



# Diseño de Red Neuronal Artificial para la Predicción de la Demanda Eléctrica

## *Design of an Artificial Neural Network for Load Consumption Forecasting*

Ernesto Carlos Casals Cunill<sup>I\*</sup>, Salvador López Hierrezuelo<sup>II</sup>, Delmar Jaime García<sup>I</sup>, Raciél David Lago Solano<sup>I</sup>

<sup>I</sup>Universidad de Oriente (UO). Santiago de Cuba, Cuba

<sup>II</sup>Empresa Filial Copextel. Santiago de Cuba, Cuba

\*Autor de la correspondencia: [ernesto.casals@uo.edu.cu](mailto:ernesto.casals@uo.edu.cu)

Recibido: 12 de septiembre de 2024    Aprobado: 20 de noviembre de 2024

Este documento posee una [licencia Creative Commons Reconocimiento-NoComercial 4.0 internacional](https://creativecommons.org/licenses/by-nc/4.0/)



### RESUMEN/ ABSTRACT

La Tienda Comercial "El Machetazo" enfrentó dificultades en la predicción de la curva de demanda eléctrica, lo que impactó negativamente la gestión de su sistema solar fotovoltaico y cumplimiento del plan de ahorro energético. Este trabajo tuvo como objetivo desarrollar una herramienta predictiva basada en redes neuronales artificiales para optimizar la gestión energética de la tienda. Para ello, se implementó un modelo de red neuronal tipo Long Short-Term Memory que empleó datos históricos de consumo eléctrico. La metodología incluyó recopilación y preprocesamiento de datos, diseño y entrenamiento del modelo, y la evaluación de su precisión. Los resultados indicaron que el modelo predijo de manera precisa la demanda eléctrica con un error medio absoluto de 0.02, lo que permitió una gestión más eficiente del sistema solar fotovoltaico y contribuyó al cumplimiento del plan energético. Este avance no solo mejoró la eficiencia operativa, sino que promovió el uso sostenible de energía.

**Palabras clave:** predicción de demanda eléctrica, redes neuronales artificiales, eficiencia energética, inteligencia artificial.

*The commercial store "El Machetazo" faced difficulties in predicting the electric demand curve, which negatively impacted the management of its photovoltaic solar system and compliance with the energy-saving plan. This work aimed to develop a predictive tool based on artificial neural networks to optimize the energy management of the store. To achieve this, a Long Short-Term Memory type neural network model was implemented using historical electricity consumption data. The methodology included data collection and preprocessing, model design and training, and evaluation of its accuracy. The results indicated that the model accurately predicted electrical demand with a mean absolute error of 0.02, enabling more efficient management of the photovoltaic solar system and significantly contributing to achieving energy goals. This advancement not only improved operational efficiency but also promoted sustainable use of energy.*

**Keywords:** Load Consumption Forecasting, Artificial Neural Networks (ANN), Energy efficiency, Artificial Intelligence (AI).

### INTRODUCCIÓN

El precio de operación de los sistemas eléctricos de potencia se ve perjudicado por componentes, como el error en la predicción de la demanda eléctrica debido a que, sin una idónea representación de las necesidades futuras de generación, los inconvenientes por exceso de capacidad o insuficiencia de capacidad, tienen la posibilidad de tener precios altos para el sistema. Gracias a esto nace la necesidad de buscar nuevos procedimientos que permitan obtener predicciones de demanda de electricidad con niveles de error mínimos con relación a los niveles reales.

Cómo citar este artículo:

Ernesto Carlos Casals Cunill y otros. Diseño de Red Neuronal Artificial para la Predicción de la Demanda Eléctrica. Ingeniería Energética. 2024. 45 (3), septiembre/diciembre. ISSN 1815-5901.

Sitio de la revista: <https://rie.cujae.edu.cu/index.php/RIE/index>

En los últimos años, el uso de redes neuronales artificiales para la predicción de la demanda eléctrica ha sido un tema de creciente interés debido a la importancia de mejorar la eficiencia energética y la sostenibilidad. La capacidad de predecir de manera precisa la demanda eléctrica no solo ayuda a optimizar los recursos energéticos, sino que también contribuye a la reducción de costos y emisiones en sistemas eléctricos. En [1], propusieron un enfoque para la predicción de demanda de energía a largo plazo mediante el uso de métodos computacionales avanzados. Utilizando técnicas de modelado estadístico, lograron una mejora significativa en la precisión de las predicciones de consumo. Este método tiene la ventaja de ser preciso en escenarios a largo plazo, pero presenta limitaciones cuando se trata de predicciones en tiempo real debido a la complejidad de los datos involucrados.

En [2], implementaron diversas técnicas de predicción para la generación de energía eléctrica en el sector industrial. Su enfoque se basaba en el uso de redes neuronales artificiales para optimizar la producción y el consumo de energía en tiempo real. Este método destacó por su flexibilidad y precisión, aunque puede resultar costoso en términos computacionales. En [3], llevaron a cabo una investigación centrada en la predicción de la demanda eléctrica en sistemas de distribución de energía utilizando redes neuronales. Este enfoque se aplicó en redes de distribución urbana, donde la demanda es altamente variable y difícil de predecir mediante métodos tradicionales. Destacaron cómo las redes neuronales, al ser entrenadas con grandes volúmenes de datos históricos, pueden superar a los métodos de predicción estadística. Sin embargo, el principal desafío identificado fue el tiempo requerido para entrenar las redes neuronales, lo que puede ser un factor limitante cuando se necesitan predicciones rápidas.

En [4], propusieron un modelo basado en redes LSTM para la predicción de la demanda eléctrica a corto plazo, con especial énfasis en su capacidad para gestionar la variabilidad temporal de los datos. El estudio mostró que las redes LSTM, diseñadas para capturar dependencias a largo plazo en las series temporales, mejoran significativamente la precisión de las predicciones frente a modelos tradicionales como ARIMA. Además, se destacó que el uso de LSTM es particularmente adecuado en sistemas donde las variaciones horarias y diarias de la demanda son comunes. No obstante, este enfoque tiene la desventaja de requerir un extenso proceso de ajuste de hiperparámetros para evitar problemas de sobreajuste, lo que puede complicar su implementación en aplicaciones más amplias.

