

	GESTIÓN DE SERVICIOS ACADÉMICOS Y BIBLIOTECARIOS		CÓDIGO	FO-GS-15
	ESQUEMA HOJA DE RESUMEN		VERSIÓN	02
			FECHA	03/04/2017
			PÁGINA	1 de 1
ELABORÓ	REVISÓ	APROBÓ		
Jefe División de Biblioteca	Equipo Operativo de Calidad	Líder de Calidad		

**RESUMEN TRABAJO DE
GRADO**

AUTOR(ES): NOMBRES Y APELLIDOS COMPLETOS

NOMBRE(S): JORMAN HERNANDO APELLIDOS: GALVIS PLATA

FACULTAD: INGENIERIA

PLAN DE ESTUDIOS: INGENIERIA ELECTROMECHANICA

DIRECTOR:

NOMBRE(S): GLORIA ESMERALDA APELLIDOS: SANDOVAL MARTINEZ

NOMBRE(S): DARWIN ORLANDO APELLIDOS: CARDOZO SARMIENTO

TÍTULO DEL TRABAJO (TESIS): MODELO PREDICTIVO DE CONSUMOS DE ENERGÍA ELÉCTRICA APLICANDO REDES NEURONALES ARTIFICIALES

En el presente proyecto de grado, se representa la aplicación de las redes neuronales artificiales (RNA) para el modelado de una variable de gran importancia en el análisis de eficiencia energética: el consumo total de potencia activa, así mismo su predicción, informe y grafica a través de una interfaz. Por una parte, se ha propuesto dos modelos de RNA para la predicción programadas en *Matlab*, la primera arquitectura es basada en el aprendizaje automático, la red retroalimentada de dos capas y la segunda arquitectura basada en el aprendizaje profundo, la red de memoria a corto y largo plazo. Por otra parte, el diseño de una interfaz capaz recopilar datos fiables de consumo total de potencia activa, enviar esta información a la RNA seleccionada, ilustrar sus predicciones, y asimismo su precisión comparada con datos reales.

PALABRAS CLAVES: MÁXIMO 5

REDES NEURONALES ARTIFICIALES, MODELOS PREDICTIVOS, CONSUMO DE ENERGIA ELECTRICA, MACHINE LEARNING, DEEP LEARNING

CARACTERISTICAS:

PÁGINAS: 87 PLANOS: ILUSTRACIONES: 19 CD ROOM:

MODELO PREDICTIVO DE CONSUMOS DE ENERGÍA ELÉCTRICA APLICANDO
REDES NEURONALES ARTIFICIALES

JORMAN HERNANDO GALVIS PLATA

UNIVERSIDAD FRANCISCO DE PAULA SANTANDER

FACULTAD DE INGENIERÍA

PLAN DE ESTUDIOS DE INGENIERÍA ELECTROMECAÁNICA

SAN JOSÉ DE CÚCUTA

2022

MODELO PREDICTIVO DE CONSUMOS DE ENERGIA ELECTRICA APLICANDO
REDES NEURONALES ARTIFICIALES

Presentado por:

JORMAN HERNANDO GALVIS PLATA

Proyecto de grado presentado como requisito para optar el título de Ingeniero Electromecánico
con modalidad de proyecto de investigación.

Director:

MsC. Gloria Esmeralda Sandoval Martínez

Codirector:

MsC. Darwin Orlando Cardozo Sarmiento

UNIVERSIDAD FRANCISCO DE PAULA SANTANDER

FACULTAD DE INGENIERÍA

PLAN DE ESTUDIOS DE INGENIERÍA ELECTROMECAÁNICA

SAN JOSÉ DE CÚCUTA

2022

**ACTA DE SUSTENTACIÓN PROYECTO DE GRADO
MODALIDAD INVESTIGATIVO**

FECHA: 15 de septiembre de 2022

HORA: 09:00 a.m.

LUGAR: SC 302 Ufps

PLAN DE ESTUDIOS: INGENIERÍA ELECTROMECAÁNICA

TÍTULO DEL TRABAJO DE GRADO: "MODELO PREDICTIVO DE CONSUMOS DE ENERGÍA ELÉCTRICA APLICANDO REDES NEURONALES ARTIFICIALES"

JURADOS Phd: JULIAN FERREIRA JAIMES
Phd: FRANCISCO ERNESTO MORENO GARCIA

DIRECTOR: Mg: GLORIA ESMERALDA SANDOVAL MARTÍNEZ
Codirector: Mg: DARWIN ORLANDO CARDOZO SARMIENTO

APROBADA

NOMBRE DEL ESTUDIANTE:	CÓDIGO	CALIFICACION
JORMAN HERNANDO GALVIS PLATA	1091178	4.4

FIRMA DE LOS JURADOS:



VOBO. COORDINADOR COMITÉ CURRICULAR

Margarita Ch.

Dedicatoria

Este proyecto de grado es dedicado y agradecido primeramente a Dios por esta oportunidad de culminación de mis estudios de pregrado, a mi familia, que me apoyo en el recorrido académico y a mis seres queridos que estuvieron conmigo en todo momento, resaltando a Zenaida Plata, a Jorge Galvis, a Yulieth Carreño, y a Ismael Montero, quienes me dieron toda su ayuda y motivación.

