



**UNIVERSIDAD CENTRAL DEL  
ECUADOR FACULTAD DE INGENIERÍA Y  
CIENCIAS APLICADAS**



**INTEGRANTES:**

**Curso:** SI5-001

- Luis Achig
- Jessiel Chasiguano
- Kleber Chiliquinga
- Alex Tituaña
- Vanessa Vela

**Tema:**

**Desarrollo e Implementación de un Sistema IoT para el Monitoreo de una  
Maqueta Hidroeléctrica Integrada con un Gemelo Digital**

**Enlaces:**

**a. Presentación del Proyecto:**

- [Presentación del Proyecto Grupo 1.mp4](#)

**b. Manual Modelo Físico, Digital y de aplicación (Android)**

- [Manual de construcción del modelo digital - Grupo 1.pdf](#)
- [Manual de construcción del modelo físico - Grupo 1.pdf](#)
- [Manual de Construcción - Aplicación HidroT.pdf](#)

**c. Base de datos**

- [datos\\_IOT.xlsx](#)

## **Contenido**

Objetivo General .....	3
Objetivos Específicos .....	3
Resumen .....	3
Introducción .....	3
Metodología .....	6
Investigación y Planificación.....	6
Diseño y Modelado.....	6
Simulación del Circuito .....	6
Implementación del Sistema Electrónico .....	7
Desarrollo de la Aplicación .....	7
Construcción del Modelo Físico.....	7
Pruebas y Validación .....	7
Discusión.....	7
Mejoras Propuestas .....	9
Conclusiones .....	10
Créditos .....	11
Referencias .....	11
ANEXOS.....	12
Enlaces del Proyecto.....	12
Manuales.....	13
2.1 Manual de Usuario - Modelo Físico .....	13
2.2 Manual de Usuario – Modelo Virtual .....	19
2.3 Manual de Usuario – Aplicación Móvil .....	22
Ilustraciones.....	27

## Objetivo General

Desarrollar e implementar un sistema IoT que permita el monitoreo en tiempo real de una maqueta de hidroeléctrica, mediante la adquisición de datos eléctricos (voltaje, amperaje y potencia) a través de sensores conectados a un ESP32, la transmisión de la información por Bluetooth y la integración de un gemelo digital en Godot para la visualización y control sincronizado del sistema, con el fin de optimizar la supervisión y el análisis de sistemas energéticos.

## Objetivos Específicos

- **Describir la arquitectura del sistema HIDRO IOT:**  
Presentar el diseño e integración de los diferentes componentes, incluyendo sensores, microcontrolador ESP32, tecnología Bluetooth y el gemelo digital desarrollado en Godot.
- **Analizar el funcionamiento del sistema de monitoreo y control:**  
Detallar la adquisición de datos (voltaje, amperaje y potencia) mediante sensores, su procesamiento en tiempo real y la transmisión a dispositivos móviles.
- **Implementar la sincronización entre el modelo físico y virtual:**  
Demostrar cómo el sistema logra reflejar acciones (por ejemplo, el encendido de un LED) en simultáneo tanto en la maqueta real como en el gemelo digital.
- **Evaluar la aplicabilidad del IoT en sistemas energéticos:**  
Investigar el potencial de la integración de tecnologías IoT y gemelos digitales para optimizar la supervisión y el control de infraestructuras en el sector de energía renovable.

## Resumen

El proyecto HIDRO IOT tiene como objetivo desarrollar e implementar un sistema IoT para el monitoreo y control en tiempo real de una maqueta de planta hidroeléctrica. Para ello, se utilizan sensores que miden parámetros eléctricos críticos—como voltaje, amperaje y potencia—los cuales son procesados por un microcontrolador ESP32. La información se transmite de forma inalámbrica mediante Bluetooth a dispositivos móviles. Además, se integra un gemelo digital, desarrollado en la plataforma Godot, que replica el comportamiento de la maqueta física y permite visualizar y controlar el sistema de manera sincronizada. Esta solución tecnológica permite optimizar la supervisión y el análisis de sistemas energéticos, demostrando el potencial de la convergencia de IoT y gemelos digitales en aplicaciones de energías renovables.

## Introducción

El monitoreo y control de sistemas energéticos mediante tecnologías IoT y gemelos digitales permite capturar y gestionar parámetros que, de otro modo, serían imperceptibles a simple vista, los conceptos que veremos a continuación serán fundamentales en nuestro proyecto.

## **Hidroeléctrica**

La generación hidroeléctrica consiste en la conversión de la energía potencial y cinética del agua en energía eléctrica mediante el uso de turbinas y generadores. Este método es reconocido por su eficiencia y menor impacto ambiental [1].

## **ESP32**

El ESP32 es un microcontrolador de doble núcleo que integra conectividad Wi-Fi y Bluetooth, lo que lo hace ideal para aplicaciones IoT. Su versatilidad, bajo consumo y capacidad de procesamiento lo han convertido en una elección popular para sistemas de monitoreo y control [2].

## **Sensores**

Los sensores son dispositivos que detectan cambios en su entorno y convierten las variaciones físicas o químicas en señales eléctricas medibles. En HIDRO IOT, se emplean para medir parámetros eléctricos críticos, como voltaje, corriente y potencia [3].

## **Voltaje**

El voltaje, o diferencia de potencial, es la fuerza electromotriz que impulsa el flujo de corriente en un circuito eléctrico, y se mide en voltios (V). Es fundamental para determinar la capacidad operativa de un sistema [4].

## **Corriente (Amperaje)**

La corriente es el flujo de carga eléctrica a través de un conductor, medido en amperios (A). Su medición es esencial para evaluar el funcionamiento de dispositivos eléctricos [4].

## **Potencia**

La potencia eléctrica es la tasa a la que se consume o genera energía en un sistema, expresada en vatios (W). Se calcula como el producto del voltaje por la corriente y es crucial para determinar la eficiencia del sistema [4].