Por otro lado, [5], desarrolló una metodología para la predicción de demanda energética utilizando RNA aplicadas a sistemas de distribución de energía eléctrica. Su estudio incluyó la comparación de diferentes técnicas de aprendizaje profundo, y concluyó que las redes neuronales artificiales eran superiores a otros métodos en términos de precisión y flexibilidad. El estudio también exploró el uso de modelos híbridos, que combinan RNA con métodos estadísticos, lo que permitió mejorar los tiempos de predicción en algunos casos. Aunque el enfoque híbrido presenta ventajas significativas, también implica una mayor complejidad en su implementación y requiere más recursos computacionales. De la revisión bibliográfica realizada, se observa que el uso de redes neuronales LSTM es el método más adecuado para la predicción de la demanda eléctrica, ya que ha demostrado ser efectivo en la captura de patrones complejos en series temporales. En este trabajo, aplicaremos este enfoque con la variante de optimizar su entrenamiento utilizando técnicas de ajuste de hiperparámetros y validación cruzada, lo que permitirá mejorar su precisión y reducir el error en las predicciones.

Se propone la ejecución de este proyecto de control de la demanda eléctrica en los locales de la entidad comercial El Machetazo perteneciente a la empresa filial Copextel, y que sirva a la vez de muestra para su comercialización en otras entidades del territorio. Esta instalación está destinada a la venta de equipamiento informático, electrodomésticos, TV, Refrigeradores, Lavadoras, cajas decodificadoras, así como partes y piezas para el ahorro energético como celdas fotovoltaicas, calentadores de agua y tecnología de redes de datos, telefonía, sistemas tecnológicos de clima y refrigeración. El local en cuestión y que es la base de nuestro Proyecto asegura estos procesos, así como la atención y servicios técnicos de la garantía comercial de dicho equipamiento. Se cuenta con un sistema fotovoltaico instalado de 2.5 kW, conformado por 10 paneles solares de 250 Wp conectados en serie y un inversor SMA de 2.5 kW.

La implementación de las redes neuronales artificiales se requiere debido al problema que existe en la predicción de la demanda eléctrica, el cual afecta el desempeño del flujo de carga, análisis de seguridad, plan de mantenimiento preventivo de los generadores solares y despacho económico, por lo que se ha optado en el desarrollo de un algoritmo, basado en redes neuronales artificiales mediante el software Python, utilizando datos reales de la Tienda Comercial El Machetazo- COPEXTEL SA en Santiago de Cuba, esto permitirá realizar un análisis del comportamiento de la demanda eléctrica horaria con el objetivo de obtener una predicción óptima. Actualmente, la demanda contratada es de 60 kW con un plan de 220 kW/h diarios, con la tarifa M1A, esto quiere decir que la entidad comercial está contratando un servicio sobredimensionado puesto que no lo está consumiendo. Si contrataran un servicio de 30 kW, se reducirían las pérdidas por exceso de demanda contratada, y con la implementación de la herramienta de predicción se puede evitar el incumplimiento del plan establecido. En la figura 1, se observa que en el año 2023 la demanda máxima leída fue de 30 kW.



Fig. 1. Demanda de El Machetazo por año. Fuente: Elaboración Propia

Las redes neuronales artificiales (RNA) son modelos computacionales inspirados en la forma en que funciona el cerebro humano y diseñados para reconocer patrones y aprender de los datos. Su configuración básica incluye un conjunto de neuronas artificiales organizadas en capas: una capa de entrada, una o más capas ocultas y una capa de salida. Cada neurona recibe información, la procesa y envía salida a las neuronas de la siguiente capa. A las conexiones entre neuronas se les asignan pesos y se ajustan durante el proceso de aprendizaje para minimizar los errores de predicción [6]. Matemáticamente, una neurona artificial calcula una suma ponderada de sus entradas y aplica una función de activación como sigmoidea o ReLU (Unidad Lineal Rectificada). Esto introduce no linealidad en el modelo, permitiéndole capturar relaciones complejas.

#### Cálculo de la Entrada y Activación de una Neurona

Cada neurona recibe múltiples entradas, cada una con un peso asociado. La entrada neta  $z$  a una neurona se calcula como la suma ponderada de las entradas  $x_i$  más un sesgo  $b$ . Ver ecuación (1).

$$z = \sum_{(i=1)}^n w_i x_i + b \quad (1)$$

donde:

- $w_i$  es el peso de la  $i$ -ésima entrada,
- $x_i$  es la  $i$ -ésima entrada,
- $b$  es el sesgo,
- $n$  es el número de entradas.

#### Función de Activación

La salida de la neurona  $a$  se determina aplicando una función de activación  $f$  al valor de entrada neta  $z$ . Las funciones de activación comunes incluyen la sigmoide, ReLU y la tangente hiperbólica. Ver ecuaciones (2-4).

Función Sigmoide:

$$a = \sigma(z) = \frac{1}{(1 + e^{(-z)})} \quad (2)$$

#### Función ReLU (Rectified Linear Unit)

$$a = \text{ReLU}(z) = \max(0, z) \quad (3)$$

#### Función Tangente Hiperbólica

$$a = \tanh(z) = \frac{(e^z - e^{(-z)})}{(e^z + e^{(-z)})} \quad (4)$$

### Propagación Hacia Adelante

En una RNA con múltiples capas, las salidas de una capa se utilizan como entradas para la siguiente capa. Para una red con L capas, la salida de la capa l se calcula como se muestra a continuación. Ver ecuaciones (5, 6).

$$z^{(l)} = W^{(l)} a^{(l-1)} + b^{(l)} \quad (5)$$

$$a^{(l)} = f(z^{(l)}) \quad (6)$$

donde:

- $W^{(l)}$  es la matriz de pesos de la capa l,
- $a^{(l-1)}$  es la salida de la capa anterior l-1,
- $b^{(l)}$  es el vector de sesgos de la capa l.

### Función de Pérdida

La función de pérdida L mide la discrepancia entre las predicciones de la red  $\hat{y}$  y los valores verdaderos y. Una función de pérdida común es el error cuadrático medio (MSE), [7]. Ver ecuación (7).

$$L(\hat{y}, y) = \frac{1}{m} \sum_{(i=1)}^m (\hat{y}_i - y_i)^2 \quad (7)$$

donde m es el número de muestras.

### Retropropagación y Actualización de Pesos

Durante el entrenamiento, los pesos se ajustan para minimizar la función de pérdida utilizando el algoritmo de retropropagación y el gradiente descendente. El gradiente de la función de pérdida respecto a cada peso w se calcula y los pesos se actualizan en la dirección opuesta al gradiente. Ver ecuación (8).

$$w \leftarrow w - \eta \frac{\partial L}{\partial w} \quad (8)$$

donde  $\eta$  es la tasa de aprendizaje.

