



Universidad de Málaga

Grado en ingeniería en tecnologías industriales

Trabajo Fin de Grado

# Modelos Predictivos de Aprendizaje Profundo y Aprendizaje Automático para la Mejora de la Eficiencia en Instalaciones de Energías Renovables: Parques Eólicos y Solares

*Autor*

Muelas de la Linde, José Manuel

*Tutora*

Burgueño Caballero, Lola

Área de Conocimiento: Lenguajes y Ciencias de la Computación

23 de junio de 2025

# Índice

# Índice de figuras

# Abstract

In the current context of energy transition and the fight against climate change, the search for solutions that enhance the efficiency and reliability of renewable energies has become a priority at both scientific and industrial levels. The ability to predict energy production with greater accuracy not only enables more efficient management of available resources, but also significantly contributes to the integration of renewable sources into electrical systems, reducing dependence on conventional sources, and fostering a more sustainable energy model.

This Bachelor's thesis addresses the development and evaluation of predictive models based on machine learning and deep learning techniques aimed at optimizing operational efficiency in renewable energy facilities, in particular, those focused on wind and solar farms. The inherent variability of weather conditions and the uncertainty of power generation pose critical challenges to the integration and stability of these systems. Being aware of this issue, we employ advanced algorithms that integrate external variables, such as wind speed, solar irradiance and temperature, among many others, to predict energy production more accurately.

The proposed methodology blends a state-of-the-art review of predictive techniques with the implementation of both classical machine learning models and deep learning architectures, assessed through performance metrics such as $R^2$, MAE, and MSE. The results reveal substantial improvements in predictive capability, translating into more efficient operational planning, optimized resource use, and reduced energy-management costs. This study not only confirms the feasibility of applying predictive models to the renewable energy sector but also lays the foundation for future work aimed at integrating intelligent systems into large-scale energy management.

**Keywords:** Machine Learning, Deep Learning, Predictive Models, Renewable Energy, Wind Farms, Solar Farms, Operational Efficiency.

# Resumen

En el contexto actual de transición energética y lucha contra el cambio climático, la búsqueda de soluciones que incrementen la eficiencia y la fiabilidad de las energías renovables se ha convertido en una prioridad tanto a nivel científico como industrial. La capacidad de predecir con mayor precisión la producción energética no solo permite optimizar la gestión de los recursos disponibles, sino que también contribuye de manera significativa a la integración de las fuentes renovables en los sistemas eléctricos, reduciendo la dependencia de fuentes convencionales y favoreciendo un modelo energético más sostenible.

Este Trabajo Fin de Grado aborda el desarrollo y evaluación de modelos predictivos basados en técnicas de aprendizaje automático y profundo, orientados a optimizar la eficiencia operativa en instalaciones de energías renovables, en este caso, centrado parques eólicos y solares. La variabilidad inherente a las condiciones meteorológicas y la incertidumbre en la generación energética representan retos críticos para la integración y estabilidad de estos sistemas. Conscientes de esta problemática, se han empleado algoritmos avanzados que, mediante la integración de variables externas tales como, la velocidad del viento, la irradiación solar y la temperatura, entre muchas otras, permiten predecir de forma más precisa la producción de energía.

La metodología propuesta combina la revisión del estado del arte en técnicas predictivas con la implementación de modelos tanto clásicos de machine learning como de arquitecturas de deep learning, evaluados a través de métricas de rendimiento como $R^2$, MAE y MSE. Los resultados obtenidos evidencian mejoras sustanciales en la capacidad de predicción, lo que se traduce en una planificación operativa más eficiente, una optimización de recursos y una reducción en los costes asociados a la gestión de la energía. Este estudio no solo confirma la viabilidad de aplicar modelos predictivos en el sector de las energías renovables, sino que también sienta las bases para trabajo futuro orientado a la integración de sistemas inteligentes en la gestión energética a gran escala.

**Palabras clave:** Aprendizaje Automático, Aprendizaje Profundo, Modelos predictivos, Energías Renovables, Parques Eólicos, Parques Solares, Eficiencia Operativa.

# 1 | Introducción

## 1.1. Contexto y Motivación

En el contexto actual, es urgente la necesidad de acabar con el cambio climático y buscar la transición hacia modelos energéticos sostenibles. Esto ha llevado a un renovado interés en las fuentes de energía renovable. Debido a su gran adopción en un gran número de países y al hecho de que es una energía limpia, la energía solar y eléctrica se han consolidado como unas de las alternativas estratégicas a los combustibles fósiles, cuyo uso prolongado contribuye significativamente a la emisión de CO2 [9]. Sin embargo, a pesar de sus muchos beneficios, la integración de estas fuentes de energía en las redes eléctricas presentan desafíos inevitables debido a su naturaleza cambiante. Estos son causados por factores ambientales como la radiación solar, la velocidad del viento, la temperatura o la presión atmosférica.

Esta situación obliga a desarrollar soluciones que permitan una predicción más precisa de la generación de energía, optimizando así la planificación operativa y reduciendo los costos de gestión de estas instalaciones.

Existe una creciente complejidad de los sistemas energéticos modernos que hace que se necesite el uso de herramientas sofisticadas y optimizadas, que en este contexto, el aprendizaje automático y el aprendizaje profundo surgen como alternativas prometedoras. Mi interés personal y el compromiso con un futuro sostenible han sido motivadores clave para investigar cómo estas técnicas podrían contribuir a soluciones innovadoras en el campo de la energía renovable.

## 1.2. Problema Planteado

En estos últimos años han aparecido diferentes avances en diseño y operación de parques eólicos y solares, aunque la predicción de la generación de energía sigue siendo un reto importante. Las condiciones meteorológicas cambian y las interacciones entre las diferentes variables ambientales son muy complejas, esto hace que los modelos predictivos más tradicionales tengan poco acierto y dejen mucho que desear. Esta inexactitud lleva a grandes diferencias entre la oferta y la demanda de energía.

El problema empeora cuando te das cuenta cómo las soluciones convencionales, actualmente, no tienen en cuenta lo dinámicos que son los entornos en los que estos sistemas operan, cada día es diferente al anterior. Por ello, es de gran necesidad desarrollar modelos predictivos que puedan adaptarse y aprender continuamente, ya que es de gran ayuda incorporar datos en tiempo real e integrar eficazmente variables externas para mejorar estos modelos. En este contexto, el enfoque que se da al aprendizaje automático y aprendizaje profundo ofrece una manera de sortear

los problemas de los métodos tradicionales y reducir los errores. Estos nos dan una herramienta poderosa para realizar predicciones más precisas y, como resultado, hacer el proceso más eficiente.

## 1.3. Objetivo del Trabajo Fin de Grado

El objetivo de este Trabajo Fin de Grado es crear modelos predictivos que nos ayuden a estimar, con una precisión aceptable, la generación de energía en parques eólicos y solares. Para alcanzar este objetivo, se han establecido los siguientes objetivos específicos:

- **Creación y comparación de diferentes enfoques para la creación de modelos predictivos**. Se examinan tanto los algoritmos clásicos de Machine Learning como las arquitecturas avanzadas de Deep Learning para encontrar los que mejor se adaptan a la complejidad de los datos y a las condiciones cambiantes del entorno. Para ello usaremos métodos para obtener los mejores parámetros de cada modelo y realizar validaciones cruzadas para asegurar que no exista sobreajuste.
- **Encontrar y combinar variables ambientales importantes**. Se usan variables como la temperatura, la velocidad del viento o la irradiación solar, para alimentar el conocimiento de los modelos predictivos, esto permite una estimación más precisa de la producción energética.
- **Evaluar el rendimiento de los modelos**. Se usan métricas conocidas en el campo de la predicción, como el coeficiente de determinación ($R^2$), el error absoluto medio (MAE) y el error cuadrático medio (MSE), para medir el desempeño de cada modelo.
- **Visualización y análisis de resultados**. Se usan gráficos que facilitan la comparación entre los valores reales y las predicciones obtenidas, proporcionando una exposición visual que da apoyo a la interpretación de los resultados y la toma de decisiones operativas.
- **Proponer futuras líneas de investigación**. Se hablará sobre las limitaciones encontradas en los resultados y se sugerirán diferentes formas de ajustar los nuevos métodos para seguir mejorando la predicción de la energía renovable.

## 1.4. Estructura de la Memoria

El resto de este documento se estructura de la siguiente forma: El capítulo 2 trata sobre el Marco Teórico, el capítulo 3 sobre los Requisitos de Diseño, el capítulo 4 sobre la Metodología, el capítulo 5 sobre los Modelos de Aprendizaje Supervisado, el capítulo 6 sobre la Evaluación y Optimización de Modelos Predictivos, el capítulo 7 sobre los Modelos de Aprendizaje Automático, el capítulo 8 sobre los Modelos de Aprendizaje Profundo y por último, el capítulo 9 concluye y detalla el trabajo futuro.

# 2 | Marco Teórico

En esta sección se explican los conceptos fundamentales y bases teóricas necesarias para la elaboración de modelos predictivos y predicción de la generación de energía en parques solares y eólicos. En primer lugar, se comenta el entorno de las energías renovables y los retos operativos que se generan de su variabilidad. Posteriormente, se detallan los conceptos fundamentales de Aprendizaje Automático y Aprendizaje Profundo, enfocándose en la utilización de estos métodos en el sector energético y se habla del estado actual del arte, analizando investigaciones anteriores y tecnologías recientes en el género hablado. En última instancia, el enfoque adoptado en este estudio se justifica considerando tanto la exactitud técnica como la pertinencia práctica en contextos reales.

## 2.1. Energías Renovables y Retos Operativos

Está surgiendo un cambio estos últimos años hacia fuentes de energía renovable, donde estas han tomado un rol crucial en los intentos de reducir el daño del cambio climático y garantizar la sostenibilidad energética. Se puede destacar que en el conjunto de las energías renovables, la energía eólica y solar sobresalen entre todas debido a su capacidad para producir energía limpia y su 'fácil' obtención. No obstante, su incorporación a los sistemas eléctricos plantea varios desafíos tanto técnicos como operativos.

Uno de los desafíos más complicados de resolver, es que estas fuentes son altamente fluctuantes, es decir, cambian diariamente y no se puede asegurar su estabilidad, ya que se basa en elementos climáticos complejos como la rapidez del viento, la irradiación solar o las condiciones del clima general.

Estas características dificultan la organización y el correcto funcionamiento de las instalaciones, impactando en la estabilidad de la red eléctrica y en los gastos relacionados. Estos problemas son particularmente notorios en sistemas con gran cantidad de energía renovable, donde la falta de habilidad para realizar proyecciones exactas puede provocar que la estabilidad energética no se cumpla. Investigaciones que se han realizado por la IEA, indican que la incorporación de información en tiempo real puede disminuir los gastos de operación en un 20 % en parques solares y eólicos. Esto se debe a que si los modelos predictivos son nutridos con datos actualizados, estos tienen un mejor desempeño.

Para ello, la necesidad de obtener unas predicciones precisas, se convierte en el factor clave para mejorar la eficiencia y la viabilidad económica de los proyectos renovables en un futuro.

## 2.2. Aprendizaje Automático y Aprendizaje Profundo

El Aprendizaje Automático (Machine Learning) se basa en la aplicación de algoritmos que pueden identificar patrones de valor en un conjunto de datos para su aplicación en el hacer predicciones o clasificaciones. Por ejemplo, en el contexto de las fuentes de energía renovable, se utiliza data histórica sobre producción de energía, junto con la velocidad del viento, irradiación solar o temperatura como factores ambientales, para entrenar modelos predictivos para producir una estimación de cuánta energía se va a producir por un parque eólico o solar. Para lograr una salida que se alinee con el valor real, cada modelo modifica una secuencia de parámetros internos. Por ejemplo, modificando los coeficientes de una regresión lineal, modificando la estructura de los árboles de decisión o modificando los pesos en los métodos de ensamblaje. Conforme el modelo recibe más información curada, aprende y optimiza su comportamiento en base a los datos proporcionados. Este aspecto es fundamental para optimizar el uso de los recursos y planificar la energía que se va a generar.

El Aprendizaje Profundo (Deep Learning) depende de crear algoritmos que puedan aprender a detectar valiosos patrones en ejemplos o entrenamiento datos y aplicarlos para hacer predeciones o clasificaciones. Por ejemplo, en el ámbito de las energías renovables, podemos emplear información histórica acerca de la producción de energía, junto con factores ambientales como la velocidad del viento, la irradiación solar y la temperatura, para instruir a los modelos en la estimación de la cantidad de energía que producirá un parque eólico o solar. Para lograr una salida que se alinee con el valor real, cada modelo modifica una secuencia de parámetros internos. Por ejemplo, modifica los coeficientes de una regresión lineal, modifica la estructura de los árboles de decisión o modifica los pesos en los métodos de ensamblaje. Conforme el modelo recibe más información, aprende y optimiza su comportamiento. Es posible que también efectúe generalizaciones sobre información novedosa que no se empleó durante la fase de entrenamiento. Este aspecto es fundamental para optimizar la utilización de los recursos y planificar las operaciones en las instalaciones de energía renovable.

El uso de estas técnicas ha creado varios avances significativos en campos como la visión por ordenador, el procesamiento del lenguaje natural y la robótica autónoma [7]. En el caso de la predicción de generación energética, las redes neuronales recurrentes (RNN) resultan especialmente útiles para modelar series temporales de velocidad del viento o irradiación solar, mientras que las redes neuronales convolucionales (CNN) pueden extraer patrones relevantes de imágenes por satélite o mapas meteorológicos.

### 2.2.1. Estado del Arte

Numerosas investigaciones han analizado la fluctuación en la producción de energía renovable mediante el uso de modelos predictivos, como regresión linear o bosques aleatorios. Estos operaran de manera eficiente en condiciones estables y cuando el volumen de datos era moderado. No obstante, estos métodos no logran identificar patrones no lineales asociados con la velocidad del viento, la radiación solar y otras variables meteorológicas debido a la complejidad del entorno.

Actualmente, se está observando que las arquitecturas de redes neuronales de tipo feed-forward pueden mejorar la precisión de la predicción al modelar relaciones complejas entre sus múltiples capas. Estos modelos profundos, entrenados con conjuntos de datos representativos y optimizados, mediante métodos de regularización y ajuste sistemático de hiperparámetros, superan a los métodos tradicionales en la identificación de tendencias sutiles y en la generalización a datos nuevos.

Los expertos han examinado también esquemas híbridos que fusionan algoritmos tradicionales con el aprendizaje profundo. Por ejemplo, la combinación de predicciones fundamentadas en la regresión o en el ensamblaje con las salidas de una red neuronal MLP facilita la utilización de la estabilidad de la primera y la flexibilidad no lineal de la segunda. Cuando se emplean en conjunto, estas estrategias mixtas disminuyen de manera significativa el número de error en las predicciones en comparación con la utilización de cada método de manera individual.

A pesar de estas mejoras, aún persisten grandes desafíos por abordar, ya puede ser la demanda de una gran cantidad de datos de alta calidad, la elevada carga computacional que implica el entrenamiento de redes profundas o la complejidad de calibrar hiperparámetros en entornos variables. Estos problemas evidencian la relevancia de emplear un método iterativo para las pruebas y validaciones, además de proporcionar recursos adecuados para el preprocesamiento y la selección de características.

## 2.3. Justificación del Enfoque Adoptado

El enfoque que se ha tomado se basa en la integración de variables externas clave (velocidad del viento, irradiación solar, temperatura, presión atmosférica, etc.), identificadas como determinantes para la generación energética. A partir de la revisión de las investigaciones previas, se reafirma la importancia de combinar datos históricos y en tiempo real para capturar las fluctuaciones diarias y estacionales de estas fuentes renovables. Aunque hay que comprobar periódicamente los datos históricos por si ha habido alguna anomalía en el pasado.

Mediante el uso de técnicas avanzadas de aprendizaje profundo y automático, el objetivo es superar las limitaciones de los enfoques tradicionales, ofreciendo modelos que no solo sean precisos, sino también aplicables en contextos reales a nivel global. Se busca así:

- Incrementar la exactitud de las predicciones, reduciendo la diferencia entre la oferta y la demanda energética.
- Reducir los costes operativos, aprovechando la planificación anticipada y la gestión dinámica de recursos.
- Promover la sostenibilidad técnica y económica de las instalaciones eólicas y solares, integrando de manera eficiente las fuentes renovables en la red eléctrica.

En síntesis, la combinación de un análisis exhaustivo de variables externas con algoritmos de Machine Learning y Deep Learning robustos ofrece una vía prometedora para mejorar la gestión y la eficiencia en instalaciones de energías renovables.

# 3 | Requisitos de Diseño

En este capítulo se describen las herramientas y el entorno de desarrollo empleados para la implementación de los modelos predictivos, así como el hardware utilizado y la fuente principal de datos. Se justifica la elección de cada software y se explica brevemente su papel en la metodología de trabajo.

## 3.1. Hardware

El equipo usado en este trabajo ha sido el **MacBook Air M2**

El motivo de elección ha sido a raíz de que se trata de un equipo portátil con arquitectura Apple Silicon (M2), que ofrece un buen equilibrio entre potencia de procesamiento y eficiencia energética. El procesador M2 incluye una GPU integrada que permite acelerar algunas operaciones de predicción de manera básica.

Algunas características destacadas son:

- Procesador Apple M2 con CPU de 8 núcleos y GPU integrada.
- Memoria unificada (RAM) con alta velocidad de acceso.
- Sistema operativo macOS 15.1.1.

Aunque no se trate del mejor equipo posible en el mercado, ha resultado suficiente para el desarrollo de modelos de complejidad moderada .

## 3.2. Fuente de Datos

La obtención de datos ha sido a través del repositorio online de Kaggle, plataforma ampliamente reconocida que ofrece conjuntos de datos de acceso público, además de competencias y recursos relacionados con el análisis de datos.

## 3.3. Software

El desarrollo del proyecto se lleva a cabo utilizando lenguajes, librerías y herramientas de software ampliamente reconocidas en el ámbito del análisis de datos y el aprendizaje automático.

Entre las principales se incluyen:

### 3.3.1. Python

Se ha usado Python como lenguaje de programación principal gracias a su facilidad de programación y lectura, y un gran ecosistema de bibliotecas. Se ha trabajado con librerías base como Pandas, NumPy, Matplotlib, Scikit-learn y TensorFlow, que ha hecho fácil el manejo de los datos, implementación de algoritmos predictivos y visualización de resultados.


**Figura 3.1.** Python Logo

### 3.3.2. Conda

Conda ha sido usada como herramienta para la gestión de entornos virtuales y dependencias. Su uso garantiza la compatibilidad entre bibliotecas, evitando conflictos durante el desarrollo y la ejecución del proyecto.


**Figura 3.2.** Conda Logo

### 3.3.3. Visual Studio Code

El entorno de desarrollo integrado (IDE) utilizado ha sido Visual Studio Code. Se ha valorado especialmente su compatibilidad con Python, el control de versiones integrado, y la posibilidad de ampliar su funcionalidad mediante extensiones como Jupyter.

**Figura 3.3.** Visual Studio Code Logo

### 3.3.4. GitHub

Para el control de versiones y la gestión del código fuente se utilizó GitHub. Esta plataforma permite mantener un historial claro de los cambios realizados en la nube y así asegurar de que el proyecto no se elimine accidentalmente. También facilita la colaboración y asegurar la trazabilidad del desarrollo.


**Figura 3.4.** GitHub Logo

### 3.3.5. Uso de Visual Studio Code, Conda y GitHub

En este proyecto, se han usado archivos `.ipynb` (notebooks de Jupyter) para desarrollar y organizar el código de forma interactiva.

#### Visual Studio Code con Notebooks

Visual Studio Code se ha configurado para trabajar cómodamente con archivos `.ipynb`. Los pasos que se han tomado son:

1. Descargar e instalar VS Code desde https://code.visualstudio.com/.
2. Instalar las extensiones necesarias:
   *a*) La extensión de Python.
   *b*) La extensión de Jupyter (esta permite abrir y trabajar directamente con archivos `.ipynb` en VS Code).
3. Abrir un notebook en VS Code:
   *a*) Crear un archivo con extensión `.ipynb`.
   *b*) Abrirlo en VS Code para usar la interfaz interactiva, donde cada celda de código puede ejecutarse individualmente.
4. Configurar el intérprete de Python:
   *a*) Presionar `Ctrl + Shift + P` (o `Cmd + Shift + P` en Mac).
   *b*) Escribir "Python: Select Interpreter" y elegir el entorno Conda configurado.

### Conda para Notebooks

Conda se ha utilizado para crear y gestionar el entorno donde se ejecutaron los notebooks. Los pasos principales son:

1. Crear un entorno con Python e instalar Jupyter:

```
conda create --name TFG2025 python=3.9
conda activate TFG2025
conda install numpy pandas matplotlib tensorflow scikit-learn
```

2. Abrir los notebooks desde VS Code con el entorno creado previamente.

### GitHub con Notebooks

Para gestionar los archivos `.ipynb` en GitHub, se han empleado los siguientes pasos:

1. Inicialización del repositorio local:

```
git init
```

2. Agregar y versionar notebooks:

```
git add archivo_notebook.ipynb
git commit -m "Agregado notebook inicial"
```

3. Subir los notebooks a un repositorio remoto en GitHub:

```
git remote add origin https://github.com/usuario/repositorio.git
git push -u origin main
```

### Beneficios del Uso de Notebooks

El formato `.ipynb` permite combinar celdas de código con visualizaciones y explicaciones escritas en texto (Markdown), haciendo que el desarrollo del proyecto sea más organizado e interactivo. Además, este formato facilita la depuración de código y la generación de informes visuales directamente desde el entorno de trabajo.

# 4 | Metodología

En este capítulo se proporciona una descripción detallada de los métodos usados para la creación y prueba de los modelos predictivos propuestos. El procedimiento elegido se fundamenta en un proceso estructurado que abarca diversas fases, cada una estuvo planificada para garantizar que los resultados sean los mas exactos posibles. Inicialmente, se llevó a cabo una etapa de recopilación y limpieza de datos, comprobando los datos erróneos y vacíos.

Posteriormente, se llevó a cabo un análisis exploratorio exhaustivo con el fin de identificar las características más relevantes de los datos recopilados y seleccionar las variables que resultarían más importantes para los modelos. Se empleó el análisis de correlación, para conocer que variables tienen mayor importancia en relación con esta selección.

Una vez identificadas las variables de mayor importancia, el siguiente paso consistió en diseñar y construir los modelos predictivos mediante el uso de métodos avanzados de Aprendizaje Automático y Aprendizaje Profundo. Se trabajó en esta sección del proyecto de forma iterativa e incremental, lo cual nos permite revisar y modificar los parámetros y configuraciones de los modelos en cada iteración. El método iterativo facilitó la identificación de problemas potenciales y posibilitó la implementación de modificaciones continuas para optimizar la habilidad de los modelos para realizar predicciones exactas.

También se implementó una planificación rigurosa a través de la elaboración de cronogramas de trabajo, los cuales se revisaban y actualizaban de manera regular en función de los resultados que se iban produciendo. Las reuniones periódicas con la tutora sirvieron de gran ayuda a la hora de tener una orientación académica y metodológica.

Finalmente, se procedió a la fase de validación y evaluación del rendimiento de los modelos predictivos mediante métricas importantes como el error cuadrático medio (RMSE), el error absoluto medio (MAE) y el coeficiente de determinación ($R^2$), garantizando así una evaluación completa y rigurosa del desempeño de los modelos [5].

## 4.1. Descripción de los Datos del Parque Eólico

El conjunto de datos, denominado `Data_Wind.csv`, incluye unas mediciones clave para el análisis del desempeño de instalaciones eólicas. En este conjunto de datos se captura tanto la generación de energía como las condiciones ambientales.

Este conjunto de datos se obtuvo de la plataforma Kaggle, específicamente del Dataset titulado Wind energy forecasting and maintenance data

(https://www.kaggle.com/datasets/pythonafroz/wind-energy-forecasting-and-maintenance

-data)

A continuación, se detallan los datos del Dataset, los pasos seguidos para la carga y la preparación inicial de estos datos.

### 4.1.1. Descripción Detallada del Dataset del Parque Eólico

El conjunto de datos utilizado, denominado `Data_Wind.csv`, contiene un total de 58,496 filas y 7 columnas. Las columnas del dataset son las siguientes:

- **index:** Registro temporal de las mediciones, en formato fecha y hora (UTC).
- **ActivePower_value_KWh:** Potencia activa generada, medida en kilovatios-hora (kWh). Es la variable objetivo principal del estudio predictivo.
- **AmbientTemperature_value:** Temperatura ambiente medida en grados Celsius (°C).
- **NacelleAngle_value:** Ángulo de orientación de la góndola, medido en grados (°).
- **RotorSpeed_value:** Velocidad de rotación del rotor, medida en revoluciones por minuto (rpm).
- **WindDirection_value:** Dirección del viento, expresada en grados (°).
- **WindSpeed_value:** Velocidad del viento, medida en metros por segundo (m/s).

Todas estas variables, son cuantitativas y están relacionadas directamente con las condiciones operativas y ambientales, permitiendo modelar con precisión la generación energética y evaluar su variabilidad temporal. El análisis preliminar nos confirma la relevancia de estas variables para entender cómo diferentes factores meteorológicos impactan en la eficiencia energética de los parques eólicos, constituyendo la base para la creación de modelos robustos.

### 4.1.2. Cargar el Dataframe

- Se utiliza la función `pd.read_csv()` de la biblioteca Pandas para cargar los datos desde un archivo CSV, facilitando así su manipulación y posterior análisis.

### 4.1.3. Manipulación de la Columna Index

- Se duplica la columna `index`, que contiene fechas y horas de las mediciones, para posteriormente convertirla a formato `datetime` con `pd.to_datetime()`, permitiendo una descomposición eficiente en componentes de fecha y hora (`year`, `month`, `day`, `hour`).
- Posteriormente, se elimina la columna inicial `index` para evitar redundancias y simplificar el DataFrame.

### 4.1.4. Preprocesamiento

1. **Lectura y consolidación de datos**

   - Todos los datos son consolidados en un único DataFrame para unificar la información y facilitar su manipulación.

2. **Limpieza de datos**

   - Se identifican y corrigen valores vacíos, inconsistentes o erróneos. Específicamente, los valores nulos en la columna `WindSpeed_value` se imputan usando el promedio de la columna.

     La decisión de reemplazar los valores vacíos mediante el promedio se tomó con el fin de preservar la integridad del conjunto de datos, evitando la pérdida innecesaria de información y reduciendo así el posible sesgo que podría producirse al eliminar filas completas. Este enfoque es particularmente adecuado cuando el número de datos faltantes es relativamente pequeño respecto al tamaño total del Dataset, permitiendo así mantener una cantidad representativa y robusta de información para el entrenamiento y evaluación de los modelos predictivos.

3. **Transformación y normalización**

   - Se aplican transformaciones a las variables para adaptarlas a la escala requerida por los algoritmos de Machine Learning. Las variables numéricas se normalizan y las categóricas se convierten en formatos numéricos para facilitar la convergencia de los modelos durante el entrenamiento.

     Por ejemplo, la variable numérica WindSpeed_value, que representa la velocidad del viento con valores entre 0 y 33.5 m/s, se transforma utilizando la técnica de estandarización (StandardScaler). Esta técnica ajusta la variable para que tenga una media de cero y una desviación estándar de uno, lo que mejora significativamente la convergencia y estabilidad del entrenamiento del modelo.

### 4.1.5. Análisis de los Datos Obtenidos

Para construir un modelo robusto es fundamental conocer en profundidad los datos con los que se cuenta. Por ello, se utilizan representaciones gráficas, ya que permiten interpretar la información de manera visual y comprensible.

En el primer análisis, figura 4.1, se verifica la ausencia de valores nulos; la representación gráfica lo confirma claramente, ya que en un gráfico sin anomalías se observa una uniformidad en el color (en este caso, un tono morado homogéneo). La aparición de alguna zona en amarillo indicaría la presencia de valores nulos, lo cual no ocurre en nuestro caso.

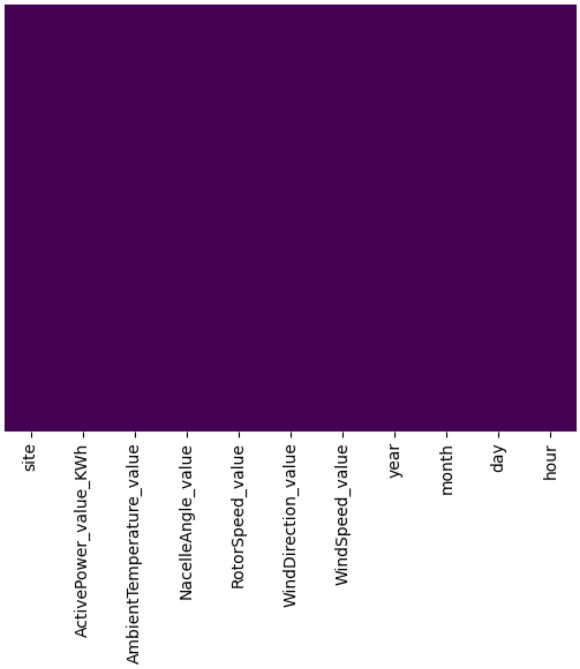


**Figura 4.1.** Análisis de Valores Nulos en el Dataset

Una vez comprobada la integridad de los datos, se procede a analizar la variable correspondiente al viento, la cual resulta crucial para nuestro modelo. Aunque en algunos casos puede no haber datos literalmente faltantes, en la figura 4.2, se ha observado que los registros correspondientes al año 2022 presentan un mayor porcentaje de datos incompletos en comparación con los años 2021 y 2023. Sin embargo, la ausencia de información en ciertos periodos no compromete la validez del análisis global, siempre que la relación entre el resto de los datos se mantenga coherente. Además, se ha realizado un estudio detallado de la distribución temporal del valor del viento, figura 4.3, desglosándolo a niveles diarios y mensuales para identificar con precisión los periodos con registros.

