

Proyecto: Generador Eólico

Universidad Central del Ecuador, Facultad de Ingeniería y Ciencias Aplicadas, Ingeniería en Sistemas de Información, Infraestructura de Ti - II

27 – 02 - 2025

I. OBJETIVOS

Objetivo General:

Desarrollar e implementar un modelo digital en Unity y un modelo físico de un generador eólico con el fin de analizar su eficiencia y aplicabilidad en la generación de energía renovable. Evaluar el comportamiento de un aerogenerador en condiciones virtuales y experimentales para establecer comparaciones de rendimiento y factibilidad técnica.

Objetivos Específicos

- Diseñar y construir un modelo tridimensional de un parque eólico en Unity con funcionalidades de animación y simulación de condiciones ambientales.
- Implementar un sistema de adquisición de datos en tiempo real mediante Google Sheets para el monitoreo y análisis de rendimiento del modelo virtual.
- Construir un generador eólico físico a escala con Arduino, sensores de medición y transmisión de datos vía Bluetooth.
- Evaluar el desempeño de ambos modelos en diversas condiciones experimentales, comparando los resultados obtenidos.
- Analizar la factibilidad de escalar los modelos para su aplicación en proyectos reales de energía eólica.

II. MARCO TEÓRICO

La energía eólica es una de las fuentes de energía renovable más utilizadas en el mundo. Su funcionamiento se basa en la conversión de la energía cinética del viento en energía mecánica mediante rotores, la cual posteriormente se transforma en

energía eléctrica mediante generadores. Este proceso permite generar electricidad sin emitir gases de efecto invernadero ni depender de combustibles fósiles.

Los aerogeneradores pueden clasificarse en dos grandes grupos: de eje horizontal y de eje vertical. Los aerogeneradores de eje horizontal son los más comunes y eficientes, ya que pueden aprovechar mejor la dirección del viento y generar mayor potencia. Sus componentes principales incluyen el rotor (con aspas diseñadas aerodinámicamente para captar el viento), el generador eléctrico, la torre de soporte y un sistema de control que regula la operación según las condiciones ambientales.

En el ámbito de la simulación digital, Unity es una de las plataformas más utilizadas para la creación de entornos tridimensionales interactivos. En este proyecto, Unity se emplea para modelar un parque eólico virtual, permitiendo analizar visualmente el comportamiento de las turbinas bajo distintas velocidades y direcciones del viento. La integración con Google Sheets facilita la simulación de datos en tiempo real, reflejando condiciones reales y proporcionando una herramienta de análisis interactiva.

Por otro lado, en el modelo físico del generador eólico, se hace uso de microcontroladores Arduino para la adquisición y transmisión de datos. Este sistema incorpora sensores de corriente, voltaje y velocidad de rotación (RPM), que permiten evaluar el rendimiento del aerogenerador. La transmisión de datos vía Bluetooth permite un monitoreo en tiempo real mediante software especializado, como PLX-DAQ en Excel, proporcionando información clave sobre la eficiencia del sistema.

* Elaborado por: Andrade Rodney, Coello Antony, Coronel Jhonatan , Moyon Kevin, Viracucha Jefferson.

El uso combinado de un modelo digital y un modelo físico permite una validación cruzada de resultados, asegurando que los datos obtenidos en la simulación sean coherentes con la realidad. Además, el análisis de ambos enfoques permite mejorar el diseño y la eficiencia del generador eólico, contribuyendo al desarrollo de tecnologías más sostenibles y eficientes.

DESARROLLO DEL PROYECTO DEL DATA CENTER

1. SITUACIÓN INICIAL Y PLANIFICACIÓN

Se identificaron los requerimientos del proyecto, dividiendo las tareas en dos enfoques: virtual y físico. Se establecieron cronogramas de desarrollo y pruebas.

2. IMPLEMENTACIÓN

El Modelo 3D en Unity:

- Se diseñó el entorno del parque eólico utilizando la herramienta Terrain de Unity para modelar el terreno.
- Se importaron modelos 3D de turbinas en formato. fbx y se ajustaron sus materiales y texturas para lograr mayor realismo.
- Se programó en C# la animación de las aspas, vinculándolas con datos en tiempo real obtenidos desde Google Sheets.
- Se implementaron sistemas de visualización interactiva para observar el comportamiento de cada turbina según los datos de velocidad del viento y revoluciones por minuto (RPM).

Modelo Físico con Arduino:

- Se ensambló una turbina de eje horizontal con aspas fabricadas en impresión 3D.
- Se utilizó un motor DC como generador para convertir la energía mecánica en eléctrica.
- Se incorporaron sensores ACS712 para la medición de corriente y un encoder para registrar la velocidad de giro del rotor.
- Se desarrolló un código en Arduino IDE para procesar los datos obtenidos y enviarlos vía Bluetooth a una PC,

donde se registraron en una hoja de cálculo mediante PLX-DAQ.