La derivada de la función de pérdida con respecto a los pesos y sesgos se calcula utilizando la regla de la cadena. Ver ecuación (9).

$$\frac{\partial L}{\partial w_i^{(l)}} = \frac{\partial L}{\partial a^{(l)}} \cdot \frac{\partial a^{(l)}}{\partial z^{(l)}} \cdot \frac{\partial z^{(l)}}{\partial w_i^{(l)}} \quad (9)$$

Estas fórmulas matemáticas forman la base del funcionamiento y entrenamiento de las redes neuronales artificiales, permitiendo que estas redes aprendan patrones complejos en los datos y realicen predicciones precisas [8].

## MATERIALES Y MÉTODOS

### Procedimiento para realizar la predicción de la demanda eléctrica mediante RNA

- 1.- Comprender el comportamiento de la demanda en el alimentador, está relacionado a buscar toda la información que permita conocer de mejor manera la operación del alimentador (curvas de la demanda eléctrica, semanal fin de semana y diaria). Como resultado final de este paso permitirá entender la base de datos adquirida.
- 2.- Seleccionar variables de entrada de las bases de datos acorde a la aplicación. En base al conocimiento adquirido en el paso 1, es necesario escoger las variables que más influyan en el comportamiento de la demanda de energía eléctrica.
- 3.- Validación y preprocesamiento de los datos de entrada. La red neuronal artificial aprende y predice en función de los datos históricos de las variables consideradas, si en estas bases de datos existen datos erróneos los resultados de las predicciones no serán las más acertadas, es por ello que previo al ingreso de las bases históricas de las variables al modelo de la red neuronal artificial es necesario eliminar la mayoría de datos erróneos que pudiesen existir.
- 4.- Definir la RNA (número de neuronas de entrada, número de capas ocultas, número de neuronas en las capas ocultas, número de neuronas de salida, número de interconexiones entre las diferentes neuronas, funciones de activación). El paso más importante a llevar a cabo, depende muchas veces de las experiencias del predictor, se hace necesario ir comparando entre diferentes estructuras de red neuronal artificial para encontrar el que mejor se ajuste a la aplicación.

- 5.- Entrenamiento de la RNA (criterios de aprendizaje, número de iteraciones) con un conjunto de datos de entrenamiento. En este paso se asignarán los valores finales, tanto el número de retrasos y el algoritmo de entrenamiento, minimizando el valor del criterio de aprendizaje seleccionado.
- 6.- Verificación de la red neuronal artificial, con un conjunto de datos de prueba. La verificación de la funcionalidad de la red neuronal artificial se la realiza al predecir valores de la demanda eléctrica y compararlas con datos reales.
- 7.- Almacenamiento de los parámetros de la red neuronal artificial una vez verificada. Es necesario almacenar la red neuronal artificial para su uso en el futuro, sin tener la necesidad de repetir los dos últimos pasos mencionados al menos que los responsables lo crean necesario.
- 8.- Implementación de la red neuronal artificial para la predicción de la demanda de energía eléctrica futura mediante la programación del algoritmo para proporcionar resultados [9].

## Desarrollo del Algoritmo de Predicción

### Descripción de los Datos

Conjunto de Datos: Para realizar la predicción de la demanda eléctrica, es crucial disponer de un conjunto de datos adecuado [10]. En este caso, utilizamos un archivo CSV que contiene registros de demanda eléctrica en intervalos de 60 minutos, proveniente de los sensores instalados en el inversor fotovoltaico de la Tienda El Machetazo. Este archivo debe ser leído y cargado en un entorno de programación, como Python, para su posterior análisis y procesamiento. Las variables incluyen demanda eléctrica (W), y la fecha y hora del día [11].

### Selección de Columnas Relevantes

El primer paso en el procesamiento de datos es seleccionar las columnas relevantes. De todo el conjunto de datos disponible, nos centramos en las columnas que contienen las marcas de tiempo y los valores de la demanda eléctrica. Esto nos permite enfocarnos en la información necesaria para el análisis y la predicción, eliminando datos innecesarios que podrían complicar el modelo.

### Visualización de los Datos

La visualización de datos es una etapa crítica en el análisis de series temporales, especialmente en el contexto de la predicción de demanda eléctrica. Mediante gráficos y otras técnicas visuales, es posible obtener una comprensión más profunda y rápida de las características y patrones presentes en los datos. La visualización ayuda a detectar tendencias, patrones estacionales, anomalías y posibles puntos de cambio que pueden influir en la precisión de los modelos predictivos [12].

### Gráfico de la Demanda a lo Largo del Tiempo

Para comprender la evolución de la demanda eléctrica, se genera un gráfico de línea que muestra los valores de la demanda a lo largo del tiempo (Ver figura 2). Este tipo de gráfico es esencial porque: Identificación de Tendencias: Permite observar si hay una tendencia general al alza o a la baja en la demanda eléctrica a lo largo del periodo estudiado. Detección de Patrones Estacionales: Los patrones repetitivos en intervalos regulares, como variaciones diarias, semanales o estacionales, se hacen evidentes. Reconocimiento de Anomalías: Cualquier valor atípico o anomalía que pueda indicar errores en los datos o eventos extraordinarios se puede detectar fácilmente [13].

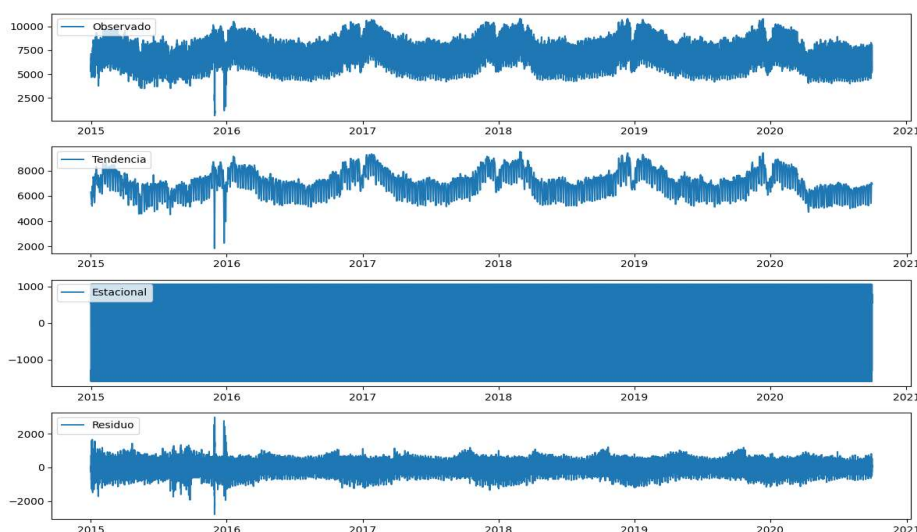


Fig. 2. Descomposición Estacional del Conjunto de Datos. Fuente: Elaboración Propia

## Preprocesamiento de Datos

El preprocesamiento de datos es una etapa crítica en cualquier proyecto de análisis y modelado predictivo. Consiste en la transformación y preparación de los datos en un formato adecuado para ser utilizado por los algoritmos de machine learning. Este proceso incluye la limpieza de datos, la gestión de valores faltantes, la normalización de variables y la creación de nuevas características temporales [14]. A continuación, se detallan los fundamentos teóricos y los pasos específicos realizados en este estudio y su explicación.

