

alimentación autónomo, una de las ventajas de la red de sensores es que no se emplearan radios RF en específico sino que se emplean el radio WiFi del propio módulo ESP32-S2 mediante una Red Mesh para su comunicación, lo que reduce el consumo de corriente de cada boya de sensado.



Figura 2. Boya de sensado

Una de las ventajas de este proyecto es que no existen sistemas de alerta que sean accesibles a la comunidad cercana a una PTAR, esto permite alertar de altos niveles de Metano a la población y pueda contrarrestar posibles efectos dañinos, así como también a los operadores de una PTAR puedan tener un control de la misma a través de los datos que recaba el sistema propuesto.

Para el desarrollo del prototipo se considera utilizar sistemas embebidos avanzados como la serie de módulos ESP32-S2 que contiene un microprocesador Xtensa single core de 32 bits, módulo WiFi con soporte en 802.11 b/g/n, una velocidad de procesamiento de 240Mhz, una capacidad de memoria de 320 KB en SRAM y 128 KB en ROM, este módulo es altamente recomendado para aplicaciones de IoT [1].

El ESP32-S3 contiene un microprocesador Xtensa dual core de 32 bits, módulo WiFi con soporte en 802.11 b/g/n, módulo Bluetooth BLE 5.0, una velocidad de procesamiento de 240Mhz, una capacidad de memoria de 512 KB en SRAM y 384 KB en ROM, este módulo es altamente recomendado para aplicaciones de IoT y reconocimiento de imágenes [2].

El principal valor agregado que busca el desarrollo del proyecto es prevenir daños en la salud de la población debido a los gases provenientes de la PTAR implementando para ello un sistema de alerta que indique cuando se encuentran altas concentraciones de gases.

Se busca contar con un diseño robusto de sensado en una PTAR para llevar un mejor manejo de dicha planta a través de los datos recabados y no solo reutilizar el agua tratada sino también cuidar todos los aspectos que conlleva su proceso, principalmente la salud de los habitantes aledaños a una PTAR.

1.1 Planta de tratamiento de aguas residuales

La planta de tratamiento de aguas residuales Oriente es la que se pretende monitorear esta PTAR tiene una capacidad de 2000 L/seg y un volumen de tratamiento anual de 45,1186,272 m³/año. El tipo de tratamiento es mediante la técnica de lagunas aireadas donde se requiere de suministrar el oxígeno requerido para biodegradar la materia orgánica mediante equipos que demandan energía eléctrica. En la figura 3 se muestra la ubicación de las PTAR de la ciudad de Durango [3].

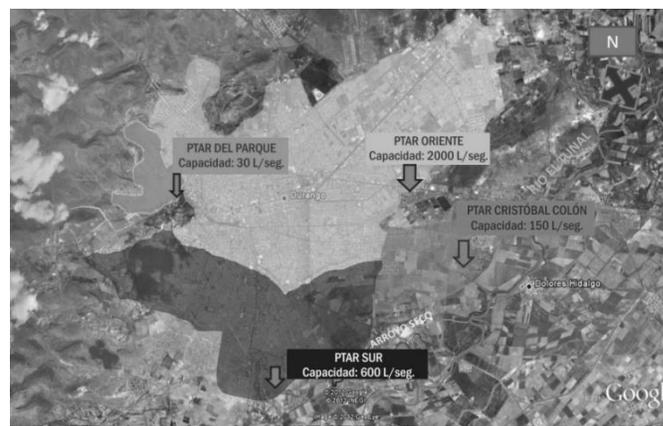


Figura 3. PTAR de la ciudad de Durango.

La PTAR Oriente es un sistema centralizado donde sus influentes tienen una demanda bioquímica de oxígeno (DBO) promedio de 146.23 mg/l y una DBO del efluente con promedio de 38.06 mg/l.

La característica de este tipo de PTAR es que ocupan grandes extensiones de terreno para su operación además la técnica de tratamiento por lagunas aireadas provocan contaminación de gas Metano en el ambiente lo que puede afectar la salud de los pobladores que están cerca de las PTAR. Además impactan la calidad de vida de los centros urbanos cercanos por lo que es necesario utilizar técnicas que reduzcan los efectos nocivos hacia la población. Es por esto que surge la necesidad de contar con un sistema de alerta sobre los niveles de gases emitidos por la PTAR [4].