Tabla de Contenido

Resumen	13
1. Introducción	14
2. Objetivos.	16
2.1. Objetivo General.	16
2.2. Objetivos Específicos.	16
3. Estado del Arte	17
4. Desarrollo del Modelo Basado en Redes Neuronales Artificiales y la Interfaz en Matlab.	22
4.1. Obtención de la Base de Datos.	22
4.2. Caracterización de las Variables de Entrada y Salida (Inputs/Targets).	23
4.3. Identificación de las Propiedades de las Redes neuronales Artificiales.	27
4.3.1. Modelos Neuronales y Algoritmos de Aprendizaje Seleccionados.	27
4.3.2. Criterio de Parada.	31
4.3.3. Implementación de las Redes Neuronales con Matlab	32
4.3.4. Modelos de Redes Neuronales con Neural Network Toolbox.	33
4.3.5. Modelos de Redes Neuronales con Matlab Edit.	36
4.3.6. Diseño la Interfaz Gráfica de Predicción.	39
4.3.7. Indicador de Calidad del Comportamiento de las Redes.	42
5. Análisis de Resultados.	43

5.1. Selección del Modelo Final de la RNA.	43
5.2. Demostración de la Aplicación Final de la Interfaz de Predicción de Consumos de Potencia Activa por Medio de Redes Neuronales	46
Conclusiones	48
Recomendaciones	50
Trabajos Futuros	51
Referencias	52
Anexos	60

Lista De Figuras

Figura 1. Diagrama de Flujo sobre el desarrollo de la RNA y su interfaz.	22
Figura 2. Muestra de la base de datos implementada.	23
Figura 3. Grafica de los Datos de Entrada.	24
Figura 4. Demostración del arreglo de datos.	26
Figura 5. Esquema de la arquitectura de las Redes Neuronales Artificiales del tipo Prealimentada.	29
Figura 6. Esquema de la arquitectura de las Redes Neuronales Artificiales del tipo Memoria a Corto y Largo plazo.	30
Figura 7. Inicio del GUI nnstart y nftool.	33
Figura 8. Selección del conjunto de variables (izquierda) y su distribución (derecha).	34
Figura 9. Ejemplo de la modificación de la cantidad de neuronas en la capa oculta.	35
Figura 10. Selección del algoritmo de optimización de Levenberg-Marquardt (izquierda) y ejemplo del registro de entrenamiento (derecha).	35
Figura 11. Códigos de parametrización de la RNA y del entrenamiento.	37
Figura 12. Ejemplo de la ventana del progreso de entrenamiento.	38
Figura 13. Código de iniciación de predicción.	38
Figura 14. Desarrollo del aplicativo en el App Design de Matlab.	40
Figura 15. Ejemplo de la distribución de datos para su inclusión en el aplicativo	40
Figura 16. Diseño final del Aplicativo de Predicción de Consumos de Potencia Activa por Medio de Redes Neuronales.	42
Figura 17. Graficas de los resultados devueltos por Neural Network Toolbox sobre la RNA.	43
Figura 18. Graficas requerida en Matlab Edit sobre la RNA.	44

Lista De Tablas

Tabla 1. Propiedades seleccionadas para la RNA en el aprendizaje automático.	36
Tabla 2. Propiedades seleccionadas para la RNA en el aprendizaje profundo.	39
Tabla 3. Exposición de los resultados de las RNA diseñadas.	45

Lista de Anexos

Anexo 1. Graficas de los Resultados Obtenidos por Neural Network Toolbox Sobre la RNA de 40 a 100 Neuronas en la Capa Oculta.	60
Anexo 2. Ventana del progreso de entrenamiento de la RNA con Arquitectura LSTM con 150 Neuronas Ocultas y 150 a 300 Épocas Máximas.	64
Anexo 3. Ventana del progreso de entrenamiento de la RNA con Arquitectura LSTM con 200 Neuronas Ocultas y 150 a 300 Épocas Máximas.	66
Anexo 4. Ventana del progreso de entrenamiento de la RNA con Arquitectura LSTM con 250 Neuronas Ocultas y 150 a 300 Épocas Máximas.	68
Anexo 5. Ventana del progreso de entrenamiento de la RNA con Arquitectura LSTM con 300 Neuronas Ocultas y 150 a 300 Épocas Máximas.	70
Anexo 6. Graficas de Resultados de la RNA con Arquitectura LSTM con 150 Neuronas Ocultas y 150 a 300 Épocas Máximas.	72
Anexo 7. Graficas de Resultados de la RNA con Arquitectura LSTM con 200 Neuronas Ocultas y 150 a 300 Épocas Máximas.	74
Anexo 8. Graficas de Resultados de la RNA con Arquitectura LSTM con 250 Neuronas Ocultas y 150 a 300 Épocas Máximas.	76
Anexo 9. Graficas de Resultados de la RNA con Arquitectura LSTM con 300 Neuronas Ocultas y 150 a 300 Épocas Máximas.	78
Anexo 10. Graficas de Resultados de la RNA con Arquitectura LSTM Actualizada con los Datos de la Salida Anterior usando 150 Neuronas Ocultas y 150 a 300 Épocas Máximas.	80
Anexo 11. Graficas de Resultados de la RNA con Arquitectura LSTM Actualizada con los Datos de la Salida Anterior con 200 Neuronas Ocultas y 150 a 300 Épocas Máximas.	82

Anexo 12. Graficas de Resultados de la RNA con Arquitectura LSTM Actualizada con los Datos de la Salida Anterior con 250 Neuronas Ocultas y 150 a 300 Épocas Máximas. 84

Anexo 13. Graficas de Resultados de la RNA con Arquitectura LSTM Actualizada con los Datos de la Salida Anterior con 300 Neuronas Ocultas y 150 a 300 Épocas Máximas. 86

Resumen

En el presente proyecto de grado, se representa la aplicación de las redes neuronales artificiales (RNA) para el modelado de una variable de gran importancia en el análisis de eficiencia energética: el consumo total de potencia activa, así mismo su predicción, informe y grafica a través de una interfaz. Por una parte, se ha propuesto dos modelos de RNA para la predicción programadas en *Matlab*, la primera arquitectura es basada en el aprendizaje automático, la red retroalimentada de dos capas y la segunda arquitectura basada en el aprendizaje profundo, la red de memoria a corto y largo plazo. Por otra parte, el diseño de una interfaz capaz recopilar datos fiables de consumo total de potencia activa, enviar esta información a la RNA seleccionada, ilustrar sus predicciones, y asimismo su precisión comparada con datos reales.

