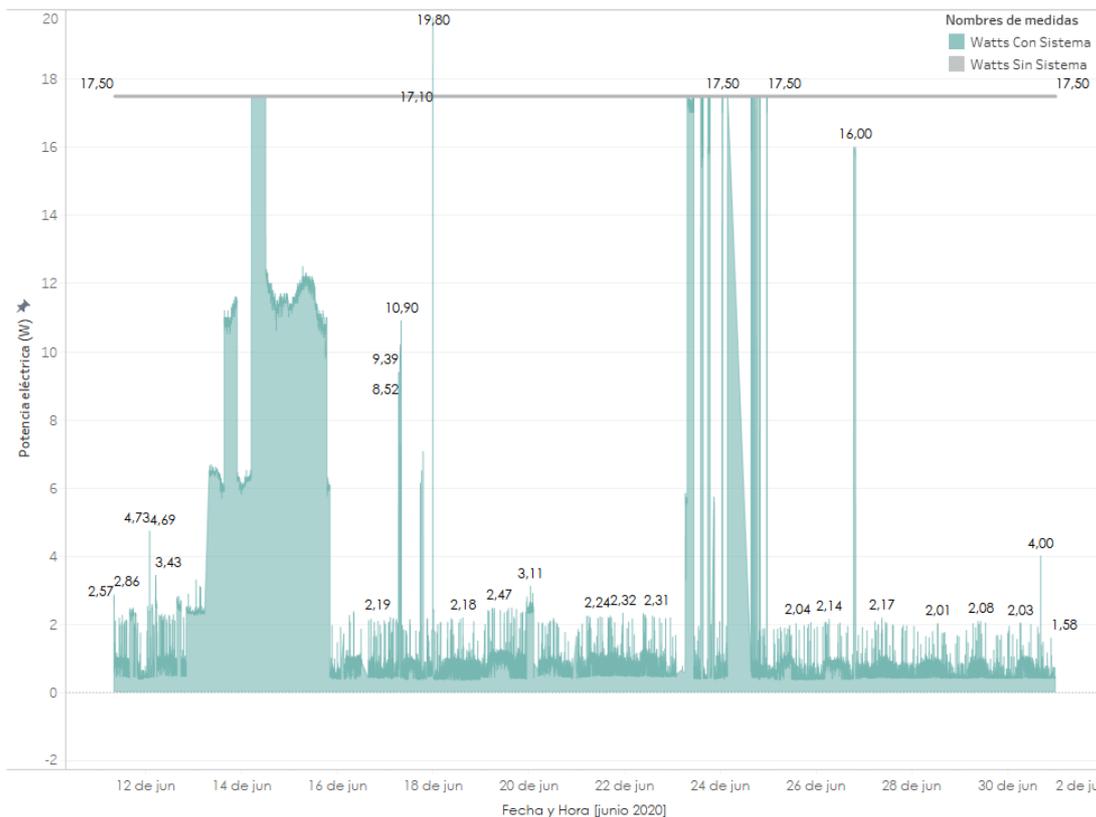


Figura 52. Potencia eléctrica con eficiencia del 75% en el transcurso de 1 mes.

Realizando el análisis del consumo eléctrico con los datos obtenidos de thingspeak se ha obtenido el siguiente resultado:

Tabla 15.

Comparativa entre la aplicación del sistema y no aplicación.

| 16 Watts Blowers | Sin Automatización | Con Automatización |
|------------------|-----------------------------|---------------------------|
| Día | 16 W x 24 horas = 384 Wh | 16 W x 6 horas = 96 Wh |
| Semana | 384 Wh x 7 días = 2688Wh | 96 Wh x 7 días = 672Wh |

| | | |
|----------------------|--------------------------------|-----------------------------|
| Mes (30 días) | 384 Wh x 30 días = 11520 Wh | 96 Wh x 30 días= 2880 Wh |
|----------------------|--------------------------------|-----------------------------|

La interpretación de los datos de consumo eléctrico cuando no hay un control el consumo es constante, pero aplicando el sistema se puede pasar de un consumo mensual de 11520 Wh a 2880 Wh donde se obtiene un ahorro energético de al menos 75%

Cuando no existe algún sistema que cense el oxígeno disuelto en el agua, se registra un consumo eléctrico mensual de 11,52 KWh; pero cuando se implementa el sistema de control de consumo eléctrico censando el oxígeno disuelto en el agua se obtiene un consumo eléctrico mensual de 2,88 KWh obteniendo un ahorro energético y económico del 80% con relación al no aplicar el sistema.