1.2 Red Mesh

En una conexión WiFi un solo nodo (enrutador) está conectado a todos los demás nodos (estaciones). Cada nodo puede comunicarse entre sí utilizando el punto de acceso. Sin embargo, esto se limita a la cobertura WiFi del punto de acceso. Cada estación debe estar dentro del rango para conectarse directamente al punto de acceso. Esto no sucede en una Red Mesh [5].

En una Red Mesh los nodos no necesitan conectarse a un nodo central. Los nodos son responsables de retransmitir las transmisiones de los demás. Esto permite que varios dispositivos se distribuyan en un área física grande. Los nodos pueden auto organizarse y comunicarse dinámicamente entre sí para garantizar que el paquete llegue a su destino de nodo final. Si se elimina algún nodo de la red, este puede auto organizarse para asegurarse de que los paquetes lleguen a su destino [5].

La Red Mesh tiene un área de cobertura mucho mayor, ya que los nodos pueden lograr la interconectividad sin necesidad de estar dentro del alcance del nodo central.

La Red Mesh es una red de múltiples saltos (multi-hop), lo que significa que los nodos pueden transmitir paquetes a otros nodos en la red a través de uno o más saltos inalámbricos. Por lo tanto, los nodos en esta red no solo transmiten sus propios paquetes, sino que sirven simultáneamente como repetidores para otros nodos. Siempre que exista una ruta entre dos nodos

en la capa física (a través de uno o más saltos inalámbricos), cualquier par de nodos dentro de una Red Mesh puede comunicarse [6].

Es por esto que se elige esta arquitectura para el desarrollo del proyecto, ya que las distancias en las lagunas aireadas son grandes y la red WiFi no tendría cobertura hacia cada nodo.

Otro de los motivos de implementar este tipo de red es que no se emplearía un radio RF externo para la transmisión de los datos con lo que se reduce en gran medida el consumo de energía de cada nodo. En la figura se observa el funcionamiento de dicha red que tendrá conectividad a la web mediante el protocolo MQTT.

1.2.1 Red Mesh con ESP32-S2 y ESP32-S3

La red Mesh que se implementa es una red de tres capas (ver figura 4) empleando para ello en el nodo raíz un módulo ESP32-S3 y en las capas 2 y 3 nodos con módulos ESP32-S2. A continuación se describe la función de cada nodo.

Nodo raíz: el nodo raíz es el nodo superior de la red y sirve como la única interfaz entre la red Mesh y una red IP externa. El nodo raíz está conectado a un enrutador WiFi convencional y transmite paquetes hacia la red IP externa a los nodos dentro de la red Mesh. Solo puede haber un nodo raíz dentro de una red Mesh y la conexión ascendente del nodo raíz solo puede ser con el enrutador. En referencia a la figura 4, el nodo A es el nodo raíz de la red [5].

Tipos de nodos

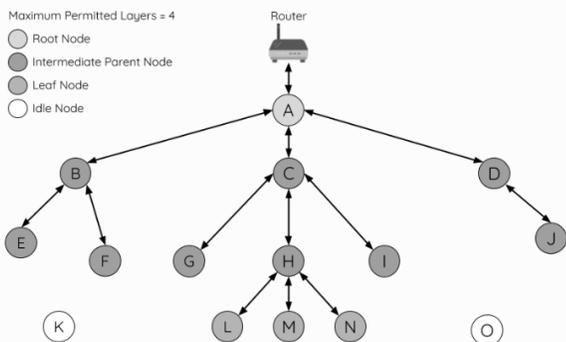


Figura 4. Tipos de nodos en una Red Mesh.

Nodo padre intermedio: los nodos conectados que no son ni el nodo raíz ni un nodo hoja son nodos padre intermedios. Un nodo padre intermedio debe tener una sola conexión ascendente (un solo nodo principal), pero puede tener de cero a varias conexiones descendentes (de cero a varios nodos secundarios). Por lo tanto, un nodo padre intermedio puede transmitir y recibir paquetes, pero también reenviar paquetes enviados desde sus conexiones ascendentes y descendentes. Con referencia a la figura 4, los nodos B al J son nodos padre intermedios. Los nodos padre intermedios sin conexiones descendentes, como los nodos E/F/G/I/J, no son equivalentes a los nodos hoja, ya que todavía se les permite formar conexiones descendentes en el futuro [5].

Nodos de hoja: un nodo hoja es un nodo al que no se le permite tener nodos secundarios (sin conexiones descendentes). Por lo tanto, un nodo hoja solo puede transmitir o recibir sus propios paquetes, pero no puede reenviar los paquetes de otros nodos. Si un nodo está situado en la capa máxima permitida de la red, se le asignará como nodo hoja. Esto evita que el nodo forme conexiones descendentes, lo que garantiza que la red no agregue una capa adicional. En referencia a la figura 4, los nodos L/M/N están situados en la capa máxima permitida de la red, por lo que se han asignado como nodos hoja [5].