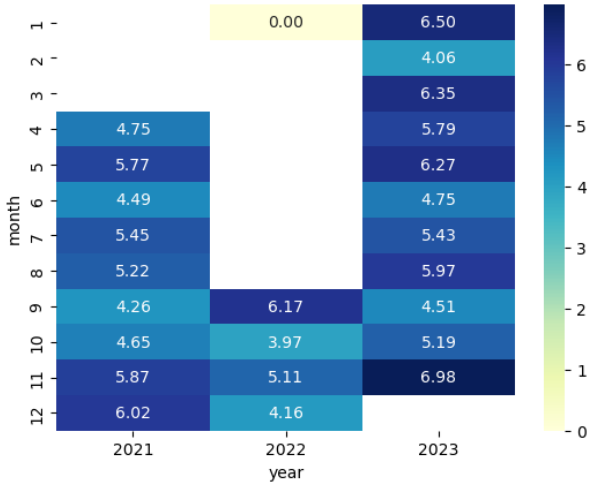


**Figura 4.2.** Valores del Wind_Speed_value en Años y Meses

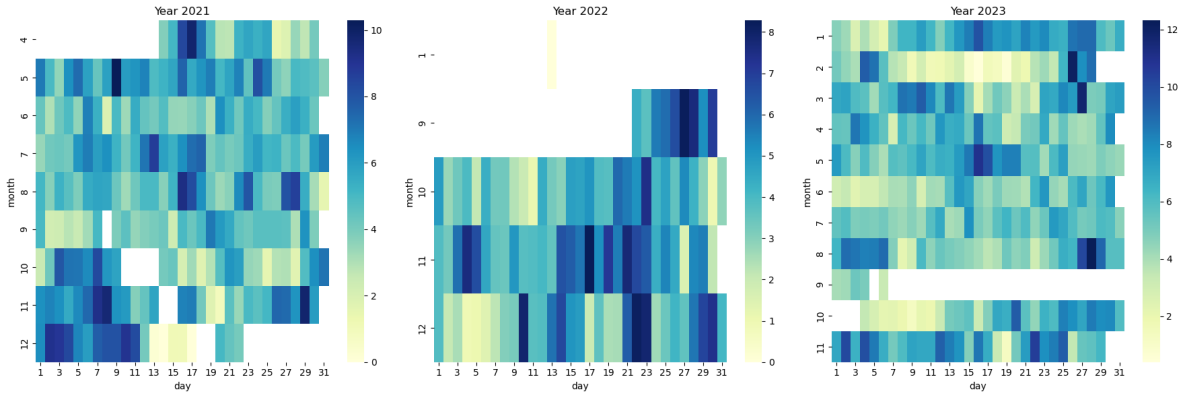


**Figura 4.3.** Valores del WindSpeed_value en Años, Meses y Días

Como se puede observar en la figura 4.4, la distribución del ángulo de la góndola (NacelleAngle) muestra agrupaciones destacadas alrededor de los 0°-10° y entre los 250°-300°, indicando posiciones mas comunes adoptadas por las turbinas en respuesta a la dirección predominante del viento. Por otra parte, la figura 4.5 presenta la distribución de la velocidad del rotor (RotorSpeed), donde destaca un comportamiento con picos en torno a los 7 y 12 revoluciones por minuto (rpm).

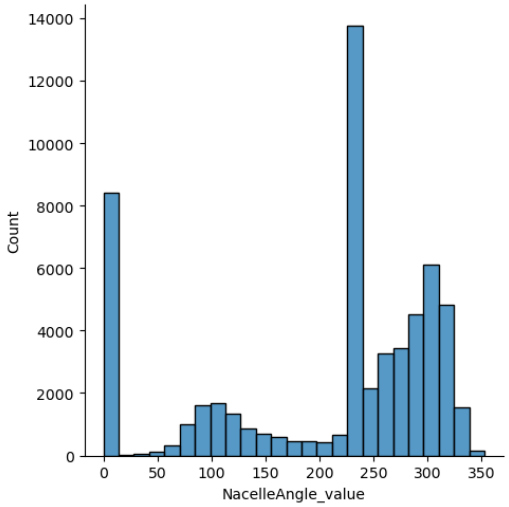


**Figura 4.4.** Distribución del NacelleAngle

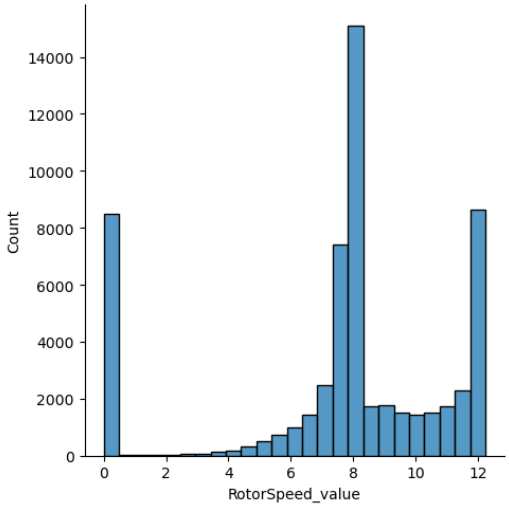


**Figura 4.5.** Distribución del RotorSpeed

La Figura 4.6 muestra la distribución de la velocidad del viento (WindSpeed), que presenta una clara asimetría positiva, concentrando la mayoría de sus valores en velocidades bajas y medias (0-10 m/s), lo que es típico en entornos eólicos reales. Por otro lado, en la Figura 4.7, la distribución de la dirección del viento (WindDirection) evidencia un predominio en el rango entre 150° y 250°, con un pico adicional significativo alrededor de los 0°, indicando que el viento tiende a provenir de direcciones específicas con mayor frecuencia, información esencial para la optimización de la orientación y operación de las turbinas eólicas.

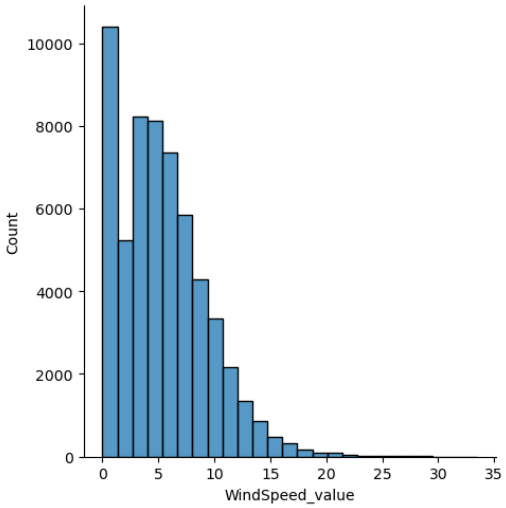


**Figura 4.6.** Distribución del WindSpeed

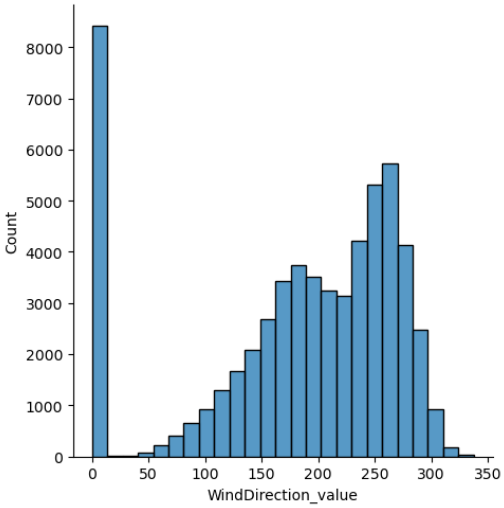


**Figura 4.7.** Distribución del WindDirection

Como puede observarse en la figura 4.8, el eje X representa el ángulo de la góndola (NacelleAngle_value), medido en grados (°), mientras que el eje Y muestra la velocidad del viento (WindSpeed_value), en metros por segundo (m/s). La gráfica muestra mediante curvas de densidad que ciertas combinaciones del ángulo de la góndola y velocidades del viento son más frecuentes. Destaca especialmente una alta concentración en ángulos alrededor de los 200°-300° para velocidades del viento entre 5 y 15 m/s, lo que sugiere que las turbinas tienden a orientarse a estas posiciones para maximizar la captación energética en estas condiciones predominantes.

En la Figura 4.9, el eje X muestra la velocidad de rotación del rotor (RotorSpeed_value), en revoluciones por minuto (rpm), y el eje Y representa la velocidad del viento (WindSpeed_value), en m/s. Se observa que las velocidades del rotor tienden a aumentar junto con la velocidad del viento, concentrándose principalmente entre 7 y 12 rpm para velocidades del viento en torno a 5-15 m/s. Este comportamiento es esperado ya que mayores velocidades del viento generan una mayor rotación del rotor, incrementando la producción energética.

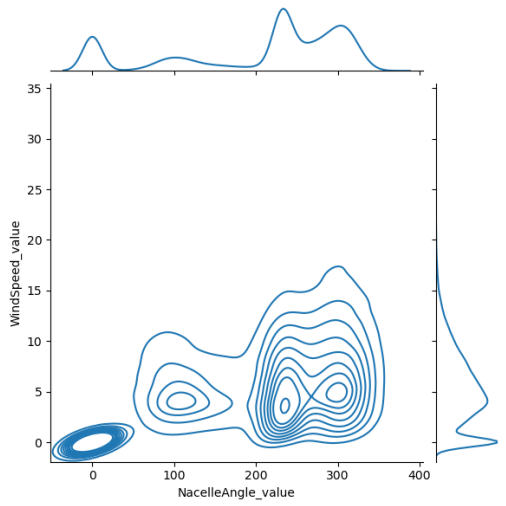


**Figura 4.8.** NacelleAngle vs WindSpeed

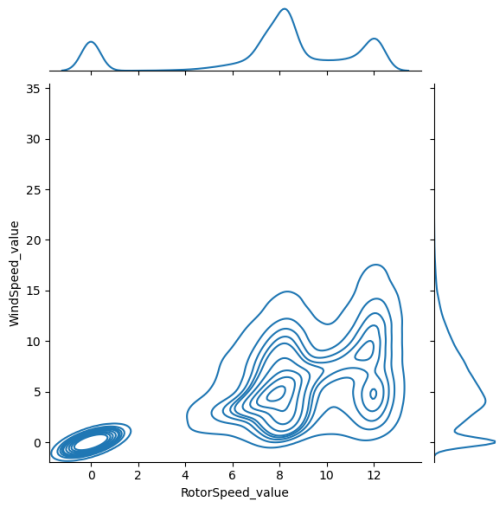


**Figura 4.9.** RotorSpeed vs WindSpeed

La Figura 4.10 muestra en el eje X la dirección del viento (WindDirection_value), medida en grados (°), y en el eje Y la velocidad del viento (WindSpeed_value), en m/s. La gráfica indica claramente una mayor densidad en direcciones del viento alrededor de 150°-250°, con velocidades predominantemente entre 5 y 10 m/s. Esto implica que el viento prevalente proviene principalmente de estas direcciones específicas y velocidades moderadas, información crucial para orientar y optimizar la operación de las turbinas.

Finalmente, en la Figura 4.11, el eje X representa la potencia activa generada (ActivePower_value_KWh), medida en kilovatios hora (kWh), y el eje Y muestra nuevamente la velocidad del viento (WindSpeed_value), en m/s. Aquí se observa una clara tendencia ascendente en la generación de potencia conforme aumenta la velocidad del viento, destacando especialmente en el rango de 5 a 15 m/s. Esta relación refleja cómo la potencia generada depende fuertemente de la velocidad del viento, lo que valida la relevancia de esta variable para los modelos predictivos desarrollados.

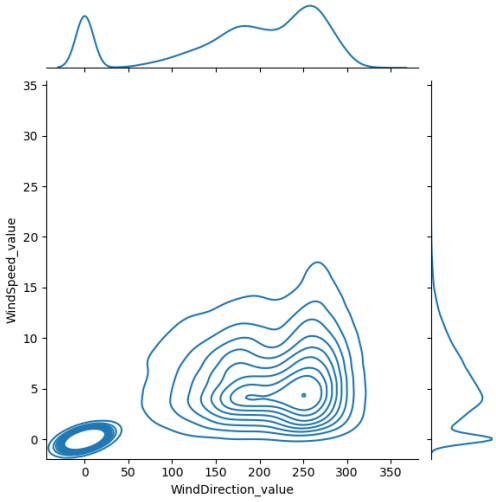


**Figura 4.10.** WindDirection vs WindSpeed

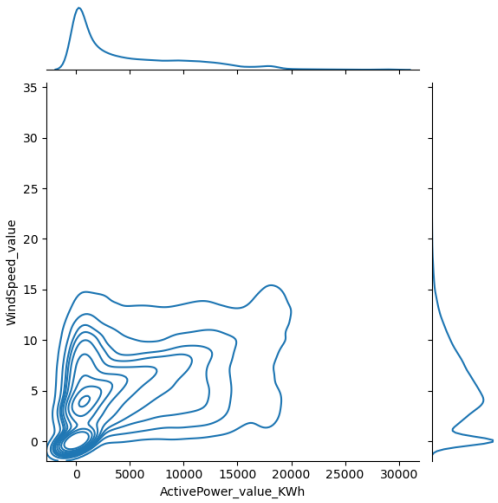


**Figura 4.11.** ActivePower vs WindSpeed

Por último, las correlaciones presentadas en el mapa de calor, figura 4.12, han sido calculadas utilizando el coeficiente de correlación de Pearson. Este coeficiente mide el grado de relación lineal entre dos variables numéricas, proporcionando valores entre -1 y 1. Un valor cercano a 1 indica una fuerte correlación positiva, es decir, cuando una variable aumenta, la otra también lo hace. Un valor cercano a -1 refleja una fuerte correlación negativa, indicando que cuando una variable aumenta, la otra disminuye. Por otro lado, valores cercanos a 0 sugieren ausencia de correlación lineal entre las variables.

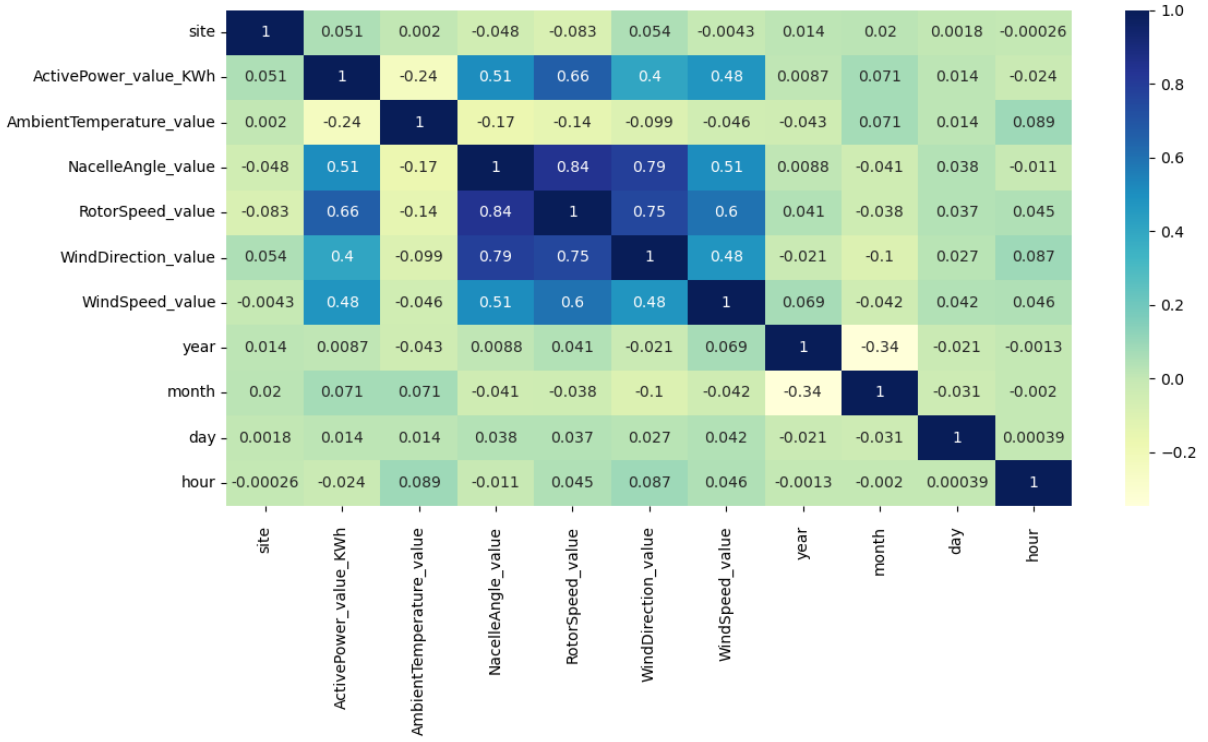


**Figura 4.12.** Mapa de Correlaciones del Parque Eólico

### 4.1.6. Análisis de Valores Atípicos

Se han realizado unas visualizaciones utilizando `boxplots` para identificar valores atípicos en variables como `WindSpeed_value`, `WindDirection_value`, `AmbientTemperature_value`, y `NacelleAngle_value`. Se puede observar en la figura 4.13, que existen varios valores atípicos, esto nos da a entender que necesitamos realizar ajustes adicionales para mejorar la calidad de los datos. Para ello, imputamos los valores atípicos por la media, dando un resultado limpio de valores atípicos que se puede ver en la figura 4.14.

Este proceso nos asegura que los datos están preparados para su uso en modelos predictivos, permitiendo un análisis más preciso y eficiente de la generación energética.

## 4.2. Descripción de los Datos de los Parques Solares N1 y N2

Los datasets, `Plant_Solar_Generation_Data` y `Plant_Solar_Weather_Sensor_Data`, incluyen mediciones clave para el análisis del desempeño de instalaciones solares, capturando tanto la generación de energía como las condiciones ambientales.

Este conjunto de datos se obtuvo de la plataforma Kaggle, específicamente del dataset titulado "Solar Power Generation Data"

(https://www.kaggle.com/datasets/anikannal/solar-power-generation-data/code?datasetId=836676&sortBy=voteCount)

A continuación, se han detallado los pasos seguidos para la carga y la preparación inicial de

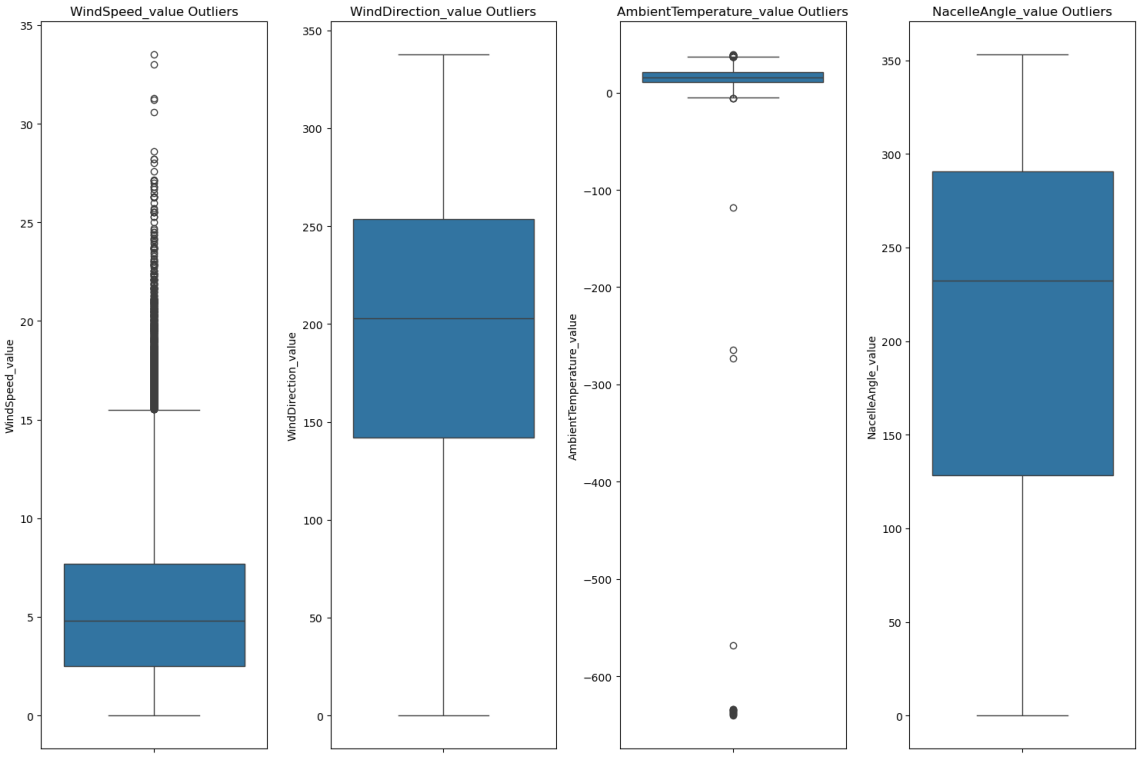


**Figura 4.13.** Análisis de Valores Atípicos para Variables Clave del Dataset

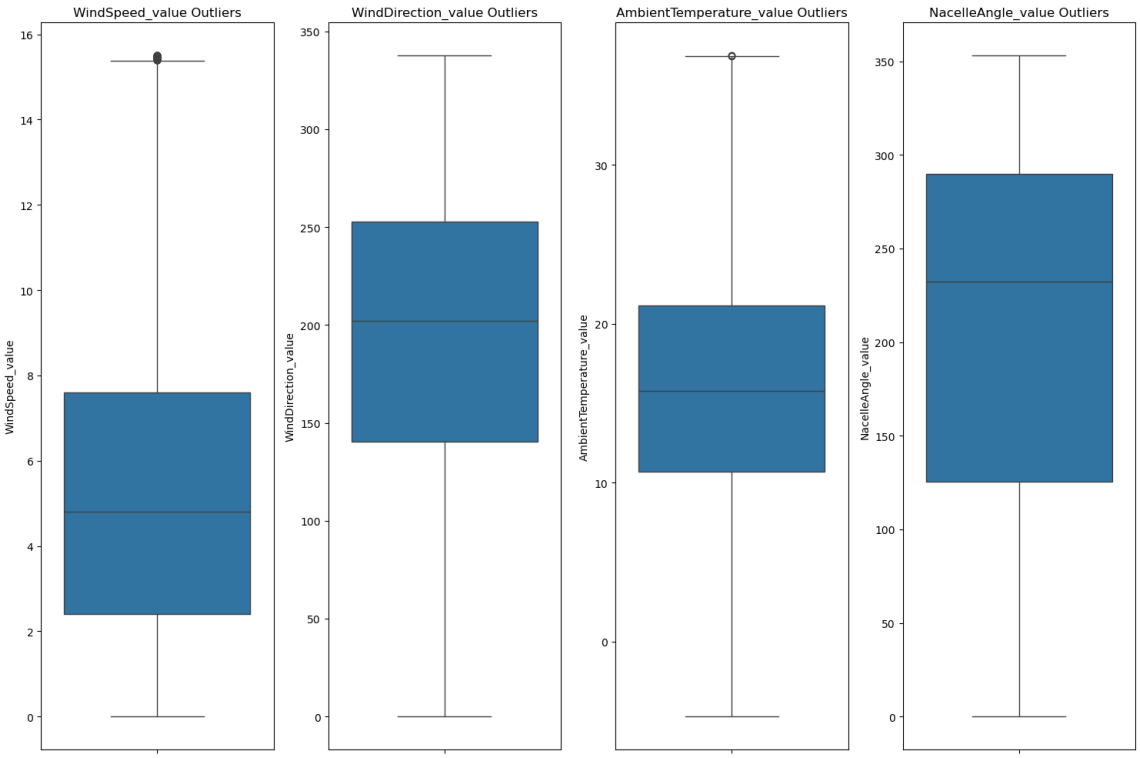


**Figura 4.14.** Distribución de Variables tras la Eliminación de Valores Atípicos

estos datos:

### 4.2.1. Cargar el DataFrame

Se utiliza la función `pd.read_csv()` de la biblioteca Pandas para cargar los datos desde un archivo CSV, facilitando así su manipulación y análisis posterior.

### 4.2.2. Manipulación de la Columna DATE_TIME

La columna `DATE_TIME`, que contiene fechas y horas de las mediciones, se convierte a formato `datetime` con `pd.to_datetime()`, permitiendo una descomposición eficiente en componentes de fecha y hora (`year`, `month`, `day`, `hour`). Posteriormente, se elimina cualquier columna redundante para simplificar el DataFrame.

### 4.2.3. Preprocesamiento

1. **Lectura y consolidación de datos:** Todos los datos son consolidados en un único DataFrame para unificar la información y facilitar su manipulación.
2. **Limpieza de datos:** Se identifican y corrigen valores faltantes, inconsistentes o erróneos. Específicamente, los valores nulos en columnas críticas se imputan usando el promedio de la columna.
3. **Transformación y normalización:** Se aplican transformaciones a las variables para adaptarlas a la escala requerida por los algoritmos de predicción. Las variables numéricas se normalizan y las categóricas se convierten en formatos numéricos para facilitar la convergencia de los modelos durante el entrenamiento.

### 4.2.4. Análisis de los Datos Obtenidos en el Parque Solar N1

En la gráfica 4.15 se puede ver la generación de corriente continua en en el parque solar N1, que alberga 22 placas solares. También se puede apreciar que se empieza a generar corriente desde las 5:45 de la mañana aproximadamente, que coincide con la salida del sol, hasta las 18:30, que coincide con la puesta de sol. Es cierto que aparece a medio día valores nulos, pero puede estar relacionado con la posibilidad que a esa hora estuviese nublado, haciendo que algunas placas solares no pudieran obtener energía.

A continuación se expone las gráficas de otras variables influyentes que permiten una comparación detallada y un análisis profundo.

Como se puede observar en la Figura 4.16, la distribución de la potencia en corriente continua (DC_POWER) presenta una fuerte concentración en valores bajos, especialmente próximos a cero, y una larga cola hacia valores más altos. En esta gráfica, el eje X representa la potencia generada en vatios (W), mientras que el eje Y muestra el número de registros correspondientes a cada intervalo. Esta distribución sugiere que durante la mayor parte del tiempo, la planta solar produce niveles bajos de energía, lo cual es coherente con los ciclos naturales de iluminación y la presencia de condiciones no ideales.

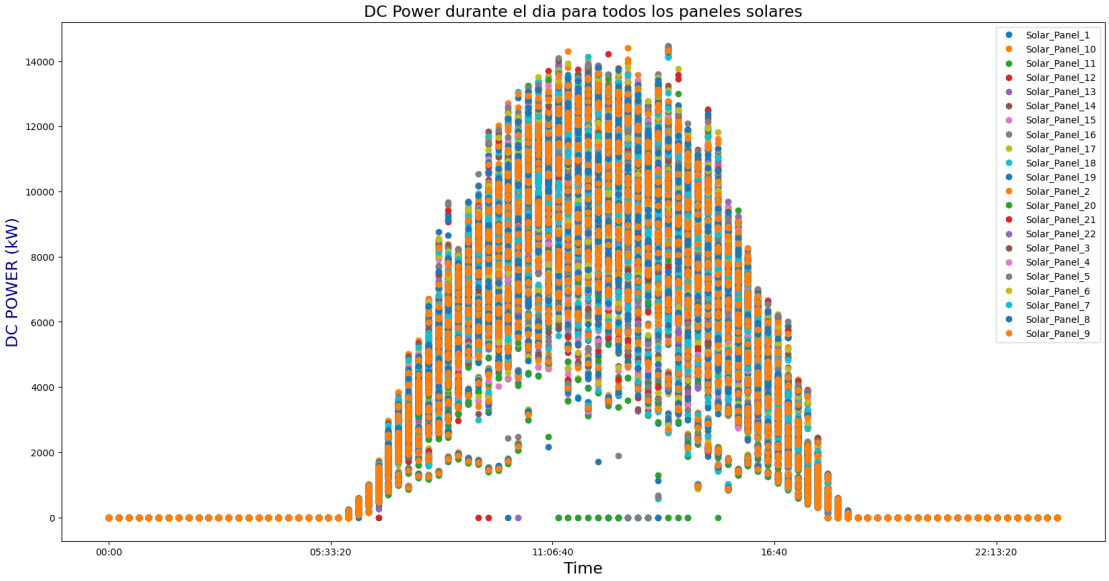


**Figura 4.15.** Energía Generada Durante un Día en el Parque Solar N1

Por otra parte, la Figura 4.17 muestra la distribución de la temperatura ambiente (AMBIENT_TEMPERATURE), medida en grados Celsius. El eje X representa la temperatura, mientras que el eje Y indica la frecuencia de ocurrencia. Se observa una forma de distribución asimétrica hacia la derecha, con un mayor número de observaciones entre los 22°C y 26°C, lo que indica que estas son las condiciones térmicas predominantes en el entorno de la instalación solar.

