

# Sensado y Control de Procesos para una Planta de Remoción de Arsénico en Pipinas

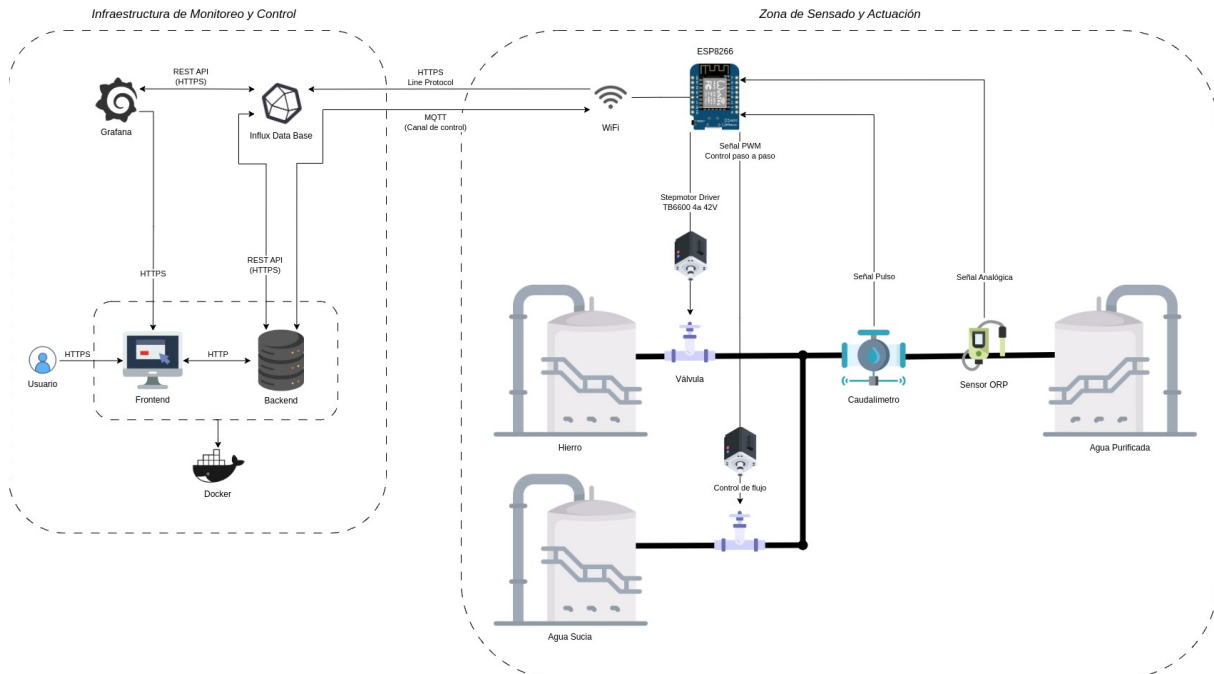
*LEICI + UIDET*

## Avance

1. Arquitectura del Sistema.....	2
2. Desarrollo y Avances.....	4
2.1. Hardware.....	4
2.2. Software.....	4
2.2.1. Firmware del MCU.....	5
2.2.2. Desarrollo de frontend.....	6
2.2.3. Desarrollo de backend.....	6
2.2.4. Despliegue de la Aplicación.....	8
3. Próximos Pasos.....	8
4. Conclusión.....	9

# 1. Arquitectura del Sistema

## Esquema General



El sistema se divide en dos bloques funcionales claramente diferenciados,

- **Zona de Sensado y Actuación**

En este bloque se agrupan todos los elementos relacionados con la adquisición y procesamiento de datos en el campo. Los sensores (ORP y caudalímetro) capturan las señales físicas, que son enviadas al microcontrolador ESP8266. Este MCU procesa y filtra las señales analógicas, se conecta a la red WiFi y transmite los datos a InfluxDB mediante solicitudes HTTPS. Además, el bloque integra los elementos de actuación: cuando el backend detecta la necesidad de intervención, envía mensajes a través de un broker MQTT al microcontrolador para gestionar los actuadores del sistema.

- **Infraestructura de Monitoreo y Control**

Este bloque centraliza el análisis y gestión del sistema. El backend, desarrollado en Node.js con Express, consulta los datos almacenados en InfluxDB para detectar tendencias o anomalías, en base a este análisis se generan alertas que se muestran en el frontend o se envían comandos de control al microcontrolador mediante un canal bidireccional MQTT.