Nodos inactivos: los nodos que aún no se han unido a la red se asignan como nodos inactivos. Los nodos inactivos intentarán formar una conexión ascendente con un nodo principal intermedio o intentarán convertirse en el nodo raíz en las circunstancias. En referencia a la figura 4, los nodos K y O son nodos inactivos [5]. En la figura 5 se aprecia la red Mesh de tres capas conformada por un nodo raíz y 6 nodos padre intermedios.

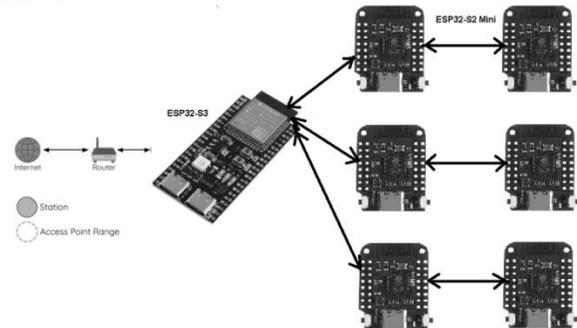


Figura 5. Red Mesh con nodos ESP32-S2 y ESP32-S3

1.2.2 Sensores de gas metano

Los sensores de gas Metano empleados para el desarrollo del proyecto son los sensores MQ2 y MQ4, son sensores de la familia MQ diseñados para detectar una o más gases como por ejemplo detección de gases inflamables, calidad del aire o detección de alcohol en aire respirado.

Los sensores de gases MQ suelen proporcionarse con una placa de medición estándar con el comparador LM393 o similar, que permite obtener la lectura tanto como un valor analógico, como un valor digital cuando se supera un cierto umbral regulado a través de un potenciómetro ubicado en la placa.

2. DESARROLLO

2.1 Diagrama general

En la figura 6 se muestra el diagrama general de este proyecto, y a continuación se describen cada una de las etapas que intervienen.

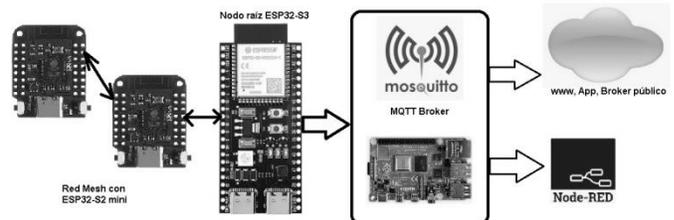


Figura 6. Diagrama general del prototipo.

2.2 Red Mesh con ESP32-S2 mini y ESP32-S3

Red de tres capas Mesh

Como se mencionó en la introducción la red que se implementa es una red de tres capas que contiene un nodo raíz y seis nodos padre intermedios.

Nodo raíz con ESP32-S3

El nodo raíz (ver figura 7) es un módulo ESP32-S3 es el encargado de recibir los mensajes de cada uno de los nodos que están dentro de la red Mesh ya sea de nodos padre intermedios y/o nodos padre sin conexión descendente. También este nodo es el encargado de conectarse a un Broker local para ser monitoreado a través de mensajes MQTT localmente o a través de un Broker en la nube y en específico para el caso de este proyecto puede conectarse a una página web ó app para mostrar alertas.

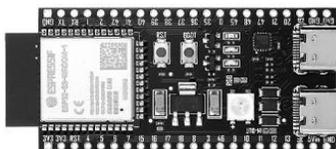


Figura 7. ESP32-S3

Para implementar la red Mesh se emplea la librería `painlessMesh` en versión 1.5 para Arduino, está diseñada para trabajar con hardware ESP8266 y ESP32, en nuestro caso se emplean ESP32-S2 y ESP32-S3 y soportan el uso de la librería.