Para ejecutar las RNA, se implementaron los datos compilados de potencia activa para crear los modelos neuronales, también se realizó variaciones de los hiperparámetros, como la cantidad de neuronas ocultas o las épocas de entrenamiento en la exploración de un modelo eficiente, además se ilustro el informe de aprendizaje de las RNA, conjuntamente el cálculo de la precisión con el error cuadrático medio y la regresión lineal, comparando los valores y seleccionando la que presento mejor respuesta.

La interfaz propuesta se encamino a ser manejable e intuitiva, capaz de importar datos de archivos Excel, predecir utilizando la RNA seleccionada, graficar, tabular e indicar la precisión de los datos, y por último exportar los valores predichos en un archivo Excel.

1. Introducción

La gestión energética, debe presentarse como la optimización de la energía identificando acciones de su uso racional y eficiente, sin mitigar la oferta del servicio o producción. Por otro lado, es un sistema muy aplicado en la industria y edificaciones, con fines económicos o concientización energética, debido al aumento continuo de la demanda por el crecimiento poblacional o puesta en práctica de nuevos procesos industrializados, así mismo, la eficiencia energética en la industria, se reconoce como un medio rentable que contribuye al desarrollo sostenible y a la competitividad industrial. Con la implementación de prácticas de gestión, se vuelve aún más imperativa para los países en desarrollo, considerando sus tendencias de uso energético como base del desarrollo económico (Hasan, Tuhin, Ullah, Sakib, Thollander & Trianni, 2021).

Debido a la actual necesidad de la industria y sectores productivos en generar ahorros energéticos y en seguir las normas de gestión de energías, se ha impulsado el desarrollo de actividades efectivas, en eficiencia energética, sin la necesidad de realizar grandes inversiones en consensos de gestión de energías, con una inversión enfocada en generar las estrategias y acciones en eficiencia energética basándose en predicciones precisas, conllevan a beneficios en la producción, competitividad y reducción de consumo, por consiguiente en la búsqueda de herramientas de pronósticos precisos, la utilización de las redes neuronales artificiales se resalta, siendo esta óptima alternativa para la gestión de energías con fines de eficiencia.

Por otro lado, para llevar a cabo ese ahorro energético, se debe actuar bajo las normas de gestión energética como por ejemplo la ISO 50001 para las industrias, ya que deben estar certificadas, implicando mejoras en productividad y reconocimiento en el mercado, aplicando estrategias de eficiencia energética y modificando hábitos de consumo en procesos de

fabricación. Por ello, utilizando los pronósticos de consumo energético se pueden tomar decisiones operativas y estrategias con mayor precisión, además de disminuir costes en el proceso (Flores, 2018), definiendo cuantitativamente la demanda futura, procurando que los resultados sean lo más exactos posibles, para esto la tecnología de las RNA, constituyen un potente instrumento en la modelización de fenómenos complejos.

La aplicación de las redes neuronales artificiales son una tecnología bastante eficiente y de buena ejecución en la acción de pronósticos usando datos recolectados y factores de importancia, obteniendo resultados con márgenes de error bajo, además de su constante aprendizaje con nuevos datos, acercando sus pronósticos a los valores reales (Arcos, 2016), su implementación en la predicción de consumo energético, logra facilitar la identificación de momentos o zonas de impacto con fidelidad, ideal para la toma de decisiones operativas y estratégicas, disminuyendo costos en el proceso.

El uso de los métodos de pronóstico se lleva a cabo en tres partes: a corto, mediano o largo plazo, analizando los resultados, buscando los sobre picos de consumo y evitar los daños en equipos por sobreesfuerzo, con un ahorro económico desactivando los equipos que realicen consumo en tiempos fuera de producción, así mejorando la gestión energética de estas construcciones.

2. Objetivos.

2.1. Objetivo General.

Desarrollar un modelo predictivo de consumos de energía eléctrica aplicando redes neuronales artificiales.

2.2. Objetivos Específicos.

Realizar un análisis del estado del arte sobre sistemas de gestión de energía y tecnología aplicadas a modelos predictivos de consumos eléctricos.

Definir un escenario de prueba a partir de la construcción y clasificación de consumo como parámetro energético.

Implementar un aplicativo de simulación con redes neuronales artificiales que permita la evaluación y verificación del modelo predictivo de consumos usando la data recopilada.

3. Estado del Arte

La Comisión Europea tiene una visión estratégica a largo plazo para liderar la transición hacia una economía climáticamente neutra para el año 2050 en consonancia con los objetivos del Acuerdo de París. En 2009 la UE adoptó un amplio paquete de alcance donde ubicaba la legislación para garantizar que la UE cumple sus objetivos climáticos y energéticos para el año 2020. Son los llamados objetivos '20-20-20' de la UE e 20% de aumento en la eficiencia energética, reducción del 20% en las emisiones de gases de efecto invernadero (desde los niveles de 1990), y 20% de renovables para 2020. En 2016, en Energía Limpia para toda Europa (Comisión Europea, 2016). Además, se establecieron tareas para cumplir con la legislación energética alineada con el nuevo modelo energético basado en la Agenda 2030, objetivos que enfatizan la 'eficiencia energética primero' como uno de los principios de la Unión de la Energía. Esto se debe a que la eficiencia energética es la forma más rentable de reducir las emisiones, mejorar la seguridad energética, aumentar la competitividad y reducir el consumo de energía, haciéndola más asequible para todos los consumidores. También presenta una de las claves para lograr los objetivos energéticos y climáticos a largo plazo. Con el Artículo 194 del TFUE (Tratado de Funcionamiento de la Unión Europea) la UE promueve la eficiencia energética, el ahorro de energía, además del desarrollo de formas de energía nuevas y renovables. Específicamente, la Directiva de Eficiencia Energética (EED) de 2012 (Union Oj, 2012) es una directiva, que introdujo una serie de medidas, políticas, requisitos y herramientas (es decir, esquemas de obligaciones de eficiencia energética o medidas alternativas) para permitir que la UE alcanzase su 20% de eficiencia energética de acuerdo con los objetivos de eficiencia energética de la agenda 2020.