### Conversión y Manipulación de Timestamps

Uno de los primeros pasos en el preprocesamiento de datos es asegurar que las marcas de tiempo (timestamps) estén en un formato adecuado para el análisis de series temporales. En este caso, los datos vienen con una columna de timestamps en formato UTC (Coordinated Universal Time).

- **Conversión a Fecha y Hora:** La columna `utc_timestamp` se convierte al formato `datetime` usando la función `pd.to_datetime`, especificando el formato ISO 8601, que es un estándar internacional para representar fechas y horas.
- **Establecimiento del Índice de Tiempo:** La columna de tiempo se establece como el índice del DataFrame con `df.set_index("time")`. Esto es crucial para las series temporales, ya que permite acceder y manipular los datos basándose en el tiempo.
- **Ordenación del DataFrame:** Se ordena el DataFrame por el índice de tiempo con `df.sort_index(inplace=True)`, garantizando que los datos estén en secuencia cronológica.
- **Eliminación de la Columna Original:** La columna original `utc_timestamp` se elimina con `df.drop(columns=['utc_timestamp'], inplace=True)` para evitar redundancias y reducir el espacio de memoria.

### Gestión de Valores Faltantes

Los datos reales a menudo contienen valores faltantes, que pueden sesgar los resultados si no se manejan adecuadamente. La gestión de estos valores es esencial para mantener la integridad del análisis [15].

- **Identificación de Valores Faltantes:** Se revisan todas las columnas para identificar la cantidad de valores faltantes (NaNs) en cada una. Esto se hace iterando sobre las columnas y utilizando `isna().sum()`, lo que proporciona un conteo de los valores faltantes en cada columna.
- **Diferencias Temporales:** Adicionalmente, se calcula la diferencia entre los tiempos consecutivos en segundos para verificar la consistencia temporal de los datos.
- **Verificación de Consistencia Temporal:** La diferencia entre tiempos consecutivos (`df.index.to_series().diff().dt.total_seconds()`) se analiza para asegurarse de que los datos estén uniformemente espaciados. Esto ayuda a identificar cualquier irregularidad en la frecuencia de los datos, lo cual es crucial para el correcto funcionamiento de modelos de series temporales.

### Imputación de Valores Faltantes

Para manejar los valores faltantes, se utiliza la interpolación, una técnica que estima los valores faltantes basándose en los valores conocidos más cercanos.

Se aplica la interpolación utilizando el método 'bfill' (backwardfill), que llena los valores faltantes con el siguiente valor no nulo en la columna. Esta técnica es útil cuando se asume que los datos cercanos en el tiempo son similares.

El método más conocido y el que recomendamos utilizar es la Interpolación Lineal, una técnica simple donde se estima el valor faltante asumiendo que el cambio entre dos puntos conocidos es lineal. Si tenemos dos puntos  $(x_1, y_1)$  y  $(x_2, y_2)$ , y queremos estimar el valor  $y$  en  $x$  que está entre  $x_1$  y  $x_2$ , la fórmula es la siguiente. Ver ecuación (10).

$$y = y_1 + (y_2 - y_1) \cdot \frac{x - x_1}{x_2 - x_1} \quad (10)$$

### División del Conjunto de Datos

La división del conjunto de datos en subconjuntos de entrenamiento, validación y prueba es un paso fundamental en la construcción de modelos predictivos robustos y generalizables. Esta división permite evaluar el rendimiento del modelo y ajustar sus hiperparámetros sin incurrir en sobreajuste (overfitting). Cada uno de estos subconjuntos tiene un propósito específico que contribuye a la efectividad y confiabilidad del modelo [16].

### Overfitting y Underfitting

Overfitting o Sobreajuste ocurre cuando un modelo aprende demasiado bien los detalles y el ruido del conjunto de datos de entrenamiento, de manera que su desempeño es muy bueno en esos datos, pero muy malo en datos nuevos o de validación. Esto sucede porque el modelo es demasiado complejo y se ajusta excesivamente a las peculiaridades del conjunto de datos de entrenamiento, capturando también el ruido.

### Conjunto de Entrenamiento

El conjunto de entrenamiento se utiliza para ajustar los parámetros del modelo. Es crucial que este subconjunto contenga una cantidad suficiente de datos para que el modelo pueda aprender las características y patrones subyacentes de la serie temporal. La elección de un tamaño adecuado para el conjunto de entrenamiento asegura que el modelo tenga una buena base de datos para aprender, lo que se traduce en una mayor capacidad predictiva.

### Conjunto de Validación

El conjunto de validación se emplea para ajustar los hiperparámetros del modelo y evaluar su rendimiento durante el proceso de entrenamiento. La validación cruzada en este conjunto permite detectar el sobreajuste, asegurando que el modelo no solo se adapte bien a los datos de entrenamiento, sino que también tenga un buen desempeño en datos no vistos. La separación clara entre los conjuntos de entrenamiento y validación es esencial para la afinación precisa del modelo, lo que a su vez mejora la generalización del mismo.

### Conjunto de Prueba

El conjunto de prueba se utiliza exclusivamente para evaluar el rendimiento final del modelo después de que todos los ajustes y entrenamientos se han completado. Este conjunto proporciona una medida imparcial del desempeño del modelo en datos completamente nuevos, lo que simula su comportamiento en escenarios del mundo real. Evaluar el modelo en el conjunto de prueba asegura que los resultados reportados reflejen de manera realista su capacidad predictiva y robustez.