La Figura 4.18 representa la distribución de la irradiancia solar (IRRADIATION). En este caso, el eje X muestra los niveles de irradiancia en kW/m$^2$, y el eje Y el número de registros. La mayoría de los valores se sitúan próximos a cero, lo que se explica por la presencia de numerosas mediciones tomadas durante horas sin luz solar (por la noche o en condiciones nubladas), mientras que los valores altos son menos frecuentes y corresponden a los momentos de mayor exposición solar.

La Figura 4.19 muestra la distribución de la producción energética diaria (DAILY_YIELD), expresada en kilovatios hora (kWh). El eje X representa los valores de energía diaria generada, y el eje Y indica la frecuencia. Se observa un número considerable de días con producciones bajas, mientras que los valores más altos son menos comunes, lo que puede estar relacionado con condiciones climatológicas específicas o limitaciones operativas puntuales.

Finalmente, la Figura 6.20 presenta la distribución de la temperatura de los módulos fotovoltaicos (MODULE_TEMPERATURE). En el eje X se encuentra la temperatura en grados Celsius y en el eje Y la frecuencia. La distribución se concentra principalmente entre 20°C y 35°C, pero alcanza valores de hasta 60°C.

Por otro lado se comprueba la relación que tienen diferentes variables y su importante relación.

Como puede observarse en la Figura 4.21, el eje X representa la potencia en corriente continua generada (DC_POWER), medida en vatios (W), mientras que el eje Y muestra la temperatura ambiente (AMBIENT_TEMPERATURE), en grados Celsius (°C). La gráfica revela que las mayores concentraciones de generación energética se producen cuando la temperatura ambiente oscila entre los 25°C y 30°C, lo que sugiere que dichas condiciones térmicas son óptimas para el funcionamiento eficiente de los paneles solares. Esta información resulta útil para identificar los umbrales ambientales en los que se maximiza el rendimiento del sistema fotovoltaico.

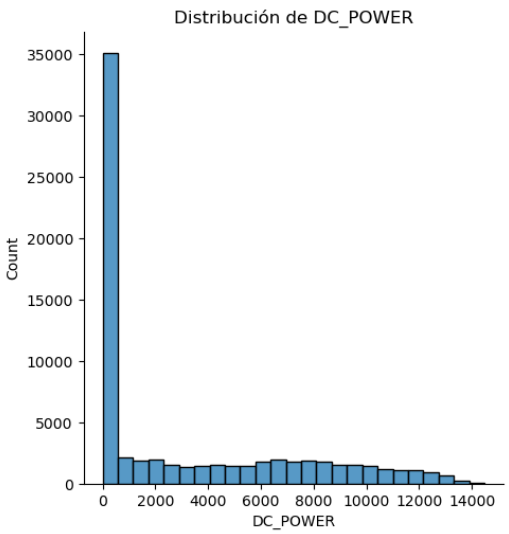


**Figura 4.16.** Distribución de DC POWER

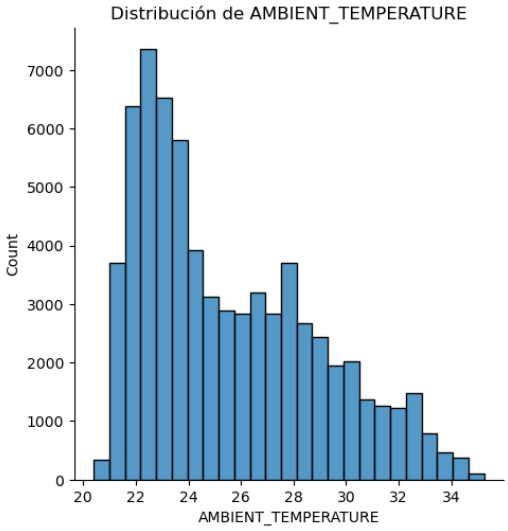


**Figura 4.17.** Distribución de AMBIENT TEMPERATURE

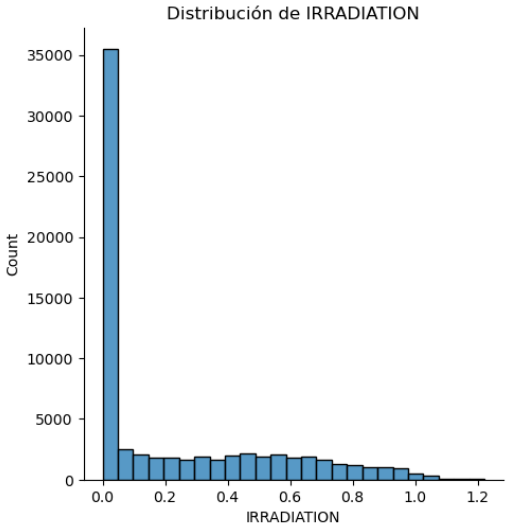


**Figura 4.18.** Distribución de IRRADIATION

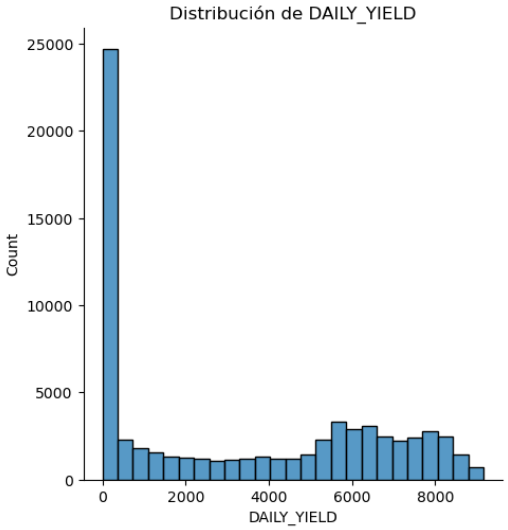


**Figura 4.19.** Distribución de DAILY YIELD

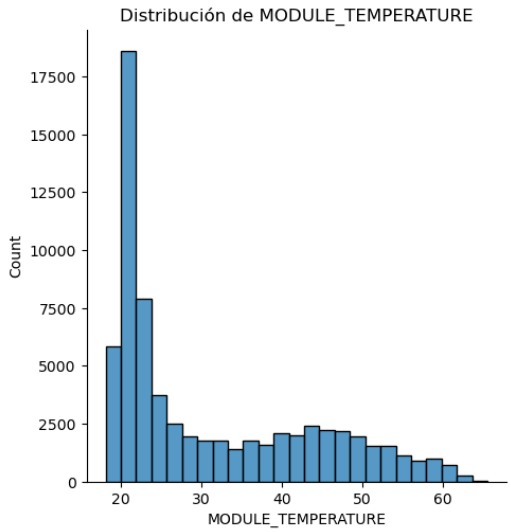


**Figura 4.20.** Distribución de MODULE TEMPERATURE

En la Figura 4.22, el eje X representa nuevamente la potencia DC_POWER (en W), y el eje Y corresponde a la temperatura del módulo (MODULE_TEMPERATURE), también medida en grados Celsius. Se observa una clara relación creciente entre ambas variables, mostrando que a medida que aumenta la potencia generada, la temperatura de los módulos también se incrementa. Este comportamiento es coherente con el hecho de que el funcionamiento intensivo de los paneles solares conlleva un aumento térmico, lo cual debe ser considerado ya que las temperaturas elevadas pueden afectar la eficiencia del sistema a largo plazo.

La Figura 4.23 presenta en el eje X la potencia DC_POWER y en el eje Y la irradiancia solar (IRRADIATION), medida en kilovatios por metro cuadrado ($kW/m^2$). La gráfica muestra una fuerte correlación positiva: a mayor irradiancia, mayor generación de potencia. Este resultado es lógico y esperado, ya que la energía solar disponible es el principal factor que determina la cantidad de energía que puede generar un panel fotovoltaico. La alta densidad a lo largo de la diagonal confirma esta dependencia directa.

Finalmente, en la Figura 4.24, se aprecia una forma similar a una 'U' invertida desplazada hacia la izquierda, lo que sugiere un patrón temporal implícito en los datos. Este comportamiento refleja el ciclo diario de generación solar: cuando la energía DC_POWER es cero, indica que aún no ha salido el sol o que ya ha anochecido. A medida que transcurre el día y aumenta la irradiancia, se observa un incremento en el valor de DAILY_YIELD, hasta alcanzar un punto máximo, seguido de una disminución progresiva conforme la luz solar desaparece. Este patrón queda aún más claro en la Figura 4.25, donde se representa de manera más explícita la evolución diaria del rendimiento energético.

Las correlaciones mostradas en la Figura 4.26 han sido calculadas mediante el coeficiente de Pearson, que mide la relación lineal entre variables numéricas con valores entre -1 y 1. Valores cercanos a 1 indican una fuerte correlación positiva; valores próximos a -1, una correlación negativa; y valores cercanos a 0, una ausencia de relación lineal. Este análisis se realizó con la función corr() de la biblioteca Pandas en Python, sobre datos previamente normalizados.

En el mapa de calor se observa que DC_POWER presenta una correlación muy alta con IRRADIATION (0.99) y MODULE_TEMPERATURE (0.95), lo que confirma la influencia directa de la irradiancia solar y la temperatura del módulo en la generación energética. También se aprecia una correlación significativa con AMBIENT_TEMPERATURE (0.72). En cambio,

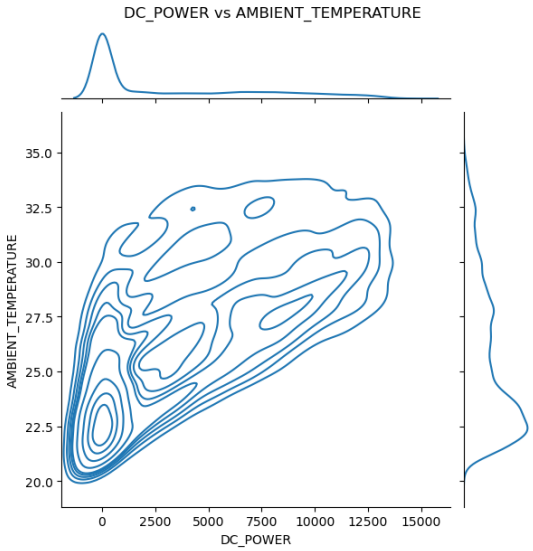


**Figura 4.21.** AMBIENT TEMPERATURE vs DC POWER

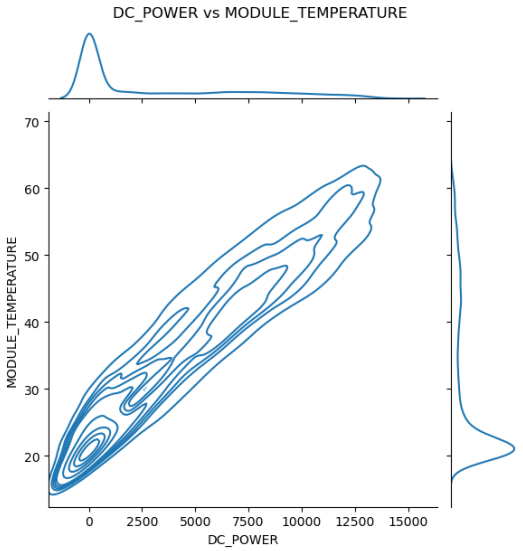


**Figura 4.22.** MODULE TEMPERATURE vs DC POWER

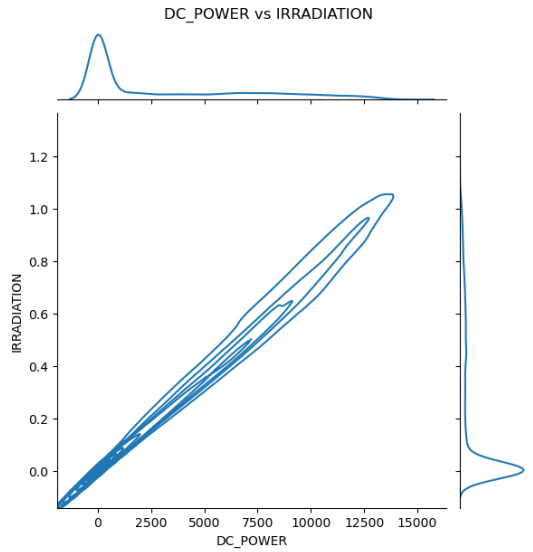


**Figura 4.23.** IRRADIATION vs DC POWER

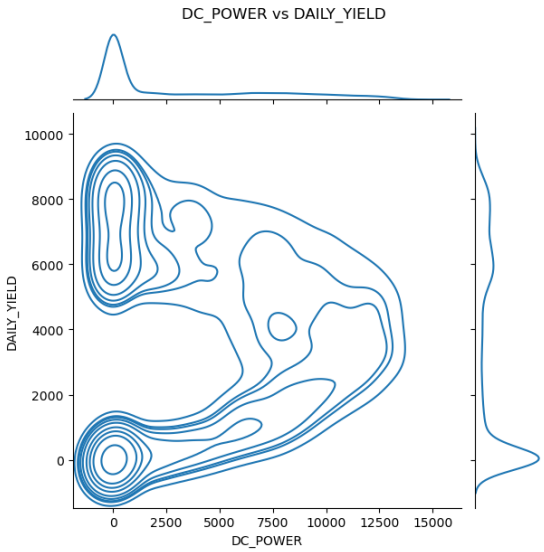


**Figura 4.24.** DAILY YIELD vs DC POWER

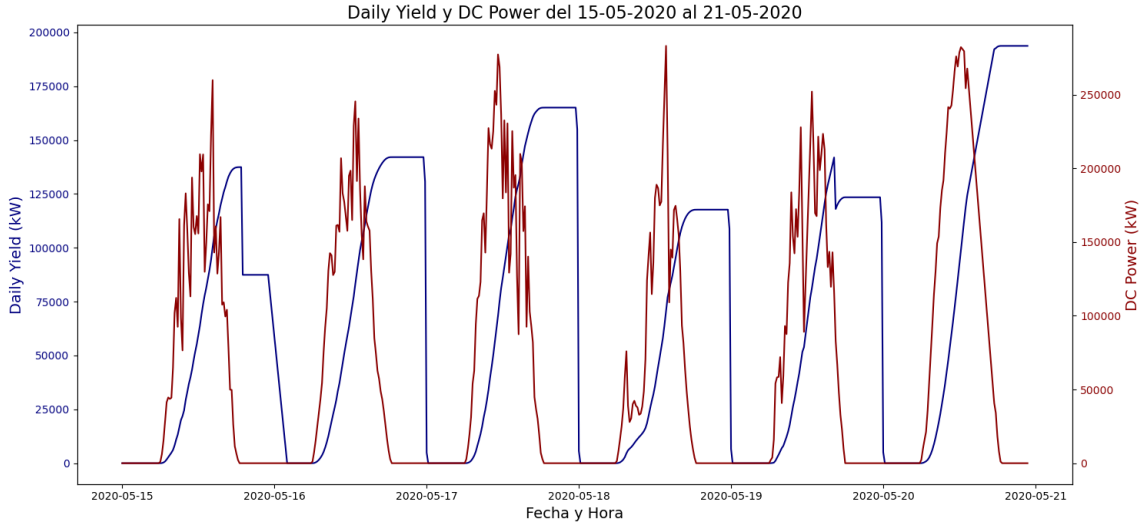


**Figura 4.25.** DAILY YIELD vs DC POWER

DAILY_YIELD muestra valores bajos de correlación, ya que se trata de una variable acumulativa y no instantánea. Este análisis permite identificar qué variables son más relevantes para el modelado predictivo del rendimiento en plantas solares.

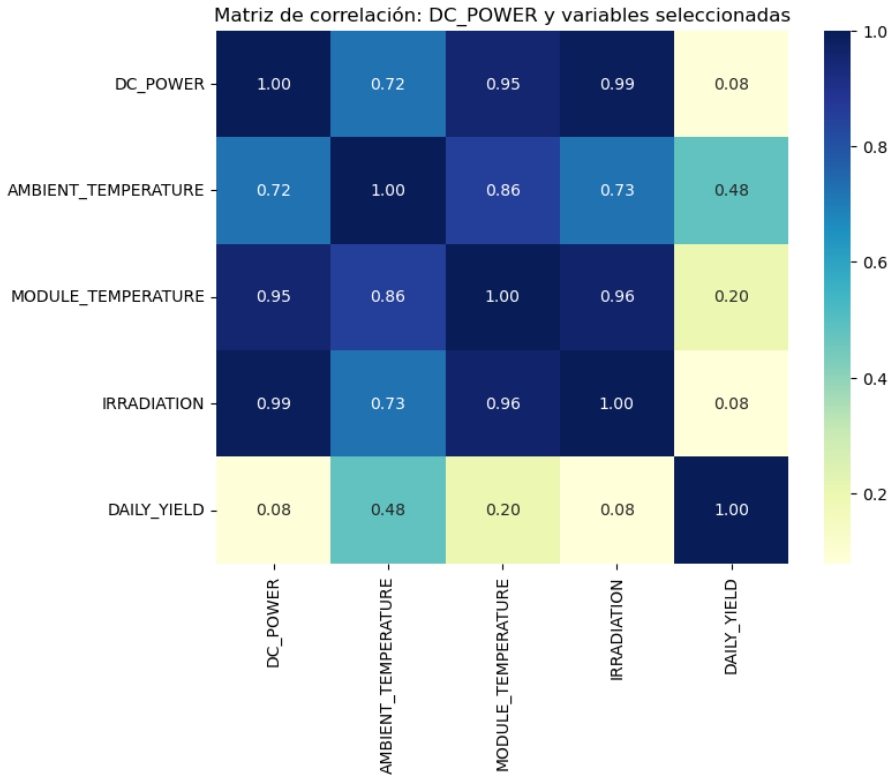


**Figura 4.26.** Mapa de Correlaciones del Parque Solar N1

### 4.2.5. Análisis de Valores Atípicos en el Parque Solar N1

Se realizan visualizaciones utilizando `boxplots` para identificar valores atípicos en variables como `AMBIENT_TEMPERATURE`, `MODULE_TEMPERATURE`, `IRRADIATION`, y `DAILY_YIELD`, se puede observar en la figura 4.27. Esto permite realizar ajustes adicionales para mejorar la calidad de los datos, en este caso imputamos los valores atípicos por la media. El resultado de la manipulación de los outliners se puede ver en la figura 4.28.

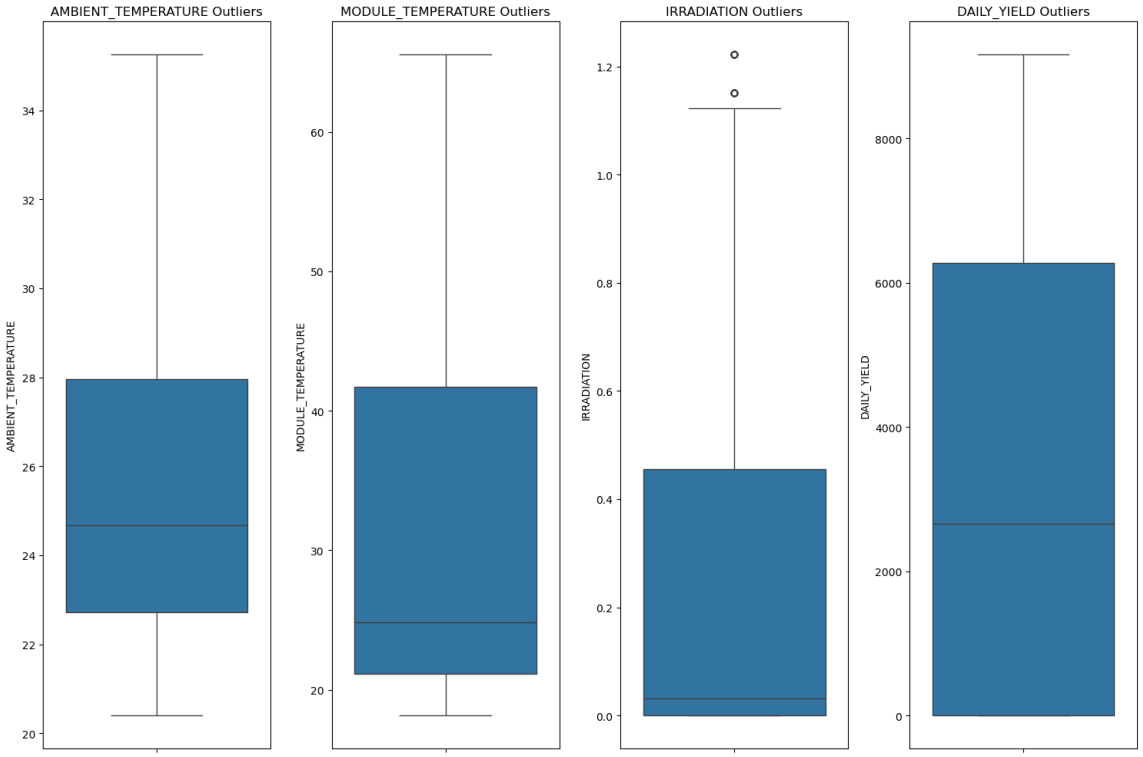


**Figura 4.27.** Análisis de Valores Atípicos para Variables Clave del Dataset del Parque Solar N1

Este proceso detallado asegura que los datos estén preparados adecuadamente para su uso en modelos predictivos, permitiendo un análisis más preciso y eficiente de la generación energética y las condiciones operativas en parques solares.

### 4.2.6. Análisis de los Datos Obtenidos en el Parque Solar N2

El análisis inicial de los datos comienza con una representación gráfica que permite observar el comportamiento de la generación de corriente continua en el parque solar N2, que, al igual que el parque N1, está equipado con 22 paneles solares. La figura 4.29 muestra cómo la generación energética inicia aproximadamente a las 5:45 horas, coincidiendo con el amanecer, y finaliza alrededor de las 18:30 horas, hora aproximada de la puesta del sol. Se observa también un aumento de valores cercanos a cero, fenómeno probablemente asociado a condiciones climáticas desfavorables como días nublados.

Además de la potencia generada, es importante realizar un estudio exhaustivo del comportamiento del resto de variables clave.

Como puede observarse en la Figura 4.30, la distribución de la potencia en corriente continua

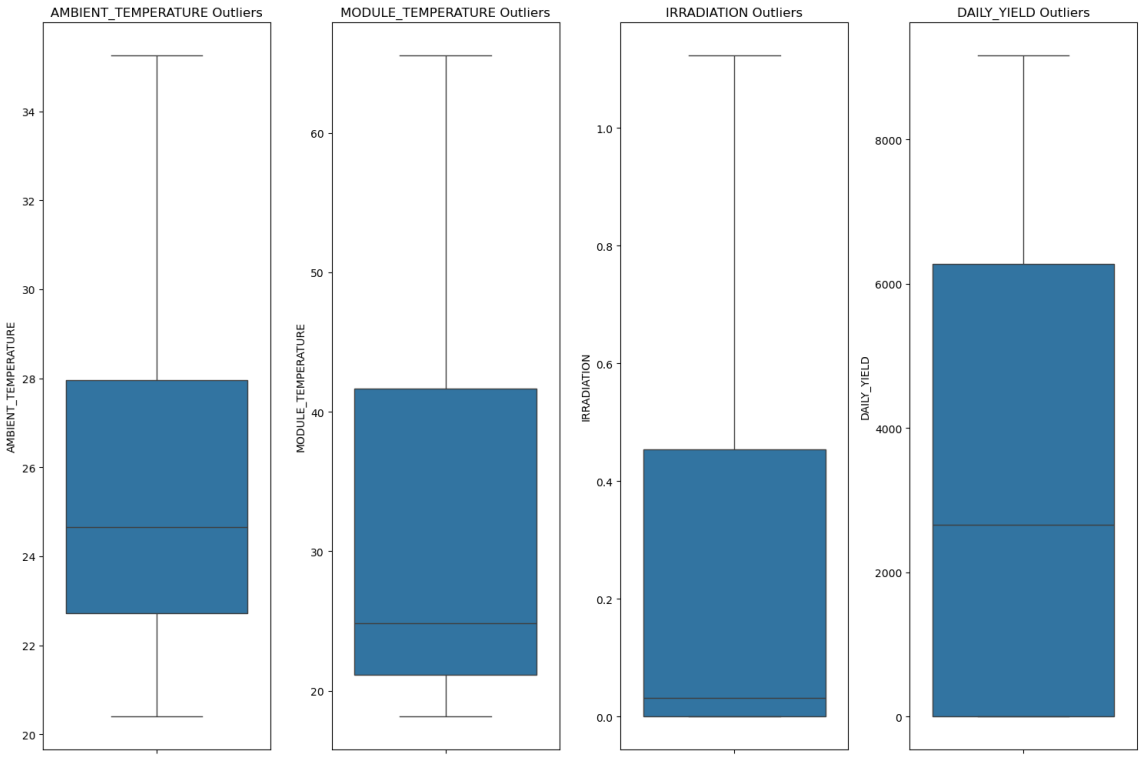


**Figura 4.28.** Distribución de Variables tras la Eliminación de Valores Atípicos del Parque Solar N1

(DC_POWER) presenta una alta concentración de valores en torno a cero, con una caída rápida hacia valores mayores. El eje X representa la potencia generada en vatios (W), mientras que el eje Y indica la frecuencia de observación. Esta distribución refleja que la mayoría de los registros corresponden a momentos de baja o nula generación, típicamente asociados a las primeras horas del día, al anochecer o a condiciones de baja irradiación.

En la Figura 4.31 se muestra la distribución de la temperatura ambiente (AMBIENT TEMPERATURE), medida en grados Celsius (°C). El eje X indica el valor de la temperatura y el eje Y la cantidad de registros. La gráfica revela una mayor frecuencia de temperaturas entre 24°C y 30°C, lo que sugiere que estas son las condiciones predominantes en la ubicación del parque solar. A medida que la temperatura aumenta, la frecuencia disminuye progresivamente.

La Figura 4.32 presenta la distribución de la irradiancia solar (IRRADIATION), medida en kilovatios por metro cuadrado ($kW/m^2$). El eje X representa los niveles de irradiancia y el eje Y la frecuencia. Se observa una gran acumulación de valores en torno a cero, lo cual es consistente con las mediciones realizadas durante la noche o en condiciones de nubosidad intensa. El resto de valores se distribuyen de forma más dispersa, indicando que los periodos de irradiancia elevada son menos frecuentes.

La Figura 4.33 muestra la distribución de la producción energética diaria (DAILY_YIELD), en kilovatios hora (kWh). El eje X representa la energía generada acumulada por día, mientras que el eje Y indica el número de ocurrencias. Al igual que en los casos anteriores, hay una gran cantidad de días con producción muy baja o nula, probablemente asociada a condiciones meteorológicas desfavorables. La distribución muestra un patrón escalonado, reflejando la variabilidad diaria en la generación solar a lo largo del tiempo.