Los Estados miembros de la UE están obligados a utilizar energía de manera más eficiente en todas las etapas de la cadena energética, incluyendo generación, transmisión, distribución y consumo final de energía. Se establecieron revisiones adicionales para las trayectorias a largo plazo (Unión Oj, 2018) con el objetivo de eficiencia energética para 2030 fijado de al menos el 32,5 % (a alcanzar colectivamente en toda la UE). Dentro del marco del Acuerdo de París, la Comisión Europea también emitió una estrategia para una economía climáticamente neutral en el 2050, actualizando su hoja de ruta hacia una descarbonización sistemática (Comisión, 2018) y proporcionando una trayectoria rentable hacia la consecución del objetivo de emisiones netas cero. La UE también opera los sistemas de comercio de los derechos de emisión de la UE, que pone un precio a emisiones de gases de efecto invernadero para crear incentivos financieros para la industria y empresas para reducir las emisiones (Malinauskaite, Jouhara, Egilegor, Al-Mansour, Ahmad & Pusnik, 2019).

A nivel internacional, las políticas y legislaciones de eficiencia energética continúan aumentando en el mercado global (Guo, Wu, Ding, Feng & Zhu, 2016), a través de programas de investigación de eficiencia energética (Du, Wang & Zhang, 2018), además la aplicación de estándares y etiquetas para electrodomésticos, así como también edificios (Kangas, Lazarevic & Kivimaa, 2018). Sin embargo, los países en vías de desarrollo, siguen teniendo políticas y programas de eficiencia energética muy carentes (Tumbaz & Mogulkoç, 2018), que promueven estándares de eficiencia energética del lado de la demanda y protocolos (Koskimaki, 2012). En línea con esto, Colombia estableció varias metas en el sector de la construcción, como el aumento de la eficiencia energética en los edificios, incluido el uso directo y el diseño. La Unidad de Planeación Minero Energética estima que el sector de la construcción, junto con el residencial y comercial, representa el 23% del consumo energético del país (UPME, 2016)

(principalmente en forma de electricidad). En 2015 Colombia lanzó un estándar de construcción sostenible, establecía el rendimiento con parámetros que se deben cumplir en términos de energía y agua para cualquier edificio nuevo construido después de 2015. Los edificios antiguos no estaban obligados a alcanzar estándares de construcción sostenible, pero los propietarios podrían voluntariamente renovar los edificios y solicitar incentivos de eficiencia energética (UPME, 2020).

El gobierno colombiano ha estimado un potencial de ahorro energético de 10 a 40% a nivel de edificio, dependiendo del tipo de edificación y clima (Ministerio de Vivienda Ciudad y Territorio, 2015). Sin embargo, los ahorros reales de energía que podrían alcanzarse en el nivel de la ciudad aún no están claros. El ahorro futuro de energía depende del número de edificios nuevos que cumplen con el estándar, edificios antiguos que adoptan las medidas de eficiencia energética y qué tan rápido sucede esto. Por lo tanto, hay una necesidad de analizar el impacto de las nuevas normas de construcción sostenible y medidas de eficiencia energética en el ahorro de energía y entender cómo dicha adopción podría desarrollarse con el tiempo, considerando no solo los nuevos edificios, sino también el parque inmobiliario actual, y teniendo en cuenta limitaciones debido a datos incompletos o inexistentes sobre los edificios como las categorías, edades o características de diseño.

En la última década se han desarrollado varios modelos para analizar la eficiencia energética de los parques de edificios a nivel nacional (Fazeli & Davidsdottir, 2017; Sahlol, Elbeltagi, Elzoughiby & Abd Elrahman, 2020). Uno de los métodos más utilizados para estimar el consumo de energía es el análisis del ciclo de vida (Dixit, 2017; Geng, Wang, Zuo, Zhou, Du & Mao, 2017), análisis de flujo de materiales (Cao, Liu, Duan, Xi, Liu & Yang, 2019; Condeixa, Haddad, Boer, 2017) y enfoques híbridos (Cabeza, Rincón, Vilaríno, Pérez, Castell, 2014;

Rincón, Castell, Pérez, Solé, Boer, Cabeza, 2013). Estos enfoques metodológicos se utilizan para el análisis a nivel de edificio, que son más precisos para estimar consumo de energía o emisiones. Sin embargo, tales análisis no consideran la evaluación de políticas en casos con poco o nula cantidad de datos. Los modelos de control predictivo (MPC) han alcanzado cada vez más interés en la investigación, ya que una de sus principales aplicaciones, ha sido su empleo en el desarrollo de sistemas avanzados de control de procesos multivariantes, debido a su elevada capacidad para calcular las acciones de control óptimas, basadas no solo en las mediciones de estado instantáneo sino también la respuesta anticipada del proceso (Ren, Alhajeri, Luo, Chen, Abdullah, Wu, Christofides, 2022).

Convencionalmente, las redes neuronales artificiales que se emplean para el reconocimiento de patrones, implican una estructura que presenta números ocultos en capas, pero ese número de unidades presentes en cada una de esas capas, están completamente interconectadas. El problema es que ese número de redes neuronales funciona bien para los problemas de menor espacio de entrada, pero a medida que aumenta la complejidad de entrada en la multicapa, disminuyen los términos de clasificación y convergencia (Xue, Jiang, Neri & Liang, 2021). Para resolver dicho problema de reconocimiento de patrones de entrada mayores, lo que se hace es descomponerlo en subproblemas más pequeños y estos subproblemas son resueltos individualmente y simultáneamente en componentes individuales de redes neuronales modulares. Otras soluciones a dichos problemas es reconstruirlos para obtener una decisión final de la red neuronal global (Sinh & Singh, 2021).