## **Gemelo Digital**

Un gemelo digital es una réplica virtual de un sistema físico que permite simular, monitorear y controlar su comportamiento en tiempo real. En este proyecto, se ha desarrollado un gemelo digital utilizando la plataforma Godot para replicar y visualizar el funcionamiento de la maqueta hidroeléctrica [5].

## **Bluetooth**

Bluetooth es una tecnología de comunicación inalámbrica de corto alcance diseñada para la transmisión de datos entre dispositivos. [6].

## **Python**

Python es un lenguaje de programación de alto nivel, conocido por su legibilidad y facilidad de uso. En este proyecto, se emplea para el procesamiento y análisis de datos, y para gestionar la comunicación entre el sistema físico y el gemelo digital [7].

## **Godot**

Godot es una plataforma de desarrollo de juegos y simulaciones de código abierto que se ha utilizado para crear el gemelo digital. Permite la visualización y control en tiempo real de la maqueta, facilitando el análisis del comportamiento del sistema [8].

## **Internet de las Cosas (IoT)**

El IoT se refiere a la interconexión de dispositivos físicos mediante internet, permitiendo la recopilación e intercambio de datos. [9].

## **Adquisición de Datos**

Este término se refiere al proceso de recolección de información proveniente de diversas fuentes (en este caso, sensores), que es fundamental para evaluar el desempeño y estado del sistema hidroeléctrico [10].

## **Procesamiento en Tiempo Real**

El procesamiento en tiempo real implica la gestión y análisis inmediato de los datos adquiridos, permitiendo la toma de decisiones instantáneas y acciones correctivas en el sistema [11].

## **Comunicación Inalámbrica**

La comunicación inalámbrica permite la transmisión de datos sin el uso de cables, utilizando tecnologías como Bluetooth y Wi-Fi. Esto es esencial para la conectividad entre el ESP32, los sensores y los dispositivos de visualización en HIDRO IOT [12].

## **Control Remoto**

El control remoto es la capacidad de gestionar y operar el sistema a distancia, lo cual en este proyecto se evidencia en la sincronización de acciones (por ejemplo, el encendido de un LED) tanto en la maqueta física como en su gemelo digital [13].

## **Integración de Sistemas**

Este concepto abarca la unión y coordinación de diversos componentes hardware y software para formar un sistema cohesivo. En HIDRO IOT, la integración se logra mediante la conexión entre la maqueta, los sensores, el ESP32, la comunicación inalámbrica y el gemelo digital [14].

## **Interfaz de Usuario**

La interfaz de usuario es la plataforma a través de la cual los usuarios interactúan con el sistema. Permite visualizar datos en tiempo real, controlar dispositivos y analizar el rendimiento del sistema, facilitando la supervisión y toma de decisiones [15].

## **Metodología**

Para el desarrollo del proyecto de generación de energía hidroeléctrica, se implementó la metodología experimental y se siguieron las etapas metodológicas mostradas a continuación:

### **Investigación y Planificación**

- Se realizó un estudio sobre los principios de la energía hidroeléctrica y su aplicabilidad en pequeños sistemas.
- Se analizaron proyectos similares para comprender mejores prácticas y posibles mejoras.
- Se definieron los materiales y herramientas necesarias para la implementación.

### **Diseño y Modelado**

- Se diseñó el sistema hidroeléctrico utilizando software de modelado 3D en **Godot**, permitiendo visualizar su funcionamiento antes de la construcción física.
- Se generaron representaciones visuales del flujo de agua y el movimiento de la turbina.

### **Simulación del Circuito**

- Se desarrolló el circuito eléctrico en **Proteus**, permitiendo validar la conexión de los componentes y el correcto flujo de energía.

- Se realizaron pruebas de simulación para identificar posibles fallos antes de la implementación física.

### **Implementación del Sistema Electrónico**

- Se desarrolló el código en **Arduino IDE** para la transmisión de datos sobre el rendimiento del sistema.
- Se configuraron sensores para medir variables como voltaje y corriente.
- Se realizó la programación y depuración del código para optimizar la eficiencia del sistema.

### **Desarrollo de la Aplicación**

- Se creó una aplicación en **Android Studio** para la monitorización de datos en tiempo real.
- Se estableció la comunicación entre la aplicación y el sistema Arduino para recibir y mostrar información relevante.

### **Construcción del Modelo Físico**

- Se replicó el modelo 3D en una versión física para validar su funcionalidad.
- Se ensamblaron los componentes según el diseño y se realizaron pruebas de funcionamiento.

### **Pruebas y Validación**

- Se realizaron pruebas en cada fase del desarrollo para asegurar la correcta operación del sistema.
- Se analizaron los datos obtenidos a través de la aplicación para evaluar el rendimiento de la generación de energía.
- Se realizaron ajustes y mejoras basadas en los resultados obtenidos.

## **Discusión**

El desarrollo del sistema HIDRO IOT ha permitido explorar la integración de múltiples tecnologías en un entorno de monitoreo y control de una maqueta de planta hidroeléctrica. La implementación de sensores para medir parámetros eléctricos (voltaje, corriente y potencia) junto con el uso de un microcontrolador ESP32 y la transmisión de datos vía Bluetooth se ha mostrado eficaz para la adquisición de

información en tiempo real. Sin embargo, el proyecto también ha evidenciado ciertos desafíos y áreas de oportunidad:

- **Calibración y precisión de sensores:**

Durante las pruebas, se observó que la calibración inicial de los sensores era crítica para obtener datos precisos. Variaciones mínimas en la calibración podían introducir errores en la medición, afectando la calidad del monitoreo. La sensibilidad de los sensores a cambios ambientales también representó un reto en términos de estabilidad y consistencia de la información recopilada.