III. METODOLOGÍA

La Se utilizó un enfoque experimental, combinando simulación digital y pruebas físicas para evaluar la eficiencia del aerogenerador. Se definieron tres fases:

- ❖ Diseño y desarrollo: Creación del entorno en Unity y ensamblaje del modelo físico.
- ❖ Pruebas y ajustes: Simulación de diferentes condiciones de viento en Unity y evaluación del rendimiento del modelo físico mediante pruebas en laboratorio.
- ❖ Análisis de resultados: Comparación de los datos obtenidos en ambas metodologías para validar la coherencia entre la simulación y la realidad.

IV. RESULTADOS DEL PROYECTO

Durante la ejecución del proyecto, se recopilaron datos significativos sobre el rendimiento del modelo físico y su comparación con la simulación digital. En la simulación en Unity, se observó que la velocidad de las aspas variaba en función de la intensidad del viento, replicando de manera precisa las condiciones reales de un parque eólico. La interacción con los datos de Google Sheets permitió visualizar en tiempo real los cambios en la generación de energía, mejorando la comprensión del sistema.

Por otro lado, el modelo físico demostró ser funcional en la generación de energía, aunque se identificaron diferencias con respecto a la simulación. Los sensores instalados en la turbina permitieron obtener mediciones precisas de corriente, voltaje y RPM. Sin embargo, factores como la resistencia mecánica, la fricción en el eje y la calidad de los materiales influyeron en la eficiencia del sistema, lo que resultó en valores de generación de energía ligeramente inferiores a los esperados según el modelo teórico.

Se llevaron a cabo pruebas bajo diferentes condiciones de viento, observando que la eficiencia del generador aumentaba en velocidades de viento entre 5 y 10 m/s, mientras que a

velocidades menores la generación era mínima. Además, la orientación de las aspas afectó significativamente el rendimiento, indicando la necesidad de ajustes en el diseño estructural.

V. DISCUSIÓN

El Los resultados obtenidos confirman que la energía eólica es una fuente viable de generación de electricidad, aunque la eficiencia del sistema depende de varios factores clave. La simulación en Unity permitió visualizar de manera efectiva el comportamiento de las turbinas en diferentes escenarios de viento, validando la utilidad de las herramientas digitales en la fase de diseño y análisis previo.

El modelo físico, a pesar de haber demostrado funcionalidad, presentó limitaciones debido a la resistencia mecánica, la fricción y la calidad de los materiales. Se observó que la generación de energía se ve afectada por el diseño de las aspas y la aerodinámica general del sistema.

Otro factor relevante fue la transmisión de datos mediante Bluetooth, que en algunos casos presentó pérdidas de información debido a interferencias. Este problema podría resolverse optimizando la configuración del sistema de comunicación y utilizando protocolos de transmisión más robustos.

Finalmente, se determinó que la integración entre el modelo físico y el virtual puede ser una herramienta poderosa para el análisis de energía eólica, permitiendo identificar oportunidades de mejora en el diseño y optimización del sistema.

VI. CONCLUSIONES Y RECOMENDACIONES

Conclusiones:

- ❖ La simulación en Unity es una herramienta efectiva para la predicción del comportamiento de turbinas eólicas en distintas condiciones climáticas.
- ❖ El modelo físico permitió verificar la viabilidad de la generación de energía, aunque con eficiencia menor a

la esperada debido a restricciones mecánicas y materiales.

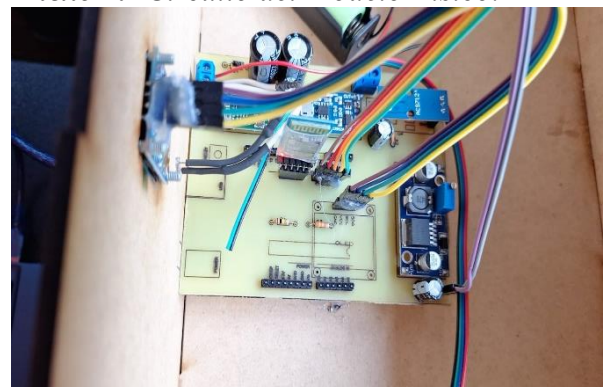
- ❖ La integración de sensores y la transmisión de datos en tiempo real facilitaron el monitoreo del sistema, aunque se recomienda mejorar la robustez de la comunicación inalámbrica.