5.3. Análisis de Errores

La Figura 53 muestra los datos tanto en la base local como en la nube, permitiendo conocer la diferencia de la toma de los datos, dicha variación de tiempos depende del tráfico de datos del proveedor de internet que está en un rango de 5, 15 o 30 segundos y el tiempo en que se tarda para obtener el dato de los sensores.

Base Local

| Shell | fecha | oxigeno | fecha | watts |
|--------------|---------------------|---------|---------------------|-------|
| DO2 = 5.46 | 2020-06-26 13:56:23 | 5.44 | 2020-06-26 13:56:01 | 15.8 |
| 200 OK | 2020-06-26 13:56:07 | 6.47 | 2020-06-26 13:55:50 | 15.7 |
| Watts = 15.9 | 2020-06-26 13:55:51 | 5.43 | 2020-06-26 13:55:44 | 15.9 |
| 200 OK | 2020-06-26 13:55:43 | 5.46 | 2020-06-26 13:55:42 | 15.9 |
| Irms = 0.13 | 2020-06-26 13:55:37 | 5.45 | 2020-06-26 13:55:25 | 16.0 |
| 200 OK | 2020-06-26 13:55:20 | 5.47 | 2020-06-26 13:54:59 | 15.9 |
| DO2 = 5.46 | 2020-06-26 13:54:54 | 5.46 | 2020-06-26 13:54:53 | 15.8 |
| 200 OK | 2020-06-26 13:54:53 | 5.49 | 2020-06-26 13:54:47 | 16.0 |
| Watts = 15.7 | | | 2020-06-26 13:54:46 | 15.8 |
| 200 OK | | | 2020-06-26 13:54:29 | 15.9 |
| Irms = 0.13 | | | | |
| 200 OK | | | | |

Base en la Nube



Figura 53. Comparativa de datos tanto en base local como en la nube.

El instrumento CrocSee AC 80-260V 100A CRS-022B equipado con un transformador de corriente alterna que funciona como multímetro de voltaje, corriente, potencia activa; el rango de funcionamiento es de: 80 hasta 260 V de corriente alterna, el rango de potencia va desde los 0 hasta 22000 Watts y es compatible con la frecuencia de Ecuador porque el rango de funcionamiento es: 45 hasta 65 Hz. En la figura 54 se muestra el instrumento CrocSee conectado y sin conectar el sensor hacia las bombas de la maqueta mostrando los valores reales de potencia para ser comparados con los obtenidos.



Figura 54. Instrumento para medir la potencia, corriente y voltaje para voltaje AC.

6. CONCLUSIONES Y RECOMENDACIONES

6.1. Conclusiones

Este proyecto se trata de la implementación de un prototipo que controle el consumo eléctrico de una bomba y blowers en cualquier planta de tratamiento, para obtener un funcionamiento eficiente utilizando los sensores de oxígeno y consumo eléctrico, permitiendo presentar los resultados en la nube y en tiempo real utilizando tecnología IOT.

La implementación de esta solución en una planta de tratamiento permitirá procesar eficiente y adecuadamente las aguas residuales que llegan de las cañerías y reutilizarlas para el riego de cultivos.

Los principales elementos que miden esta solución permiten conocer los niveles de concentración de oxígeno disuelto que se encuentran en los tanques homogenizadores con la finalidad de que los datos sean visualizados de manera remota en la plataforma en la nube.

Este sistema permite optimizar el rendimiento de la planta evitando el consumo excesivo de electricidad y el deterioro de esta.

La realización de este prototipo podrá facilitar al momento de ser implementado, que las personas encargadas de la planta de tratamiento de aguas residuales puedan optimizar tiempo y exponerse a riesgos.

Una base local permite tener una autoría total de los datos almacenados a diferencia de la nube, si se pierde o vence el usuario no tendría acceso a los datos.

Este sistema es escalable permitiendo la agregación de más campos y características como: sensores, tiempo de envío de datos y visualización.