Maziar, Perdikaris & Karniadakis (2019), introdujeron una red neuronal de información física (PINN) que está entrenada para resolver aprendizaje supervisado- de tareas de aprendizaje sobre leyes físicas dadas descritas por ecuaciones diferenciales parciales no lineales generales. Al

resolver los problemas directo e inverso de las ecuaciones diferenciales parciales, propusieron dos métodos de forma continua y forma discreta, y obtuvo una mejor precisión de resolución, y este método resolvió los problemas directos e inversos de ecuaciones diferenciales parciales a la misma velocidad. En consecuencia, Jin, Cai, Li & Karniadakis (2021), adoptaron PINN para codificar directamente las ecuaciones gobernantes en el sistema neuronal profundo. a través de la diferenciación automática, a fin de superar algunas limitaciones para simular incompresible flujos laminares y turbulentos. Sheng & Yang (2021), propusieron una red neuronal sin penalización (PFN-N), que puede resolver eficazmente una clase de problemas de valores en la frontera de segundo orden en complejos geométricos.

Los experimentos numéricos mostraron que PFNN era superior a varios métodos existentes en precisión, flexibilidad y robustez. Recientemente, más y más estudiosos han comenzado a prestar atención al uso de métodos de redes neuronales para resolver ecuaciones diferenciales parciales fraccionarias y literatura relevante surgen uno tras otro. Pang, Lu & Karniadakis (2019), plantean redes neuronales para resolver ecuaciones fraccionarias de advección-difusión del espacio-tiempo, y estudió sistemáticamente su convergencia. El elemento novedoso es el enfoque híbrido que introdujeron para construir el residual en la función de pérdida, utilizando tanto la diferenciación automática para los operadores de orden entero y discretizado para los operadores fraccionarios. Entonces Pang, D'Elia, Parks & Karniadakis (2020), construyen modelos de redes neuronales de inferencia de funciones para ecuaciones integrales, como modelos no locales de Poisson y de turbulencia, además se refirió a ellos como PINN no locales (nPINN), para obtener más instrucciones sobre los métodos de redes neuronales. Sin embargo, hay poca literatura sobre el uso de métodos de redes neuronales profundas para resolver problemas directos e inversos de ecuaciones de difusión fraccionaria de tiempos conformes.

4. Desarrollo del Modelo Basado en Redes Neuronales Artificiales y la Interfaz en *Matlab*.

La figura 1, representa el diagrama de flujo indicando el desarrollo del modelo predictivo y la interfaz usando redes neuronales artificiales (Artificial Neural Networks – ANN), primero el análisis de los datos utilizados, segundo, la selección de características de la ANN, tercero, la implementación de los datos en el aprendizaje de la ANN, cuarto, la extracción de los modelos de ANN aceptables con buena precisión, quinto y último, la creación y utilización de la ANN en la aplicación final.

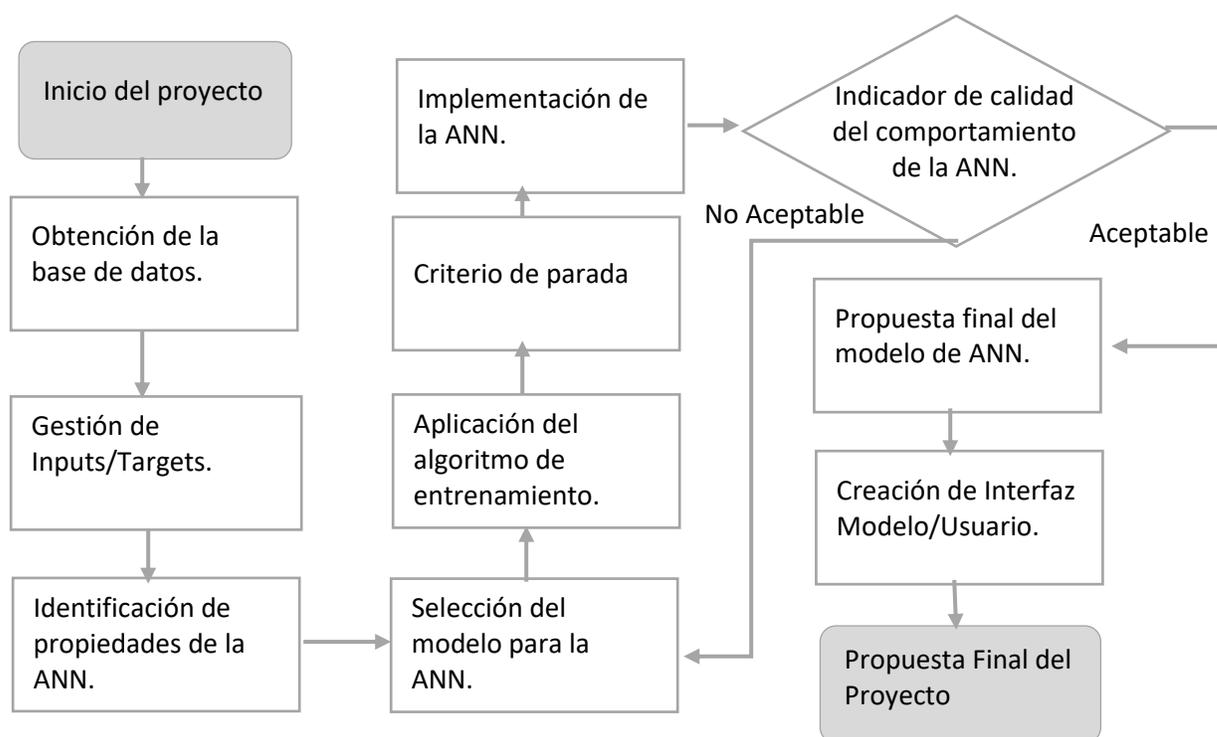


Figura 1. Diagrama de Flujo sobre el desarrollo de la RNA y su interfaz.