En el nodo raíz se define el nombre de la red Mesh, la contraseña y el puerto, en todos los nodos de la red Mesh deben contar con esta definición.

```
#define MESH_PREFIX "RedMesh12345678"  
#define MESH_PASSWORD "iotluis12345678"  
#define MESH_PORT 5555
```

En nodo raíz es el puente de conexión hacia un router es por esto que se define la SSID y contraseña de la red a la cual se va a conectar.

```
#define STATION_SSID "PLC_Luis"  
#define STATION_PASSWORD "xxxxxxxx"
```

Enseguida es necesario crear la estructura mesh de la librería `painlessMesh`.

```
painlessMesh mesh;
```

En la configuración de la red Mesh se configura el nombre de la red, contraseña puerto, la red SSID y contraseña a la que se conecta el nodo raíz, así como también se define el modo de operación del nodo, así como el modo de recepción de datos continuo.

```
mesh.init( MESH_PREFIX, MESH_PASSWORD, MESH_PORT, WIFI_AP_STA, 6 );  
mesh.stationManual(STATION_SSID, STATION_PASSWORD);  
mesh.setRoot(true);  
mesh.setContainsRoot(true);  
mesh.onReceive(&receivedCallback);
```

En el lazo principal solo se actualiza continuamente el estado de la red.

```
void loop() {  
  mesh.update();  
}
```

La función principal que ejecuta en el nodo raíz es la de recepción continua de los datos, aquí se especifica el dato recibido de cada uno de los nodos.

```
void receivedCallback( uint32_t from, String &msg ) {  
  Serial.printf("Puente: Dato del nodo %u msg=%s\n", from, msg.c_str());  
}
```

Nodos padre intermedios con ESP32-s2 mini

Estos nodos (ver figura 8) tienen la función de monitorear empleando un sensor de gas metano y visualizar el valor en una pantalla Oled I2C, también tienen la función de ser repetidor ya que puede enviar datos de otros nodos padre dentro de la red Mesh.

Para la configuración dentro de la red Mesh hay que definir el nombre de la red y contraseña que es la misma que en todos los nodos, además de habilitar la función de envío de datos, así como la tarea para enviar los datos.

En la función de envío de datos se realiza la lectura analógica del canal A0 del ESP32-S2, la resolución del ADC es de 13 bits y el ajuste para obtener la lectura del sensor de gas metano.

En la configuración para iniciar la red Mesh se define el nombre de la red y contraseña y el puerto.

Se añade y se configura la tarea a realizar y también se habilita tanto el envío de datos como la recepción de datos.

En el lazo principal se actualiza continuamente el estado de la red.

Otra función principal del nodo padre intermedio es la de recibir datos de otros nodos y retransmitir los datos hacia el nodo raíz, para ello utiliza el comando `mesh.sendSingle`.

```
void receivedCallback( uint32_t from, String &msg ) {  
  mesh.sendSingle( from, msg );  
}
```

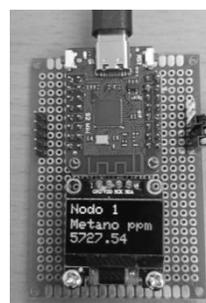


Figura 8. Nodo padre con ESP32-S2 mini

Nodos padre sin conexión descendente con ESP32-s2 mini

Estos nodos padre solo tienen la función de monitorear empleando un sensor de gas metano y visualizar el valor en una pantalla Oled I2C. La configuración es la misma que la de un nodo padre intermedio con la diferencia de que no se la activa la recepción de datos continua ni el reenvío de datos.

Por lo tanto la función principal de estos nodos es la de envío de datos.

2.3 Broker MQTT con Mosquitto

MQTT es popular porque es liviano y proporciona una forma efectiva de intercambiar datos entre varios dispositivos, incluso en un entorno con poco ancho de banda.

Los dispositivos pueden enviar y recibir datos a través de un broker MQTT. El broker no almacena ni modifica y genera ningún dato. Solo transmite los mensajes entre varios dispositivos que están conectados a él. Los dispositivos se comunican entre sí mediante un sistema de publicación y suscripción. Un dispositivo puede publicar mensajes sobre un tópico específico. Los dispositivos que están suscritos a ese tópico pueden recibir el mensaje.

Para implementar el Broker MQTT se utiliza Mosquitto ya que este se puede instalar fácilmente en un sistema operativo de Raspberry Pi, cabe mencionar que se pretende utilizar otros Broker más avanzados. Los comandos para instalar Mosquitto son:

```
sudo apt install -y mosquitto
sudo apt install -y mosquitto-clients
```

Una vez instalado en la Raspberry se puede consultar el estado usando el comando:

```
sudo systemctl status mosquitto.service
```

Para monitorear los mensajes hacia el broker MQTT se utiliza Node-Red, para ello se emplean los nodos de conexión MQTT in y MQTT out.