6.2. Recomendaciones

En los niveles de oxígeno se recomienda tener en un rango de 8 a 12 mg/l para obtener una buena oxigenación debido a que si es menor a 4 mg/l tendremos una disminución de oxígeno mala y si es mayor a 12 mg/l existirá una saturación de este.

Se debe realizar backup mensuales externos para evitar ocupar un mayor consumo de memoria y poder tener optimización de los procesos en la Raspberry Pi 4.

Se sugiere que en el segundo tanque homogenizador se coloque el sensor de concentración de oxígeno ya que en dicho tanque se encuentra el blower permitiendo oxigenar el agua que está en el interior.

Para el consumo eléctrico es necesario que se utilice el sensor PZCT-02 ya que éste, de acuerdo con sus características nos brinda la facilidad para poder recolectar los datos precisos en el prototipo.

Se propone la creación de una cuenta de correo usuario estudiantil de la carrera de Telecomunicaciones para prolongar el tiempo de vida del servicio de API porque se es asociado con la cuenta del estudiante, pero esa cuenta puede dejar de existir.

REFERENCIAS

- Alexander, J., Hembach, N., & Schwartz, T. (2020). Evaluation of antibiotic resistance dissemination by wastewater treatment plant effluents with different catchment areas in Germany. *Scientific Reports*, 10(1), 1–9. <https://doi.org/10.1038/s41598-020-65635-4>.
- Avilés, A. B. L., Del Cerro Velázquez, F., & Del Riquelme, M. L. P. (2020). Methodology for energy optimization in wastewater treatment plants. Phase II: Reduction of air requirements and redesign of the biological aeration installation. *Water* (Switzerland), 12(4). <https://doi.org/10.3390/W12041143>.
- Angulo Infante, C. L. (2019). Monitoreo, Adquisición Y Transmisión De Datos En Tiempo Real, Mediante IoT, De Las Variables Medidas Por Un Prototipo De Caracterización De Paneles Fotovoltaicos Y De Medición De Variables Meteorológicas (Doctoral dissertation, Universidad Industrial de Santander, Escuela de Ingeniería Eléctrica, Electrónica y Telecomunicaciones).
- C. M. Lopez-Vazquez, G. B. (2017). Tratamiento biológico de aguas residuales: principios, modelación y diseño (ilustrada ed.). (M. C. Mogens Henze, Ed.) WA PUBLISHING, Obtenido de <https://books.google.es/books?hl=es&lr=&id=lxNBDwAAQBAJ&oi=fnd&pg=PP1&dq=aparicion+de+las+plantas+de+tratamiento+de+aguas+residuales+historia&ots=ygmL7WBcwO&sig=wQbfdFibXQS1jiXeszOllouD5a8#v=onepage&q=aparicion%20de%20plantas%20de%20tratamiento%20de>.

- Cabarna González, J. J., Chiriach, M., & Uña García, J. M. de. (2010). Sistema de adquisición de datos. *Revista Española de Electrónica*, 665, 58–70. http://www.redeweb.com/_txt/665/abril_2010.pdf.
- Cálculo, D. M. De, Fernando, L., & Castaño, C. (2009). Desarrollo del Hardware de una Red de Sensores de Medidas de Consumo Eléctrico.
- Chen, J., Liao, J., & Wei, C. (2020). Coking wastewater treatment plant as a sources of polycyclic aromatic hydrocarbons (PAHs) in sediments and ecological risk assessment. *Scientific Reports*, 10(1), 1–8. <https://doi.org/10.1038/s41598-020-64835-2>.
- Collivignarelli, M. C., Baldi, M., Abbà, A., Caccamo, F. M., Miino, M. C., Rada, E. C., & Torretta, V. (2020). Foams in wastewater treatment plants: From causes to control methods. *Applied Sciences (Switzerland)*, 10(8). <https://doi.org/10.3390/APP10082716>.
- Conagua. (2016). Monitoreo de la calidad del agua en México. De los sitios muestreados por la Red Nacional de Monitoreo del 2012-2015. Comisión Nacional Del Agua, 9.
- Deacutis, C. F. (2016). Dissolved oxygen. *Encyclopedia of Earth Sciences Series*, 202–203. https://doi.org/10.1007/978-94-017-8801-4_72.
- Doukas, c, (2012), Building Internet of Things with the ARDUINO. CreateSpace Independent Publishing Platform.
- Enríquez Lenis, A., Hamilton Ortiz, J., & Taha Ahmed, B. (2014). Banda Ancha Inalámbrica. Banda Ancha Inalámbrica. <https://doi.org/10.3926/oms.195>.
- ENRIQUEZ HERRADOR, R. (2009). Guía de Usuario de Arduino. Universidad de Cordoba, 1, 8–10.