Por último, la Figura 4.34 muestra la distribución de la temperatura del módulo fotovoltaico

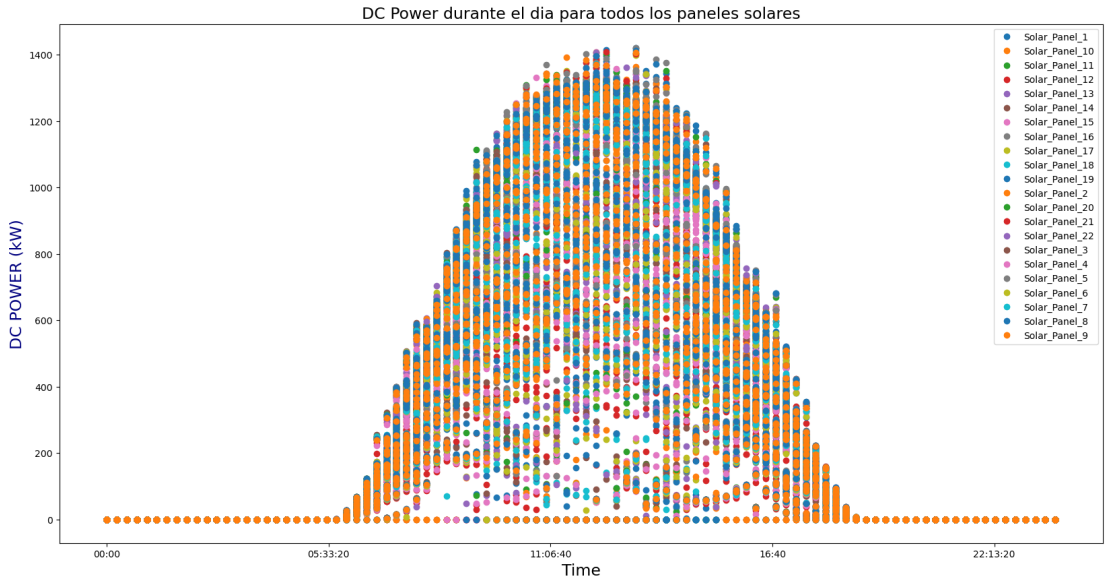


**Figura 4.29.** Energía Generada Durante un Día en el Parque Solar N2

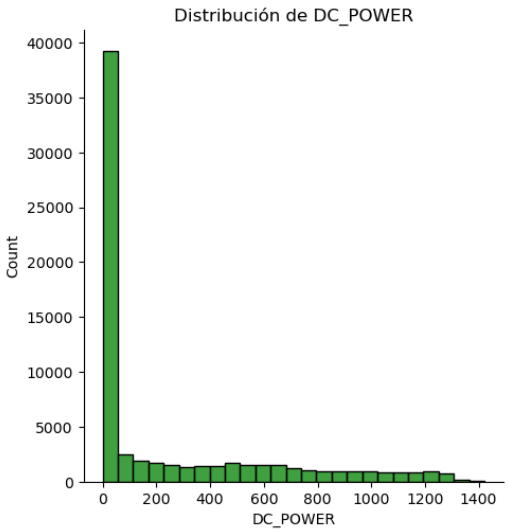


**Figura 4.30.** Distribución de DC POWER

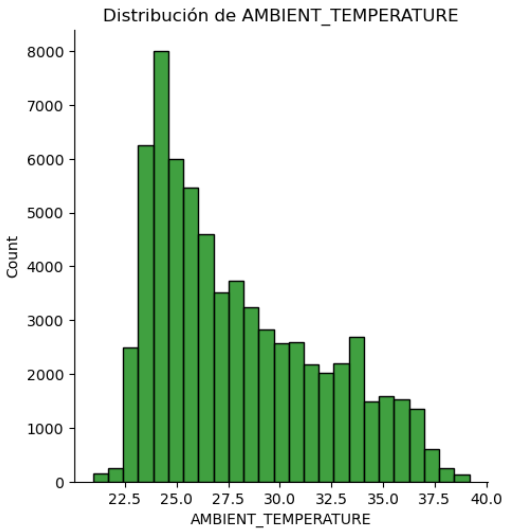


**Figura 4.31.** Distribución de AMBIENT TEMPERATURE

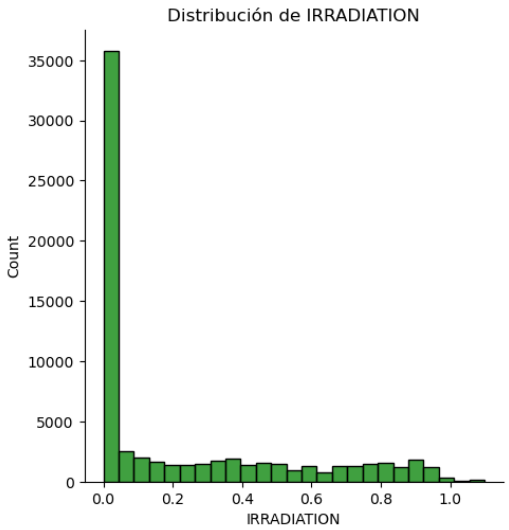


**Figura 4.32.** Distribución de IRRADIATION

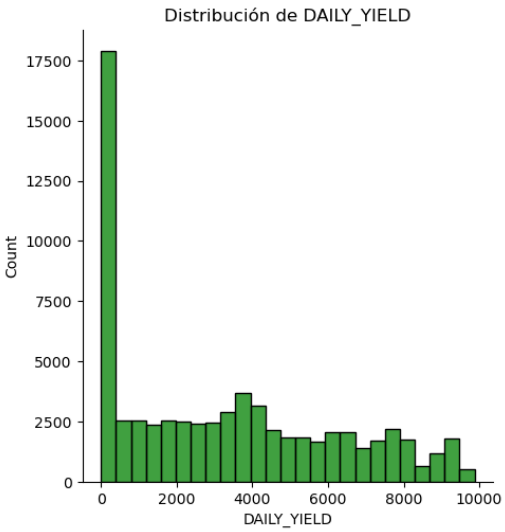


**Figura 4.33.** Distribución de DAILY YIELD

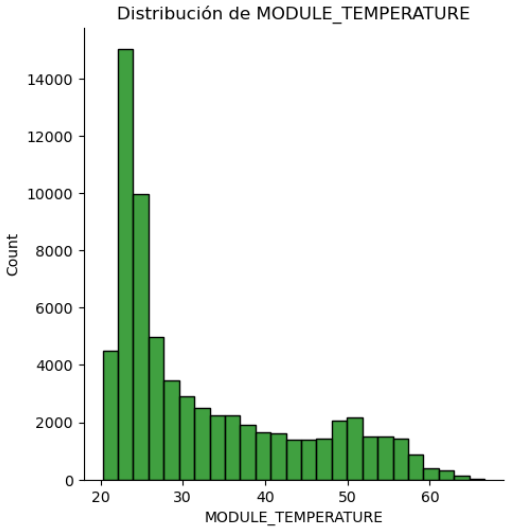


**Figura 4.34.** Distribución de MODULE TEMPERATURE

(MODULE_TEMPERATURE), medida en grados Celsius (°C). El eje X representa la temperatura de los módulos, mientras que el eje Y indica la frecuencia de aparición de dichos valores. Se observa una clara asimetría hacia la derecha, con una alta concentración de temperaturas en el rango de 20°C a 30°C, que corresponden a condiciones de operación moderadas. A partir de los 35°C, la frecuencia disminuye progresivamente, aunque se detectan algunos picos secundarios que podrían estar asociados a condiciones de alta irradiación o a acumulación térmica durante periodos prolongados de exposición solar.

Para profundizar en la influencia de estas variables sobre la potencia generada, se presentan correlaciones directas entre cada una de ellas y la variable `DC_POWER`.

Se observa en la Figura 4.35 cómo el incremento en la temperatura ambiente repercute positivamente en la generación energética medida como DC_POWER. En esta gráfica, el eje X representa la potencia en corriente continua (W) y el eje Y la temperatura ambiente en grados Celsius (°C). Las curvas de densidad muestran que, a medida que aumenta la temperatura hasta

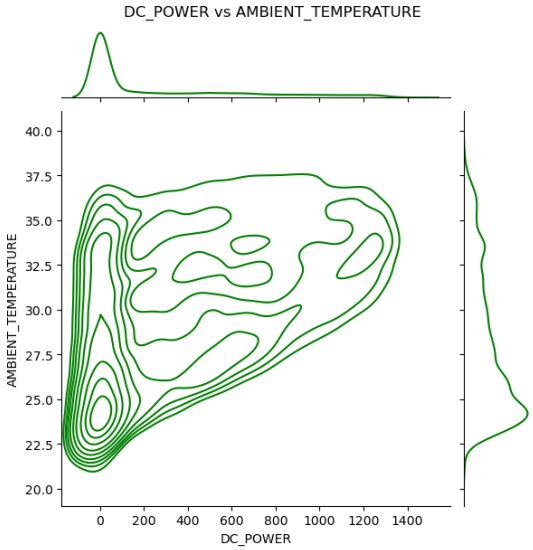


**Figura 4.35.** AMBIENT TEMPERATURE vs DC POWER

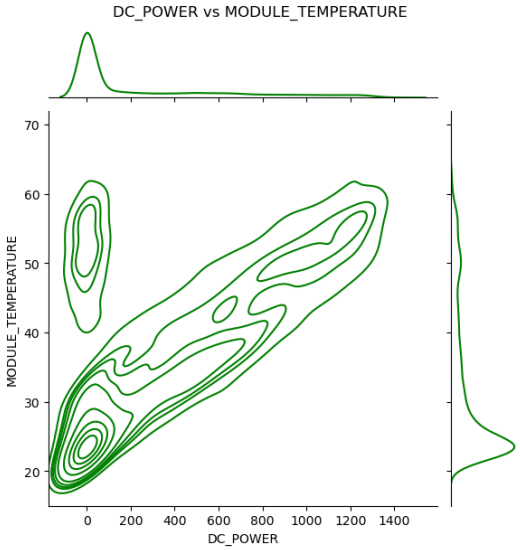


**Figura 4.36.** MODULE TEMPERATURE vs DC POWER

aproximadamente 32°C, también se incrementa la potencia generada, lo que sugiere que estas condiciones térmicas favorecen el funcionamiento del sistema fotovoltaico. No obstante, a partir de ciertos umbrales térmicos más altos, la relación se estabiliza o incluso puede tender a decrecer ligeramente.

De igual forma, en la Figura 4.36 se muestra una tendencia similar con respecto a la temperatura del módulo (MODULE_TEMPERATURE). En este caso, también se aprecia una correlación creciente: a mayores temperaturas del módulo, se observa un aumento paralelo en la potencia generada. Sin embargo, esta relación está sujeta a un efecto límite, ya que temperaturas excesivas podrían llegar a afectar negativamente la eficiencia de los módulos. Este tipo de análisis resulta clave para definir los rangos de temperatura operativa óptimos.

La Figura 4.37 representa la relación entre DC_POWER e IRRADIATION, donde el eje Y muestra la irradiancia solar en kW/m$^2$. Se aprecia una correlación fuertemente positiva, con un patrón casi lineal: cuanto mayor es la irradiación, mayor es la potencia generada. Este comportamiento era esperado, dado que la irradiación solar es el principal insumo energético para el sistema fotovoltaico.

Finalmente, en la Figura 4.38 se muestra la relación entre DC_POWER y la producción acumulada diaria (DAILY_YIELD). La gráfica presenta una forma de "U" invertida, similar a observaciones anteriores, lo que indica que la generación diaria se acumula progresivamente durante las horas de mayor luz solar. Esta visualización permite captar cómo las mediciones instantáneas de potencia (DC_POWER) se reflejan en la acumulación energética diaria, reforzando la coherencia del comportamiento entre ambas variables.

La Figura 4.39 muestra la evolución temporal conjunta de las variables DAILY_YIELD y DC_POWER a lo largo de varios días consecutivos. En el eje X se representa la fecha y hora, mientras que el eje Y izquierdo indica la energía acumulada diaria (DAILY_YIELD, en kWh) y el eje Y derecho muestra la potencia generada instantáneamente (DC_POWER, en W). Se aprecia cómo la curva de DAILY_YIELD presenta un comportamiento escalonado que refleja el incremento acumulativo de energía conforme avanza el día, mientras que DC_POWER sigue un patrón cíclico característico de la producción solar, con picos durante las horas centrales y valores

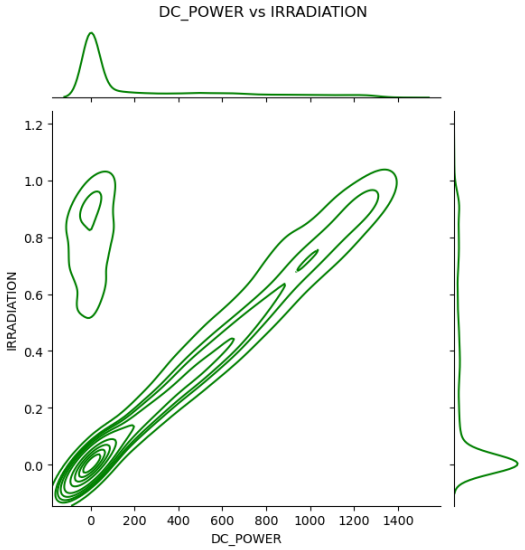


**Figura 4.37.** IRRADIATION vs DC POWER

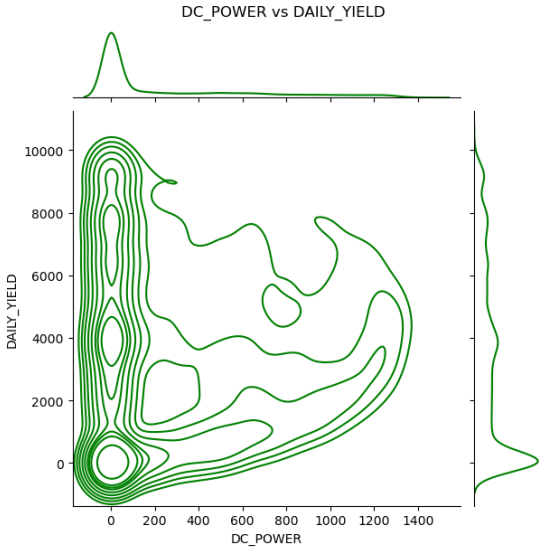


**Figura 4.38.** DAILY YIELD vs DC POWER

nulos durante la noche. Esta representación permite visualizar cómo la potencia instantánea influye directamente en la acumulación energética diaria.

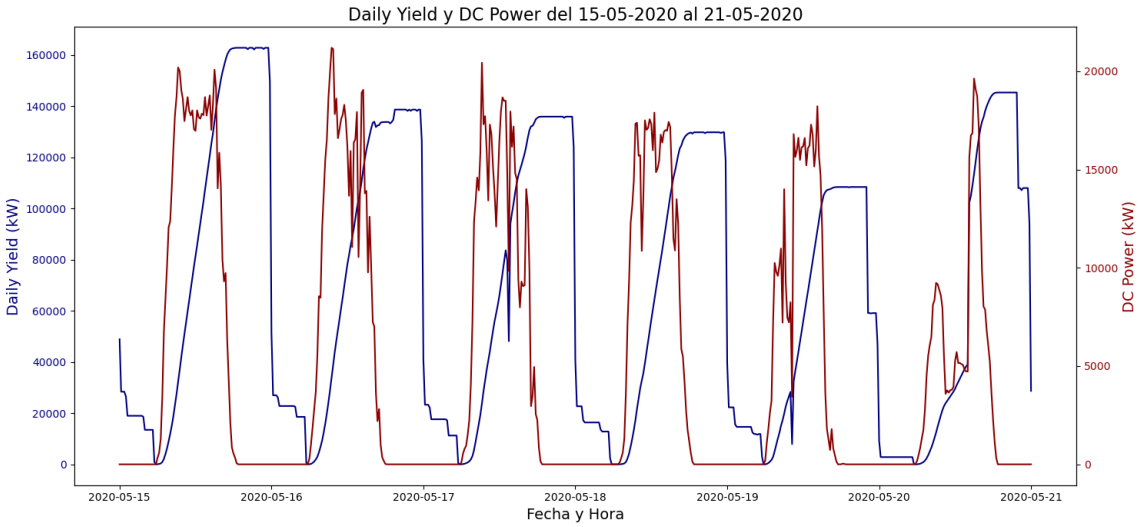


**Figura 4.39.** DAILY YIELD vs DC POWER

Finalmente, se muestra un mapa de correlaciones, Figura 4.40, que permite identificar y cuantificar las interacciones entre las variables clave del sistema. Las correlaciones fueron calculadas mediante el coeficiente de Pearson, utilizando la función corr() de la biblioteca Pandas en Python, sobre datos previamente normalizados. Este coeficiente toma valores entre -1 y 1, donde 1 indica una relación lineal positiva perfecta, -1 una relación negativa perfecta, y valores cercanos a 0 ausencia de correlación lineal.

En este caso, se observa que DC_POWER se correlaciona positivamente con IRRADIATION (0.78), MODULE_TEMPERATURE (0.75) y AMBIENT_TEMPERATURE (0.56), confirmando la influencia directa de estas variables en la producción de energía. La fuerte relación entre MODULE_TEMPERATURE e IRRADIATION (0.95) destaca cómo la radiación solar con-

tribuye al calentamiento de los paneles. Por otro lado, DAILY_YIELD presenta correlaciones más débiles, ya que representa un valor acumulado y no instantáneo, con una ligera correlación con AMBIENT_TEMPERATURE (0.32) y prácticamente nula con DC_POWER (0.01). Este análisis proporciona una base sólida para la selección de variables en los modelos predictivos y contribuye a una mejor comprensión del funcionamiento del sistema fotovoltaico.


**Figura 4.40.** Mapa de Correlaciones del Parque Solar N2

### 4.2.7. Análisis de Valores Atípicos en el Parque Solar N2

Para asegurar la calidad de los datos, se realizó un análisis mediante `boxplots` (figura 4.41) para identificar valores atípicos. Posteriormente, estos valores se ajustaron mediante imputación por la media, observándose el resultado en la figura 4.42, garantizando así la precisión y fiabilidad del análisis posterior.

Este proceso es crucial para optimizar los modelos predictivos posteriores, mejorando así la calidad general de los resultados obtenidos en el estudio energético.

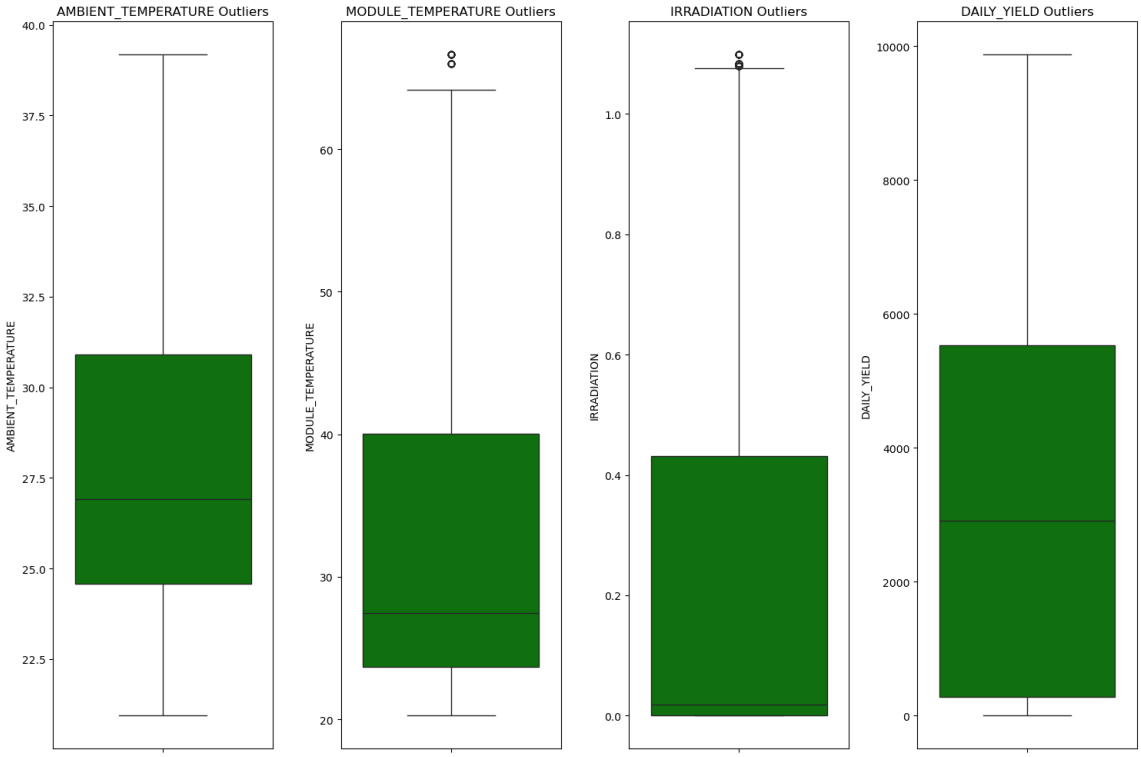


**Figura 4.41.** Análisis de Valores Atípicos para Variables Clave del Dataset del Parque Solar N2

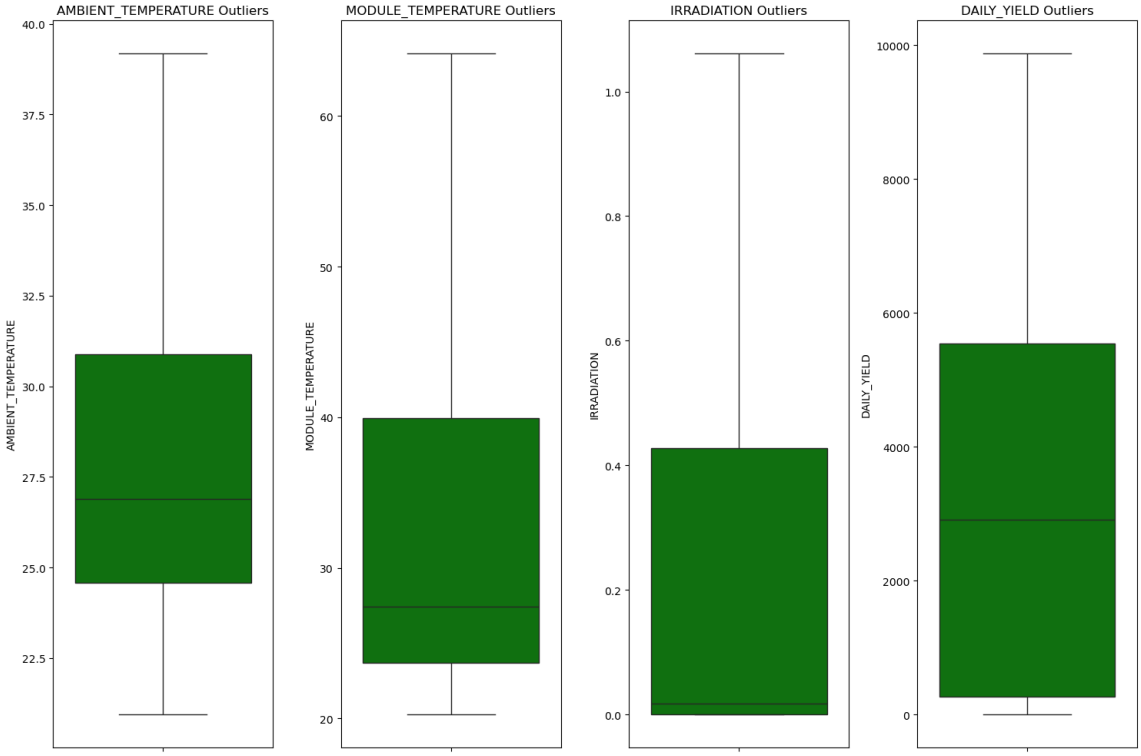


**Figura 4.42.** Distribución de Variables tras la Eliminación de Valores Atípicos del Parque Solar N2

# 5 | Modelos de Aprendizaje Supervisado

## 5.1. Adaptive Boosting Regressor

El método **Adaptive Boosting Regressor** (AdaBoostRegressor) es una técnica de aprendizaje supervisado diseñada para problemas de regresión. Es una extensión del algoritmo AdaBoost, que fue inicialmente desarrollado para tareas de clasificación. El objetivo de AdaBoost es combinar varios modelos débiles para formar un modelo más robusto, reduciendo el error general del sistema.

### 5.1.1. Fundamentos Matemáticos

Dado un conjunto de datos de entrenamiento $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, se asignan pesos iniciales iguales a cada observación:

$$w_i^{(1)} = \frac{1}{N}, \quad \text{para} \quad i = 1, 2, \ldots, N$$

En cada iteración $m$, se entrena un modelo débil $h_m(x)$ utilizando los pesos actuales. El error ponderado del modelo se calcula como:

$$\epsilon_m = \frac{\sum_{i=1}^{N} w_i^{(m)} \left| y_i - h_m(x_i) \right|}{\sum_{i=1}^{N} w_i^{(m)}}$$

El peso de cada modelo débil se calcula con:

$$\alpha_m = \frac{1}{2} \ln \left( \frac{1 - \epsilon_m}{\epsilon_m} \right)$$

Los pesos de las observaciones se actualizan para dar mayor importancia a los errores:

$$w_i^{(m+1)} = w_i^{(m)} \exp \left( \alpha_m \left| y_i - h_m(x_i) \right| \right)$$

y se normalizan para que la suma sea 1:

$$w_i^{(m+1)} = \frac{w_i^{(m+1)}}{\sum_{j=1}^{N} w_j^{(m+1)}}$$

La predicción final del modelo es una combinación ponderada de todos los modelos débiles:

$$F(x) = \sum_{m=1}^{M} \alpha_m h_m(x)$$

## 5.2. Bootstrap Aggregating Regressor

El método **Bootstrap Aggregating Regressor** (BaggingRegressor) es una técnica de ensamblado que busca reducir la varianza de los modelos individuales combinando múltiples predictores entrenados en diferentes subconjuntos del mismo conjunto de datos.

### 5.2.1. Fundamentos Matemáticos

Dado un conjunto de datos de entrenamiento $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, el método de bagging crea $M$ subconjuntos $\mathcal{D}_m$ mediante muestreo aleatorio con reemplazo, donde cada subconjunto puede contener observaciones repetidas. Esto se expresa como:

$$\mathcal{D}_m = \{(x_{i_1}, y_{i_1}), (x_{i_2}, y_{i_2}), \ldots, (x_{i_{N'}}, y_{i_{N'}})\}, \quad i_j \sim \text{Uniform}(1, N)$$

Cada subconjunto se utiliza para entrenar un modelo base $h_m(x)$, y las predicciones finales se calculan como el promedio de las salidas de todos los modelos:

$$F(x) = \frac{1}{M} \sum_{m=1}^{M} h_m(x)$$

Esto reduce la varianza del estimador final sin aumentar significativamente el sesgo, mejorando la estabilidad del modelo. Es particularmente efectivo cuando los modelos base tienen alta varianza, como los árboles de decisión.

## 5.3. Categorical Boosting Regressor

El método **Categorical Boosting Regressor** (CatBoostRegressor) es un algoritmo de aprendizaje supervisado basado en árboles de decisión con boosting de gradiente. Está diseñado para manejar eficientemente variables categóricas y reducir el sobreajuste en conjuntos de datos pequeños y medianos, manteniendo al mismo tiempo una alta precisión.

### 5.3.1. Fundamentos Matemáticos

Dado un conjunto de datos de entrenamiento $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, el objetivo de CatBoost es minimizar el error cuadrático medio (MSE) definido como:

$$L = \frac{1}{N} \sum_{i=1}^{N} (y_i - F(x_i))^2$$

El modelo se construye de forma iterativa, añadiendo árboles que corrigen los errores residuales del modelo anterior:

$$F_t(x) = F_{t-1}(x) + \eta \cdot h_t(x)$$

donde:

- $F_{t-1}(x)$ es el modelo actual,
- $\eta$ es la tasa de aprendizaje,
- $h_t(x)$ es el nuevo árbol ajustado para minimizar el error residual.

CatBoost es especialmente eficiente en el manejo de variables categóricas, utilizando técnicas como target statistics y ordered boosting para reducir el sobreajuste y preservar la estructura temporal de los datos.

## 5.4. ElasticNet

El método **ElasticNet** es una técnica de regresión lineal regularizada que combina las penalizaciones de las regresiones **Ridge** (regresión de crestas) y **Lasso** (Least Absolute Shrinkage and Selection Operator). Es especialmente útil cuando se espera que solo un subconjunto de las características sea relevante para el modelo, pero estas pueden estar correlacionadas.

### 5.4.1. Fundamentos Matemáticos

Dado un conjunto de datos de entrenamiento $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, el objetivo de ElasticNet es minimizar la siguiente función de costo:

$$L(\beta) = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=1}^{p} \beta_j^2$$

donde:

- $\hat{y}_i = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij}$ es la predicción del modelo,

- $\lambda_1$ controla la penalización L1 (Lasso),
- $\lambda_2$ controla la penalización L2 (Ridge).

Esto se puede reescribir en términos de los hiperparámetros $\alpha$ y $\rho$:

$$L(\beta) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 + \alpha\left(\rho\sum_{j=1}^{p}|\beta_j| + (1-\rho)\sum_{j=1}^{p}\beta_j^2\right)$$

donde:

- $\alpha = \lambda_1 + \lambda_2$ es el parámetro de regularización total,
- $\rho$ controla el balance entre L1 y L2.

ElasticNet es particularmente efectivo cuando se espera que las variables predictores estén correlacionadas, ya que combina las propiedades de selección de características de Lasso y la estabilidad de Ridge.

## 5.5. Extremely Randomized Trees Regressor

El método **Extremely Randomized Trees Regressor** (ExtraTreesRegressor) es una técnica de ensamblado basada en árboles de decisión que busca reducir la varianza del modelo y mejorar su precisión combinando múltiples árboles entrenados de forma independiente. Es similar a **RandomForestRegressor**, pero introduce mayor aleatoriedad en la construcción de los árboles, lo que mejora su capacidad de generalización.

### 5.5.1. Fundamentos Matemáticos

Dado un conjunto de datos de entrenamiento $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, ExtraTreesRegressor crea $M$ árboles de decisión $T_m$, donde cada árbol se entrena seleccionando aleatoriamente tanto las características como los puntos de corte para las divisiones internas. A diferencia de RandomForestRegressor, no utiliza muestreo con reemplazo, es decir, cada árbol se entrena con todo el conjunto de datos.

La predicción final se calcula como el promedio de las salidas de todos los árboles:

$$F(x) = \frac{1}{M}\sum_{m=1}^{M} T_m(x)$$

Este enfoque introduce mayor aleatoriedad a nivel de nodo, reduciendo la correlación entre los árboles y disminuyendo así la varianza del modelo final, lo que mejora la precisión y estabilidad del estimador.

## 5.6. LightGBM Regressor

El método **LightGBM Regressor** (LGBMRegressor) es un algoritmo de boosting de gradiente desarrollado para ser altamente eficiente tanto en velocidad como en uso de memoria. Utiliza una técnica de crecimiento de árboles basada en hojas (leaf-wise) en lugar de profundidad (level-wise), lo que le permite reducir el error de manera más rápida y precisa en comparación con otros métodos de boosting.

### 5.6.1. Fundamentos Matemáticos

Dado un conjunto de datos de entrenamiento $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, el objetivo de LGBMRegressor es minimizar una función de pérdida, típicamente el error cuadrático medio (MSE):

$$L = \frac{1}{N} \sum_{i=1}^{N} (y_i - F(x_i))^2$$

El modelo se construye de forma iterativa, añadiendo árboles que corrigen los errores residuales del modelo anterior:

$$F_t(x) = F_{t-1}(x) + \eta \cdot h_t(x)$$

donde:

- $F_{t-1}(x)$ es el modelo actual,
- $\eta$ es la tasa de aprendizaje,
- $h_t(x)$ es el nuevo árbol ajustado para minimizar el error residual.