## **Flujo de Datos,**

### **1. Inicio en los sensores**

Los sensores instalados en la planta capturan parámetros críticos y los envían al microcontrolador.

### **2. Procesamiento en el MCU**

El ESP8266 adquiere y procesa las señales analógicas, aplicando algoritmos de filtrado para obtener lecturas precisas. Una vez procesados, el MCU se conecta a la red WiFi y envía estos datos a InfluxDB mediante HTTPS.

### **3. Almacenamiento y visualización**

InfluxDB almacena los datos históricos y en tiempo real, y Grafana consulta esta base mediante su API para generar dashboards dinámicos, que luego se integran en el frontend y permiten a los usuarios visualizar la información de forma clara y actualizada.

## **Flujo de Control,**

El control y la toma de decisiones se centralizan en el backend, que se encarga de,

- **Análisis de datos**

El backend consulta InfluxDB para extraer datos históricos y actuales, y aplica algoritmos de procesamiento que permiten detectar anomalías o tendencias.

- **Generación de alertas y envío de comandos**

En caso de detectar condiciones que requieran intervención, el backend genera alertas o comandos de control.

Esta arquitectura modular, que separa la adquisición de datos en campo de la lógica de análisis y control centralizada, permite que el sistema sea escalable, robusto y fácil de mantener, asegurando que el microcontrolador se concentre en recopilar y transmitir datos mientras el backend gestiona el procesamiento avanzado y la intervención remota.

## 2. Desarrollo y Avances

El desarrollo del sistema se ha abordado en dos grandes secciones: hardware y software. A continuación, se detalla el avance logrado en cada una.

### 2.1. Hardware

En esta sección, se ha reutilizado la PCB del proyecto anterior, mencionada en el informe “plan-inicial”, la cual ha demostrado ser confiable en entornos de laboratorio. El montaje se realizó conectando el microcontrolador a la PCB y alimentando el sistema (provisoriamente conectado el MCU a un puerto USB de una computadora). Además, la PCB cuenta con una entrada para un sensor analógico, inicialmente diseñada para un sensor de pH, actualmente, esta se utiliza para conectar el sensor ORP. *\* Agregar imagen del montaje. \**

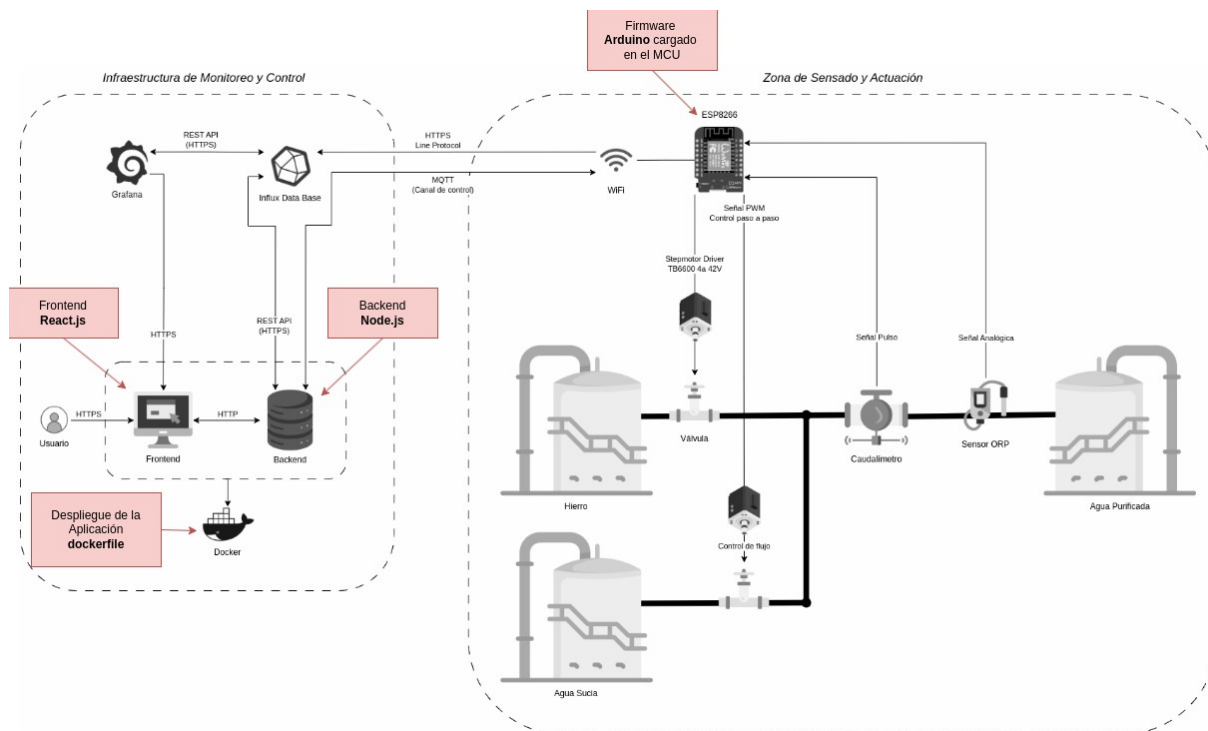
El firmware, cargado en el MCU mediante código Arduino, permite adquirir los datos analógicos de los sensores, procesarlos mediante algoritmos de filtrado (como un promedio móvil) y enviarlos a InfluxDB.