- Galicia, C A. Eu, Lopez, W. G., & Mallette, R, F. G. Implementación de plataforma para el internet de las cosas en un ambiente de nube pública IOT platform development in a public cloud environment.
- Hernandez, J. M. i. (2005). Software Libre Técnicamente Viable, Económicamente Sostenible y Socialmente Justo. Infonomia - Re de Innovadores, 1–191. <http://www.infonomia.com/img/pdf/lilibrejmas.pdf>.
- León-García, A., & Widjaja, I. (2002). Redes de comunicación. 1–137.
- Nikuze, M. J., Niyomukiza, J. B., Nshimiyimana, A., & Kwizera, J. P. (2020). Assessment of the efficiency of the wastewater treatment plant: A case of Gacuriro Vision City. IOP Conference Series: Earth and Environmental Science, 448(1). <https://doi.org/10.1088/1755-1315/448/1/012046>.
- Nolasco, D, A. (2010), Desarrollo de proyectos MDL en plantas de tratamiento de aguas residuales, Obtenido de BID Mejorando Vidas, Obtenido de <https://publications.iadb.org/publications/spanish/document/Desarrolode-proyectos-MDL-en-plantas-de-tratamiento-de-aguas-residuales.pdf>.
- Ramírez, L. G, C., Jiménez, G. S. A., & Carreño, J, M. (2014). Sensores y actuadores. Grupo Editorial Patria.
- Rey, G., Mario, J., & Johann, F. (2012). Medición inalámbrica de concentración de oxígeno para productos vitivinícolas Wireless measurement of oxygen Resumen.

Rodríguez, G. E. (2014). Diseño y desarrollo de un prototipo de riego automático controlado con Raspberry Pi y Arduino. Sergioguillen.Com, 1–45. http://sergioguillen.com/wp-content/uploads/2015/05/Guillen_Cavero_TFG_vFinal.pdf

Sensors, H., Sensors, M., & Code, S. (2017). Gravity : Analog Dissolved Oxygen Sensor SKU : SEN0237. 0237, 1–7.

Siatou, A., Manali, A., & Gikas, P. (2020). Energy consumption and internal distribution in activated sludge wastewater treatment plants of Greece. *Water (Switzerland)*, 12(4), 1–15. <https://doi.org/10.3390/W12041204>.

SOMMERVILLE, I. (2004). *Ingeniería del Software 7ma. Ed.* - Ian Sommerville.pdf (p. 691).

Schallenberg, J. C., Gonzalo, R., Izquierdo, P., Hernández Rodríguez, C., Unamunzaga, P., Ramón, F., Déniz, G., Díaz, M., Delia, T., Pérez, C., Martel Rodríguez, G., Pardilla, J., Vicente, F., & Ortin, S. (2008). *Energías renovables y eficiencia energética.*

Tarigan, P. B. (2013). Servicios en red. *Journal of Chemical Information and Modeling*, 53(9), 1689–1699. <https://doi.org/10.1017/CBO9781107415324.004>.

Vega Luna, J. I., Sánchez-Rangel, F. J., & Cosme-Aceves, J. F. (2019). Sistema de monitorización de puertas y ventanas de un centro de datos con IoT. *Ingenius*, 22, 72–79. <https://doi.org/10.17163/ings.n22.2019.07>

ANEXOS

Anexo 1: DataSheet del sensor de oxígeno disuelto



 Environmental Robotics

 V 4.2

Revised 3/20

Gen 2

Dissolved Oxygen Probe

| | |
|-----------------------------|------------------------------|
| Reads | Dissolved Oxygen |
| Range | 0 – 100 mg/L |
| Accuracy | +/- 0.05 mg/L |
| Response time | ~0.3 mg/L/per sec |
| Temperature range °C | 1 – 60 °C |
| Max pressure | 3,447 kPa (500 PSI) |
| Max depth | 343 Meters (1,125 ft) |
| Connector | Male SMA / Male BNC |
| Cable length | 1 meter |
| Internal temperature sensor | No |
| Time before recalibration | ~1 Year |
| Life expectancy | 5 Years + |
| Maintenance | ~18 Months |





Anexo 2: Detalles del circuito embebido para la lectura de datos del sensor de oxígeno disuelto.