### Preparación del Dataset para LSTM

Uno de los primeros pasos en la preparación del dataset es transformar los datos en un formato supervisado. Esto implica crear pares de entrada y salida que el modelo LSTM puede utilizar para el aprendizaje. En el contexto de series temporales, esto se traduce en tomar una ventana de tiempo pasada (input) y asociarla con el valor futuro que se desea predecir (output) [17]. Este proceso se realiza de la siguiente manera: Selección de la Ventana de Entrada y Salida: Se elige una longitud fija de la ventana de tiempo de entrada (por ejemplo, 24 horas) y una longitud fija para la predicción (por ejemplo, 4 horas). Generación de Pares de Entrenamiento: Para cada punto en el tiempo, se crean pares de datos tomando los valores de la serie temporal correspondientes a la ventana de entrada y asociándolos con los valores futuros para la ventana de salida. Este proceso se repite para toda la serie temporal, generando un conjunto de ejemplos de entrenamiento. Este formato supervisado permite que la LSTM aprenda las relaciones y patrones en los datos temporales, mejorando su capacidad de hacer predicciones precisas.

### Normalización de los Datos

La normalización de los datos es un paso crítico en el preprocesamiento de datos para redes neuronales, incluyendo las LSTM. La normalización consiste en escalar los datos a un rango específico, comúnmente entre -1 y 1, por varias razones importantes: Estabilidad Numérica: Los algoritmos de optimización utilizados en el entrenamiento de redes neuronales funcionan mejor cuando los datos están normalizados. Ayuda a prevenir problemas numéricos y mejora la estabilidad del entrenamiento. Convergencia Rápida: La normalización de los datos puede acelerar el proceso de convergencia durante el entrenamiento. Los gradientes de las funciones de costo son más consistentes y permiten que los optimizadores encuentren mejores soluciones de manera más eficiente. Comparabilidad de las Características: Cuando se tienen múltiples características (inputs) con diferentes escalas, la normalización las lleva a un rango común, haciendo que el modelo trate todas las características de manera equitativa [18].

El proceso de normalización se realiza de la siguiente manera:

Cálculo de Parámetros de Normalización: Se calcula el mínimo y máximo (o media y desviación estándar) de los datos de entrenamiento.

La Media de una característica  $x$  se calcula como se muestra en la ecuación (11).

$$\mu_x = \frac{1}{N} \sum_{(i=1)}^N x_i \quad (11)$$

Y la Desviación Estándar de una característica  $x$  se calcula como se ve en la ecuación (12).

$$\sigma_x = \sqrt{\frac{1}{N} \sum_{(i=1)}^N (x_i - \mu_x)^2} \quad (12)$$

Donde:

- $\mu_x$  es la media de la característica  $x$ .
- $N$  es el número total de muestras.
- $x_i$  es el valor de la característica  $x$  en la muestra  $i$ .
- $\sigma_x$  es la desviación estándar de la característica  $x$ .

### Arquitectura del Modelo

El proceso de diseño de un modelo LSTM incluye varias decisiones clave sobre su arquitectura:

1. Número de Capas LSTM: La profundidad de la red puede variar dependiendo de la complejidad del problema. En este caso, se suele empezar con una sola capa LSTM y, si es necesario, agregar más capas para capturar características más complejas de la serie temporal.
2. Unidades en la Capa LSTM: Las unidades LSTM son neuronas especializadas que permiten la memorización de patrones a largo plazo. Se debe elegir un número adecuado de unidades para cada capa. Esto se puede ajustar posteriormente mediante técnicas de afinación.
3. Capas Densas Posteriores: Después de las capas LSTM, es común añadir capas densas (fullyconnectedlayers) que ayudan a mapear las representaciones aprendidas por las LSTM a las predicciones finales.
4. Capa de Salida: La capa de salida del modelo debe tener tantas neuronas como el número de pasos de tiempo que se desean predecir. En este caso, una única capa densa con una activación lineal es suficiente.

### Compilación del Modelo

Una vez definida la arquitectura, se procede a compilar el modelo. La compilación implica elegir una función de pérdida adecuada y un optimizador:

La función de pérdida utilizada, el error cuadrático medio (RMSE), es una métrica común para evaluar el rendimiento en problemas de regresión. Minimizar el RMSE durante el entrenamiento ayuda a asegurar que las predicciones del modelo sean lo más precisas posibles. El optimizador RMSprop se seleccionó por su efectividad en entrenar modelos LSTM [19].

## RESULTADOS Y DISCUSIÓN

### Resultados Después del Primer Entrenamiento

En esta sección se presentan los resultados obtenidos del modelo de predicción de demanda eléctrica utilizando redes LSTM. Se evaluó el rendimiento del modelo en los conjuntos de entrenamiento, validación y prueba para garantizar la generalización y precisión del mismo. El proceso de entrenamiento del modelo se llevó a cabo durante 80 épocas. A lo largo del entrenamiento, se monitoreó la función de pérdida (RMSE) tanto en el conjunto de entrenamiento como en el de validación. Los resultados del entrenamiento final se resumen a continuación (Ver figura 3 y tabla 1).