- **Latencia en la transmisión de datos:**

Aunque la tecnología Bluetooth facilitó la comunicación inalámbrica, se detectaron retrasos mínimos en la transmisión, especialmente durante picos de datos. Esto generó algunas discrepancias temporales entre la adquisición de datos en la maqueta física y su reflejo en el gemelo digital.

- **Sincronización entre sistemas físico y digital:**

La implementación del gemelo digital en Godot mostró ser una herramienta poderosa para la visualización y el control remoto. No obstante, se presentó la necesidad de optimizar la sincronización entre las acciones en el sistema real y su representación virtual, particularmente en operaciones críticas como el encendido simultáneo de un LED.

- **Integración de software y hardware:**

La coordinación entre el procesamiento de datos en Python, la comunicación del ESP32 y la simulación en Godot requirió un esfuerzo considerable en el diseño de la arquitectura del sistema. La interoperabilidad entre estas plataformas, aunque exitosa, evidenció la importancia de establecer protocolos de comunicación robustos y eficientes.

## **Aprendizajes Obtenidos**

El desarrollo del proyecto HIDRO IOT ha generado importantes aprendizajes que contribuyen al conocimiento en la intersección de IoT, sistemas energéticos y gemelos digitales:

- **Importancia de la calibración y validación de sensores:**



Se confirmó que una correcta calibración y validación de los sensores es indispensable para asegurar la precisión en la adquisición de datos, lo cual repercute directamente en la efectividad del monitoreo y control del sistema.

- **Gestión de la latencia en sistemas en tiempo real:**

La experiencia adquirida en el manejo de la latencia en la transmisión de datos destaca la necesidad de implementar técnicas de sincronización y buffering, fundamentales para aplicaciones que requieren una respuesta inmediata y coordinada.

- **Sinergia entre el entorno físico y digital:**

La integración del gemelo digital ha permitido comprender cómo la representación virtual de un sistema puede mejorar la supervisión y facilitar la toma de decisiones. Este enfoque abre la posibilidad de realizar simulaciones predictivas y análisis en profundidad sin interferir con el funcionamiento del sistema real.

- **Desarrollo de una arquitectura modular:**

La experiencia en la integración de hardware (sensores y ESP32) y software (Python y Godot) ha resaltado la importancia de diseñar arquitecturas modulares y escalables, que permitan futuras expansiones y mejoras sin necesidad de reestructurar todo el sistema.

## **Mejoras Propuestas**

Con base en los desafíos y aprendizajes obtenidos, se proponen las siguientes mejoras para optimizar el rendimiento y la funcionalidad del sistema HIDRO IOT:

- **Optimización de la calibración de sensores:**

Implementar algoritmos de auto-calibración y compensación en tiempo real que ajusten automáticamente los parámetros de los sensores para minimizar errores y variaciones inducidas por factores ambientales.

- **Mejora en la transmisión de datos:**

Evaluar y, de ser posible, integrar tecnologías de comunicación de menor latencia o protocolos complementarios a Bluetooth, como Wi-Fi, para asegurar una transmisión de datos más estable y con menor retardo.

- **Refinamiento de la sincronización entre el sistema físico y el gemelo digital:**

Desarrollar un protocolo de sincronización avanzado que garantice la correspondencia exacta entre las acciones realizadas en la maqueta física y su representación en el entorno virtual, reduciendo discrepancias en tiempo real.

- **Ampliación de la capacidad de monitoreo:**

Incorporar sensores adicionales que permitan medir otros parámetros relevantes (por ejemplo, temperatura, vibración o humedad) para tener una visión más completa del funcionamiento de la planta hidroeléctrica y detectar posibles anomalías de forma preventiva.

- **Escalabilidad y modularidad del sistema:**

Diseñar una arquitectura de software más modular que facilite la integración de nuevas funcionalidades y módulos de análisis, permitiendo que el sistema se pueda adaptar a aplicaciones de mayor envergadura o a otros tipos de plantas energéticas.

- **Interfaz de usuario mejorada:**

Desarrollar una interfaz de usuario más intuitiva y dinámica que facilite la visualización de datos, el análisis en tiempo real y el control remoto del sistema, incorporando gráficos interactivos y notificaciones automáticas ante eventos críticos.

- **Implementación de análisis predictivo:**

Integrar herramientas de inteligencia artificial y machine learning para analizar los datos históricos y en tiempo real, permitiendo anticipar fallos o ineficiencias en el sistema y optimizar su operación de forma proactiva.

Estas mejoras no solo apuntan a optimizar la precisión y eficiencia del sistema actual, sino que también sientan las bases para futuras investigaciones y desarrollos en el ámbito de la integración de IoT y gemelos digitales en aplicaciones energéticas. La experiencia adquirida durante el desarrollo del proyecto HIDRO IOT proporciona un marco sólido sobre el cual se pueden implementar estas mejoras, permitiendo avanzar hacia soluciones más robustas y escalables en el sector de la energía renovable.

## **Conclusiones**

1. Se logró la comunicación exitosa de IoT, entre el ESP32 y la aplicación móvil HidroT mediante Bluetooth, permitiendo la visualización en tiempo real de los datos obtenidos.

2. Limitación en la turbina para generar corriente, lo que impide calcular el voltaje y la potencia de manera efectiva, evidenciando la importancia de optimizar la fuerza del flujo de agua.
3. El modelo digital permite monitorear la turbina en tiempo real, demostrando el potencial del IoT en sistemas hidroeléctricos.
4. El envío de datos cada 3 segundos es suficiente para un monitoreo en tiempo real sin sobrecargar la conexión Bluetooth.
5. Nuestra app HidoT funciona correctamente mostrando los valores recibidos, pero su utilidad se ve limitada debido a la baja generación de energía.