3. RESULTADOS

Al momento de realizar este artículo se tienen avances en la implementación de la red Mesh, cabe resaltar que las pruebas se han hecho fuera de la planta de tratamiento por lo que se trabaja en la implementación de la boya de sensado que permitirá observar si hay más latencia en condiciones de presencia de gran cantidad de agua y gases.

En la figura 9 se observa el nodo raíz conectado al Broker MQTT local que se implementa en una Raspberry, este Broker es monitoreado a través de Node-Red, cabe resaltar que los mensajes recibidos por el nodo raíz se pueden monitorear a través de un Broker público, a una app o página web.

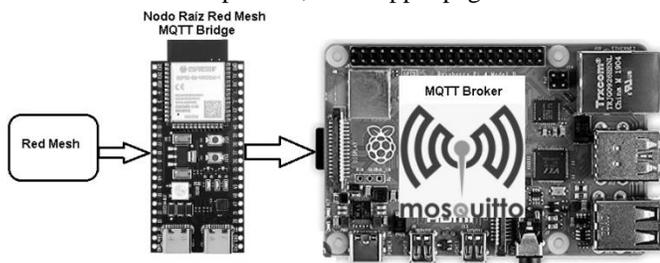


Figura 9. Conexión del nodo raíz con Mosquitto.

En la tabla 1 se muestran las pruebas realizadas considerando el número de nodos en la red Mesh, se consideraron distancias cortas entre los nodos de no más de 100 mts para las pruebas de conexión, una de las pruebas que se pretende realizar es hacer una red de 2 capas, esto es, un nodo raíz con 6 nodos padre para verificar si hay menos latencia con los datos.

Tabla 1. Pruebas en Red Mesh de 3 capas.

Tipo de red Mesh	Nodos Raíz	Nodos padre intermedios	Nodos padre sin conexión descendente	Latencia
2 capas	1	1	0	Nula
3 Capas	1	1	1	Baja
3 Capas	1	3	1	Baja
3 Capas	1	3	3	Intermitente

3.1 MQTT en nodo raíz

El nodo raíz es el encargado de conectarse a un Broker local para ser monitoreado a través de mensajes MQTT localmente o a través de un Broker en la nube. Para implementar MQTT en el nodo raíz se añaden dos librerías PubSubClient y WiFiClient, además de definir la IP y el puerto del Broker local que en este caso se utilizó el Broker Mosquitto.

```
IPAddress getLocalIP();
IPAddress myIP(0,0,0,0);
IPAddress mqttBroker(192, 168, 1, 1);
WiFiClient wifiClient;
PubSubClient mqttClient(mqttBroker, 1883, mqttCallback, wifiClient);
```

Además se activa la función de mqttCallback para la manipulación de los mensajes MQTT.

Para suscribirse a un tópico de MQTT se utiliza la función mqttClient.subscribe, como parámetro se introduce el tópico al que se suscribe.

Para publicar hacia un tópico de MQTT se utiliza la función mqttClient.publish, como parámetros se introduce el tópico al que publicara un dato además el propio mensaje.

En la figura 10 se observa la Dashboard creada en Node-Red, se aprecian los 6 nodos de monitoreo para las zonas 1 y 2 de la PTAR.

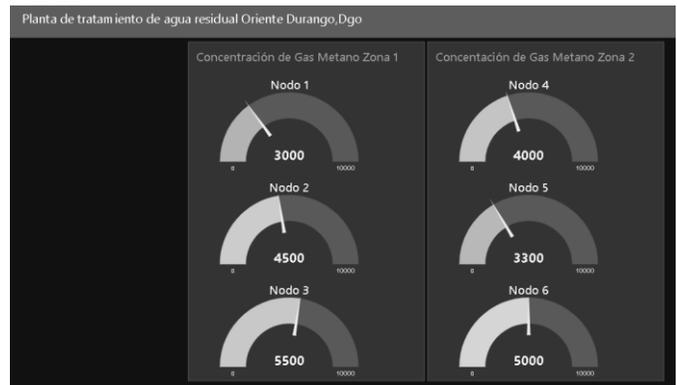


Figura 10. Dashboard para el monitoreo de la Red Mesh.

3.2 Boya de sensado

Para la implementación de la boya se tienen pruebas de alimentación mediante baterías de tipo Li-ion de 3.7V como se muestra en la figura 11, se pretende añadir un módulo de recarga de batería mediante panel solar que contendrá la boya. Para la estructura de la boya se trabaja en el diseño que sea compacto, resistente a la intemperie y con sistema de flotación. Se trabaja en ser eficientes en los periodos de sensado para así optimizar el consumo de energía, esto es, programar ciclos de lectura de datos en los periodos con mayor presencia de gases en la PTAR.