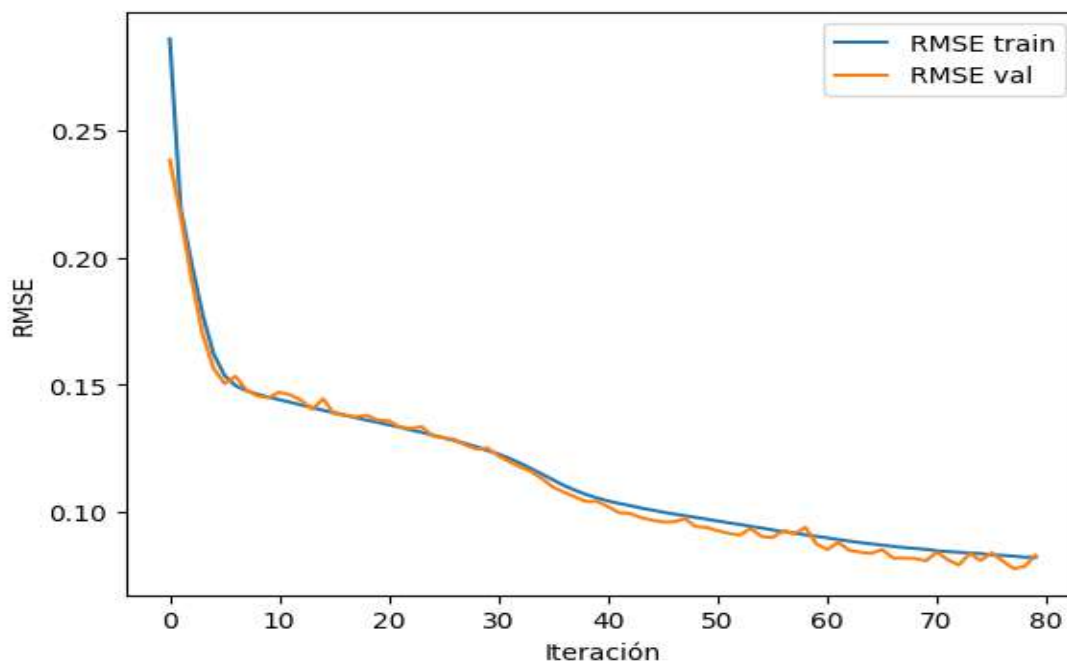


Fig. ¡Error! Utilice la ficha Inicio para aplicar 0 al texto que desea que aparezca aquí. **Tabla 1.** Valor del Error en los Conjuntos de Entrenamiento y Validación sin ajustar la Red. Fuente: Elaboración propia

Épocas	80/80
Pérdida en el conjunto de entrenamiento (loss)	0.0822
Pérdida en el conjunto de validación (val_loss)	0.0830
Tiempo por época	668ms/epoch

### Evaluación del Modelo

Para evaluar el rendimiento del modelo, se calcularon los valores del RMSE (Root Mean Squared Error) en los conjuntos de entrenamiento, validación y prueba. Los resultados son los siguientes (Ver tabla 2):

**Tabla 2.** Valor del RMSE después de entrenar la Red. Fuente: Elaboración propia

RMSE en el conjunto de entrenamiento	0.081
RMSE en el conjunto de validación	0.079
RMSE en el conjunto de prueba	0.074

### Afinación de Hiperparámetros

La afinación de hiperparámetros es un proceso crítico para mejorar el rendimiento del modelo. Involucra ajustar parámetros que no se aprenden directamente del proceso de entrenamiento, como el número de unidades LSTM, la tasa de aprendizaje, y el número de neuronas en las capas densas. Este proceso se puede realizar mediante técnicas de búsqueda automatizada como el RandomSearch o el GridSearch[20].

1. Definición de Espacios de Búsqueda: Se definen los rangos de valores posibles para cada hiperparámetro. Por ejemplo, el número de unidades LSTM puede variar entre 50 y 200, y la tasa de aprendizaje puede ser 0.1, 0.01, o 0.001.
2. Ejecución de Búsqueda: Se utiliza una herramienta de afinación como KerasTuner para probar diferentes combinaciones de hiperparámetros. Este proceso puede implicar entrenar y validar el modelo múltiples veces con diferentes configuraciones.
3. Selección del Mejor Modelo: La configuración de hiperparámetros que produce el menor error en el conjunto de validación se selecciona como la mejor. Esta configuración se utiliza para entrenar el modelo final.

### Evaluación y Validación

Una vez afinado el modelo, se evalúa su rendimiento en el conjunto de validación y prueba:

Entrenamiento y Validación: El modelo se entrena utilizando el conjunto de datos de entrenamiento y se valida continuamente en el conjunto de validación. Se monitorean métricas de rendimiento como el RMSE, MSE, MAPE y  $R^2$  para asegurar que el modelo no está sobreajustando.

Test Final: Finalmente, se evalúa el modelo en el conjunto de prueba para obtener una medida imparcial de su rendimiento. El error en el conjunto de prueba debe ser comparable al error en el conjunto de validación si el modelo está correctamente afinado.

### Resultados del Entrenamiento Después de la Afinación de Hiperparámetros

El modelo se entrenó durante 10 épocas de 1260 iteraciones, y los resultados del entrenamiento y la validación se presentan en la tabla 3.

**Tabla 3.** Resultados del Entrenamiento del Modelo Afinado. Fuente: Elaboración Propia

Épocas	Conjunto de Entrenamiento		Conjunto de Validación	
	Pérdida	Error Medio Absoluto	Pérdida	Error Medio Absoluto
1	0.0032	0.0384	0.0024	0.0339
2	0.0025	0.0334	0.0014	0.0278
3	0.0023	0.0320	0.0013	0.0260
4	0.0022	0.0314	0.0015	0.0291
5	0.0021	0.0307	0.0013	0.0266
6	0.0021	0.0304	0.0018	0.0294
7	0.0021	0.0300	0.0017	0.0314
8	0.0020	0.0296	0.0030	0.0404
9	0.0020	0.0296	0.0013	0.0257
10	0.0020	0.0294	0.0012	0.0251

### Evaluación del Modelo

El desempeño del modelo se evaluó utilizando varias métricas en los conjuntos de entrenamiento, validación y prueba. Los resultados se resumen en la tabla 4.

**Tabla 4.** Comparativo de Desempeños. Fuente: Elaboración propia

Métrica	Valor
MSE	31823.909657
RMSE	178.392572
MAPE	0.020613
$R^2$	0.975252

Luego se compararon las curvas de datos históricos con las predicciones del modelo, y se calculó el porcentaje de error para cada valor predicho. (Ver las figuras 4 y 5).

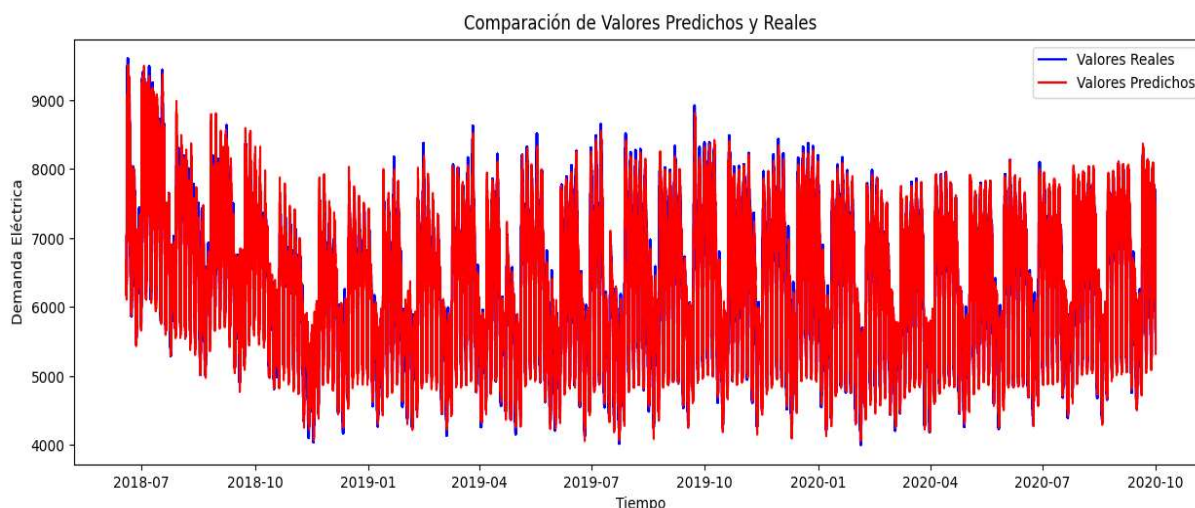


Fig. 4. Comparación de Valores Predichos y Reales en función del Tiempo. Fuente: Elaboración Propia