4.1. Obtención de la Base de Datos.

La serie histórica de variables de consumo eléctrico de este estudio fueron obtenidos por medio del Servicio Nacional de Aprendizaje (SENA) en la localidad de Cúcuta, seleccionando el edificio Administrativa del barrio Pescadero como objeto de estudio.

El muestreo de datos se realizó con el analizador trifásico PowerPad® III Modelo 8336, conectado entre la fecha y tiempo del 15 de febrero del 2022 a las 3:45:00 p.m. hasta el 16 de marzo del 2022 a las 5:41:00 p.m., cuantificando hasta 40 variables diferentes con un periodo de 60 segundos, de las cuales se extrajeron solo las variables consideradas óptimas para el modelo neuronal, que fueron la fecha, el tiempo y la potencia activa total, con un total de 41877 datos por variable.

	A	B	C
1	Fecha	Hora:minutos	Potencia Act. (W)
2	15/02/2022	3:45:00 p. m.	100165
3	15/02/2022	3:46:00 p. m.	100639
4	15/02/2022	3:47:00 p. m.	100787
5	15/02/2022	3:48:00 p. m.	98953
6	15/02/2022	3:49:00 p. m.	98667
7	15/02/2022	3:50:00 p. m.	99096
8	15/02/2022	3:51:00 p. m.	99070
9	15/02/2022	3:52:00 p. m.	98330
10	15/02/2022	3:53:00 p. m.	98146
11	15/02/2022	3:54:00 p. m.	98834
12	15/02/2022	3:55:00 p. m.	97362
13	15/02/2022	3:56:00 p. m.	97216
14	15/02/2022	3:57:00 p. m.	97741
15	15/02/2022	3:58:00 p. m.	97894
16	15/02/2022	3:59:00 p. m.	97753
17	15/02/2022	4:00:00 p. m.	97088

Figura 2. Muestra de la base de datos implementada.

4.2. Caracterización de las Variables de Entrada y Salida (Inputs/Targets).

La selección de características es el proceso de reducir el número de variables predictoras al desarrollar un modelo predictivo, ayudando a reducir los costes computacionales de los modelos de entrenamiento y mejorando el rendimiento del modelo (Packwood, Nguyen, Cesana, Zhang, Staykov, Fukumoto & Nguyen, 2022).

Los pronósticos se pueden diferenciar en cuatro tipos, los de muy corto plazo, los de corto plazo, los de mediano plazo y los de largo plazo, definidos por el código de redes del Sistema Interconectado Nacional (SIN), avalado por la Comisión de Regulación de Energía y Gas (CREG), como el periodo cubierto en la etapa de planeamiento operativo, en donde el muy corto plazo desde la hora actual hasta el fin del día, el corto plazo desde las 24 horas a cinco semanas, el mediano plazo de cinco semanas a cinco años y el largo plazo a más de cinco años.

Como se disponía de 30 días en datos para la creación del modelo de predicción, se tomó como objetivo de salida un pronóstico a corto plazo con la mayor precisión experimentada. A las variables se le aplicó un promedio de datos reduciendo su cantidad, facilitando su manejo, transformando los datos de minutos a horas, disminuyendo la cantidad de datos a 697 por variable.

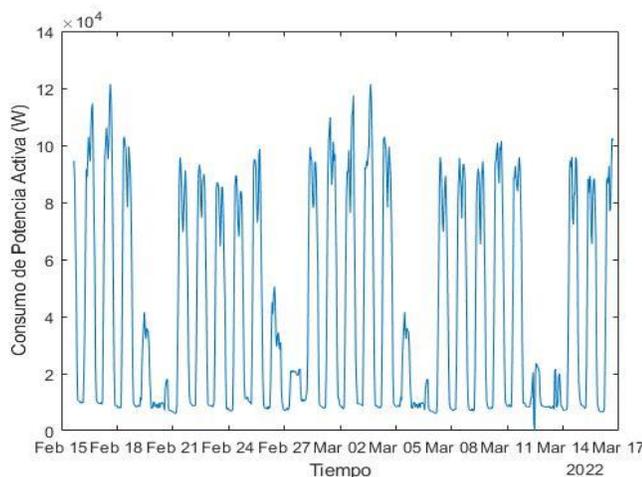


Figura 3. Grafica de los Datos de Entrada.

Las redes neuronales artificiales exigen cierta cantidad de datos para ser entrenada y evaluada, para este caso, se fraccionó el total de datos seleccionando el 90% para su entrenamiento y el 10% restante para evaluarlo.

Para mejorar el modelo de la red neuronal artificial es necesario hacer un preprocesamiento en los datos de entrada, como la minimizando de las diferencias entre datos, la integración, transformación y reducción de valores, disminuyendo resultados divergentes en su entrenamiento y evaluación, mejorando la calidad de la red para alcanzar la predicción, posterior al entrenamiento, la salida de la red neuronal artificial se invierte según el preprocesamiento seleccionado (Rodríguez & Ugalde, 2021).

Los preprocesamientos frecuentes son el escalado, como el rango posterior al centrado de cada columna, que es dividido por su rango máximo y mínimo, el normalizado, aplicando la desviación estándar como factor de escala (Rodríguez & Ugalde, 2021), y el estandarizado, permitiendo una media cero y unificación de la desviación estándar (Carbajal-Hernández, Sánchez-Fernández, Hernández-Bautista & Hernández-López, 2017), siendo este último el utilizado para el preprocesamiento de datos de entrenamiento y evaluación en el aprendizaje profundo.

En el aprendizaje automático se realizó una ordenanza de los datos de entrada, como es habitual con las series de tiempo en problemas del tipo supervisado, al tener una sola fila de datos en el tiempo, se convierten en varias filas, que se puedan dividir en entradas de tiempos pasados y salidas de tiempos futuros.