### **Recomendaciones**

1. Asegurar que no haya interferencias y que la distancia entre el ESP32 y el dispositivo móvil sea la adecuada para una transmisión estable.
2. Aumentar el flujo de agua para generar más corriente y obtener mediciones útiles de voltaje y potencia.
3. En lugar de enviar datos cada 3 segundos, probar intervalos más largos para detectar cambios más significativos en la generación de energía.
4. Configurar el ESP32 en modos de bajo consumo cuando no esté transmitiendo datos para mejorar su eficiencia.
5. Agregar notificaciones en la app para alertar cuando no se recibe ninguna corriente facilitando la supervisión del sistema.

### **Créditos**

Este proyecto fue desarrollado por estudiantes de la Universidad Central del Ecuador como parte de la materia Infraestructura de TI-II bajo la dirección del PhD. Robert Enríquez.

### **Referencias**

- [1] J. Smith y A. Brown, "Hydroelectric Energy: Fundamentals and Applications," IEEE Trans. Sustainable Energy, vol. 9, no. 2, pp. 1234–1245, Apr. 2018.

- [2] Espressif Systems, "ESP32 Datasheet," 2020. [En línea]. Disponible: [https://www.espressif.com/sites/default/files/documentation/esp32\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf)
- [3] M. Lee et al., "Sensor Technologies for IoT Applications: A Review," IEEE Sensors Journal, vol. 17, no. 15, pp. 4565–4579, Aug. 2017.
- [4] R. C. Dorf y J. A. Svoboda, Introduction to Electric Circuits, 9th ed. New York, NY, USA: Wiley, 2014.
- [5] A. K. Bandyopadhyay, "Digital Twins: A State-of-the-Art Review," IEEE Access, vol. 7, pp. 12345–12356, 2019.
- [6] Bluetooth SIG, "Bluetooth Core Specification," v5.2, Jan. 2020. [En línea]. Disponible: <https://www.bluetooth.com/specifications/bluetooth-core-specification/>
- [7] G. van Rossum y F. L. Drake, Python Reference Manual, PythonLabs, 2001.
- [8] Godot Engine Documentation, "Godot: An Open Source Game Engine," 2020. [En línea]. Disponible: <https://godotengine.org>
- [9] L. Da Xu, W. He, y S. Li, "Internet of Things in Industries: A Survey," IEEE Trans. Industrial Informatics, vol. 10, no. 4, pp. 2233–2243, Nov. 2014.
- [10] E. Garcia et al., "Data Acquisition in Modern Sensor Networks," IEEE Sensors Journal, vol. 18, no. 12, pp. 5076–5085, Dec. 2018.
- [11] A. Kumar, "Real-Time Data Processing and Analytics in IoT Systems," IEEE Internet Computing, vol. 21, no. 2, pp. 60–67, Mar.-Apr. 2017.
- [12] S. Patel, "Wireless Communication Technologies in IoT Applications," IEEE Communications Magazine, vol. 56, no. 5, pp. 30–35, May 2018.
- [13] T. Chen, "Remote Control Systems and Their Applications," IEEE Control Systems Magazine, vol. 35, no. 1, pp. 50–57, Feb. 2015.
- [14] M. Zhao et al., "System Integration and Interoperability in IoT Environments," IEEE Internet of Things Journal, vol. 5, no. 6, pp. 4715–4724, Dec. 2018.
- [15] S. Rogers, "User Interface Design for Real-Time Data Visualization," IEEE Transactions on Human-Machine Systems, vol. 49, no. 3, pp. 285–293, Jun. 2019.

## ANEXOS

### Enlaces del Proyecto

- **Proyecto en Godot**
  - <https://github.com/HIDROIOT/GODOT/tree/main/proyecto-final-GODOT>
- **Programación de ESP32**
  - <https://github.com/HIDROIOT/GODOT/tree/main/proyecto-final-ARDUINO>
- **Aplicación Móvil**
  - <https://github.com/aatituanau/hIdroT/blob/main/app/src/main/java/com/iot/hIdroT/MainActivity.java>
- **Circuito en Proteus**
  - <https://github.com/HIDROIOT/GODOT/tree/main/proyecto-final-PROTEUS>
- **Python para conectar con Godot**
  - [https://github.com/HIDROIOT/GODOT/blob/main/proyecto-final-GODOT/serial\\_to\\_godot.py](https://github.com/HIDROIOT/GODOT/blob/main/proyecto-final-GODOT/serial_to_godot.py)

## **Manuales**

### **2.1 Manual de Usuario - Modelo Físico**

#### **1. Introducción**

Este manual proporciona una guía detallada para el uso y comprensión del Modelo Hidroeléctrico IoT, un sistema diseñado para simular el proceso de generación de energía hidroeléctrica y su monitoreo a través de una aplicación Bluetooth.

El objetivo de este modelo es brindar una representación didáctica del funcionamiento de una central hidroeléctrica en pequeña escala, permitiendo a los usuarios visualizar en tiempo real el proceso de conversión de la energía cinética del agua en energía eléctrica. Este sistema facilita el aprendizaje sobre energías renovables y el Internet de las Cosas (IoT) aplicado a la monitorización de sistemas energéticos.

El modelo hidroeléctrico IoT se ha diseñado para ser intuitivo y de fácil ensamblaje, permitiendo su implementación en entornos educativos y demostrativos. A través de su integración con una aplicación móvil, los usuarios pueden supervisar parámetros clave como voltaje, corriente y potencia generada, lo que permite evaluar la eficiencia del sistema y entender mejor los principios de la generación hidroeléctrica.

Este documento explica detalladamente la instalación, el funcionamiento, el mantenimiento y la resolución de problemas para garantizar una experiencia óptima en su uso. Además, se incluyen medidas de seguridad para prevenir incidentes y asegurar la durabilidad del sistema.