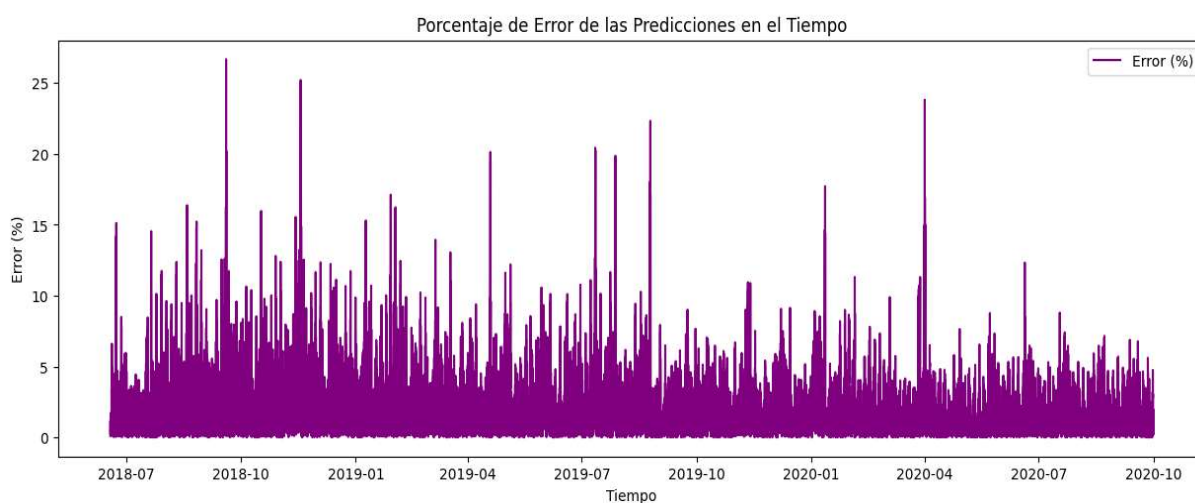


Fig. 5. Porcentaje de Error de las Predicciones en el Tiempo. Fuente: Elaboración Propia

### Interpretación de las Métricas

- MSE (Mean Squared Error): El MSE es una medida de la magnitud promedio de los errores al cuadrado. Un MSE de 31823.909657 indica que, en promedio, los errores cuadráticos entre las predicciones y los valores reales son relativamente bajos, lo que sugiere que el modelo tiene una buena capacidad predictiva.
- RMSE (Root Mean Squared Error): El RMSE es la raíz cuadrada del MSE y proporciona una medida de error en las mismas unidades que los datos originales. Un RMSE de 178.392572 es relativamente bajo, lo que confirma que el modelo tiene un buen ajuste a los datos.
- MAPE (Mean Absolute Percentage Error): El MAPE mide el error porcentual medio absoluto entre las predicciones y los valores reales. Un MAPE de 0.020613 indica que, en promedio, las predicciones del modelo están a un 2.0613% de los valores reales, lo cual es bastante preciso para aplicaciones de predicción de demanda eléctrica.
- R<sup>2</sup> (Coeficiente de Determinación): El R<sup>2</sup> es una medida de cuán bien las predicciones del modelo explican la variabilidad de los datos. Un valor de R<sup>2</sup> de 0.975252 sugiere que el modelo explica el 97.52% de la variabilidad en los datos de demanda eléctrica, lo que es un excelente indicador de la capacidad predictiva del modelo.

### CONCLUSIONES

Se desarrolló un algoritmo basado en una red neuronal recurrente, concretamente LSTM, que mostró un rendimiento superior en comparación con los métodos tradicionales como ARIMA y la regresión lineal. Los resultados obtenidos permiten concluir que el modelo desarrollado no sólo cumple con los objetivos planteados, sino que también supera las expectativas en términos de reducción de errores de pronóstico, con un error absoluto promedio de 0.02 y adaptabilidad a diferentes patrones de consumo de energía. Los resultados obtenidos demuestran la efectividad de las redes neuronales artificiales en la predicción de sistemas eléctricos, fortaleciendo así el método de diseño e implementación de RNA predictivas en entidades con alto consumo de energía.