The figure consists of two screenshots of spreadsheets. The top screenshot, titled "Datoseneltiempo", shows a 1x100 string array with columns 1-11 and rows 1-12. Row 1 contains T0, T1, T2, T3, T4, T5, T6, T7, T8, T9, T10. Row 2 contains T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11. The bottom screenshot, titled "Arreglodatos", shows a 50x50 string array with columns 1-11 and rows 1-12. Row 1 contains T0, T1, T2, T3, T4, T5, T6, T7, T8, T9, T10. Row 2 contains T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11. The rest of the rows are filled with a grid of time steps shifted by one column.

Figura 4. Demostración del arreglo de datos.

Para el aprendizaje profundo se establecieron los valores de entrenamiento como pasos de tiempo en secuencia, especificando las entradas como las secuencias de tiempo sin el último paso en el tiempo, y las respuestas como el valor del siguiente paso de tiempo, es decir, las respuestas son secuencias de entrenamiento con valores desplazados un paso de tiempo.

Los datos de fecha y tiempo al ser de corto plazo, presentan poca importancia como entrada al no realizar cambios realmente relevantes para una red neuronal artificial, aunque presento una naturaleza cíclica de tiempo, las fechas aisladas que no se puedan comparar con otros datos, pueden generar más confusión que aciertos para la RNA, por tal motivo se prefirió utilizarlos solo para la graficación y tabulación, excluyéndolos como entradas de la red neuronal artificial, se procedió a convertirlos en variables del tipo *datetime*, y unirlos para su interpretación.

Los datos de potencia activa se convirtieron en variables del tipo *doble*, al ser un conjunto de datos contiguos en el tiempo conformados por observaciones históricas, se clasificaron como valores de series de tiempo, ideal para realizar pronósticos o estimaciones (Urrego, 2022).

4.3. Identificación de las Propiedades de las Redes neuronales Artificiales.

4.3.1. Modelos Neuronales y Algoritmos de Aprendizaje Seleccionados.

El campo que combina la inteligencia artificial y la ciencia de datos, es el aprendizaje automático (Machine Learning) usando la informática y la estadística en el desarrollo de algoritmos de aprendizaje, que se centra en conseguir que un ordenador resuelva un problema, sin necesidad que las personas le muestren como hacerlo (Mosquera, Castrillón & Parra, 2018) (Sudha, Suresh & Nagesh, 2022).

Las redes neuronales artificiales son la aplicación más destacada del aprendizaje automático, capaces de clasificar e imitar el modelo de sistemas complejos, con el entrenamiento no memorizan, más bien generalizan su comportamiento, al mejorar el rendimiento y la capacidad de abstracción, las RNA se extiende hacia el aprendizaje profundo (Deep Learning), que además de su habilidad de aprendizaje, también pueden encontrar el significado del contexto entrenado (Aguirre, 2019).

Los elementos de la neurona artificial estándar en el aprendizaje automático constan de las entradas $x_j(t)$, con los pesos sinápticos w_{ij} , el parámetro del bias θ_i , y n como las variables de entrada, dejando la ecuación (1) de la función de salida f_i , (Vivas, Martínez & Pérez, 2014):

$$(1)f_i = \sum_{j=1}^n w_{ij}x_i - \theta_i$$

La red neuronal artificial del aprendizaje profundo, es una percepción de capas múltiples que consiste en varias capas con cada capa conteniendo varias neuronas, la ecuación (2) representa el modelo matemático del RNA en el aprendizaje profundo:

$$(2)Z = \sum_{i=1}^n w_i + x_i + b$$

Donde z es el valor de salida predicho por el modelo de aprendizaje profundo, n son las capas de entradas, i es la anotación de la variable, x la señal de entrada, b es el sesgo y w es el peso de cada variable (Mostafa, Ramadan & Elfarouk, 2022).

Las primeras redes neuronales artificiales seleccionadas se basaron en el aprendizaje automático, diseñadas con la arquitectura del tipo Prealimentada (Feed-Forward), y entrenada con el algoritmo de retropropagación (Backpropagation) de Levenberg-Marquardt.

La red neuronal Prealimentada es de arquitectura serial, que ordena las capas neuronales una frente a la otra, la primera capa recibe el nombre de sensorial, encargada de recibir las variables de entrada, la capa intermedia u oculta determina la modificación de sus propios pesos generando el aprendizaje de la red, finalmente, la capa de salida es la responsable de entregar las respuestas del modelo neuronal (Fonseca & Sanabria, 2018).

Las capas se conectan una tras otra, de tal forma que la salida de la capa actual se utiliza como entrada en la siguiente capa, progresivamente, hasta obtenida la salida de la red.

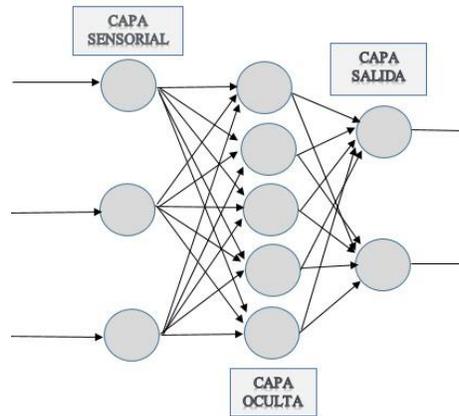


Figura 5. Esquema de la arquitectura de las Redes Neuronales Artificiales del tipo Prealimentada.

El algoritmo de entrenamiento desarrollado por Levenberg-Marquardt, usa las definiciones de retropropagación como la regla delta generalizada (generalized delta rule), la tasa de aprendizaje (Learning Rate) y el modo por lotes (Batch Mode), aplican la matriz Jacobiana en el cálculo del gradiente (Castro & Mejía, 2022), diseñado de tal manera para acercarse a la velocidad de entrenamiento de segundo orden aproximando la ecuación (3) de la matriz hessiana y (4) el gradiente a:

$$(3) H = J^T J \qquad (4) g = J^T e$$

Donde H es la matriz hessiana, J es la matriz jacobiana conteniendo las primeras derivadas de los errores de la red con respecto a los pesos y los sesgos, y e como el vector de errores de la red (Mathworks, 2022).