Recomendaciones:

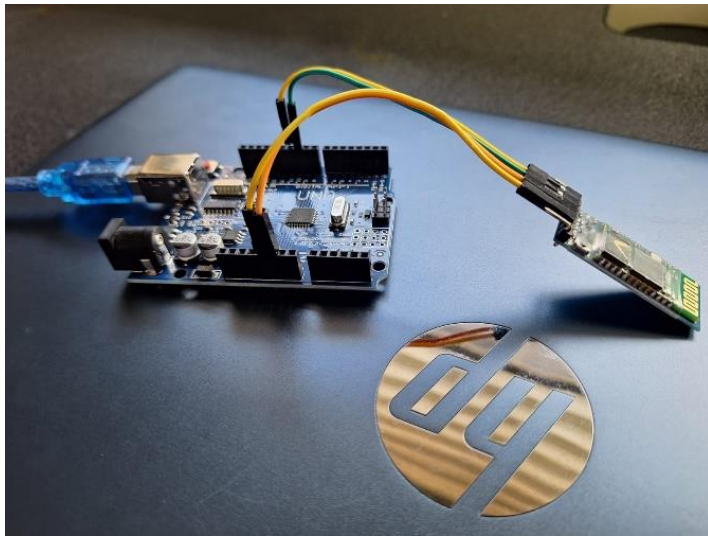
- Mejorar el diseño de las aspas para optimizar la captación de viento y reducir la fricción en el eje del rotor.
- Utilizar materiales más ligeros y resistentes en la construcción de la turbina para aumentar la eficiencia del generador.
- Implementar algoritmos de optimización para la transmisión de datos y utilizar tecnologías de conectividad más avanzadas como Wi-Fi o LoRa para mejorar la estabilidad de la comunicación.
- Realizar pruebas adicionales con diferentes configuraciones de viento para validar la eficiencia del sistema en diversos entornos.

ANEXOS

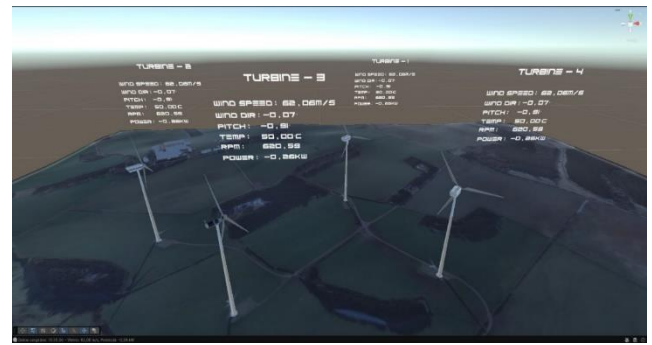
➤ Anexo 1.- Circuito del Modelo Físico.



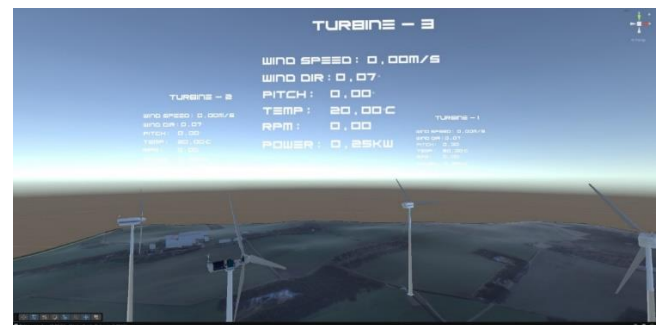
➤ *Anexo 2.- Arduino Uno y Modulo Bluetooth.*



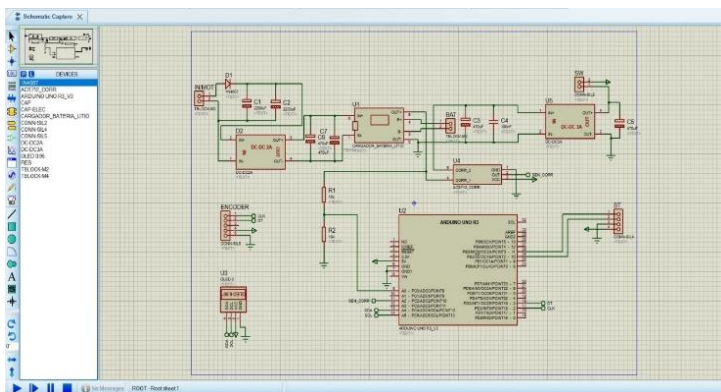
➤ *Anexo 5.- Turbinas Modelo Digital*



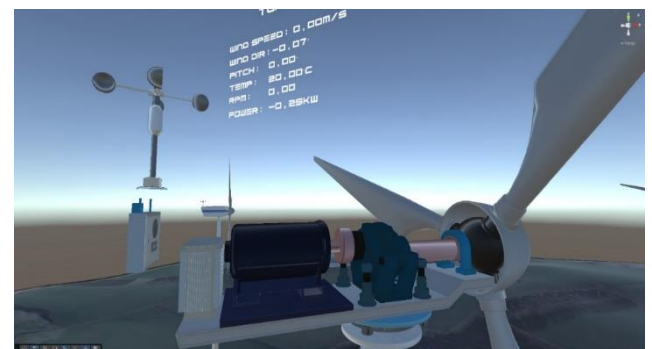
➤ *Anexo 6.- Tabla de Datos Modelo Digital.*