AtlasScientific™
Environmental Robotics
V 5.4
Revised 1/09

EZO-DO™

Embedded Dissolved Oxygen Circuit

| | |
|---|---|
| Reads | Dissolved Oxygen |
| Range | 0.01 – 100+ mg/L 0.1 – 400+ % saturation |
| Accuracy | +/- 0.05 mg/L |
| Response time | 1 reading per sec |
| Supported probes | Any galvanic probe |
| Calibration | 1 or 2 point |
| Temperature, salinity and pressure compensation | Yes |
| Data protocol | UART & I ² C |
| Default I ² C address | 97 (0x61) |
| Operating voltage | 3.3V – 5V |
| Data format | ASCII |

RoHS

PATENT PROTECTED

Written by Jordan Press
Designed by Neath Press

This is an evolving document, check back for updates.

Anexo 3: Script del arduino

```
#include "EmonLib.h" //Librería para lectura de datos analógicos
```

```
#include <SoftwareSerial.h>
```

```
#define rx 2
```

```
#define tx 3
```

```
SoftwareSerial myserial(rx, tx);
```

```

EnergyMonitor energyMonitor;           // Crear una instancia EnergyMonitor
float voltajeRed = 124.0;               // Voltaje de nuestra red eléctrica
Scientific product
String inputstring = "";
String sensorstring = "";
boolean input_string_complete = false;
boolean sensor_string_complete = false;
float DO;
is the Do

void setup() {
  Serial.begin(9600);
  myserial.begin(9600);
  inputstring.reserve(10);
  sensorstring.reserve(30);
  energyMonitor.current(0, 1.24);      //Número del análogo conectado y
  valor de calibración teórica
}

void serialEvent() {
  inputstring = Serial.readStringUntil(13);
  input_string_complete = true;
}

void loop() {

  if (input_string_complete == true) {
    myserial.print(inputstring);
    myserial.print('\r');
    inputstring = "";
  }
}

```

```
input_string_complete = false;
}

if (myserial.available() > 0) {
  char inchar = (char)myserial.read();
  sensorstring += inchar;
  if (inchar == '\r') {
    sensor_string_complete = true;
  }
}

if (sensor_string_complete == true) {
  // Obtenemos el valor de la corriente eficaz
  // Pasamos el número de muestras que queremos tomar
  double Irms = energyMonitor.calcIrms(1484);

  // Calculamos la potencia aparente
  double potencia = Irms * voltajeRed;

  // Mostramos la información por el monitor serie
  Serial.print("Watts = ");
  Serial.println(potencia);
  Serial.print("Irms = ");
  Serial.println(Irms);
  Serial.println("DO2 = "+sensorstring);
  //send that string to the PC's serial monitor

  sensorstring = "";
  sensor_string_complete = false;
}
```

```
}
```