Una característica distintiva de **LightGBM** es su método de crecimiento de árboles basado en hojas, que selecciona las hojas con mayor reducción de pérdida para expandir, en lugar de crecer de manera uniforme como en otros métodos:

$$\Delta L = \text{Gain} = \frac{G^2}{H + \lambda}$$

donde:

- $G$ es la suma de los gradientes de las muestras en el nodo,
- $H$ es la suma de los valores de segundo orden (Hessiano) de las muestras,
- $\lambda$ es un parámetro de regularización para evitar sobreajuste.

Este enfoque reduce significativamente el error al enfocarse en las particiones más prometedoras, mejorando la eficiencia del modelo.

## 5.7. Least Absolute Shrinkage and Selection Operator

El método **Least Absolute Shrinkage and Selection Operator** (Lasso) es una técnica de regresión lineal que introduce una penalización L1 para mejorar la precisión del modelo y reducir el sobreajuste. Es particularmente efectivo para seleccionar características relevantes en conjuntos de datos de alta dimensionalidad.

### 5.7.1. Fundamentos Matemáticos

Dado un conjunto de datos de entrenamiento $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, el objetivo de Lasso es minimizar la siguiente función de costo:

$$L(\beta) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

donde:

- $\hat{y}_i = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij}$ es la predicción del modelo,
- $\lambda$ es el parámetro de regularización que controla el grado de penalización L1.

La penalización L1 tiene dos efectos importantes:

- **Reducción de Coeficientes:** Fuerza a muchos coeficientes a ser exactamente cero, lo que implica una selección automática de características.
- **Simplicidad del Modelo:** Produce modelos más simples y fáciles de interpretar en comparación con otros métodos como Ridge.

## 5.8. Linear Regression

El método **LinearRegression** es uno de los enfoques más simples y ampliamente utilizados para problemas de regresión. Se basa en el principio de encontrar la línea que mejor se ajusta a los datos, minimizando la suma de los errores cuadráticos.

### 5.8.1. Fundamentos Matemáticos

Dado un conjunto de datos de entrenamiento $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, el objetivo de LinearRegression es encontrar los coeficientes $\beta$ que minimizan la siguiente función de costo:

$$L(\beta) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

donde:

- $\hat{y}_i = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij}$ es la predicción del modelo,
- $\beta_0$ es el término de sesgo (intercepto),
- $\beta_j$ son los coeficientes que representan la influencia de cada característica en la variable objetivo.

Los coeficientes $\beta$ se pueden calcular de forma cerrada utilizando la expresión:

$$\beta = (X^T X)^{-1} X^T y$$

donde:

- $X$ es la matriz de características,
- $y$ es el vector de valores objetivo.

## 5.9. Random Forest Regressor

El método **RandomForestRegressor** es un algoritmo de ensamblado basado en árboles de decisión que combina múltiples árboles para mejorar la precisión y reducir el riesgo de sobreajuste [5]. Es una extensión del enfoque de bagging, con mejoras en la selección aleatoria de características en cada nodo para reducir la correlación entre los árboles.

### 5.9.1. Fundamentos Matemáticos

Dado un conjunto de datos de entrenamiento $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, RandomForestRegressor crea $M$ árboles de decisión $T_m$, donde cada árbol se entrena utilizando un subconjunto aleatorio de los datos (con reemplazo) y un subconjunto aleatorio de características para cada nodo. Esto introduce mayor diversidad en los árboles y reduce la varianza del modelo final.

La predicción final se calcula como el promedio de las salidas de todos los árboles:

$$F(x) = \frac{1}{M} \sum_{m=1}^{M} T_m(x)$$

## 5.10. Ridge

El método **Ridge** (también conocido como regresión de crestas) es una técnica de regresión lineal que introduce una penalización L2 para mejorar la precisión del modelo y reducir el sobreajuste. Es especialmente útil cuando las características del modelo están correlacionadas, ya que estabiliza los coeficientes y reduce la varianza del modelo.

### 5.10.1. Fundamentos Matemáticos

Dado un conjunto de datos de entrenamiento $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, el objetivo de Ridge es minimizar la siguiente función de costo:

$$L(\beta) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 + \lambda\sum_{j=1}^{p}\beta_j^2$$

donde:

- $\hat{y}_i = \beta_0 + \sum_{j=1}^{p}\beta_j x_{ij}$ es la predicción del modelo,
- $\lambda$ es el parámetro de regularización que controla el grado de penalización L2.

La penalización L2 tiene dos efectos importantes:

- **Reducción de Varianza:** Evita que los coeficientes se vuelvan extremadamente grandes, reduciendo el riesgo de sobreajuste.
- **Estabilidad del Modelo:** Mejora la estabilidad numérica en modelos con características correlacionadas.

Los coeficientes $\beta$ en Ridge se calculan de forma cerrada como:

$$\beta = (X^T X + \lambda I)^{-1} X^T y$$

donde:

- $X$ es la matriz de características,
- $y$ es el vector de valores objetivo,
- $I$ es la matriz identidad.

## 5.11. Extreme Gradient Boosting Regressor

El método **Extreme Gradient Boosting Regressor** (XGBRegressor) es un algoritmo de boosting de gradiente altamente eficiente y flexible. Fue desarrollado como parte del paquete XGBoost (eXtreme Gradient Boosting) y es conocido por su velocidad, precisión y capacidad para manejar datos escasos y de alta dimensionalidad.

### 5.11.1. Fundamentos Matemáticos

Dado un conjunto de datos de entrenamiento $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, el objetivo de XGBRegressor es minimizar una función de pérdida, típicamente el error cuadrático medio (MSE), a través de un proceso iterativo que ajusta los errores residuales:

$$L = \frac{1}{N} \sum_{i=1}^{N} (y_i - F(x_i))^2$$

El modelo se construye añadiendo árboles secuencialmente, donde cada árbol $h_t(x)$ corrige los errores residuales del modelo anterior:

$$F_t(x) = F_{t-1}(x) + \eta \cdot h_t(x)$$

donde:

- $F_{t-1}(x)$ es el modelo actual,
- $\eta$ es la tasa de aprendizaje,
- $h_t(x)$ es el nuevo árbol ajustado para minimizar el error residual.

XGBRegressor introduce dos términos adicionales para mejorar el ajuste del modelo y evitar el sobreajuste:

$$L = \sum_{i=1}^{N} (y_i - F(x_i))^2 + \sum_{t=1}^{M} \left( \gamma T_t + \frac{\lambda}{2} \sum_{j=1}^{p} w_j^2 \right)$$

donde:

- $\gamma$ controla la complejidad del árbol penalizando el número de nodos,
- $\lambda$ penaliza la magnitud de los pesos para regularizar el modelo.

Esta estructura permite que XGBRegressor sea altamente flexible y efectivo en una amplia gama de problemas de regresión.

## 5.12. Feed-Forward Neural Network

Una **Feed-Forward Neural Network** es el tipo más básico de red neuronal artificial, donde la información fluye en una sola dirección desde las entradas hasta las salidas, pasando a través de una o más capas ocultas. No tiene conexiones cíclicas, lo que simplifica su entrenamiento y análisis.

### 5.12.1. Fundamentos Matemáticos

Dado un conjunto de datos de entrenamiento $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, el objetivo de una red neuronal es aprender una función que mapea los datos de entrada a los valores objetivo mediante una serie de transformaciones lineales y no lineales. Esto se expresa como:

$$\hat{y} = f(x) = \sigma(W^{(L)}\sigma(W^{(L-1)} \cdots \sigma(W^{(1)}x + b^{(1)}) \cdots + b^{(L-1)}) + b^{(L)})$$

donde:

- $W^{(l)}$ son las matrices de pesos para cada capa $l$,
- $b^{(l)}$ son los vectores de sesgo,
- $\sigma$ es la función de activación (ReLU, Sigmoid, Tanh, etc.),
- $L$ es el número de capas.

El objetivo es minimizar una función de costo, típicamente el error cuadrático medio (MSE) para tareas de regresión:

$$L = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y_i})^2$$

El entrenamiento de la red se realiza mediante retropropagación, que calcula los gradientes de esta función de costo con respecto a los pesos usando el algoritmo de descenso de gradiente:

$$W^{(l)} \leftarrow W^{(l)} - \eta\frac{\partial L}{\partial W^{(l)}}$$

$$b^{(l)} \leftarrow b^{(l)} - \eta\frac{\partial L}{\partial b^{(l)}}$$

donde $\eta$ es la tasa de aprendizaje.

# 6 | Evaluación y Optimización de Modelos Predictivos

Una vez desarrollados los modelos de aprendizaje automático y profundo, resulta esencial evaluar su rendimiento y mejorar su capacidad predictiva mediante un ajuste adecuado de sus configuraciones. Este capítulo se centra en dos aspectos clave del proceso: la evaluación mediante métricas objetivas y la optimización de hiperparámetros.

## 6.1. Métricas de Evaluación

Para determinar el rendimiento y la eficacia de los modelos predictivos desarrollados, se han seleccionado diversas métricas de evaluación que permiten cuantificar el error y la capacidad de generalización del modelo. Las métricas empleadas son:

### 6.1.1. Coeficiente de Determinación

El **Coeficiente de Determinación ($\mathbf{R^2}$)** es una métrica utilizada para evaluar la precisión de los modelos de regresión [1]. Representa la proporción de la varianza total de la variable dependiente que es explicada por las variables independientes del modelo.

#### Fundamentos Matemáticos

Dado un conjunto de datos de prueba $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, el coeficiente de determinación se define como:

$$R^2 = 1 - \frac{\text{SSE}}{\text{SST}}$$

donde:

- $\text{SSE} = \sum_{i=1}^{N}(y_i - \hat{y}_i)^2$ es la Suma de los Errores Cuadrados (Sum of Squared Errors),
- $\text{SST} = \sum_{i=1}^{N}(y_i - \bar{y})^2$ es la Suma Total de los Cuadrados (Total Sum of Squares),
- $\bar{y}$ es el promedio de los valores reales.

Una forma equivalente de calcular $R^2$ es mediante el cociente de la varianza explicada y la varianza total:

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2}$$

El coeficiente $R^2$ toma valores entre 0 y 1, donde:

- $R^2 = 1$ indica que el modelo explica perfectamente la variabilidad de los datos,
- $R^2 = 0$ indica que el modelo no explica ninguna variabilidad,
- Valores negativos pueden aparecer si el modelo es peor que un simple promedio.

### 6.1.2. Error Absoluto Medio

El **Error Absoluto Medio (MAE)** es una métrica utilizada para evaluar la precisión de los modelos de regresión. Mide el promedio de las diferencias absolutas entre los valores predichos y los valores reales, proporcionando una medida directa del error medio en las predicciones [12].

#### Fundamentos Matemáticos

Dado un conjunto de datos de prueba $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, el MAE se define como:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$$

donde:

- $y_i$ es el valor real,
- $\hat{y}_i$ es el valor predicho,
- $N$ es el número total de observaciones.

### 6.1.3. Error Cuadrático Medio

El **Error Cuadrático Medio (MSE)** es una métrica utilizada para evaluar la precisión de los modelos de regresión. Mide el promedio de los errores al cuadrado entre los valores predichos y los valores reales, penalizando fuertemente los errores grandes debido al efecto cuadrático.

#### Fundamentos Matemáticos

Dado un conjunto de datos de prueba $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, el MSE se define como:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

donde:

- $y_i$ es el valor real,
- $\hat{y}_i$ es el valor predicho,
- $N$ es el número total de observaciones.

### 6.1.4. Cross Validation

El **Cross Validation** (Validación Cruzada) es una técnica fundamental para evaluar la capacidad predictiva de un modelo de machine learning. Su objetivo es estimar el rendimiento del modelo en datos no vistos, evitando el sobreajuste y proporcionando una evaluación más robusta [4].

#### Fundamentos Matemáticos

Dado un conjunto de datos $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, el proceso de validación cruzada consiste en dividir los datos en $K$ subconjuntos (folds) aproximadamente iguales. El procedimiento que se sigue es el siguiente:

Dividir el conjunto de datos en $K$ subconjuntos disjuntos:

$$\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2 \cup \cdots \cup \mathcal{D}_K$$

Para cada fold $k$, entrenar el modelo $M_k$ usando todos los datos excepto los del fold $\mathcal{D}_k$ y validar usando $\mathcal{D}_k$:

$$\text{Error}(M_k) = \text{Error}(M_k, \mathcal{D}_k)$$

Calcular el error promedio del modelo como:

$$\text{Error Promedio} = \frac{1}{K}\sum_{k=1}^{K}\text{Error}(M_k, \mathcal{D}_k)$$

Este enfoque reduce el sesgo y la varianza de la estimación del error, proporcionando una evaluación más precisa del rendimiento del modelo.

## 6.2. Optimización de Hiperparámetros

En la fase de desarrollo de modelos, es crucial ajustar adecuadamente los hiperparámetros para mejorar el rendimiento y la generalización de los algoritmos de aprendizaje automático. Para este propósito, se han empleado dos técnicas avanzadas de optimización de hiperparámetros: **GridSearchCV** y **RandomizedSearchCV**.

### 6.2.1. GridSearchCV

El método **GridSearchCV** se utilizó para realizar una búsqueda exhaustiva a través de una especificada grilla de hiperparámetros para los modelos rápidos y moderados. Este enfoque sistemático prueba todas las combinaciones posibles de los parámetros proporcionados y valida cada combinación utilizando la técnica de validación cruzada para garantizar que los resultados sean robustos y replicables. Los modelos involucrados en este proceso incluyen regresiones lineales como **LinearRegression**, **Ridge**, **Lasso**, y **ElasticNet**, así como modelos basados en ensambles como **BaggingRegressor**, **AdaBoostRegressor**, **RandomForestRegressor** y **ExtraTreesRegressor**.

### 6.2.2. RandomizedSearchCV

Para los modelos considerados computacionalmente más intensos, como **CatBoostRegressor**, **LGBMRegressor**, y **XGBRegressor**, se utilizó **RandomizedSearchCV**. A diferencia del **GridSearchCV**, **RandomizedSearchCV** selecciona al azar combinaciones de parámetros para explorar el espacio de búsqueda de manera más eficiente. Este método es particularmente útil cuando el espacio de parámetros es grande y una búsqueda exhaustiva podría ser computacionalmente prohibitiva. La configuración del **RandomizedSearchCV** incluyó un número limitado de iteraciones por modelo, utilizando validación cruzada para asegurar la evaluación rigurosa de cada conjunto de parámetros.

### 6.2.3. Ejecución y Evaluación

Ambas técnicas se implementaron dentro de funciones definidas en Python, que automatizan la búsqueda y selección de los mejores parámetros para cada modelo. Las evaluaciones se basaron en el coeficiente de determinación ($R^2$) para cuantificar la capacidad de cada modelo de replicar los resultados observados y predecir resultados futuros basados en nuevas entradas de datos.

La aplicación de estas técnicas de optimización de hiperparámetros conduce a la identificación de los ajustes óptimos para cada modelo, contribuyendo significativamente a la mejora de su rendimiento y precisión en tareas predictivas. La configuración óptima de cada modelo se registró y utilizó para las evaluaciones subsiguientes en el conjunto de datos. Este enfoque metodológico garantiza que los modelos no solo están bien ajustados a los datos de entrenamiento, sino que también poseen la capacidad de generalizar efectivamente sobre datos no vistos, lo cual es esencial para aplicaciones prácticas de modelos predictivos en entornos reales.

### 6.2.4. Resultados de la Optimización de Hiperparámetros

Tras la implementación de las técnicas de búsqueda de hiperparámetros, se obtuvieron configuraciones óptimas que maximizan el rendimiento de cada modelo.

**Parques eólicos**

- **Linear Regression**

  {'fit_intercept': True}.

- **Ridge Regression**

  {'alpha': 10, 'solver': 'auto'}.

- **Lasso Regression**

  {'alpha':1, 'max_iter': 500, 'tol': 0.001}.

- **ElasticNet**

  {'alpha': 0.1, 'l1_ratio': 0.2, 'max_iter': 1000}.

- **Bagging Regressor**

  {'max_features': 1.0, 'max_samples': 1.0, 'n_estimators': 200}.

- **AdaBoost Regressor**

  {'learning_rate': 0.01, 'loss': 'exponential', 'n_estimators': 100}.

- **Random Forest Regressor**

  {'criterion': 'squared_error', 'max_depth': None, 'min_samples_leaf': 1,

  'min_samples_split': 2, 'n_estimators': 200}.

- **Extra Trees Regressor**

  {'bootstrap': False, 'criterion': 'friedman_mse', 'max_depth': None, 'max_features': 'log2', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 500}.

- **CatBoost Regressor**

  {'random_strength': 5, 'learning_rate': 0.3, 'iterations': 500, 'depth': 10}.

- **LGBM Regressor**

  {'num_leaves': 50, 'n_estimators': 500, 'max_depth': -1, 'learning_rate': 0.3,

  'boosting_type': 'dart'}.

- **XGB Regressor**

  {'subsample': 1, 'n_estimators': 500, 'max_depth': 5, 'learning_rate': 0.3,

  'colsample_bytree': 0.8}.

**Parques solares N1 y N2**

- **Linear Regression**
  {'fit_intercept': True}.
- **Ridge Regression**
  {'alpha': 0.1, 'solver': 'svd'}.
- **Lasso Regression**
  {'alpha': 0.1, 'max_iter': 500, 'tol': 0.001}.
- **ElasticNet**
  {'alpha': 0.1, 'l1_ratio': 0.8, 'max_iter': 1000}.
- **Bagging Regressor**
  {'max_features': 1.0, 'max_samples': 0.8, 'n_estimators': 200}.
- **AdaBoost Regressor**
  {'learning_rate': 0.01, 'loss': 'linear', 'n_estimators': 100}.
- **Random Forest Regressor**
  {'criterion': 'friedman_mse', 'max_depth': 20, 'min_samples_leaf': 1,
  'min_samples_split': 5, 'n_estimators': 50}.
- **Extra Trees Regressor**
  {'bootstrap': False, 'criterion': 'friedman_mse', 'max_depth': 20, 'max_features': 'log2', 'min_samples_leaf': 1, 'min_samples_split': 5, 'n_estimators': 300}.
- **CatBoost Regressor**
  {'random_strength': 5, 'learning_rate': 0.3, 'iterations': 500, 'depth': 10}.
- **LGBM Regressor**
  {'num_leaves': 50, 'n_estimators': 500, 'max_depth': -1, 'learning_rate': 0.3,
  'boosting_type': 'dart'}.
- **XGB Regressor**
  {'subsample': 1, 'n_estimators': 1000, 'max_depth': 5, 'learning_rate': 0.1,
  'colsample_bytree': 1}.

Estos parámetros fueron seleccionados tras una larga prueba que demostraron ser las configuraciones más eficaces para maximizar la precisión y la capacidad predictiva de los modelos. La implementación de estos parámetros optimizados permite que cada modelo alcance su máximo potencial, reflejando en una mejora significativa de su rendimiento en las tareas predictivas.

A continuación, los modelos fueron reconstruidos con sus hiperparámetros óptimos y evaluados nuevamente, confirmando así su gran desempeño gracias al proceso de optimización.

# 7 | Modelos de Aprendizaje Automático

Este capítulo presenta la implementación del código desarrollado como parte del presente trabajo, detallando su estructura, funciones principales y contribución al objetivo general del estudio. A lo largo del texto, se explicarán las decisiones metodológicas adoptadas y se proporcionará un análisis del desempeño obtenido.

Los modelos se pueden encontrar en este repositorio de GitHub.

(https://github.com/josemauma/TFG-ML-DL-WIND-SOLAR-FARMS)

- **Modelo W_ML:** Este modelo ha sido desarrollado para para identificar patrones y optimizar la predicción de la generación de energía del parque eólico. Consulta Modelo W_ML para mas información.

  (http://github.com/josemauma/TFG-ML-DL-WIND-SOLAR-FARMS/blob/main/WINDMILLS/TFG_W_ML.ipynb)

- **Modelo SP1_ML:** Este modelo ha sido desarrollado para para identificar patrones y optimizar la predicción de la generación de energía del parque solar N1. Consulta SP1_ML para mas información.

  (https://github.com/josemauma/TFG-ML-DL-WIND-SOLAR-FARMS/blob/main/SOLAR%20PANELS/TFG_SP1_ML.ipynb)

- **Modelo SP2_ML:** Este modelo ha sido desarrollado para para identificar patrones y optimizar la predicción de la generación de energía del parque solar N2. Consulta SP2_ML para más información.

  https://github.com/josemauma/TFG-ML-DL-WIND-SOLAR-FARMS/blob/main/SOLAR%20PANELS/TFG_SP2_ML.ipynb

## 7.1. Parque Eólico

En esta sección se han comparado diversos modelos de regresión con el objetivo de predecir la potencia generada en un parque eólico. La evaluación de los modelos se realizó considerando su capacidad de generalización, medida a través del coeficiente de determinación $R^2$, así como los errores cuadráticos medios (Squared Error) y absolutos medios (MAE), indicadores clave para cuantificar la precisión de las predicciones.

### 7.1.1. Resultados y Comparación de Modelos

Entre los modelos analizados, figura 7.1, el **CatBoostRegressor** se destacó como el más preciso, obteniendo un $R^2$ de 0.87 en el conjunto de prueba, lo que indica una alta capacidad predictiva y generalización. Su Mean Absolute Error (MAE) fue de 1427.13 y su Squared Error de 4,276,866.39, lo que confirma su excelente rendimiento.

Otros modelos destacados incluyen el LGBMRegressor con un $R^2$ de 0.86 y el ExtraTreesRegressor con un $R^2$ de 0.85. Sin embargo, el ExtraTreesRegressor presentó un claro sobreajuste, evidenciado por un rendimiento perfecto (100 %) en el conjunto de entrenamiento y una diferencia notable del 15.11 % en comparación con el conjunto de prueba. A pesar de ello, presentó errores competitivos, con un MAE de 1491.39 y un Squared Error de 4,817,132.24.

El **BaggingRegressor** y el **RandomForestRegressor** también mostraron un desempeño sólido, ambos con un $R^2$ de 0.81, aunque sus errores fueron ligeramente superiores respecto al **CatBoostRegressor**.

Por otro lado, modelos como **XGBRegressor** y **AdaBoostRegressor** tuvieron desempeños más modestos, con $R^2$ de 0.81 y 0.46 respectivamente, destacándose negativamente el **AdaBoostRegressor** con un MAE considerablemente alto de 3270.22.

Finalmente, los modelos basados en regresión lineal (**LinearRegression, Lasso, Ridge, y ElasticNet**) mostraron un desempeño significativamente inferior, con un $R^2$ de tan solo 0.36, indicando una insuficiente capacidad predictiva y, por ende, no recomendables para la predicción precisa de generación de energía en este contexto específico.

| | Modelo | Train score | Test score | $R^2$ score | Ratio difference | Evaluate model | Squared error | Mean Absolute Error (MAE) |
|---|---|---|---|---|---|---|---|---|
| 8 | CatBoostRegressor | 95.88% | 86.58% | 0.87 | 9.30% | good | 4276866.39 | 1427.13 |
| 9 | LGBMRegressor | 92.41% | 86.23% | 0.86 | 6.18% | good | 4389876.43 | 1424.92 |
| 7 | ExtraTreesRegressor | 100.0% | 84.89% | 0.85 | 15.11% | unknown | 4817132.24 | 1491.39 |
| 4 | BaggingRegressor | 97.42% | 81.43% | 0.81 | 16.00% | good | 5919440.31 | 1611.25 |
| 6 | RandomForestRegressor | 97.41% | 81.37% | 0.81 | 16.05% | good | 5939174.69 | 1614.70 |
| 10 | XGBRegressor | 89.7% | 81.15% | 0.81 | 8.55% | good | 6009771.64 | 1723.87 |
| 5 | AdaBoostRegressor | 45.9% | 45.64% | 0.46 | 0.26% | bad | 17327776.94 | 3270.22 |
| 0 | LinearRegression | 35.89% | 36.26% | 0.36 | -0.37% | bad | 20316901.33 | 3501.94 |
| 1 | Ridge | 35.89% | 36.26% | 0.36 | -0.37% | bad | 20316908.18 | 3501.94 |
| 2 | Lasso | 35.89% | 36.26% | 0.36 | -0.37% | bad | 20317081.58 | 3501.97 |
| 3 | ElasticNet | 35.89% | 36.25% | 0.36 | -0.37% | bad | 20319552.06 | 3502.03 |

**Figura 7.1.** Resultados Obtenidos de los Modelos Machine Learning (Parque Eólico)

### 7.1.2. Elección del Modelo Final

Con base en los resultados obtenidos, se seleccionó **CatBoostRegressor** como el modelo más adecuado para la predicción de potencia generada en el parque solar, debido a su alto $R^2$ de 0.87 y menor error absoluto medio (MAE) de 1427.13. Para validar su estabilidad, se realizó una validación cruzada, obteniendo resultados consistentes que demuestran su sólida capacidad de generalización.

Aunque el ExtraTreesRegressor mostró una alta precisión, su evidente sobreajuste limita su recomendación. Por lo tanto, se selecciona el CatBoostRegressor como el modelo más adecuado debido a su equilibrio entre precisión, generalización y bajo error. Para mejorar aún más su

rendimiento, podrían explorarse estrategias adicionales como optimización bayesiana de hiperparámetros o técnicas avanzadas de regularización.

## 7.2. Parque Solar N1

### 7.2.1. Resultados y Comparación de Modelos

Entre los modelos analizados, figura 7.2, el **BaggingRegressor** se destacó como el más preciso, obteniendo un $R^2$ de 0.99 en el conjunto de prueba y mostrando un desempeño muy sólido y estable, con una diferencia mínima del 0.70 % entre entrenamiento y prueba. Su Mean Absolute Error (MAE) fue de solo 144.13, indicando predicciones altamente cercanas a los valores reales.

Otros modelos con un rendimiento igualmente sólido fueron el **RandomForestRegressor** y el **ExtraTreesRegressor**, también con $R^2$ de 0.99 y errores reducidos. **CatBoostRegressor, LGBMRegressor** y **XGBRegressor** obtuvieron igualmente buenos resultados, con $R^2$ próximos a 0.99 y métricas competitivas, destacando el bajo error absoluto del CatBoostRegressor con 149.19.

Modelos como **LinearRegression, Ridge** y **Lasso** presentaron un desempeño ligeramente inferior con un $R^2$ de 0.98, mientras que AdaBoostRegressor tuvo un desempeño aceptable aunque con errores más altos. ElasticNet fue el modelo menos efectivo, mostrando un $R^2$ de 0.95 y errores considerablemente mayores.

| | Modelo | Train score | Test score | $R^2$ score | Ratio difference | Evaluate model | Squared error | Mean Absolute Error (MAE) |
|---|---|---|---|---|---|---|---|---|
| 4 | BaggingRegressor | 99.79% | 99.1% | 0.99 | 0.70% | good | 146256.42 | 144.13 |
| 6 | RandomForestRegressor | 99.72% | 99.13% | 0.99 | 0.58% | good | 140461.31 | 142.41 |
| 7 | ExtraTreesRegressor | 99.63% | 99.25% | 0.99 | 0.38% | good | 121805.67 | 134.08 |
| 8 | CatBoostRegressor | 99.52% | 99.22% | 0.99 | 0.30% | good | 126201.67 | 149.19 |
| 9 | LGBMRegressor | 99.41% | 99.21% | 0.99 | 0.20% | good | 127888.70 | 148.88 |
| 10 | XGBRegressor | 99.45% | 99.16% | 0.99 | 0.30% | good | 136691.98 | 150.63 |
| 0 | LinearRegression | 97.9% | 97.91% | 0.98 | -0.02% | good | 337988.71 | 265.55 |
| 1 | Ridge | 97.9% | 97.91% | 0.98 | -0.02% | good | 337988.63 | 265.55 |
| 2 | Lasso | 97.9% | 97.91% | 0.98 | -0.02% | good | 337990.66 | 265.57 |
| 5 | AdaBoostRegressor | 97.55% | 97.52% | 0.98 | 0.02% | good | 401025.05 | 317.61 |
| 3 | ElasticNet | 95.35% | 95.37% | 0.95 | -0.02% | good | 749438.75 | 505.70 |

**Figura 7.2.** Resultados Obtenidos de los Modelos ML (Parque Solar N1)

### 7.2.2. Elección del Modelo Final

En vista de los resultados obtenidos, se optó por seleccionar el **BaggingRegressor** como modelo final. Esta elección se basa en su excelente rendimiento, mostrando un $R^2$ cercano a 0.99 en el conjunto de prueba y un desempeño muy estable en la validación cruzada, reflejado en una mínima diferencia del 0.70 % entre los resultados del conjunto de entrenamiento y de prueba. Además, su Mean Absolute Error (MAE) fue de solo 144.13, lo que indica predicciones altamente precisas respecto a los valores reales.

Otros modelos, como el RandomForestRegressor y el ExtraTreesRegressor, también mostra-

ron resultados sobresalientes, con métricas muy cercanas al modelo seleccionado, consolidándose como alternativas sólidas. No obstante, la combinación de estabilidad y mínima diferencia de error posiciona al BaggingRegressor como la opción más adecuada para la predicción en este parque solar.

En resumen, la precisión, estabilidad y robustez del BaggingRegressor justifican plenamente su elección frente a otros modelos analizados, convirtiéndolo en la herramienta idónea para abordar la predicción de potencia generada en este contexto específico.