Actualmente, se han realizado ensayos utilizando un generador de señales a la entrada analógica de la PCB para simular las condiciones reales, ya que aún no se dispone del sensor ORP en campo. Los resultados han demostrado la estabilidad en la lectura y la precisión en la conversión analógica-digital, estableciendo una base sólida para futuras ampliaciones.

*\* Es importante destacar que se está desarrollando una nueva PCB que integrará, además del sensor ORP, otros sensores y actuadores previstos en el esquema global del sistema, lo que permitirá ampliar las funcionalidades de sensado y control.*

### 2.2. Software

El desarrollo del software se divide en cuatro áreas fundamentales: el firmware del microcontrolador, la aplicación web (frontend) y el backend para análisis y control avanzado. Por último, la independización de la aplicación web basada en Docker. Cada uno de estos componentes ha sido diseñado para interactuar de forma modular y complementaria.



### 2.2.1. Firmware del MCU

El firmware, programado en Arduino, es responsable de la adquisición y procesamiento de la información de los sensores. La rutina de lectura continua captura el valor crudo en el pin analógico del microcontrolador y aplica un filtro basado en un promedio móvil de las últimas diez mediciones. De esta manera, entre el filtro por software mencionado y el filtro por hardware (circuito de filtrado integrado en PCB) se logra reducir las fluctuaciones y mejorar la precisión de la lectura.

Una vez procesado el dato, el ESP8266 se conecta a la red WiFi utilizando las credenciales predefinidas y sincroniza su reloj mediante servidores NTP, lo que resulta esencial para la validación de certificados y la correcta marca temporal en la base de datos. Con estas condiciones, el microcontrolador envía la información a InfluxDB mediante solicitudes HTTPS en formato line protocol<sup>1</sup>, utilizando la API Rest de InfluxDB. La autenticación se realiza a través de un token incluido en los headers, y el firmware gestiona de forma robusta los errores de conexión o de escritura, reportándolos en el monitor serie para facilitar la supervisión y el ajuste del sistema.

<sup>1</sup> Line Protocol es un formato de texto ligero y optimizado por InfluxDB para escribir datos de series temporales.

### 2.2.2. Desarrollo de frontend

La interfaz de usuario desarrollada con React, proporciona una experiencia moderna e intuitiva para la visualización de la información por parte del usuario. El frontend se centra en la presentación de dashboards o paneles que han sido generados en Grafana, el cual consulta a InfluxDB mediante su API Rest y utiliza un token de autenticación para acceder a los datos en tiempo real y generar paneles según los requerimientos.

Estos dashboards se integran en la aplicación web mediante iframes<sup>2</sup>, permitiendo que la información se actualice en tiempo real. Durante las pruebas locales, se verificó que los paneles se muestran correctamente y que la navegación es fluida, lo que garantiza una interfaz robusta y de fácil acceso para la monitorización del sistema.

*\* Agregar imagen de panel incrustado en la aplicación. \**

Además, se ha implementado la conexión con el backend para que, en función de los análisis y alertas generados, se puedan presentar notificaciones específicas y acciones de control en la interfaz. Esta conexión se establece a través de una API Rest: La aplicación web en React realiza peticiones HTTP a endpoints<sup>3</sup> específicos definidos en el servidor (backend).

### 2.2.3. Desarrollo de backend

Se han evaluado dos alternativas para el procesamiento de datos:

- Que el MCU realice parte del análisis y envíe el resultado a InfluxDB.
- Que el MCU se dedique únicamente a enviar datos crudos, dejando el procesamiento avanzado en manos del backend.