## REFERENCIAS

- [1]. Althaus, G. W., *et al.* “Predicción de la demanda de la energía eléctrica a largo plazo: un reto en ingeniería computacional”. DYNA. 2015, vol. 90, n. 6, p. 582-584. ISSN 1989-1490. Disponible en: <https://revista-dyna.com/index.php/DYNA/article/view/7027/6699>
- [2]. Peña, A. C., *et al.* “Implementación de las técnicas de predicción en la generación de energía eléctrica en el sector industrial”. Revista de Investigación en Tecnologías de la Información: RITI. 2020, vol. 8, n. 15, p. 16-26. ISSN 2387-0893. Disponible en: <https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=https://dialnet.unirioja.es/descarga/articulo/7272003.pdf&ved=2ahUKewj5xJLe0O2IAxXTVTABHXhcMwsQFnoECBMQAO&usg=AOvVaw0kFYZwnlroEigkSzfN1YKV>
- [3]. Lozano, S., *et al.* “Predicción de demanda de energía eléctrica mediante redes neuronales artificiales”. Revista Ibérica de Sistemas e Tecnologías de Informação. 2020, n. 25, p. 505-519. ISSN 1646-9895. Disponible en: <https://www.proquest.com/openview/1956dc09ca8d362d9ed5513c6a9eb9f7/1?pq-origsite=gscholar&cbl=1006393>
- [4]. Félix, F., *et al.* “Load Prediction in Electrical Networks using LSTM”. In 2021 Mexican International Conference on Computer Science (ENC). 2021, p. 1-6. ISBN 978-1-6654-2612-1. Disponible en: <https://ieeexplore.ieee.org/document/9534794>
- [5]. Coba Louzado, A. “Metodología para pronóstico de la demanda de energía eléctrica”. Tesis de maestría. Universidad de Moa, Dr. Antonio Nuñez Jiménez, Holguín, Cuba. 2022. Disponible en: <https://ninive.ismm.edu.cu/bitstream/handle/123456789/3931/alexeym.pdf?sequence=1&isAllowed=y>
- [6]. Zhong, B. “Deep learning integration optimization of electric energy load forecasting and market price based on the ANN–LSTM–transformer method”. Frontiers in Energy Research. 2023, vol. 11, p. 1-17. ISSN 2296-598X. Disponible en: <https://www.frontiersin.org/journals/energy-research/articles/10.3389/fenrg.2023.1292204/full>
- [7]. Vanting, B., *et al.* “A scoping review of deep neural networks for electric load forecasting”. Energy Informatics. 2021, vol. 4, n. 49. ISSN 2520-8942. Disponible en: <https://link.springer.com/article/10.1186/s42162-021-00148-6>
- [8]. Janiesch, C., *et al.* “Machine learning and deep learning”. Electronic Markets. 2021, vol. 31, n. 3, p. 685-695. ISSN 1422-8890. Disponible en: <https://link.springer.com/article/10.1007/s12525-021-00475-2>
- [9]. Cabrera, C. H. L., *et al.* “Predicción de demanda eléctrica utilizando redes neuronales artificiales para un sistema de distribución de energía eléctrica”. Ciencia Latina Revista Científica Multidisciplinar. 2022, vol. 6, n. 3, p. 3871-3894. ISSN 2707-2215. Disponible en: <https://ciencialatina.org/index.php/cienciala/article/view/2502>
- [10]. Pramod, A., *et al.* “Machine learning and deep learning: Open issues and future research directions for the next 10 years”. Computational analysis and deep learning for medical care: Principles, methods, and applications. 2021, cap. 2, p. 463-490. ISBN 9781-1197-8575-0. Disponible en: <https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119785750.ch18>
- [11]. Rojas, E. M. “Machine Learning: análisis de lenguajes de programación y herramientas para desarrollo”. Revista Ibérica de Sistemas e Tecnologías de Informação. 2020, n. 28, p. 586-599. ISSN 1646-9895. Disponible en: <https://www.proquest.com/openview/c7e24c997199215aa26a39107dd2fe98/1?pq-origsite=gscholar&cbl=1006393>
- [12]. Alfaro, A. D. J., & Ospina, J. V. D. “Revisión sistemática de literatura: Técnicas de aprendizaje automático (machine learning)”. Cuaderno activa. 2021, vol. 13, n. 1, p. 113-121. ISSN 2619-5232. Disponible en: <https://ojs.tdea.edu.co/index.php/cuadernoactiva/article/view/849>
- [13]. Tealab, A. “Time series forecasting using artificial neural networks methodologies: A systematic review”. Future Computing and Informatics Journal. 2018, vol. 3, n. 2, p. 334-340. ISSN 2314-7296. Disponible en: <https://www.sciencedirect.com/science/article/pii/S2314728817300715>
- [14]. Mughees, N., *et al.* “Deep sequence to sequence Bi-LSTM neural networks for day-ahead peak load forecasting”. Expert Systems with Applications. 2021, vol. 175. ISSN 0957-4174. Disponible en: <https://www.sciencedirect.com/science/article/abs/pii/S0957417421002852>
- [15]. Yugcha, C., *et al.* “Previsión del consumo eléctrico en el cantón Salcedo mediante técnicas de aprendizaje automático”. Revista Odigos. Mayo 2024, vol. 5, n. 1, p. 9-24. ISSN 2697 – 3405. Disponible en: <https://revista.uisrael.edu.ec/index.php/ro/article/view/1134>
- [16]. Martínez, A. A. “Predicción de demanda y producción de energía eléctrica mediante redes neuronales y validación de los resultados mediante ensayos realizados en el laboratorio de recursos energéticos distribuidos de la UPV”. Tesis de maestría. Universidad de Valencia, España. 2021. Disponible en: [https://www.europeana.eu/en/item/355/https\\_hispana\\_mcu\\_es\\_lod\\_oai\\_riunet\\_upv\\_es\\_10251\\_164734\\_ent0](https://www.europeana.eu/en/item/355/https_hispana_mcu_es_lod_oai_riunet_upv_es_10251_164734_ent0)

- [17]. Forootan, M., *et al.* “Machine learning and deep learning in energy systems: A review”. Sustainability. 2022, vol. 14, n. 8. ISSN 2071-1050. Disponible en: <https://www.mdpi.com/2071-1050/14/8/4832>
- [18]. Nti, I. K., *et al.* “Electricity load forecasting: a systematic review”. Journal of Electrical Systems and Information Technology. 2020, vol. 7, n. 13, p. 1-19. ISSN 2314-7172. Disponible en: <https://jesit.springeropen.com/articles/10.1186/s43067-020-00021-8>
- [19]. Azeem, A., *et al.* “Electrical load forecasting models for different generation modalities: a review”. IEEE Access. 2021, vol. 9, p. 142239-142263. ISSN 2169-3536. Disponible en: <https://ieeexplore.ieee.org/document/9576703>
- [20]. Klyuev, R. V., *et al.* “Methods of forecasting electric energy consumption: A literature review”. Energies. 2022, vol. 15, n. 23, 8919. ISSN 1996-1073. Disponible en: <https://www.mdpi.com/1996-1073/15/23/8919>

### CONFLICTO DE INTERESES

Los autores declaran que no existe conflicto de intereses.

### CONTRIBUCIÓN DE LOS AUTORES

**Ernesto Carlos Casals Cunill:** <https://orcid.org/0009-0008-3617-4535>

Participó en el diseño de la investigación, diseño del modelo, la simulación, el procesamiento de los datos y la redacción del manuscrito, la revisión crítica de su contenido y en la aprobación final.

**Salvador López Hierrezuelo:** <https://orcid.org/0009-0005-2295-6358>

Participó en el diseño de la investigación, la redacción del manuscrito y la adquisición de los datos.

**Delmar Jaime García:** <https://orcid.org/0000-0002-8103-8523>

Participó en el diseño de la investigación y la redacción del manuscrito, la revisión crítica de su contenido y en la aprobación final.

**Raciel David Lago Solano:** <https://orcid.org/0009-0001-7901-5429>

Participó en el diseño del modelo y la simulación, la revisión crítica de su contenido y en la aprobación final.