El sistema de boya flotante que se pretende implementar al no ser sumergible evita constante mantenimiento a los sensores, pero si requiere de materiales adecuados ya que estará expuesto a aguas grises y al sol principalmente.

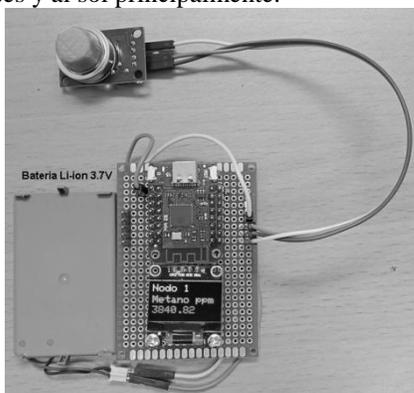


Figura 11. Nodo de monitoreo con alimentación autónoma.

4. CONCLUSIONES

Uno de los objetivos primordiales de este prototipo es realizar un sistema de alerta de alta concentración de gas metano en la PTAR oriente de la ciudad de Durango, empleando sistemas embebidos avanzados capaces de implementar una Red de sensores haciendo uso de Tecnología de redes Mesh y mensajes MQTT, dicho objetivo se logró con el uso de módulos ESP32-S2 y ESP32-S3.

La red Mesh implementada consta de 3 capas con módulos ESP32-S2 y ESP32-S3, dicha red presenta ventajas sobre otro tipo de redes que emplea módulos RF, principalmente el consumo de energía es menor.

Como siguiente trabajo se pretende incrementar el número de nodos Mesh y comprobar las ventajas que pueda tener el implementar redes Mesh con auto conexión hacia un nodo raíz y comparar el funcionamiento con redes de sensores que empleen nodos RF para su implementación.

También se concluye que el uso de mensajes MQTT para el desarrollo del proyecto es el adecuado ya que al poder trabajar con un Broker local los tiempos de latencia se reducen.

Se pretende también añadir las funcionalidades de almacenamiento en el Broker local para así tener datos relevantes de la PTAR en cuestión de emisión de gas Metano.

5. REFERENCIAS

- [1] Espressif Systems (2023), ESP32-S2 Series, Disponible: https://www.espressif.com/sites/default/files/documentation/esp32-s2_datasheet_en.pdf, Fecha de acceso: 20 de Junio 2023
- [2] Espressif Systems (2023), ESP32-S3 Series, Disponible: https://www.espressif.com/sites/default/files/documentation/esp32-s3_datasheet_en.pdf, Fecha de acceso: 20 de Junio 2023
- [3] Aguas del Municipio de Durango AMD (2012), Saneamiento, operación y mantenimiento, Disponible: https://www.gob.mx/cms/uploads/attachment/file/149692/Saneamiento_Operaci_n_y_Mantenimiento.pdf, Fecha de acceso: 30 de Junio de 2023
- [4] CONAGUA (2012), Manual de Agua Potable, Alcantarillado y Saneamiento Diseño de Plantas de Tratamiento de Aguas Residuales Municipales: Zonas Rurales, Periurbanas y Desarrollos Ecoturísticos, Disponible: <https://files.conagua.gob.mx/conagua/mapas/SGAPDS-1-15-Libro31.pdf>, Fecha de acceso: 30 de Junio de 2023
- [5] Espressif (2023), ESP WiFi Mesh, Disponible: <https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-guides/esp-wifi-mesh.html>, Fecha de acceso: 30 de Junio de 2023
- [6] R. Sharma (2023), Getting Started With ESP-MESH and ESP-8266, Disponible: <https://www.instructables.com/Getting-Started-With-ESP-MESH-and-ESP-8266/>, Fecha de acceso: 1 de Julio de 2023
- [7] L. Llamas (2016), Detector de gases con Arduino y la familia de sensores MQ Disponible: <https://www.luisllamas.es/arduino-detector-gas-mq/>, Fecha de acceso: 1 de Julio de 2023
- [8] Reyes, L (2021), Sistema de clasificación de agua jabonosa empleando redes neuronales, Universidad Politécnica de Durango, ELECTRO 2021, ISSN=1405-2172, Vol.43
- [9] Reyes, L (2022), Monitoreo para sistema de clasificación de agua residual doméstica empleando Node-Red, Universidad Politécnica de Durango, ELECTRO 2022, ISSN=1405-2172, Vol.44