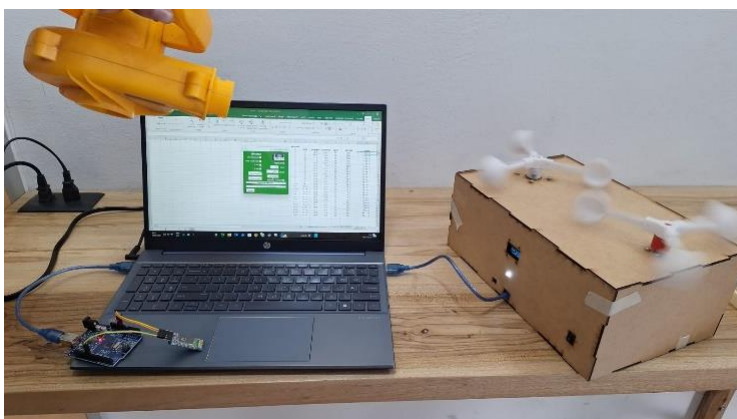
➤ *Anexo 3.- Proteus Modelo Físico.*



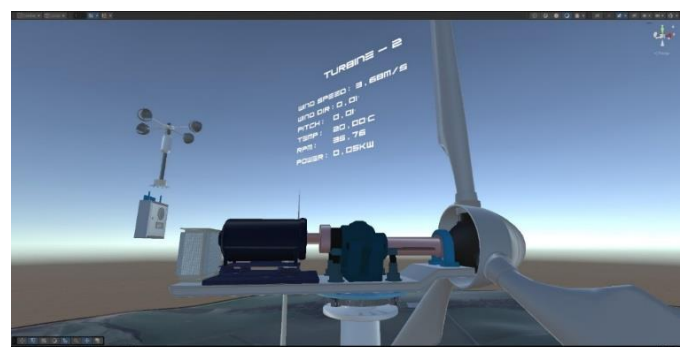
➤ *Anexo 7.- Motor de Turbina*



➤ *Anexo 4.- Modelo Físico.*



➤ *Anexo 8.- Componentes de la Turbina*



BIBLIOGRAFÍAS Y REFERENCIAS

Chavez, J. (2021). Mastering Unity 2021 Game Development. Packt Publishing.

Freeman, A. (2020). Pro C# 8 with .NET Core 3. Apress.

Gregory, J. (2018). Game Engine Architecture, Third Edition. A K Peters/CRC Press.

Google Cloud. (s.f.). Google Sheets API Documentation. Google Developers. Recuperado el 25 de febrero de 2025, de <https://developers.google.com/sheets/api>

Google Cloud. (s.f.). Google Cloud Platform Documentation. Google Developers. Recupera-do el 25 de febrero de 2025, de <https://cloud.google.com/docs>

Microsoft. (s.f.). Microsoft .NET C# Documentation. Microsoft Docs. Recuperado el 25 de febrero de 2025, de <https://learn.microsoft.com/en-us/dotnet/csharp/>

Newtonsoft. (s.f.). *Json.NET Documentation*. Recuperado el 25 de febrero de 2025, de <https://www.newtonsoft.com/json/help/html/Introduction.htm>

OpenWeatherMap. (s.f.). *OpenWeatherMap API Documentation*. Recuperado el 25 de febrero de 2025, de <https://openweathermap.org/api>

Unity Technologies. (s.f.). *Unity Documentation*. Recuperado el 25 de febrero de 2025, de <https://docs.unity3d.com/>

GameDevBeginner. (s.f.). *Coroutines in Unity – A Complete Guide*. Recuperado el 25 de febrero de 2025, de <https://gamedevbeginner.com/coroutines-in-unity/>

Unity Technologies. (s.f.). *Optimización de rendimiento en Unity*. Recuperado el 25 de febrero de 2025, de <https://unity3d.com/learn/tutorials/topics/graphics>

Towards Data Science. (s.f.). *Using C# with Google Sheets API to Read and Write Data*. Recuperado el 25 de febrero de 2025, de <https://towardsdatascience.com/using-c-with-google-sheets-api-to-read-and-write-data-97b504799380>