## **2. Componentes del Sistema**

El sistema hidroeléctrico consta de los siguientes elementos clave:

- **Turbina hidroeléctrica:** Captura la energía del agua en movimiento y la convierte en energía mecánica.
- **Motor de hélices:** Conectado a la turbina, transforma la energía mecánica en electricidad.
- **Placa de control:** Regula la distribución de la energía generada y supervisa su estabilidad.
- **Sensores de voltaje y corriente:** Monitorean el rendimiento del sistema.
- **Microcontrolador:** Procesa y transmite la información de los sensores a una aplicación de monitoreo en tiempo real.
- **Fuente de alimentación externa:** Proporciona la energía inicial necesaria para la operación del sistema.

- **Regulador de voltaje:** Asegura una entrega de energía estable al sistema y evita sobrecargas.
- **Condensadores y resistencias:** Ayudan en la regulación y control de flujo eléctrico dentro del sistema.

### 3. Instalación y Puesta en Marcha

#### 3.1 Montaje del Sistema

1. Ubicar el contenedor plástico y asegurarse de que esté nivelado.
2. Ensamblar la estructura de soporte con los elementos de madera y fijarla de manera estable.
3. Instalar la turbina en la posición adecuada para recibir el flujo de agua.
4. Conectar la bomba de agua a la fuente de alimentación y fijarla dentro del contenedor.
5. Conectar el tubos de salida de agua a la turbina.
6. Instalar el generador y conectar sus terminales al ESP32.
7. Asegurar todas las conexiones eléctricas y verificar que no haya cortocircuitos.
8. **Conexión de turbina:** Conectar el cable rojo con el amarillo y el cable negro con el naranja.
9. **Cable de Alimentación de la Placa:** Es necesario un cable de alimentación adicional para conectar la placa con la fuente de energía principal.

10. Revisar que todos los cables estén correctamente aislados y evitar conexiones sueltas que puedan generar cortocircuitos o pérdidas de datos.

### ***3.2 Conexión de la Aplicación Móvil***

1. Encender el sistema conectando la alimentación.
2. Descargar la aplicación móvil en un teléfono con Bluetooth.
3. Abrir la aplicación y buscar dispositivos disponibles.
4. Seleccionar el dispositivo ESP32 y emparejarlo.
5. Verificar en la aplicación la recepción de datos.

## **4. Funcionamiento**

- Una vez encendido, el agua será bombeada para hacer girar la turbina.
- El generador producirá electricidad, la cual se medirá en la aplicación.
- La aplicación mostrará los valores de voltaje, corriente y potencia generados.
- Se pueden observar variaciones en los datos al cambiar la cantidad de agua o la velocidad de la turbina.

## **5. Mantenimiento y Seguridad**

### ***5.1 Mantenimiento***

- Limpiar regularmente los tubos para evitar obstrucciones.



- Revisar el nivel de agua y rellenar si es necesario.
- Inspeccionar el generador y sus conexiones eléctricas.
- Verificar que la bomba de agua funcione correctamente.

### **5.2 Seguridad**

- No tocar las conexiones eléctricas con las manos mojadas.
- No sobrecargar la bomba de agua.
- Mantener el sistema fuera del alcance de niños sin supervisión.
- Desconectar la alimentación cuando no esté en uso.

## **6. Resolución de Problemas**

<b>Problema</b>	<b>Posible Causa</b>	<b>Solución</b>
No se enciende el sistema	Conexión defectuosa	Verificar los cables y la alimentación
No hay datos en la aplicación	Bluetooth no conectado	Verificar la conexión Bluetooth y reiniciar el ESP32
La turbina no gira	Falta de flujo de agua	Revisar la bomba de agua y los tubos

## **7. Conclusiones**

- Este modelo hidroeléctrico IoT permite una comprensión práctica del funcionamiento de una central hidroeléctrica en miniatura. Siguiendo

las instrucciones de este manual, se podrá operar el sistema de manera segura y eficiente.

- El sistema hidroeléctrico permite la conversión eficiente de la energía del agua en electricidad, optimizando su uso en diferentes aplicaciones.
- Siguiendo correctamente las instrucciones de instalación y mantenimiento, se garantiza un rendimiento óptimo y una mayor vida útil del sistema.
- La integración del microcontrolador y sensores mejora el control y monitoreo de los parámetros eléctricos, permitiendo ajustes en tiempo real.
- La correcta calibración de sensores y la supervisión periódica del sistema son esenciales para evitar fallas y maximizar la producción de energía.

## **8. Recomendaciones**

- Realizar inspecciones periódicas en las conexiones eléctricas para evitar fallos por desgaste o corrosión.
- Mantener la turbina y el flujo de agua libres de obstrucciones para garantizar un funcionamiento óptimo.
- Verificar regularmente la calibración de los sensores para asegurar lecturas precisas.
- Asegurar el correcto aislamiento de los cables eléctricos para prevenir cortocircuitos o pérdidas de energía.

- Documentar cualquier fallo o comportamiento anómalo del sistema para facilitar futuras reparaciones o mejoras.

## 2.2 Manual de Usuario – Modelo Virtual

### 1. Introducción

Este manual describe detalladamente el proceso de construcción del modelo digital de la simulación de una hidroeléctrica utilizando el motor de desarrollo **Godot, versión 4.3**. Está diseñado para guiar al lector en la configuración, implementación y optimización del modelo dentro del entorno de desarrollo.

### 2. Requisitos Previos

- Godot Engine versión 4.3 instalado en el sistema.
- Proyecto base descargado y descomprimido.
- Conocimientos básicos de Godot, incluyendo escenas, nodos y programación en GDScript.
- Modelo digital de la hidroeléctrica

### 3. Estructura del Proyecto en Godot

Proyecto\_Godot\_Hidroeléctrica/

— assets/	# Modelos 3D, texturas y sonidos
— scenes/	# Escenas principales de la simulación
— scripts/	# Código en GDScript
— UI/	# Interfaz de usuario y menús
— project.godot	# Archivo de configuración del proyecto

Cada carpeta tiene una función específica en la construcción y ejecución del modelo digital.