## 7.3. Parque Solar N2

### 7.3.1. Resultados y Comparación de Modelos

En la figura 7.3 se presentan los resultados de los distintos modelos evaluados para el parque solar N2. Los algoritmos **BaggingRegressor**, **RandomForestRegressor**, **ExtraTreesRegressor**, **CatBoostRegressor**, **LGBMRegressor** y **XGBRegressor** alcanzaron un $R^2$ cercano a 0,93, situándose como los de mejor desempeño. No obstante, sus métricas de error y el equilibrio entre los conjuntos de entrenamiento y prueba muestran matices relevantes.

El **BaggingRegressor** destaca por lograr el Squared Error más bajo (alrededor de 9284,41) y un Mean Absolute Error (MAE) de solo 26,91, lo que revela predicciones especialmente ajustadas a los valores reales. Modelos como **RandomForestRegressor** y **ExtraTreesRegressor** ofrecen también una alta precisión, con MAE ligeramente superiores (27,40 y 30,13, respectivamente). De forma similar, **CatBoostRegressor**, **LGBMRegressor** y **XGBRegressor** mantienen un rendimiento competitivo, con pequeñas variaciones en el error absoluto y la brecha de ajuste.

En contraste, **AdaBoostRegressor** presenta un $R^2$ menor (en torno a 0,77), mientras que las aproximaciones lineales (*LinearRegression*, *Ridge*, *Lasso* y *ElasticNet*) se sitúan aún más abajo, con $R^2$ entre 0,60 y 0,62 y errores considerablemente mayores. Estos resultados evidencian que, para el parque solar N2, los métodos basados en conjuntos (Bagging, Random Forest o Extra Trees) ofrecen estimaciones sustancialmente más fiables que las aproximaciones lineales.

| | Modelo | Train score | Test score | R² score | Ratio difference | Evaluate model | Squared error | Mean Absolute Error (MAE) |
|---|---|---|---|---|---|---|---|---|
| 4 | BaggingRegressor | 98.56% | 93.2% | 0.93 | 5.36% | good | 9284.41 | 26.91 |
| 6 | RandomForestRegressor | 97.94% | 93.1% | 0.93 | 4.84% | good | 9424.39 | 27.40 |
| 7 | ExtraTreesRegressor | 96.86% | 93.05% | 0.93 | 3.82% | good | 9491.07 | 30.13 |
| 8 | CatBoostRegressor | 97.83% | 92.82% | 0.93 | 5.01% | good | 9806.09 | 30.82 |
| 9 | LGBMRegressor | 96.97% | 92.98% | 0.93 | 3.99% | good | 9587.35 | 30.47 |
| 10 | XGBRegressor | 96.91% | 92.8% | 0.93 | 4.12% | good | 9833.02 | 31.44 |
| 5 | AdaBoostRegressor | 76.85% | 77.1% | 0.77 | -0.26% | middle | 31251.37 | 86.33 |
| 0 | LinearRegression | 61.86% | 62.13% | 0.62 | -0.28% | bad | 51684.58 | 132.94 |
| 1 | Ridge | 61.86% | 62.13% | 0.62 | -0.28% | bad | 51684.52 | 132.94 |
| 2 | Lasso | 61.85% | 62.13% | 0.62 | -0.28% | bad | 51684.19 | 132.97 |
| 3 | ElasticNet | 59.37% | 59.71% | 0.6 | -0.35% | bad | 54990.05 | 139.88 |

**Figura 7.3.** Resultados Obtenidos de los Modelos ML (Parque Solar N2)

### 7.3.2. Elección del Modelo Final

A la luz de las métricas obtenidas, se escogió el **BaggingRegressor** como modelo definitivo para el parque solar N2. Su elección se sustenta en el MAE mínimo de 26,91 y en un Squared Error de aproximadamente 9284,41, que confirman un ajuste muy preciso a los valores reales. Además, su equilibrada diferencia porcentual entre entrenamiento y prueba ($\sim$ 5,36 %) indica que el modelo no incurre en sobreajuste notable, reforzando su confiabilidad.

Otros algoritmos, como **RandomForestRegressor** y **ExtraTreesRegressor**, mostraron métricas cercanas y constituyen alternativas sólidas. Sin embargo, la combinación de baja desviación, estabilidad y mínima brecha entre conjuntos respalda firmemente la selección del BaggingRegressor como la opción más adecuada para las predicciones de potencia generada en el parque solar N2.

En síntesis, la precisión, estabilidad y robustez evidenciadas por el BaggingRegressor justifican plenamente su elección, posicionándolo como la herramienta idónea para la estimación de la producción en futuros escenarios operativos.

## 7.4. Gráficas de los Resultados Obtenidos

Para evaluar el desempeño de los modelos analizados, se presentan diversas gráficas que permiten comparar su precisión, capacidad de generalización y magnitud del error. Estas visualizaciones ofrecen una interpretación clara de cómo cada modelo se ajusta a los datos y qué tan bien logra predecir la variable objetivo.

### 7.4.1. Gráficas de Puntajes de Entrenamiento y Prueba

La figura 7.4, 7.5 y la 7.6 ilustra, de forma comparativa, el rendimiento de varios modelos tanto en el conjunto de entrenamiento como en el de prueba. Se aprecia que los mejores algoritmos muestran puntajes muy elevados y mantienen una brecha pequeña entre ambas fases, lo cual indica que su capacidad de generalización es sólida. Por otro lado, algunos presentan un ligero descenso en el conjunto de prueba frente al de entrenamiento, lo que sugiere un posible sobreajuste. En conjunto, esta visualización permite identificar rápidamente cuáles modelos ofrecen un equilibrio adecuado entre aprendizaje y capacidad predictiva con datos nuevos, sin necesidad de entrar en detalles numéricos específicos.

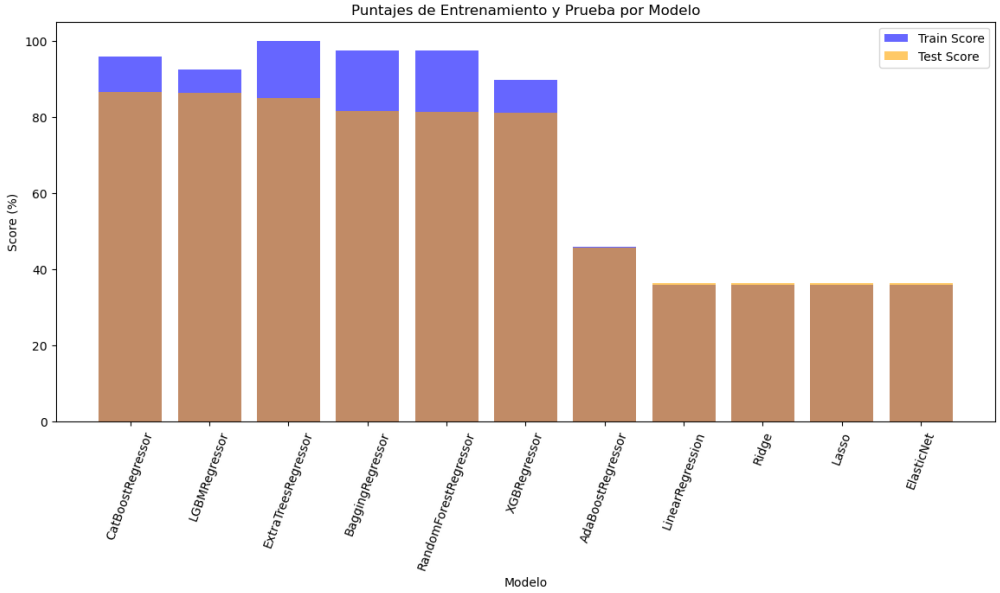


**Figura 7.4.** Resultados Obtenidos de Train Score y Test score (Parque Eólico)

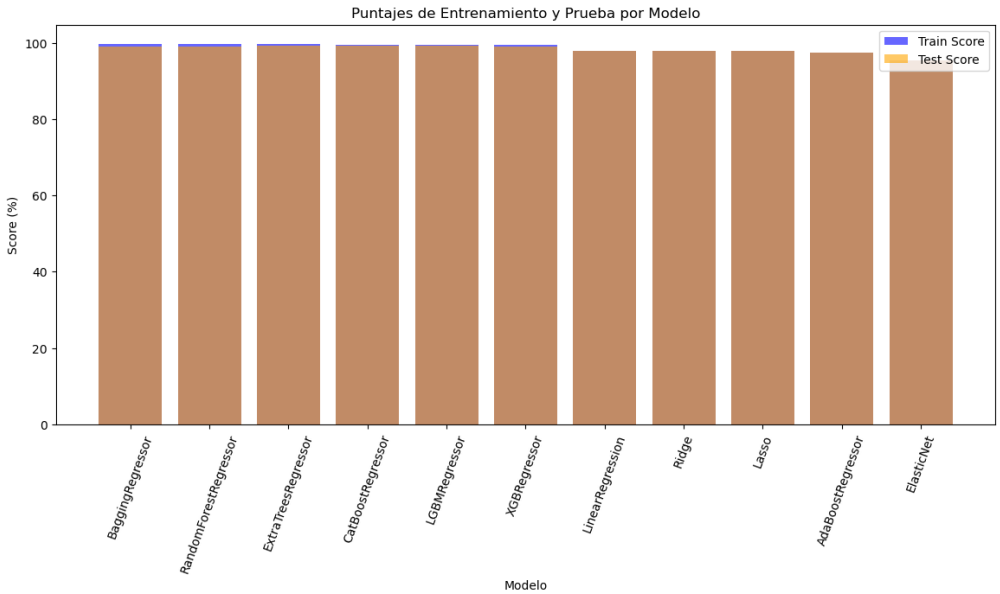


**Figura 7.5.** Resultados Obtenidos de Train Score y Test score (Parque Solar N1)

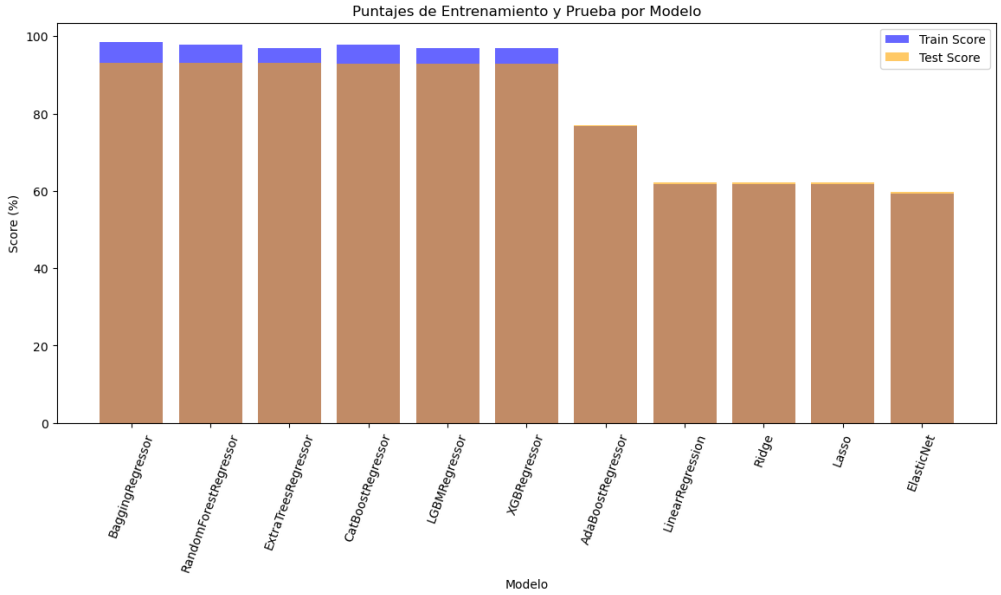


**Figura 7.6.** Resultados Obtenidos de Train Score y Test score (Parque Solar N2)

### 7.4.2. Gráficas del Coeficiente de Determinación ($R^2$ Score)

La gráficas de la figuras 7.7, 7.8 y 7.9, muestran el coeficiente de determinación ($R^2$) para cada modelo, que indica la proporción de la variabilidad de los datos explicada por sus predicciones. Los primeros modelos exhiben valores de $R^2$ elevados, lo cual sugiere un ajuste notablemente preciso y una fuerte capacidad para capturar la dinámica de los datos. Por el contrario, los modelos que se sitúan hacia la derecha presentan cifras más modestas, reflejando un desempeño menos ajustado. En conjunto, esta métrica facilita la comparación rápida de la eficacia de cada algoritmo al reproducir el comportamiento real.

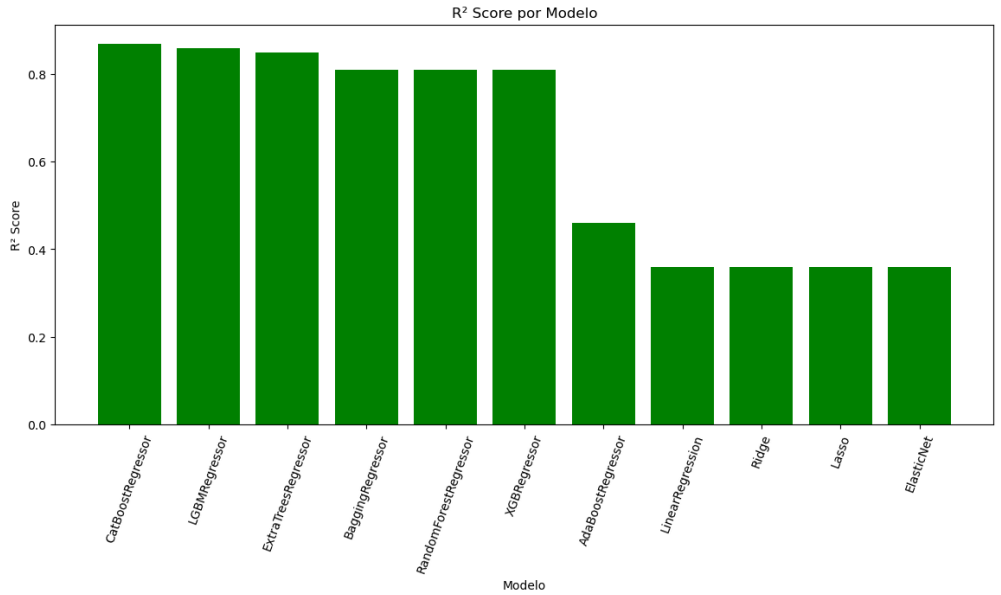


**Figura 7.7.** Resultados Obtenidos de $R^2$ (Parque Eólico)

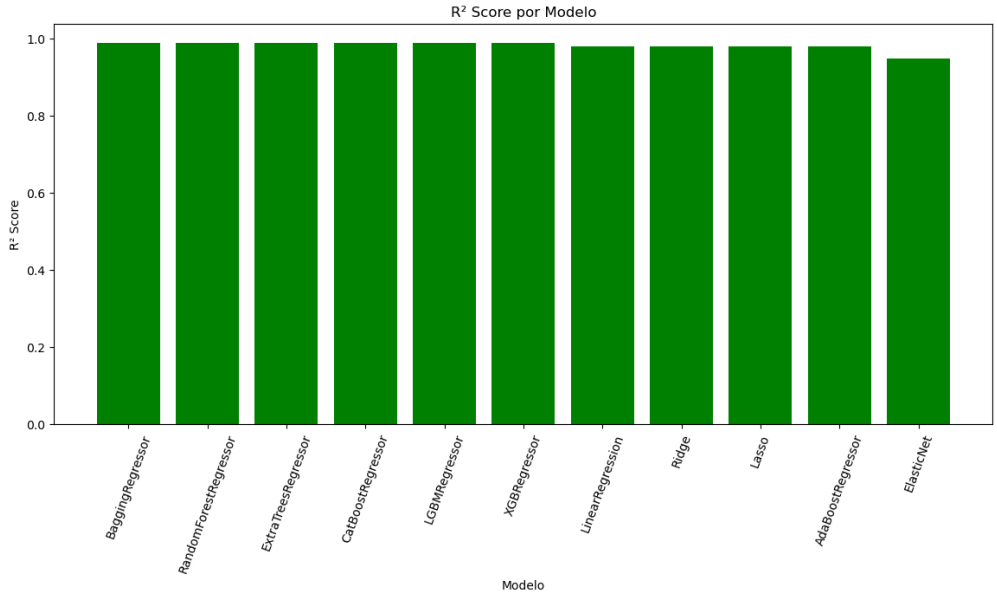


**Figura 7.8.** Resultados Obtenidos de $R^2$ (Parque Solar N1)

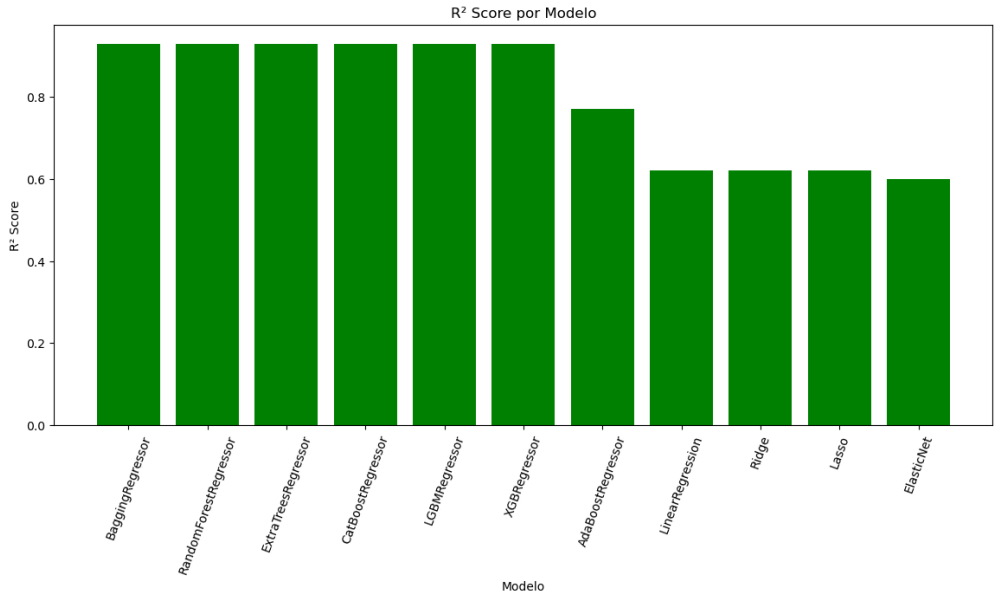


**Figura 7.9.** Resultados Obtenidos de $R^2$ (Parque Solar N2)

### 7.4.3. Gráficas del Error Cuadrático Medio (MSE)

La gráficas de la figuras 7.10, 7.11 y 7.12, muestran la magnitud de los errores al cuadrado para cada modelo, de modo que los desvíos grandes quedan penalizados con mayor intensidad. Un valor de MSE más bajo implica que, en promedio, las predicciones se ajustan mejor a los valores reales, especialmente en los casos donde las discrepancias podrían ser más pronunciadas. Por ello, analizar esta métrica permite identificar rápidamente qué modelos presentan un mejor ajuste global y cuáles pueden estar incurriendo en errores más considerables.

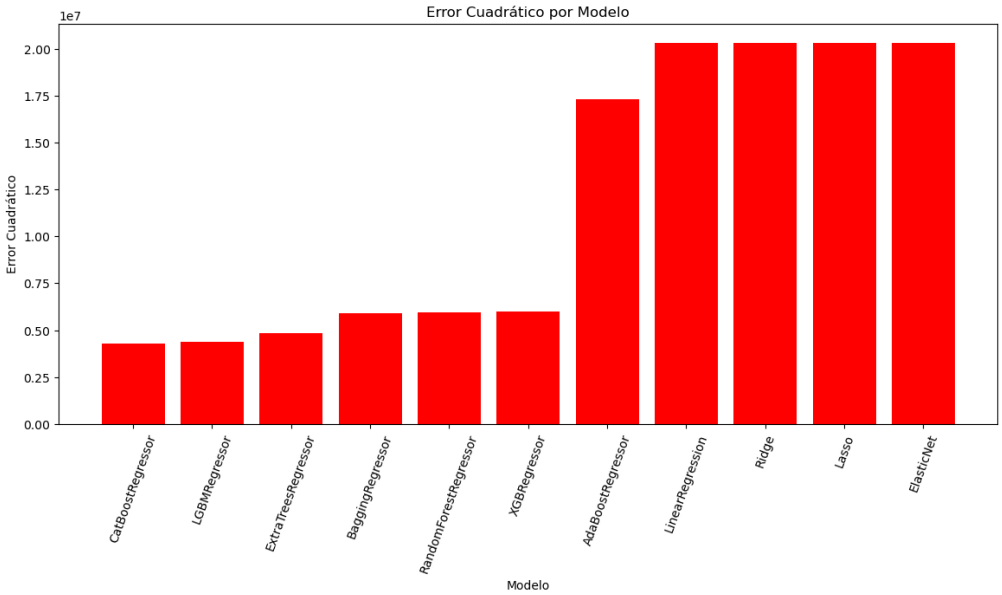


**Figura 7.10.** Resultados Obtenidos de Error Cuadrático (Parque Eólico)

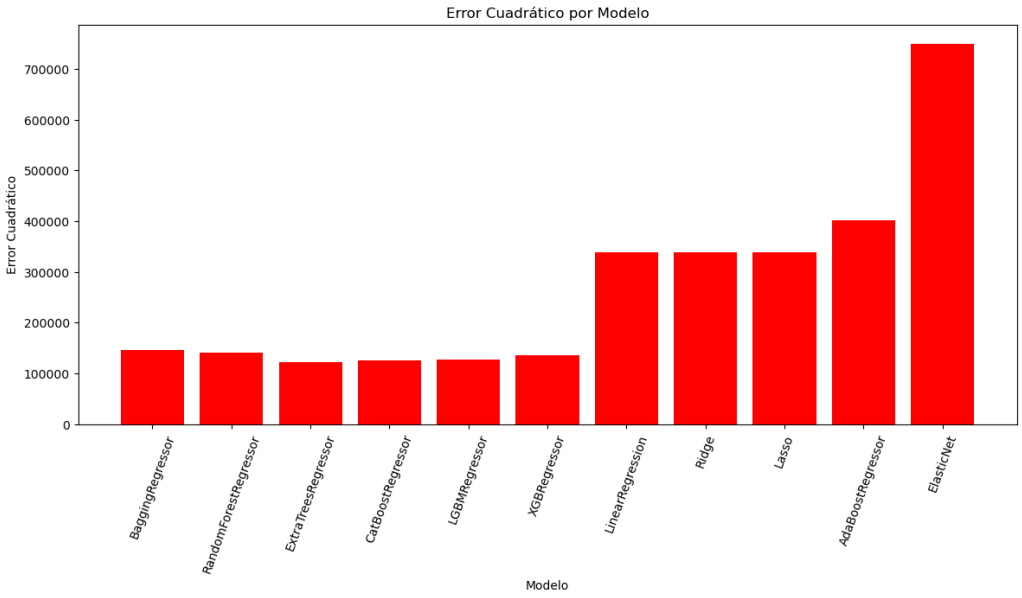


**Figura 7.11.** Resultados Obtenidos de Error Cuadrático (Parque Solar N1)

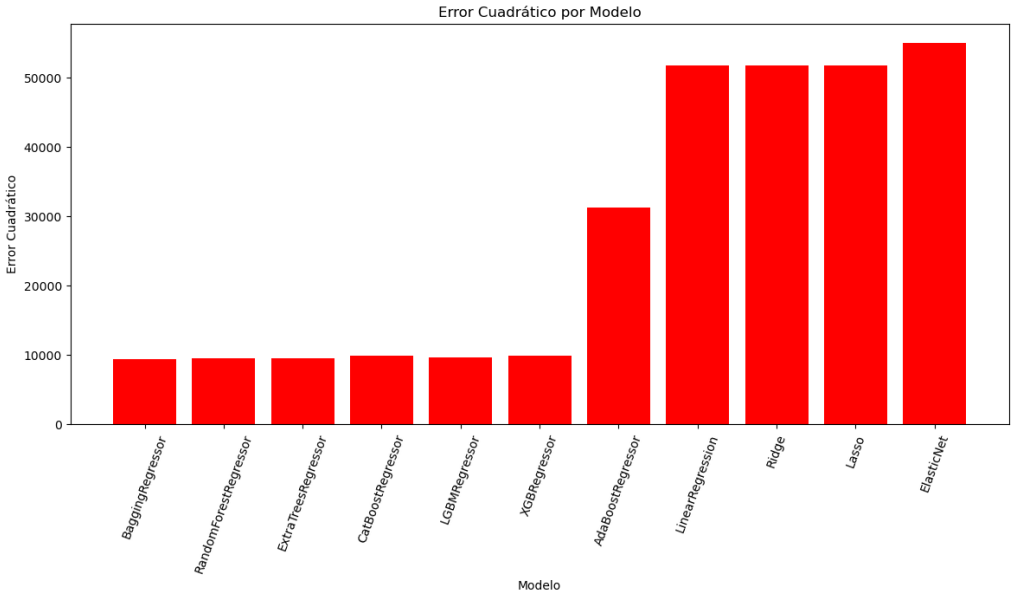


**Figura 7.12.** Resultados Obtenidos de Error Cuadrático (Parque solar N2)

### 7.4.4. Gráficas del Error Absoluto Medio (MAE)

La gráficas de la figuras 7.13, 7.14 y 7.15, representan el promedio de la diferencia absoluta entre las predicciones y los valores reales, expresado en las mismas unidades que la variable objetivo. De este modo, un MAE reducido indica que, en promedio, el modelo produce estimaciones más cercanas a la realidad y, por tanto, resulta más preciso. Además, esta métrica facilita la interpretación práctica de los errores, ya que muestra cuánto se apartan las predicciones de los datos reales en términos absolutos.

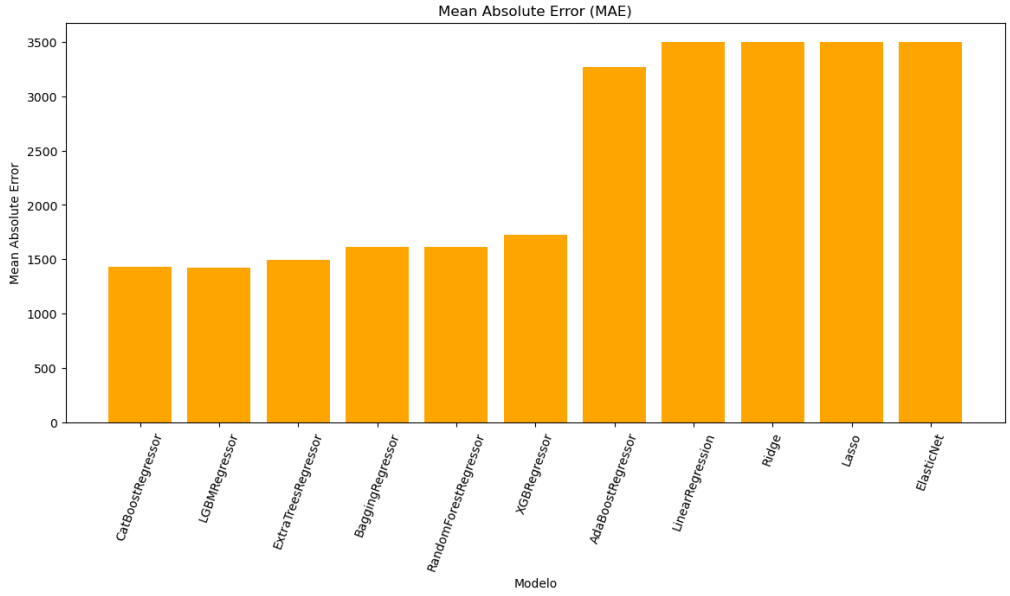


**Figura 7.13.** Resultados Obtenidos de MAE (Parque Eólico)

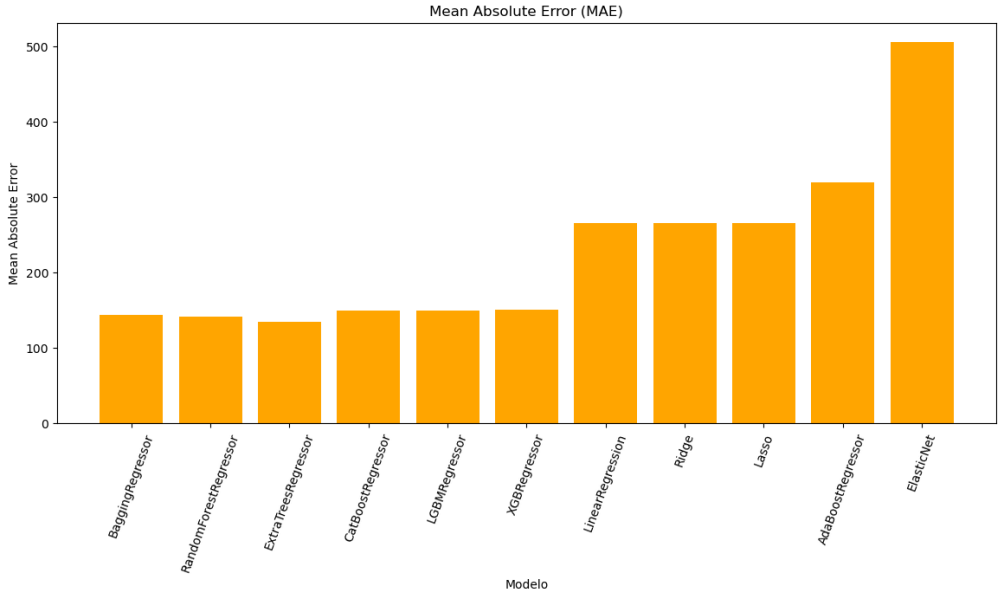


**Figura 7.14.** Resultados Obtenidos de MAE (Parque solar N1)

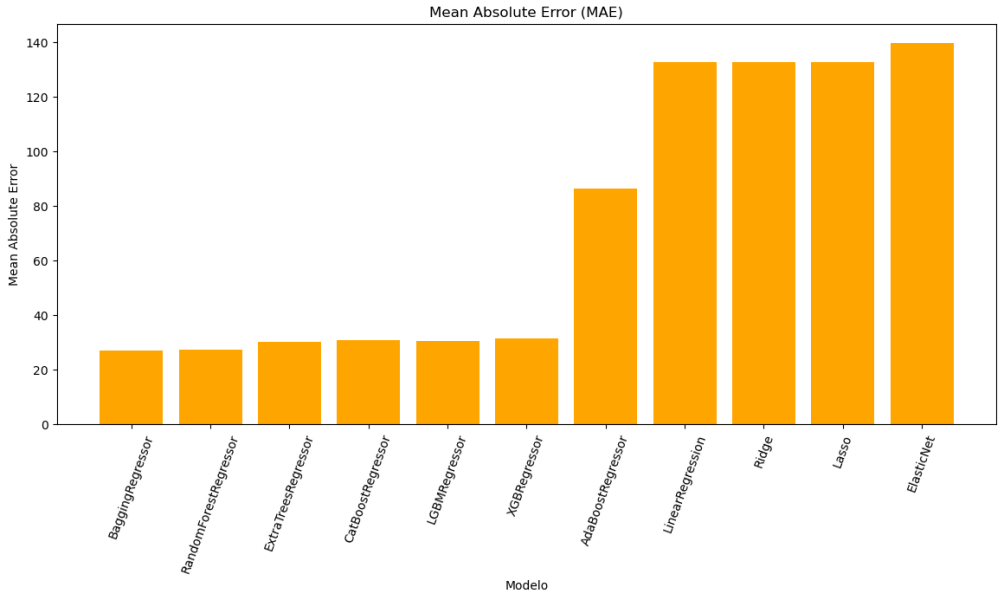


**Figura 7.15.** Resultados Obtenidos de MAE (Parque Solar N2)

### 7.4.5. Gráficas de Diferencia entre Puntajes de Entrenamiento y Prueba

La gráficas de las figuras 7.16, 7.17 y 7.18, muestran el porcentaje de separación entre el rendimiento en entrenamiento y el de prueba para cada modelo. Los valores más elevados sugieren un posible sobreajuste, es decir, que el modelo aprende en exceso los detalles del conjunto de entrenamiento. Por el contrario, diferencias más pequeñas indican que el modelo generaliza mejor y conserva un rendimiento uniforme frente a datos nuevos. Como se puede observar existen valores negativos, esto significa que el modelo obtiene un puntaje, o precisión mayor en el conjunto de

prueba que en el de entrenamiento.