Se ha optado por la segunda alternativa, ya que permite mantener el firmware del MCU simple y robusto, delegando el análisis de datos y la generación de alertas a un servidor con mayores recursos. Entonces, el backend es el cerebro central del sistema, encargado de extraer, procesar y analizar los datos almacenados en InfluxDB, y de generar alertas o enviar comandos de control al microcontrolador.

---

<sup>2</sup> iframes son elementos HTML que permiten incrustar un documento HTML dentro de otro, facilitando la inclusión de contenido externo.

<sup>3</sup> Endpoints son puntos de acceso a una API que permiten la comunicación entre clientes y servidores a través de solicitudes HTTP.

El backend está desarrollado en Node.js con Express, lo que facilita la consulta de datos a InfluxDB a través de su API Rest y, teniendo en cuenta la posibilidad de integrar a futuro la biblioteca MQTT.js, permite la comunicación bidireccional con el microcontrolador en caso de requerir acciones de control.

El proceso en el backend se resume en los siguientes pasos:

- **Definición de la Consulta:** Se arma una consulta (usando InfluxQL) que extrae datos históricos y en tiempo real de la base de datos InfluxDB.
- **Solicitud HTTP:** Mediante la librería Axios, el backend envía solicitudes GET (o POST) a la API Rest de InfluxDB, incluyendo la consulta y el token de autenticación en los headers.
- **Procesamiento de la Respuesta:** La respuesta, en formato JSON, se procesa para extraer y formatear la información. Esto incluye la aplicación de algoritmos para detectar anomalías (por ejemplo, verificar si el valor ORP supera un umbral crítico) y la generación de alertas.
- **Exposición del endpoint:** Se crea un endpoint en Express que permite al frontend consultar los datos procesados, recibiendo la información en formato JSON para su visualización o para disparar notificaciones en tiempo real en el frontend.
- **Comunicación Bidireccional:** En caso de que se detecte la necesidad de intervenir, el backend puede enviar comandos al microcontrolador a través de un canal MQTT seguro (preferiblemente con TLS).

Esta arquitectura garantiza que el procesamiento avanzado de datos se realice en un entorno centralizado, mientras que el microcontrolador se enfoca únicamente en la adquisición y transmisión de datos.

#### 2.2.4. Despliegue de la Aplicación

La estrategia de despliegue del sistema contempla la dockerización de los distintos componentes para facilitar su gestión, escalabilidad y migración a entornos de producción. Inicialmente, se ha dockerizado el frontend, lo que permite un despliegue independiente y escalable. Recientemente, se ha extendido la dockerización al backend, lo que ofrece ventajas como actualizaciones de forma independiente de los componentes, mayor seguridad y escalabilidad y acceso remoto desde otros dispositivos.

Esta solución modular y contenedorizada refuerza la robustez operativa y la eficiencia del sistema, sentando las bases para futuras ampliaciones y mejoras.

### 3. Próximos Pasos

Los pasos siguientes en el desarrollo del proyecto son los siguientes:

- **Especificar Detalles de Análisis en el Backend:** Definir los parámetros y algoritmos para analizar los datos y generar alertas o comandos de control, estableciendo umbrales críticos (por ejemplo, cuando el valor ORP supere cierto límite).
- **Desarrollar la Nueva PCB:** Completar la fabricación de una nueva PCB que permita la integración de otros sensores y actuadores, conforme al esquema global del sistema.
- **Implementar la Comunicación Bidireccional:** Una vez disponible la nueva PCB, desarrollar la comunicación mediante MQTT entre el backend y el MCU para gestionar el control remoto de los actuadores, y actualizar el firmware del MCU para incorporar esta funcionalidad.
- **Definir el Despliegue Final de la Aplicación:** Consolidar la dockerización y migrar la solución a un entorno de producción (por ejemplo, en la infraestructura de la Facultad), asegurando el acceso desde dispositivos externos y la integración con la base de datos institucional.



## **4. Conclusión**

Este informe de avance refleja el progreso obtenido en la validación del hardware y el desarrollo del software, estableciendo una base sólida para la futura integración de la comunicación entre backend y el MCU, que permitirá un análisis avanzado y el control bidireccional del sistema. La estrategia adoptada, donde el MCU se dedica exclusivamente a la adquisición y transmisión de datos, mientras que el procesamiento complejo se delega en el backend, garantiza la robustez, escalabilidad y eficiencia del proyecto. La solución final será adaptable a las necesidades de la planta de remoción de arsénico en Pipinas, permitiendo la incorporación gradual de nuevas funcionalidades y asegurando un sistema de monitoreo y control seguro y centralizado.