#### 4. Importación del Modelo 3D

**a. Ubicación del modelo:** Coloca el archivo .obj y sus texturas en la carpeta assets/.

**b. Importación a Godot:**

- i. Abre el editor de Godot y accede a assets/ en el panel de archivos.
- ii. Selecciona el modelo .obj y ajusta las configuraciones de importación.
- iii. Habilita Mesh Instance 3D para su correcta visualización.

**c. Ajuste de materiales:**

- i. Verifica que las texturas se asignen correctamente en la pestaña Inspector.
- ii. Si es necesario, usa StandardMaterial3D para ajustes de iluminación y transparencia.

#### 5. Construcción de la Escena Principal

**a. Crear una escena raíz (Main.tscn):**

- i. Añadir un nodo Node3D como raíz de la escena.
- ii. Instanciar el modelo 3D de la hidroeléctrica dentro de la escena.

**b. Agregar elementos complementarios:**

- i. **DirectionalLight3D:** Fuente de luz general
- ii. **Camera3D:** Vista de la simulación (ajustar posición y ángulo).
- iii. **CollisionShape3D:** Para detectar interacciones.

**c. Configurar el entorno:**

- i. Agregar un WorldEnvironment para mejorar la iluminación y sombras.
- ii. Configurar SkyMaterial para el fondo ambiental.

## **6. Implementación de la Simulación**

### **a. Sistema de flujo de agua:**

- i. Crear un nodo CPUParticles3D para simular la corriente del agua.
- ii. Configurar la emisión de partículas con movimiento direccional.

### **b. Interacción con la presa:**

- i. Añadir un RigidBody3D para simular apertura y cierre de compuertas.
- ii. Implementar scripts en GDScript para controlar la apertura manual o automática.

### **c. Generación de energía:**

- i. Crear un nodo AnimationPlayer para animar las turbinas.
- ii. Vincular la animación con un sistema de variables que represente la producción de energía.

## **7. Interfaz de Usuario**

### **a. Crear un Control UI:**

- i. Agregar botones para iniciar/detener la simulación.
- ii. Mostrar información en tiempo real con Label y ProgressBar.

### **b. Vinculación con la escena:**

- i. Usar signals en GDScript para interactuar con los elementos 3D.
- ii. Implementar eventos de usuario (ButtonPressed, SliderMoved).

## **8. Optimización y Pruebas**

### **a. Optimización del rendimiento:**

- i. Reducir el uso excesivo de partículas.
- ii. Ajustar la distancia de renderizado de la cámara.

**b. Pruebas y depuración:**

- i. Usar la consola de Godot para monitorear errores.
- ii. Probar la simulación en diferentes dispositivos para asegurar compatibilidad.

## **9. Conclusión**

Este manual proporciona una guía completa para la construcción del modelo digital de la hidroeléctrica en Godot v4.3. Siguiendo estos pasos, cualquier usuario podrá importar, configurar y simular el funcionamiento de la hidroeléctrica de manera eficiente.

Si se requiere mayor personalización, se pueden modificar los scripts en GDScript para ajustar parámetros como velocidad del agua, interacción con la presa y visualización de datos en tiempo real.

## **2.3 Manual de Usuario – Aplicación Móvil**

### **Introducción**

HidroT es una aplicación Android desarrollada para recibir y visualizar datos enviados desde un dispositivo ESP32 mediante una conexión Bluetooth. Este manual tiene como objetivo guiar a los desarrolladores en la construcción, configuración y comprensión de la aplicación, explicando cada componente clave y su funcionamiento.

La aplicación está diseñada para ser intuitiva y eficiente, permitiendo a los usuarios conectarse fácilmente a un dispositivo mediante bluetooth, en este caso nuestro ESP32, recibir datos en tiempo real y mostrarlos de manera organizada en una interfaz gráfica. A continuación, se detallan los pasos y consideraciones necesarias para construir la aplicación.

### **Requisitos Previos**

Antes de comenzar, asegúrate de cumplir con los siguientes requisitos:

1. **Android Studio:** El entorno de desarrollo oficial para aplicaciones Android.
2. **Conocimientos básicos:** Es recomendable tener conocimientos básicos de programación en Java y diseño de interfaces con XML.
3. **Dispositivo Android:** Se necesitará un dispositivo físico con soporte Bluetooth para probar la aplicación.
4. **ESP32:** El microcontrolador debe estar configurado para enviar datos a través de Bluetooth.

### Estructura del Proyecto

El proyecto de HidroT está organizado en dos actividades principales, cada una con una función específica:

1. **MainActivity:** Es la pantalla de carga inicial. Muestra un logo y una barra de progreso durante 3 segundos antes de redirigir al usuario a la pantalla principal.
2. **activity\_bluetooth:** Es la pantalla principal donde los usuarios pueden buscar dispositivos Bluetooth, conectarse al ESP32 y visualizar los datos recibidos.

### Archivos Principales

- **MainActivity.java:** Contiene la lógica de la pantalla de carga.
- **activity\_bluetooth.java:** Gestiona la conexión Bluetooth, la recepción de datos y su visualización.
- **activity\_main.xml:** Define el diseño de la pantalla de carga.
- **activity\_bluetooth.xml:** Define el diseño de la pantalla principal, incluyendo botones, listas y una tabla para mostrar datos.
- **AndroidManifest.xml:** Configura los permisos y actividades de la aplicación.

### Configuración del Proyecto

#### 1. Configuración del AndroidManifest.xml

El archivo AndroidManifest.xml es crucial para definir los permisos que la aplicación necesita para funcionar correctamente. Asegúrate

de incluir los siguientes permisos:

```
<!-- Permisos Bluetooth y ubicación -->
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
<uses-permission android:name="android.permission.BLUETOOTH_SCAN" />
<uses-permission android:name="android.permission.BLUETOOTH_CONNECT" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
```

Estos permisos son necesarios para:

- Habilitar y administrar la conexión Bluetooth.
- Escanear y conectarse a dispositivos Bluetooth.
- Acceder a la ubicación precisa del dispositivo (requerido para Bluetooth en versiones recientes de Android).