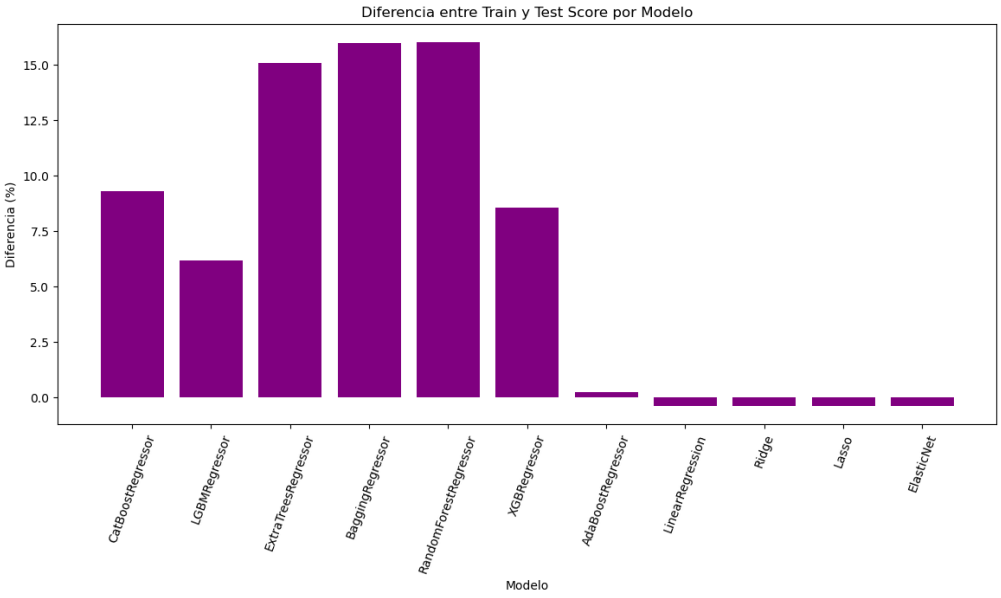


**Figura 7.16.** Diferencia entre Train Score y Test Score (Parque Eólico)

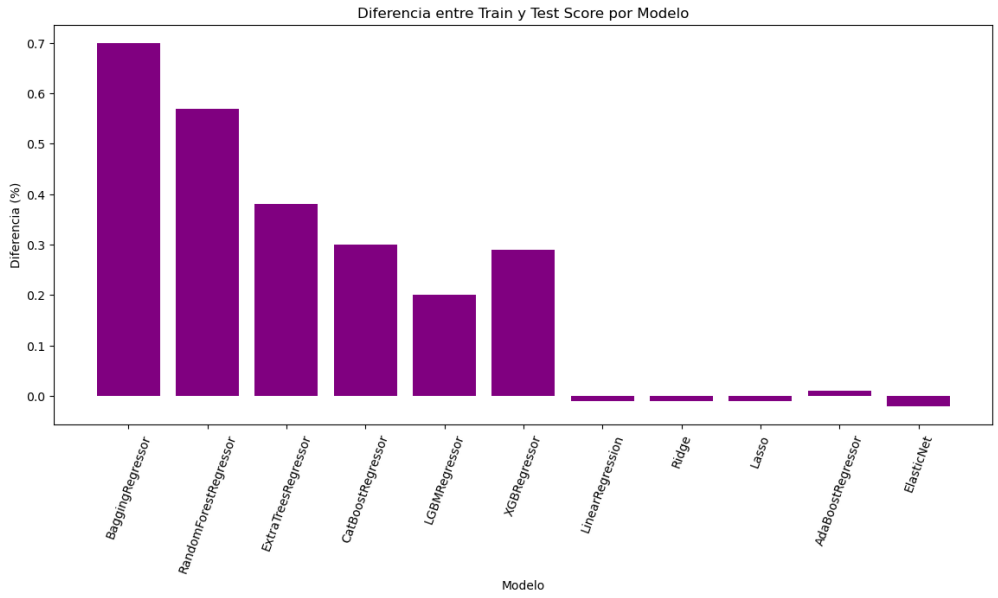


**Figura 7.17.** Diferencia entre Train Score y Test Score (Parque Solar N1)

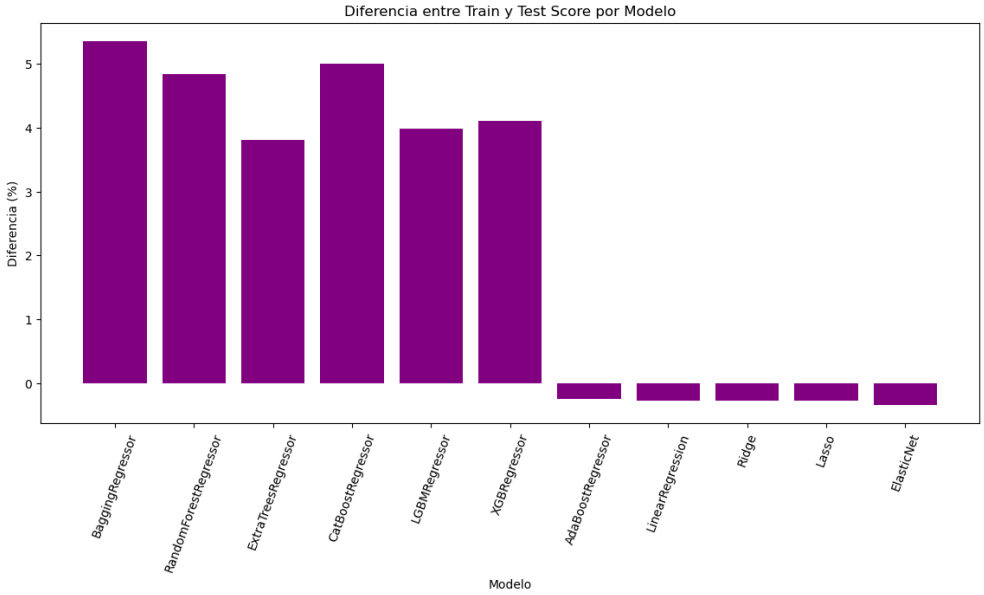


**Figura 7.18.** Diferencia entre Train Score y Test Score (Parque Solar N2)

# 8 | Modelos de Aprendizaje Profundo

El uso de modelos de aprendizaje profundo para la predicción de la generación de energía en parques eólicos y solares ha permitido mejorar la precisión y fiabilidad de las estimaciones en comparación con otros enfoques de Machine Learning. En esta sección, se presentan los resultados obtenidos tras la implementación de redes neuronales profundas optimizadas para la predicción de potencia en diferentes instalaciones de generación de energía renovable.

Los modelos se pueden encontrar en este repositorio de GitHub.

(https://github.com/josemauma/TFG-ML-DL-WIND-SOLAR-FARMS)

- **Modelo W_DL:** Este modelo ha sido desarrollado para el análisis de datos del parque eólico. Consulta Modelo W_DL para mas información.

  (https://github.com/josemauma/TFG-ML-DL-WIND-SOLAR-FARMS/blob/main/WINDMILLS/TFG_W_DL.ipynb)

- **Modelo SP1_DL:** Este modelo ha sido desarrollado para el análisis de datos del parque solar N1. Consulta SP1_DL para mas información.

  (https://github.com/josemauma/TFG-ML-DL-WIND-SOLAR-FARMS/blob/main/SOLAR%20PANELS/TFG_SP1_DL.ipynb)

- **Modelo SP2_DL:** Este modelo ha sido desarrollado para el análisis de datos del parque solar N2. Consulta SP2_DL para más información.

  (https://github.com/josemauma/TFG-ML-DL-WIND-SOLAR-FARMS/blob/main/SOLAR%20PANELS/TFG_SP2_DL.ipynb)

- **Modelo SP12_DL:** Este modelo ha sido desarrollado para el análisis de datos combinado del parque solar N1 y N2. Consulta SP12_DL para más información.

  (https://github.com/josemauma/TFG-ML-DL-WIND-SOLAR-FARMS/blob/main/SOLAR%20PANELS/TFG_SP12_DL.ipynb)

## 8.1. Diseño del Modelo

Se ha desarrollado un modelo de Deep Learning basado en una arquitectura de red neuronal profunda compuesta por múltiples capas densas. Con el objetivo de aumentar la capacidad de generalización y evitar el sobreajuste, se han integrado varias técnicas de regularización fundamentales:

- **Regularización L2**: esta técnica penaliza la magnitud excesiva de los pesos en el modelo, contribuyendo a evitar que el mismo se adapte de forma demasiado específica a los datos de entrenamiento:

$$L = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \; + \; \lambda \sum_{j=1}^{p} w_j^2$$

  **Donde:**

  - $N$ es el número de muestras.
  - $y_i$ es el valor real de la muestra $i$.
  - $\hat{y}_i$ es la predicción del modelo para la muestra $i$.
  - $\lambda$ es el parámetro de regularización L2.
  - $w_j$ es el peso $j$ del modelo.

- **Normalización por lotes (Batch Normalization)**: al normalizar la entrada de cada capa, se facilita una convergencia más estable y rápida durante el entrenamiento:

$$\hat{z} = \frac{z - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$$

  **Donde:**

  - $z$ es la activación original de la capa previa.
  - $\mu_B$ es la media del mini-lote.
  - $\sigma_B^2$ es la varianza del mini-lote.
  - $\epsilon$ es un pequeño valor para evitar división por cero.

- **Dropout**: consiste en desactivar aleatoriamente un porcentaje de neuronas durante el entrenamiento para reducir el sobreajuste:

$$x_i' = \begin{cases} 0, & \text{con probabilidad } p, \\ x_i, & \text{con probabilidad } 1-p. \end{cases}$$

  **Donde:**

  - $x_i$ es el valor de activación de la neurona $i$ antes de dropout.
  - $x_i'$ es el valor tras aplicar dropout.
  - $p$ es la tasa de dropout.

Para optimizar el modelo se empleó el algoritmo **Adam** (Adaptive Moment Estimation), con tasa de aprendizaje inicial $\eta = 0{,}001$ y las siguientes actualizaciones:

$$m_t = \beta_1 m_{t-1} + (1-\beta_1)\frac{\partial L}{\partial W}, \quad v_t = \beta_2 v_{t-1} + (1-\beta_2)\left(\frac{\partial L}{\partial W}\right)^2$$
$$\hat{m}_t = \frac{m_t}{1-\beta_1^t}, \quad \hat{v}_t = \frac{v_t}{1-\beta_2^t}$$
$$W \leftarrow W - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$$

**Donde:**

- $W$ es el vector de parámetros.
- $m_t$ y $v_t$ son las estimaciones de primer y segundo momento.
- $\beta_1, \beta_2$ son coeficientes de decaimiento exponencial.
- $\epsilon$ es un término de estabilidad numérica.

Además se aplicaron:

- **ReduceLROnPlateau**: reduce la tasa de aprendizaje $\eta$ cuando la métrica de validación deja de mejorar.
- **Early Stopping**: detiene el entrenamiento si la pérdida de validación no mejora tras varias épocas.

### 8.1.1. Estandarización

El método **StandardScaler** es una técnica de preprocesamiento crítica antes de entrenar la red. Transforma cada característica para que tenga media cero y desviación estándar uno:

$$X' = \frac{X - \mu}{\sigma}$$

**Donde:**

- $X$ es el valor original de la característica.
- $\mu$ es la media de la característica en el conjunto de datos.
- $\sigma$ es la desviación estándar de la característica.

### 8.1.2. Función de Activación ReLU

La función de activación **ReLU** (Rectified Linear Unit) introduce no linealidad:

$$\sigma(x) = \text{máx}(0, x).$$

**Donde:**

- $x$ es la entrada a la función de activación.

## 8.2. Parque Eólico

En este subcapítulo podemos ver los resultados obtenidos del modelo creado para el parque eólico.

### 8.2.1. Evaluación del Modelo

El desempeño del modelo se evaluó utilizando validación cruzada, considerando las siguientes métricas:

- **Coeficiente de Determinación ($R^2$):** Un valor medio de 0.753 indica que el modelo es capaz de explicar aproximadamente el 75.3 % de la variabilidad de la potencia generada.
- **Error Cuadrático Medio (MSE):** Con un valor de 7,819,360.8, el MSE proporciona una medida de la dispersión de las predicciones respecto a los valores reales.
- **Error Absoluto Medio (MAE):** Un MAE de 1,907.92 refuerza que, en promedio, las predicciones se desvían en esa cantidad en comparación con los datos reales.

Las gráficas de evolución de la pérdida y del MAE durante el entrenamiento y validación evidencian una convergencia estable, sin señales de sobreajuste, lo que confirma la robustez del proceso de entrenamiento.

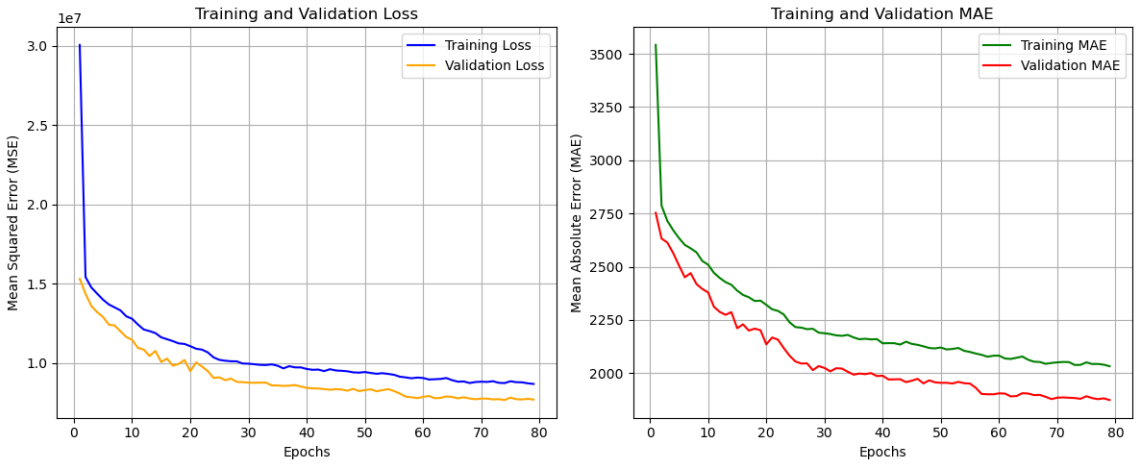


**Figura 8.1.** Evolución de las Métricas de Error (MSE y MAE) Durante el Entrenamiento y la Validación del Modelo

En la figura 8.1, las gráficas que presento, se aprecia cómo el error (tanto el cuadrático medio, MSE, a la izquierda, como el absoluto medio, MAE, a la derecha) desciende a medida que avanzan las épocas de entrenamiento. Esto indica que el modelo va aprendiendo y cada vez comete menos errores al predecir. Además, resulta positivo observar que las curvas de validación se mantienen cercanas a las de entrenamiento, ya que ello sugiere que el modelo no está "aprendiendo de memoria" los datos de entrenamiento, sino que también funciona razonablemente bien con datos nuevos.

En las primeras etapas se ven descensos muy marcados, algo normal cuando el modelo empieza a ajustarse; a partir de cierto punto, las mejoras se vuelven más sutiles, lo cual también es esperable porque a medida que avanza el entrenamiento, cada vez queda menos error por pulir. En conjunto, estas gráficas muestran que el proceso de aprendizaje se ha desarrollado de forma adecuada y que el modelo se ha vuelto más preciso con el paso de las épocas.

### 8.2.2. Análisis de los Resultados

En las gráficas de validación cruzada de la figura 8.2, puede verse que los valores de MSE (error cuadrático medio) y MAE (error absoluto medio) se mantienen relativamente consistentes a lo largo de los distintos "folds" (o particiones de los datos), lo que significa que el modelo se comporta de forma bastante estable en todas las iteraciones. El promedio de MSE ronda los 7,8 millones, mientras que el de MAE se sitúa alrededor de 1900, indicando la magnitud típica de los errores en las predicciones.

Por otro lado, el coeficiente de determinación $R^2$ promedia cerca de 0,75. Esto quiere decir que, en conjunto, el modelo explica alrededor del 75 % de la variabilidad de los datos. Es una buena señal, pues significa que el rendimiento no se debe a la casualidad y que el modelo tiene un poder predictivo razonable.

En resumen, las métricas apuntan a un desempeño estable y robusto en cada "fold", lo que sugiere que el modelo generaliza bien y no depende excesivamente de un único conjunto de entrenamiento.

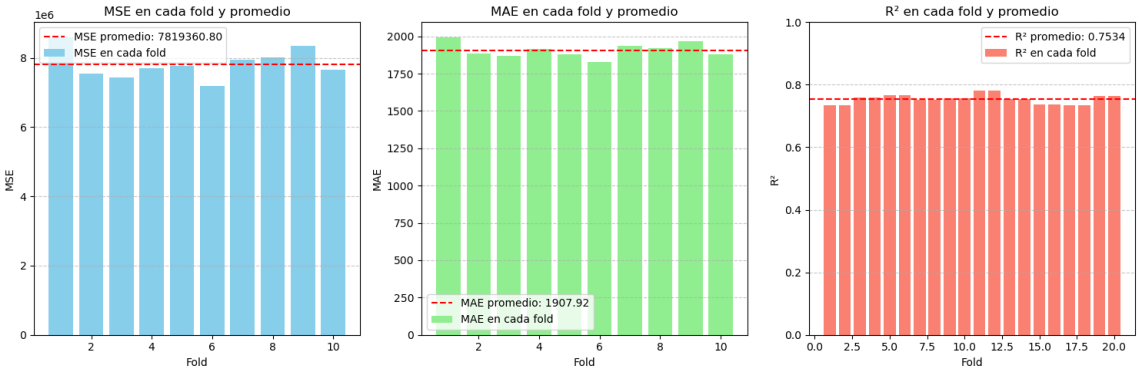


**Figura 8.2.** Resultados de MSE, MAE y $R^2$ en cada Fold y su Promedio

### 8.2.3. Evaluación de las Predicciones

La gráfica que compara los valores reales y las predicciones del modelo en el conjunto de prueba refleja una elevada concordancia entre ambas curvas. Este resultado indica que el modelo ha captado con precisión la tendencia de los datos, mostrándose sólido tanto en las variaciones generales como en los cambios más bruscos.

De forma destacable, el modelo reproduce con exactitud los picos de generación, siguiendo de cerca los valores reales sin incurrir en desfases significativos ni suavizados excesivos. Aunque se observan ligeras diferencias en algunos puntos, estas no parecen ser sistemáticas, lo que sugiere

que el modelo generaliza correctamente sin introducir sesgos notables. En conjunto, esta visualización refuerza la solidez del enfoque propuesto y su capacidad para ofrecer predicciones fiables en distintos escenarios operativos.

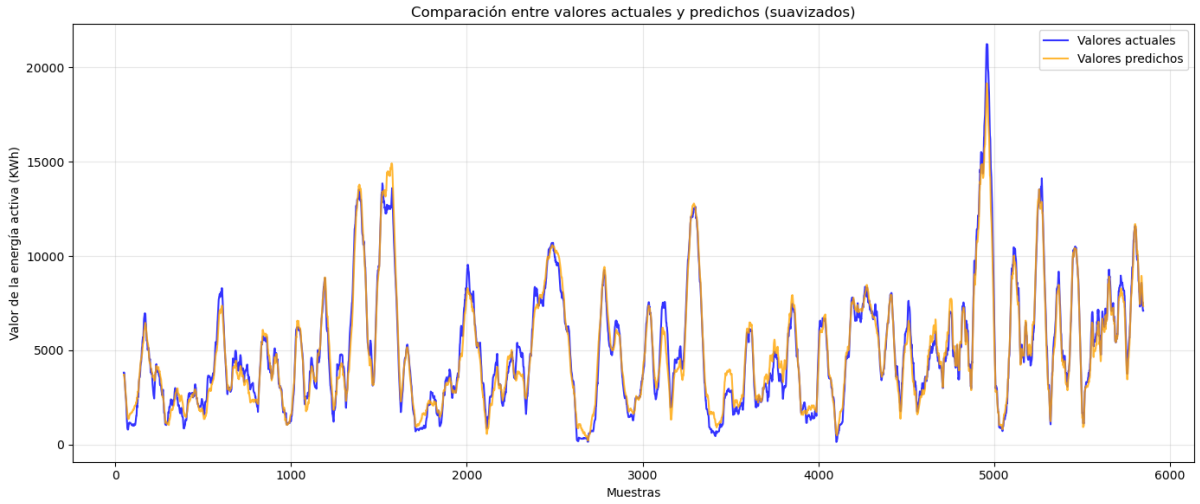


**Figura 8.3.** Evolución de las Métricas de Error (MSE y MAE) Durante el Entrenamiento y la Validación del Modelo

## 8.3. Parque Solar N1

### 8.3.1. Evaluación del Modelo

El rendimiento del modelo se evaluó mediante validación cruzada, considerando las siguientes métricas:

- **Coeficiente de determinación ($R^2$):** Un valor medio de 0.982 indica que el modelo explica más del 98 % de la variabilidad en la potencia generada.
- **Error Cuadrático Medio (MSE):** Con un valor de 290120.92, el MSE evidencia una baja dispersión en las predicciones respecto a los datos reales.
- **Error Absoluto Medio (MAE):** Un MAE de 215.72 confirma la precisión del modelo, ya que, en promedio, las predicciones se desvían poco de los valores reales.

Las curvas de evolución de la pérdida y del MAE durante el entrenamiento y la validación evidencian una convergencia estable, sin señales de sobreajuste—algo así como un modelo que se pone en forma sin comer pastelitos de más.

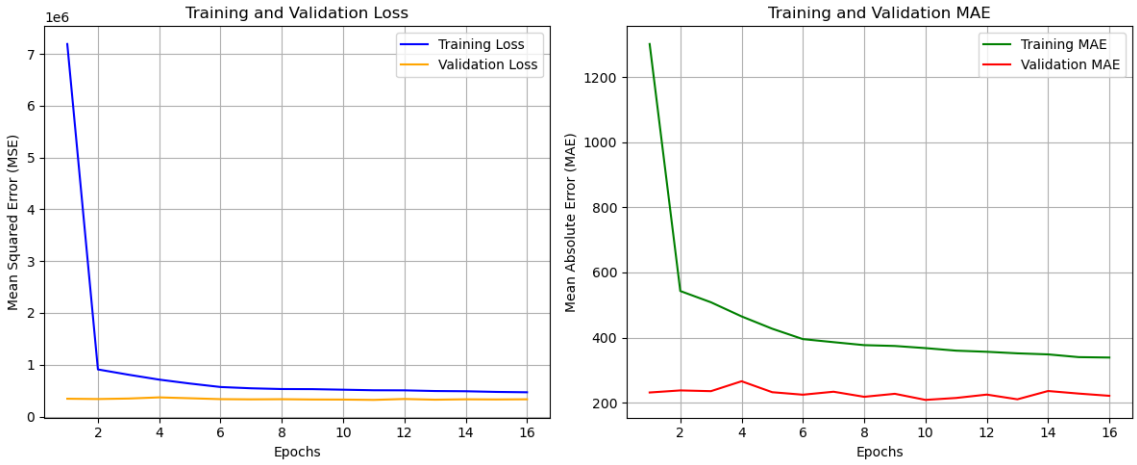


**Figura 8.4.** Evolución de las Métricas de Error (MSE y MAE) en el Parque Solar N1

### 8.3.2. Análisis de los Resultados

La combinación de técnicas de regularización y la optimización con Adam han permitido estabilizar el entrenamiento, reduciendo la varianza y mejorando la capacidad predictiva del modelo. La validación cruzada demuestra que, a través de distintos subconjuntos de datos, el modelo mantiene un desempeño consistente—un indicativo claro de que no se quedó dormido en el trabajo. La alta correspondencia entre los valores reales y las predicciones evidencia que el modelo es capaz de capturar tanto las tendencias generales como las fluctuaciones abruptas en la generación de energía.

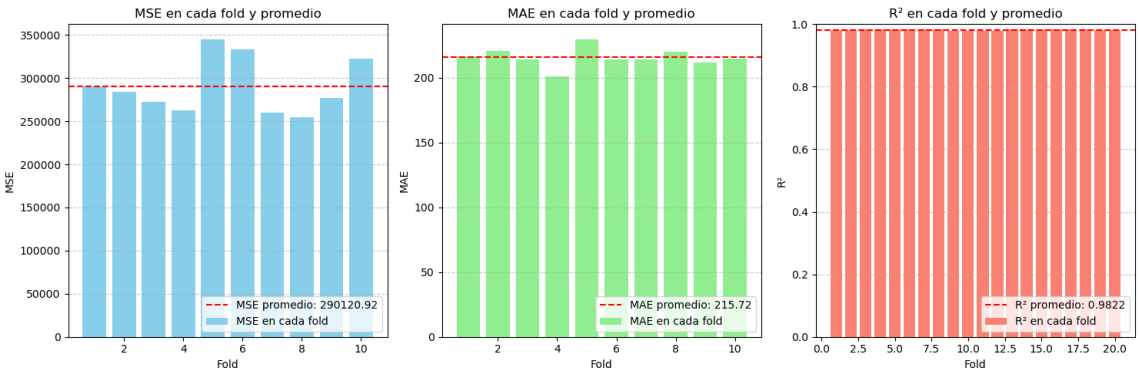


**Figura 8.5.** Resultados de MSE, MAE y $R^2$ en Cada Fold en el Parque Solar N1

### 8.3.3. Evaluación de las Predicciones

La comparación visual entre los valores reales y las predicciones en el conjunto de prueba refleja una elevada concordancia entre ambas curvas. En la figura 8.6, que abarca las primeras 750 muestras, se observa un ajuste preciso en el que se capturan de forma exacta los picos y valles característicos de la potencia generada, sin desfases significativos ni suavizados excesivos. En la figura 8.9, al extender el análisis a un mayor volumen de datos, se confirma la capacidad

del modelo para seguir la dinámica operativa de forma estable, incluso ante cambios abruptos.