Anexo 4: Script de Python llamado conexionThingSpeak.py

```
import serial
import json
import http.client
#Libreria para thingSpeak
import urllib3
from urllib.parse import urlencode
#Libreria para fecha y transforamr datos
from decimal import Decimal
from datetime import date
from datetime import datetime
#Libreria para la conexión a la base de datos MariaDB
import mysql.connector as mariadb
import time
#Lectura de datos por el puerto serial
arduino = serial.Serial('/dev/ttyACM0',9600)

key = "SoloDisponibleEnDispositivo" # API Key ThingSpeak

def Planta():

    DO2 = ""
    WATTS = ""
    IRMS = ""
    while True:

        #Conexion hacia Mariadb
        mariadb_connection = mariadb.connect(host='localhost', user='user1',
                                             password='secret', database='sensor')
        cursor = mariadb_connection.cursor()
```

```

#Obtencion de Datos
dato = arduino.readline()
#convertir a string
chare = str(dato)
#Comprobando si el dato es Oxigeno
if(chare.find("DO2 =") >= 0 and chare.find("DO2 =") >= 0):
    #Discriminando el valor de Oxígeno
    DO2_PURGUE = chare.find("DO2 =")+5
    chareb = chare.find("DO2 =")+10
    #Obteniendo y almacenando el valor de Oxígeno
    DO2 = chare[DO2_PURGUE:chareb]
    print("DO2 =",DO2)

elif(chare.find("Watts =") >= 0):
    WATTS_PURGUE = chare.find("Watts =")+7
    charea = chare.find("Watts =")+12
    WATTS = chare[WATTS_PURGUE:charea]
    print("Watts =",WATTS)

elif(chare.find("Irms =") >= 0):
    IRMS_PURGUE = chare.find("Irms =")+6
    charec = chare.find("Irms =")+11
    IRMS = chare[IRMS_PURGUE:charec]
    print("Irms =",IRMS)

#Guardando los datos en la base local de mariadb
sql = "INSERT INTO sensor.datos(fecha,oxigeno,watts,irms) values
(%s,%s,%s,%s)"
now = datetime.now()
#array de los datos
val = (now,DO2,WATTS,IRMS)
#inserción de datos en la setencia SQL
cursor.execute(sql,val)

```

```

#Guardando la sesion
mariadb_connection.commit()
#Enviando datos hacia ThingSpeak
params = urlencode({'field1': DO2,'field2': WATTS, 'key':key })
headers = {"Content-type": "application/x-www-form-
urlencoded","Accept": "text/plain"}
conn = http.client.HTTPConnection("api.thingspeak.com:80")
try:
    #Abriendo al conexión hacia ThingSpeak
    conn.request("POST", "/update", params, headers)
    #Estado de la conexión
    response = conn.getresponse()
    #print (Oxigeno)
    print (response.status, response.reason)
    data = response.read()
    #cerrando la conexión
    conn.close()
except:
    print ("connection failed")
    break
if __name__ == "__main__":
    while True:
        Planta()

```

Anexo 5: Script pyhton llamado “releAutomatizado.py”

```

import mysql.connector as mariadb
#import wiringpi2
from datetime import date
from datetime import datetime
import time
import RPi.GPIO as GPIO

```

```

while True:
    GPIO.setmode(GPIO.BCM)
    #GPIO.setwarnings(False)
    GPIO.setup(17,GPIO.OUT)
    # GPIO.output(17,GPIO.HIGH)
    mariadb_connection = mariadb.connect(host='localhost', user='user1',
password='secret', database='sensor')
    cursor = mariadb_connection.cursor()
    #cursor.execute("drop table tmp_convert")
    cursor.execute("truncate table tmp_convert")
    cursor.execute("insert into tmp_convert select cast(oxigeno as float) as DO2
from datos where oxigeno not in (") and oxigeno not like '%s%' and oxigeno not
like '%xa%' order by fecha desc limit 100")
    cursor.execute("select avg(DO2) as promDO2 from tmp_convert")
    prom = cursor.fetchone()
    datotr = str(prom)
    dato = datotr[1:datotr.find(",")]
    #print(dato)
    if (float(dato) <= 7):
        #Activa el rele para encender la oxigenación
        print("Alerta de Disminución Oxígeno =",dato)
        #GPIO.output(17,GPIO.LOW)
        GPIO.output(17,GPIO.LOW)
        #time.sleep(1)
    elif (float(dato) > 7.5):
        #Desactiva el rele de la oxigenación
        print("Concentración de oxigeno ACEPTABLE =",dato)
        GPIO.output(17,GPIO.HIGH)
        time.sleep(3600)
    else:
        print("Ok",dato)
        GPIO.output(17,GPIO.HIGH)
        #time.sleep(5)

```

mariadb_connection.commit()

Nota: Todos los scripts se encuentran en el repositorio digital de GitHub bajo la siguiente ruta: <https://github.com/Jota420/Scripts>

Anexo 5: Video de la funcionalidad del Prototipo de una Planta de Tratamiento de Aguas Residuales.



https://udlaec-my.sharepoint.com/:v:/g/personal/byron_garces_udla_edu_ec/EcLlsLu-LKRBj32IVd-5fP8B1ouM1Nc7utlLKmCwP1I24g?e=qPT8fx