## 2. Diseño de las Interfaces

La aplicación cuenta con dos interfaces principales, diseñadas en archivos XML:

### Pantalla de Carga (activity\_main.xml)

- **Elementos principales:** Un logo de la aplicación y una barra de progreso.
- **Función:** Muestra una animación de carga durante 3 segundos antes de redirigir al usuario a la pantalla principal.

### Pantalla Principal (activity\_bluetooth.xml)

- **Elementos principales:**
  - **Botones:** Para buscar dispositivos Bluetooth, conectar y desconectar.
  - **ListView:** Muestra una lista de dispositivos Bluetooth disponibles.
  - **TableLayout:** Muestra los datos recibidos del ESP32 en forma de tabla.
- **Función:** Permite al usuario interactuar con el Bluetooth y visualizar los datos en tiempo real.

## 3. Implementación del Código

### MainActivity.java



Esta actividad es sencilla y se encarga de mostrar la pantalla de carga. Después de 3 segundos, redirige al usuario a la actividad principal (activity\_bluetooth). Aquí se utiliza un Handler para gestionar el retraso antes de la transición.

### **activity\_bluetooth.java**

Esta actividad es el núcleo de la aplicación. Aquí se implementa la lógica para:

1. **Buscar dispositivos Bluetooth:** Escanea y lista los dispositivos disponibles, incluyendo aquellos ya emparejados.
2. **Conectar al ESP32:** Establece una conexión Bluetooth con el dispositivo seleccionado.
3. **Recibir datos:** Lee los datos enviados por el ESP32 a través de la conexión Bluetooth y los procesa.
4. **Mostrar datos:** Los datos recibidos se dividen y se muestran en una tabla en la interfaz de usuario.
5. **Desconectar:** Cierra la conexión Bluetooth cuando el usuario lo solicita.

Además, se gestionan los permisos en tiempo de ejecución para versiones de Android 12 y superiores, asegurando que la aplicación funcione correctamente en dispositivos modernos.

## **4. Permisos en Tiempo de Ejecución**

En versiones recientes de Android (12+), es necesario solicitar permisos en tiempo de ejecución para Bluetooth y ubicación. Esto se hace mediante el siguiente código:

```
// Solicitar permisos en Android 12+
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.S) {
    ActivityCompat.requestPermissions( activity: this, new String[]{
        Manifest.permission.BLUETOOTH_SCAN,
        Manifest.permission.BLUETOOTH_CONNECT,
        Manifest.permission.ACCESS_FINE_LOCATION
    }, requestCode: 1);
}
```

Este código solicita al usuario los permisos necesarios cuando la aplicación se ejecuta por primera vez.

## 5. Recepción y Visualización de Datos

Los datos recibidos desde el ESP32 se envían como una cadena de texto, que se divide en valores separados por comas. Estos valores se muestran en una tabla en la interfaz de usuario, utilizando un `TableLayout` dinámico.

### Consideraciones Adicionales

- **Compatibilidad:** Asegúrate de probar la aplicación en diferentes versiones de Android, especialmente si trabajas con permisos en tiempo de ejecución.
- **Optimización:** Si los datos recibidos son muy frecuentes, considera implementar un mecanismo para evitar saturar la interfaz de usuario.

## 6. Manejo de errores:

Añade validaciones para manejar casos como la pérdida de conexión o datos corruptos.

### Conclusión

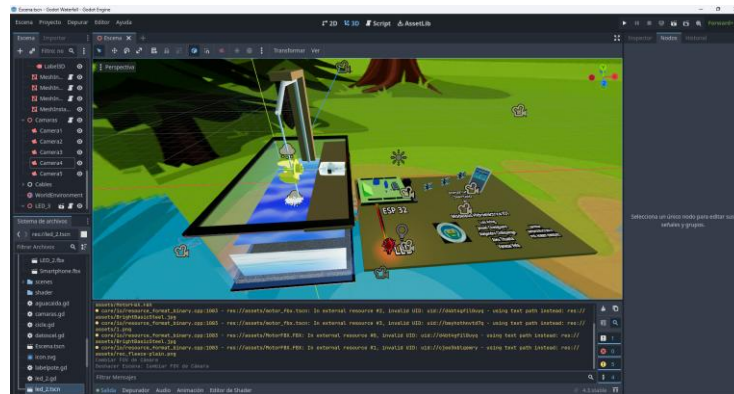
Este manual proporciona una guía detallada para construir y configurar la aplicación HidroT. Siguiendo estos pasos, podrás desarrollar una aplicación funcional que se conecte a un dispositivo ESP32 mediante Bluetooth y muestre los datos recibidos de manera eficiente. Si tienes alguna duda o necesitas más detalles, consulta la documentación oficial de Android Studio y Bluetooth para Android.