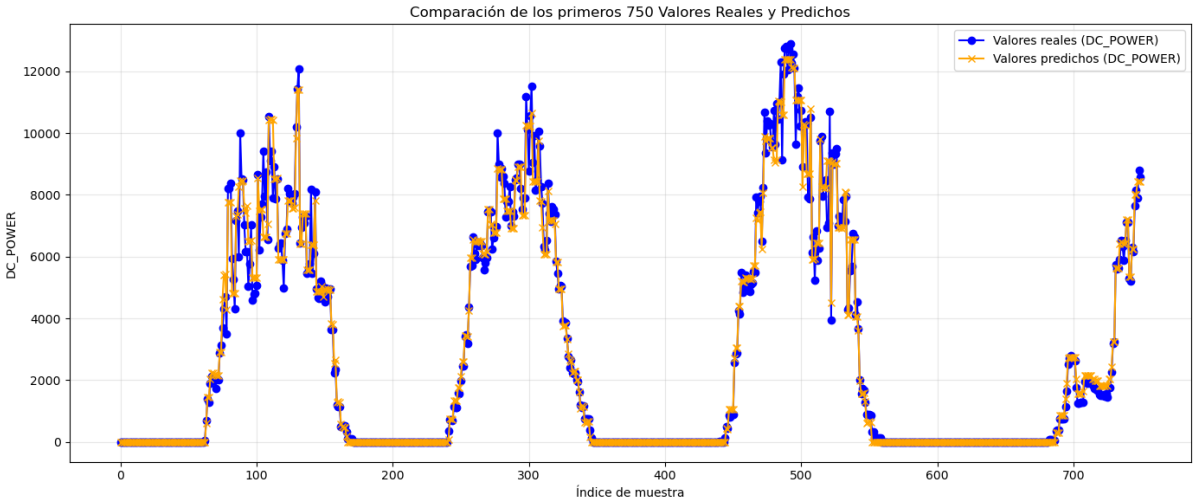


**Figura 8.6.** Comparación de los Primeros 750 Valores: Reales vs Predichos en el Parque Solar N1

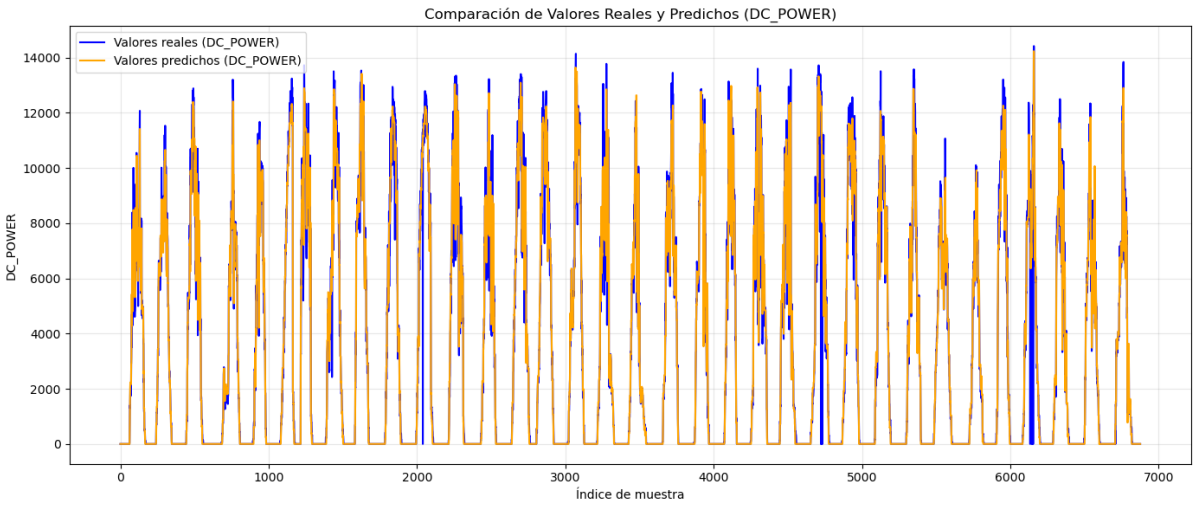


**Figura 8.7.** Comparación de los Valores: Reales vs Predichos en el Parque Solar N1

Estos resultados validan la eficacia del modelo propuesto, demostrando su aplicabilidad en escenarios operativos reales donde la precisión y la fiabilidad en la predicción de la potencia DC son factores críticos.

## 8.4. Parque Solar N2

### 8.4.1. Evaluación del Modelo

La evaluación del modelo se llevó a cabo mediante validación cruzada, obteniéndose métricas satisfactorias:

- **Coeficiente de Determinación ($R^2$):** Un valor medio de 0.866 indica que el modelo es capaz de explicar el 86.6 % de la variabilidad en la potencia generada, lo que, aunque inferior al obtenido en la primera planta, sigue siendo muy robusto.
- **Error Cuadrático Medio (MSE):** Con un valor promedio de 18,414.39, el MSE revela una baja dispersión de las predicciones en comparación con los valores reales.
- **Error Absoluto Medio (MAE):** Un MAE de 48.95 refuerza la precisión del modelo en la estimación de la variable objetivo.

Las curvas de pérdida y del MAE durante el entrenamiento muestran una rápida convergencia en las primeras iteraciones, estabilizándose en las últimas épocas, lo que evidencia que el modelo ha encontrado su ritmo sin necesidad de una maratón extra.

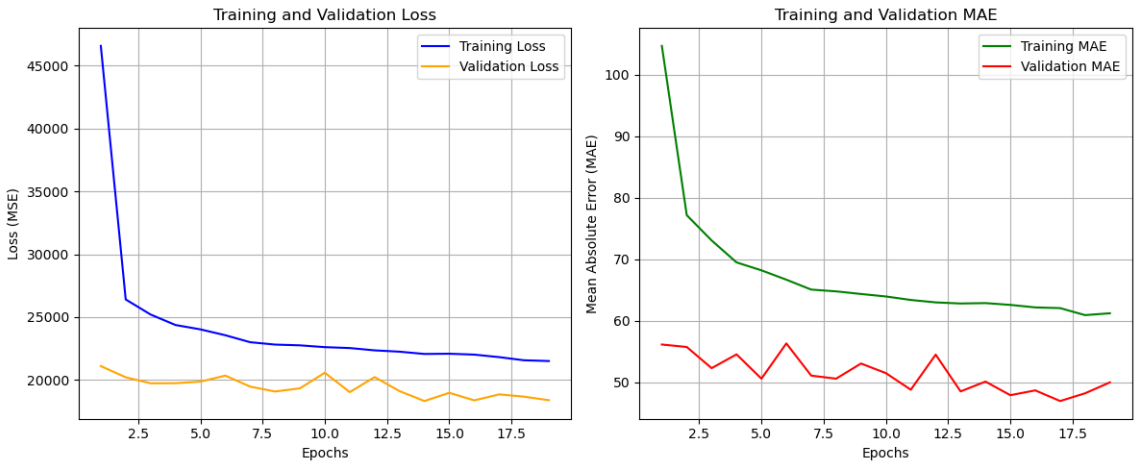


**Figura 8.8.** Curvas de Entrenamiento del Modelo DL para el Parque Solar N2

### 8.4.2. Análisis de los Resultados

La comparación entre los valores reales y los predichos demuestra un ajuste preciso en la estimación de la potencia generada. Al analizar la serie completa, el modelo capta de forma efectiva los patrones cíclicos de generación, respondiendo adecuadamente a las variaciones diarias típicas de la producción fotovoltaica. En el análisis de un subconjunto de 750 muestras, se aprecia una correspondencia estrecha entre las predicciones y los valores reales, con un seguimiento adecuado de picos y valles. Aunque se observan ligeras desviaciones puntuales, estas no afectan la precisión global del modelo.

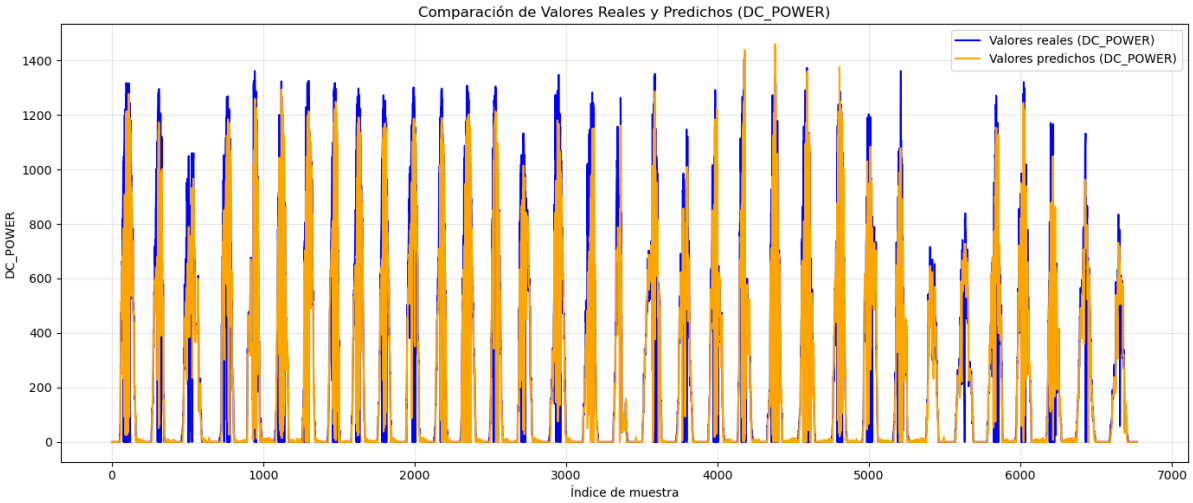


**Figura 8.9.** Predicción de la Serie Completa: Valores Reales vs Predichos

### 8.4.3. Evaluación de las Predicciones

Las gráficas permiten analizar detalladamente la capacidad del modelo para replicar el comportamiento de la variable objetivo. En la figura que muestra un subconjunto de 750 muestras, se evidencia una correspondencia minuciosa entre ambas series, donde el modelo responde de forma ágil ante cambios abruptos en la producción sin perder el ritmo en la mayoría de los puntos. Aunque se notan algunas desviaciones puntuales, la tendencia general se mantiene, reafirmando la fiabilidad del enfoque.

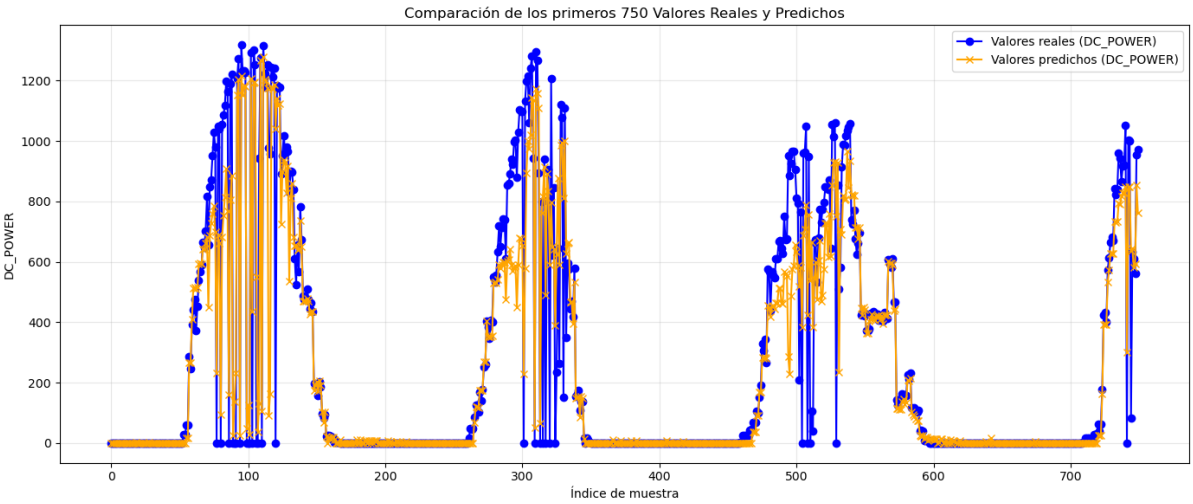


**Figura 8.10.** Comparación de los Primeros 750 Valores: Reales vs Predichos en el Parque Solar N2

Estos resultados confirman que el modelo es capaz de adaptarse a distintas plantas solares sin perder precisión en la estimación de la potencia generada. La solidez de su desempeño en diferentes escenarios sugiere que puede ser utilizado con confianza en aplicaciones de monitoreo

y predicción en entornos reales, garantizando estimaciones fiables y consistentes.

## 8.5. Modelo Combinado de los Parques Solares

En esta sección vamos a explicar el por qué y como se ha podido conseguir un modelo integrado de los dos parques para obtener una alta precisión y en obtener así, un modelo general para los dos parques.

Primeramente, al tener dos datasets con las mismas características, aunque con diferentes datos, se puede obtener un dataset conjunto que haga mas robusto al nuevo modelo.

### 8.5.1. Diferencia de Datos en los Datasets

Como se puede apreciar en la gráficas que aparecen mas abajo, las cuales comparan las características en común en los dos datasets, existen diferencias en los datos obtenidos en en los dos parques.

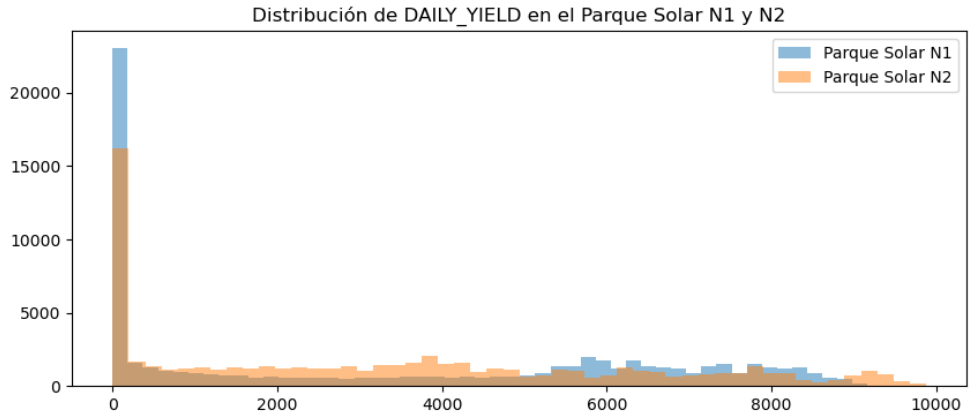


**Figura 8.11.** Distribución de DAILY YIELD en el Parque Solar N1 y N2

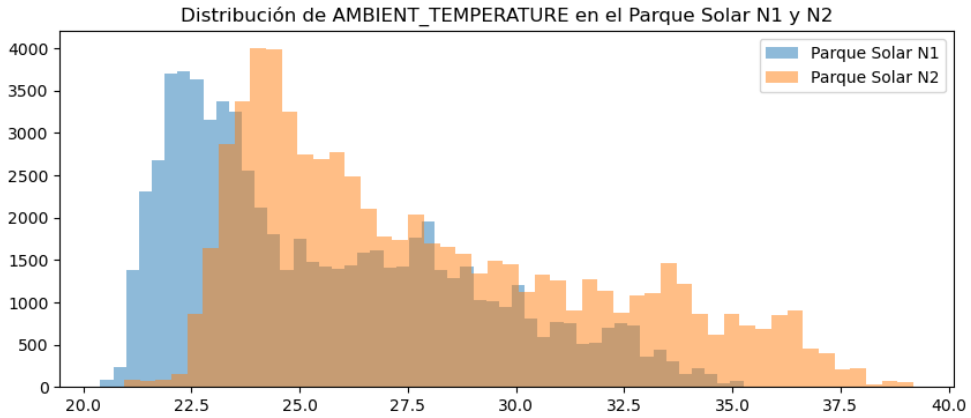


**Figura 8.12.** Distribución de AMBIENT TEMPERATURE en el Parque Solar N1 y N2

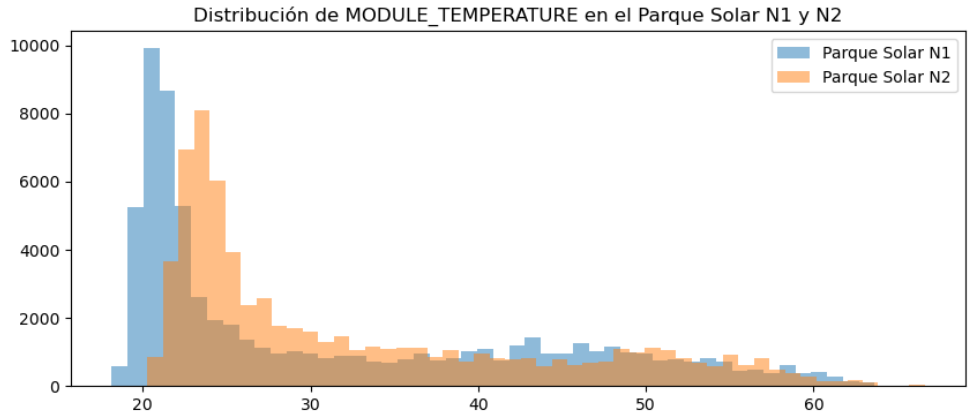


**Figura 8.13.** Distribución de MODULE TEMPERATURE en el Parque Solar N1 y N2

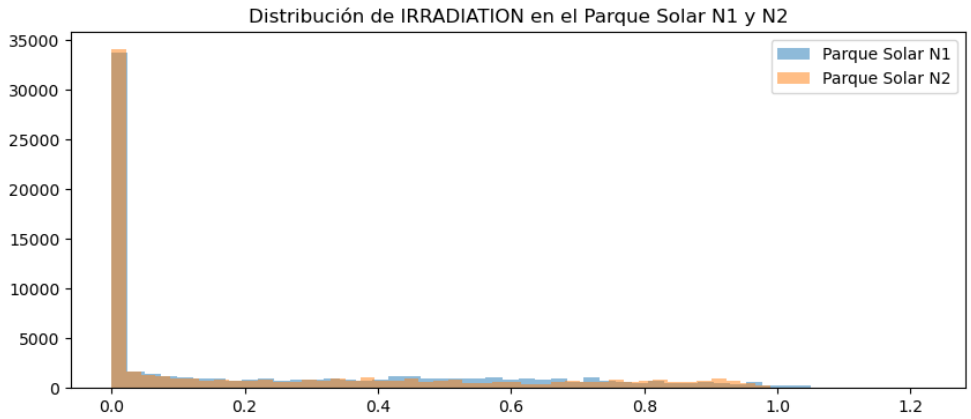


**Figura 8.14.** Distribución de IRRADATION en el Parque Solar N1 y N2

Estas diferencias entre datasets se ha podido comprobar que si se usa el del parque solar N1

para el modelo del parque solar N2 obtenemos un resultado de baja precisión, y viceversa. Es por ello que el objetivo de esta sección es el de combinar los modelos del Parque solar N1 y el Parque solar N2.

### 8.5.2. Diseño del Código

El código diseñado demuestra un proceso de integración de modelos predictivos para la generación de energía solar, donde se cargan previamente los dos modelos entrenados y sus correspondientes escaladores, uno para cada planta solar, a través de las funciones load_model y joblib.load.

Posteriormente, se definen las variables predictoras (X) y la variable objetivo (Y) a partir de los conjuntos de datos enriquecidos con información meteorológica, diferenciando entre la planta 1 y la planta 2. Para cada planta, se divide el dataset en conjuntos de entrenamiento y prueba utilizando la función train_test_split (80 % para train y 20 % para test), lo que permite evaluar la capacidad predictiva de los modelos en datos no vistos. La función predict_ensemble se encarga de realizar el ensamble de ambos modelos: primero, iguala el número de muestras a partir del menor de los dos conjuntos de características, y luego procede a escalar estas muestras mediante los escaladores correspondientes. Con los datos ya escalados, se obtiene la predicción individual de cada modelo, y el método de ensamble consiste en calcular el promedio aritmético de ambas predicciones, generando así una predicción final que se espera sea más robusta y precisa que las obtenidas de forma individual. Finalmente, se ejemplifica el uso de esta función de ensamble aplicándola a los conjuntos de prueba de ambas plantas para obtener la predicción combinada, lo que demuestra la viabilidad y eficiencia del enfoque en la integración de modelos.

### 8.5.3. Resultados del Modelo Ensamblado

Los resultados obtenidos evidencian la eficacia del enfoque de ensamblado. La magnitud del error, reflejada en el MSE de 76237.75 y el MAE de 120.18, demuestra que las diferencias entre las predicciones y los valores reales son aceptablemente bajas, permitiendo una interpretación práctica robusta de los datos. El elevado coeficiente de determinación ($R^2 = 0.9813$) indica que el modelo captura casi en su totalidad la variabilidad inherente a la generación solar, lo que respalda la estrategia de combinar las predicciones de ambos modelos para obtener un desempeño superior en comparación con enfoques individuales.

# 9 | Conclusiones

## 9.1. Conclusión

En este Trabajo Fin de Grado se ha presentado un marco predictivo que integra técnicas de aprendizaje automático y profundo para estimar la generación eléctrica en un parque eólico y dos parques solares.

Partiendo del conjunto de datos obtenidos, se aplicaron unas etapas de limpieza y normalización de datos, previas a la creación de los modelos predictivos. Posteriormente se entrenaron y se validaron once modelos predictivos clásicos más una red neuronal, para cada uno de los parques. Se han empleado $R^2$, MAE y MSE como métricas de referencia para la evaluación de la calidad de los modelos. En total se han entrenado 36 modelos.

La evaluación de los mismos ha mostrado que los métodos de ensamble BaggingRegressor proporcionan las predicciones más fiables en los parques solares, mientras que CatBoostRegressor resulta el más consistente en el entorno eólico. La red profunda, aunque todavía mejorable, demostró una robustez prometedora frente a datos incompletos. Como es de saber, cuanto mejor y mas completos sean los datos, mejor serán los modelos y se obtendrán mejores resultados.

Esta aproximación constituye la primera evaluación sistemática, que combina modelos de predictivos automáticos y profundos, listos para transferirse a una operación real de planta.

Entre las limitaciones cabe señalar la heterogeneidad temporal de las series, días con datos incompletos, y la falta de meteorología en tiempo real; ambas se mitigaron con técnicas de aumentación de datos: imputación múltiple y escenarios sintéticos. Los resultados sugieren que serían necesarios futuros trabajos hacia fuentes de datos mas grandes y un aprendizaje continuo.

En conjunto, los resultados avalan que unos modelos bien calibrados pueden mejorar de forma tangible la planificación y reducir los desvíos, acercándonos a un sistema energético renovable más fiable, eficiente y competitivo.

## 9.2. Trabajo Futuro

Durante la realización de este trabajo han surgido muchas líneas de trabajo para poder mejorarlo y darle una utilidad mas óptima. Las que más se ajustan son las que se definen a continuación:

- **Ingeniería de Datos y Calidad:** mejorar la obtención y limpieza de datos mediante

WebScraping, APIs especializadas y simulaciones sintéticas (CFD, imágenes satelitales), junto con detección automática de outliers (RANSAC, Isolation Forest).

- **Aprendizaje Continuo:** implementar reentrenamiento incremental para actualizar modelos en tiempo real, monitorizar deriva de datos y modelo y automatizar un ajuste de hiperparámetros con AutoML (Optuna, Auto-Sklearn).
- **Despliegue y Aplicación:** desarrollar microservicios Docker/Kubernetes para crear APIs de predicción, acompañado de un dashboard web (React + D3.js) y unas notificaciones inteligentes que avisen de umbrales críticos.
- **Modelos Híbridos y Explicabilidad:** combinar modelos clásicos (XGB, CatBoost) con DL (RNN, transformers) e integrar técnicas de Explainable AI (SHAP, LIME) para interpretar predicciones.
- **Expansión de Servicios:** extender la plataforma a la predicción de calidad de red y a mantenimiento predictivo; integrar lecturas IoT en tiempo real para ajustar pronósticos minuto a minuto.

## 9.3. Experiencia Personal

Durante el desarrollo de este proyecto, me he enfrentado a numerosas dificultades, particularmente cuando estaba empleando métodos avanzados de Aprendizaje Automático y Aprendizaje Profundo. Inicialmente, fue complicado manejar los algoritmos debido a su complejidad y a la gran cantidad de información disponible.

No obstante, conforme incrementaba mi lectura de artículos científicos, documentación técnica y guías prácticas, lograba construir una comprensión sólida de los conceptos fundamentales con el paso del tiempo. Para ello, he tenido que continuar con mi propio aprendizaje y experimentar con diversas bibliotecas, tales como TensorFlow, Keras y Scikit-learn, con el fin de hacer mi proceso de aprendizaje más dinámico y provechoso.

El procedimiento de elaboración de modelos fue fundamentalmente de prueba y error, y se llevó a cabo en diversas etapas. Para reducir ese tiempo, se ha convertido en una costumbre dedicar tiempo a ajustar meticulosamente los hiperparámetros, seleccionar las configuraciones óptimas, entrenar modelos de manera reiterada y verificar su funcionamiento. Han habido circunstancias en las que la frustración era ineludible, tales como cuando un modelo no alcanzaba la precisión anticipada o cuando las redes neuronales no lograban identificar patrones significativos de los datos suministrados. No obstante, estos momentos adversos fueron fundamentales para comprender la importancia de diagnosticar errores de manera correcta y continuar avanzando incluso en situaciones adversas.

Cada pequeña mejora me proporcionó un fuerte estímulo de motivación, lo cual me confió en que podía encontrar soluciones mas efectivas. Además, gracias a esta experiencia, he adquirido un gran conocimiento sobre la relevancia de preparar y procesar los datos de manera adecuada. Tomé conciencia de que la calidad de la entrada de datos tiene una influencia significativa en los resultados del modelo.

Finalmente, resultó sumamente gratificante observar que el esfuerzo puesto en el trabajo resultó en unos modelos que podían formular predicciones que eran justificables y útiles. Esta experiencia ha contribuido a la consolidación de mis conocimientos teóricos y mis competen-

cias prácticas. Además, ha originado en mí un gran interés de continuar con el aprendizaje y perfeccionamiento de estas técnicas para futuros proyectos académicos y profesionales.

# Bibliografía


[1] Pablo Andrés Buestán Andrade, Pedro Esteban Carrión Zamora, Anthony Eduardo Chamba Lara, and Juan Pablo Pazmiño Piedra. A comprehensive evaluation of ai techniques for air quality index prediction: Rnns and transformers. *Ingenius: Revista de Ciencia y Tecnología*, 33:60–75, 2025. doi: 10.17163/ings.n33.2025.06. URL http://scielo.senescyt.gob.ec/scielo.php?script=sci_arttext&pid=S1390-860X2025000100060&lng=en. Accedido: 2 de febrero de 2025.

[2] Casa del Libro. Artificial intelligence: A modern approach (3rd revised ed.). https://www.casadellibro.com/libro-artificial-intelligence-a-modern-approach-3rd-revised-ed/9781292153964/4913628?gad_source=1, n.d. Accedido: 1 de mayo de 2025.

[3] CiberseguridadMAX. Epoch. https://ciberseguridadmax.com/epoch/, n.d. Accedido: 23 de marzo de 2025.

[4] FasterCapital. Evaluación de la precisión de los modelos de series temporales. https://fastercapital.com/es/tema/evaluaci%C3%B3n-de-la-precisi%C3%B3n-de-los-modelos-de-series-temporales.html, n.d.. Accedido: 12 de abril de 2025.

[5] FasterCapital. Modelos estadísticos y datos. https://fastercapital.com/es/palabra-clave/modelos-estad%C3%ADsticos-datos.html, n.d.. Accedido: 15 de diciembre de 2024.

[6] Keras. Keras examples. https://keras.io/examples/, n.d. Accedido: 19 de marzo de 2025.

[7] Novita AI. 6 api de modelos de lenguaje de código abierto recomendadas para desarrolladores, April 2024. URL https://blogs.novita.ai/es/6-recommended-open-source-large-language-models-apis-for-developers/. Accedido: 23 de enero de 2025.

[8] Python Software Foundation. Pep 8 – style guide for python code. https://peps.python.org/pep-0008/, n.d. Accedido: 3 de abril de 2025.

[9] José A. Roca. Comienza a construirse una de las plantas de fotovoltaica flotante más grandes del mundo en singapur. https://elperiodicodelaenergia.com/comienza-a-construirse-una-de-las-plantas-de-fotovoltaica-flotante-mas-grandes-del-mundo-en-singapur/. El Periódico de la energía, 20 agosto 2020. Consultado el 2 de mayo de 2025.

[10] Federico Martín Rodríguez. Estimación de precios de mercado para celulares usados mediante técnicas de aprendizaje automático. Tesis de maestría, Universidad Torcuato Di Tella, 2024. URL https://repositorio.utdt.edu/handle/20.500.13098/12786. Accedido: 8 de diciembre de 2024.

[11] Stack Overflow. Differences in scikit-learn, keras, or pytorch. https://stackoverflow.com/questions/54527439/differences-in-scikit-learn-keras-or-pytorch/54532702#54532702, n.d. Accedido: 27 de diciembre de 2024.

[12] Toolify. Explorando métricas de regresión comunes, February 2024. URL https://www.toolify.ai/es/ai-news-es/explorando-mtricas-de-regresin-comunes-1703431. Accedido: 7 de marzo de 2025.

[13] Udemy. Python for data science and machine learning bootcamp. https://www.udemy.com/share/101WaU3@5RFSEDGyG6fTBzF2mQeOIYILr8MG3HU2ABXCPqrSeOUBL5DkGt5Lepm3qDsIG5TNQA==/, n.d. Accedido: 12 de noviembre de 2024.

[14] Wikipedia. Red neuronal prealimentada. https://es.wikipedia.org/wiki/Red_neuronal_prealimentada, n.d. Accedido: 8 de febrero de 2025.