### Repositorio en GitHub

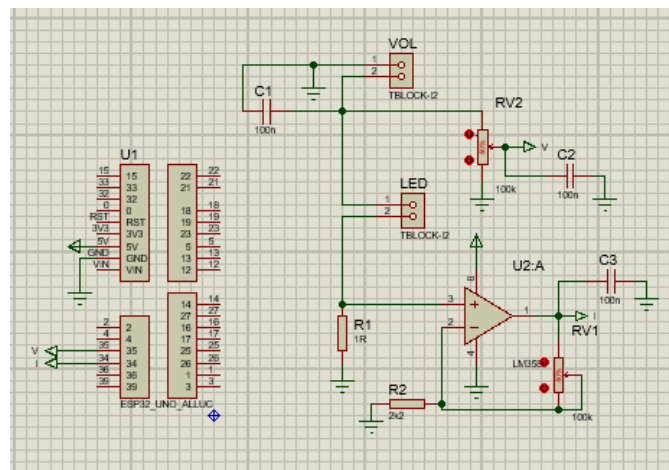
A continuación, proporcionamos el enlace al repositorio de GitHub donde hemos subido el proyecto de HidroT. Aquí podrás acceder al código fuente, realizar modificaciones y colaborar en su desarrollo:  
Enlace al repositorio de HidroT

<https://github.com/aatituanau/hIdroT/blob/main/app/src/main/java/com/iot/hIdroT/MainActivity.java>

## Ilustraciones



*Ilustración 1 - Sistema Hidroeléctrico Modelado 3D en Godot*



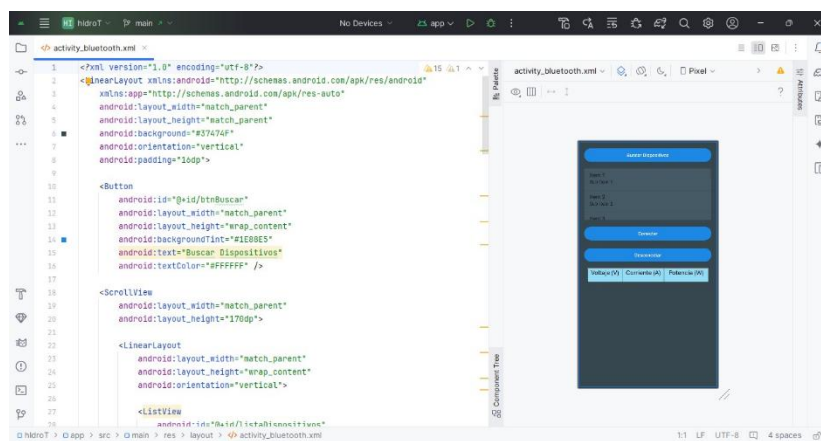
*Ilustración 2 - Desarrollo del circuito eléctrico en Proteus*

```

1 #include "BluetoothSerial.h"
2
3 #if !defined(CONFIG_BT_ENABLED) || !defined(CONFIG_BLUEDROID_ENABLED)
4 #error Bluetooth is not enabled! Please run 'make menuconfig' to enable it
5 #endif
6
7 BluetoothSerial SerialBT;
8 void setup()
9 {
10   SerialBT.begin("ESP32vip");
11 }
12
13 void loop()
14 {
15   analogSetAttenuation(ADC_6db);
16   int v=analogRead(35);
17   float voltaje=(8.0/4095.0)*v;
18
19   analogSetAttenuation(ADC_0db);
20   int i=analogRead(34);
21   float corriente=(20.0/4095.0)*i;
22   float potencia=voltaje*corriente;
23
24   SerialBT.print(voltaje);
25   SerialBT.print(",");
26   SerialBT.print(corriente);
27   SerialBT.print(",");
28   SerialBT.println(potencia);
29 }

```

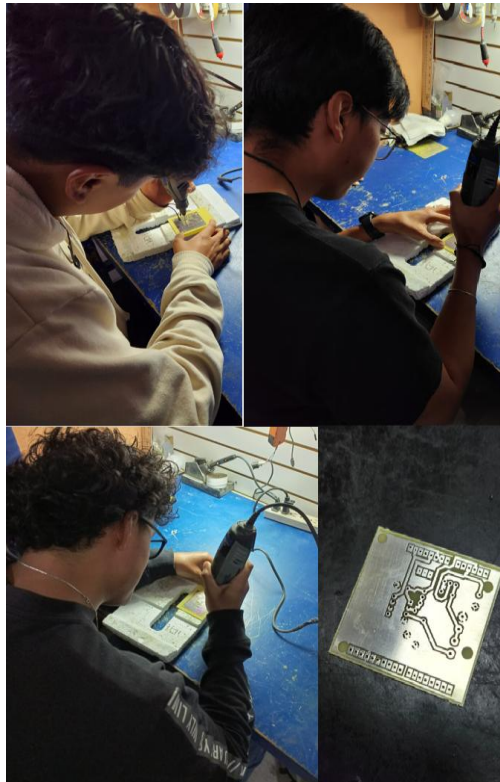
*Ilustración 3 - Desarrollo del código en Arduino IDE*



*Ilustración 4 - Desarrollo de aplicación en Android Studio*



*Ilustración 5 - Elaboración de placa para soldar componentes*



*Ilustración 6 - Perforación de placa para conectar pines del microcontrolador*

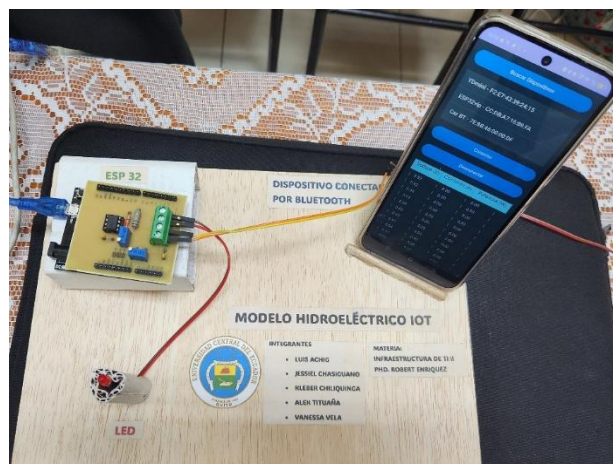




*Ilustración 7 - Pruebas de funcionamiento*



*Ilustración 8 - Ensamblaje del modelo físico - Parte 1*



*Ilustración 9 - Ensamblaje del modelo físico - Parte 2*



*Ilustración 10 - Presentación casa abierta carrera Sistemas de Información "Impulsando la era digital: Innovación y desarrollo en Sisistemas de información"*