Posteriormente con las técnicas de aprendizaje profundo se eligieron las redes neuronales recurrentes con la arquitectura de Memoria a Corto y Largo Plazo (Long Short-Term Memory) usando el algoritmo entrenador de Estimación Adaptativa de Momentos (Adaptive Moment Estimation).

Las redes neuronales recurrentes añaden un ciclo en cada neurona permitiendo, mantener información en el tiempo mediante una nueva matriz de peso, facilitando colocar en contexto los datos de tiempos pasados permitiendo hacer interferencia en el tiempo actual. Siendo la arquitectura LSTM un buen exponente de las redes neuronales recurrentes permitiendo almacenar información a corto y largo plazo, su característica es la edición en un estado de celda en el que se escribe o borra la información relevante de la serie de tiempo, modificándose con cada dato de entrenamiento, con tres compuertas importantes, llamadas olvido, recuerdo y salida, que realizan la predicción para el tiempo siguiente basándose en el estado oculto (Donoso, 2018).

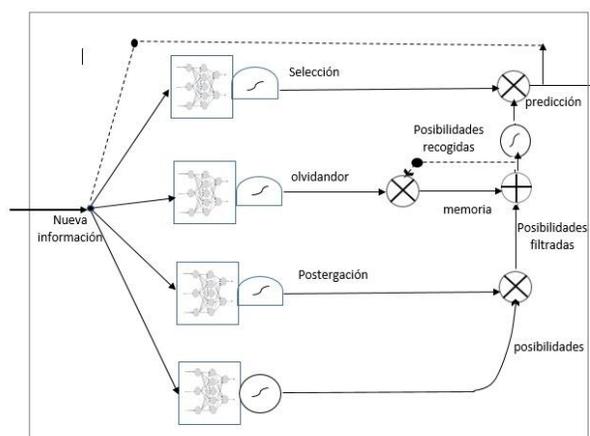


Figura 6. Esquema de la arquitectura de las Redes Neuronales Artificiales del tipo Memoria a Corto y Largo plazo.

El algoritmo de Estimación Adaptativa de Momentos es usado para el cálculo de un nuevo valor de los sesgos y los pesos del modelo, donde cada iteración intenta disminuir la compuerta de olvido. Su método es descenso de gradiente estocástico orientado en la estimación del momento adaptativo de primer y segundo orden, (Rodríguez & Monsalve, 2019) buscando la mejor hipótesis y parámetros que se adecue al modelo experimentado.

Utiliza parámetros similares a la optimización de propagación de la media cuadrática, con la diferencia de un término de impulso adicional, con un promedio cambiante de elementos, como en el gradiente de parámetros y sus valores al cuadrado, usándolos para actualizar los parámetros de la red en una dirección determinada, tales como (5):

$$(5) \quad W_{t+1} = W_t - a_t \frac{au_1}{\sqrt{v + \varepsilon}}$$

Donde

$$(6) \quad u_{t+1} = u_t\mu_1 + (1 + \mu_1)\nabla_{wt}f$$

Y

$$(7) \quad v_{t+1} = v_t\mu_2 + (1 - \mu_2)(\nabla_{wt}f)^2$$

En el cual, W es la secuencia de actualización de cada parámetro, a_t corresponde al factor de aprendizaje, μ el termino de impulso adicional, $\nabla_{wt}f$ al sentido contrario del gradiente y ε es una pequeña constante que se añade para evitar la división por cero (Mathworks, 2022).

4.3.2. Criterio de Parada.

El proceso iterativo se debe resolver durante un número significativo de épocas hasta que algún criterio de convergencia determine que la red ha aprendido la información que se le ha mostrado (Ocampo, Giraldo & Isaza, 2004).

El criterio de parada seleccionado para las redes neuronales prealimentadas es indicado con el aumento del error cuadrático medio (Mean Squared Error) de las muestras de validación que se calcula con la ecuación (8):

$$(8) \text{MSE} = \frac{1}{n} \sum_{i=1}^n (r_i - e_i)^2$$

Donde n es el número total de variables, r las variables reales y e las variables estimadas.

Las redes neuronales artificiales LSTM usan una reducción de aprendizaje durante el entrenamiento, actualizando la tasa de aprendizaje cada número de épocas determinadas por un factor, este es especificado por un escalar de 0 a 1.

4.3.3. Implementación de las Redes Neuronales con Matlab

Para empezar, se utilizó los paquetes de cálculo de redes neuronales artificiales de *Matlab* para la creación de los modelos, así mismo las RNA de aprendizaje automático con el *Neural Network Toolbox*, y de aprendizaje profundo con el *Matlab Editor*.

Lo más importante es la selección de variables, llamadas hiperparámetros, esto son las variables que definen la estructura de la red y su entrenamiento, determinándolas antes del entrenamiento y ajustándolas después de los resultados de los mismo, por ejemplo, las capas ocultas del algoritmo o la función de activación aplicada (Pérez del Barrio, Fernández-Miranda, Sanz, Lloret & Rodríguez, 2022).

La selección de la cantidad de capas ocultas y el número de neuronas de estas, de acuerdo con el teorema de aproximación universal, una red neuronal multicapa con un número finito de neuronas puede actuar como un aproximador universal, al mismo tiempo se consiguen resultados más que aceptables utilizando un número relativamente pequeño de neuronas, además de disminuir la carga computacional (Berzal, 2018), de modo idéntico, Isasi-Viñuela & Galvan-Leon (2004) destacan que los hiperparámetros deben ser seleccionadas por el diseñador, a causa de no existir

una regla o método que identifique el número óptimo de neuronas o capas ocultas en la solución de los modelos.

4.3.4. Modelos de Redes Neuronales con Neural Network Toolbox.

Para ingresar a la interfaz creada por *Matlab*, se debe ingresar el código *nstart* en *Matlab Edit*, y seleccionar el tipo de generador de red neuronal artificial que dispone la GUI (Graphical Interface Used).

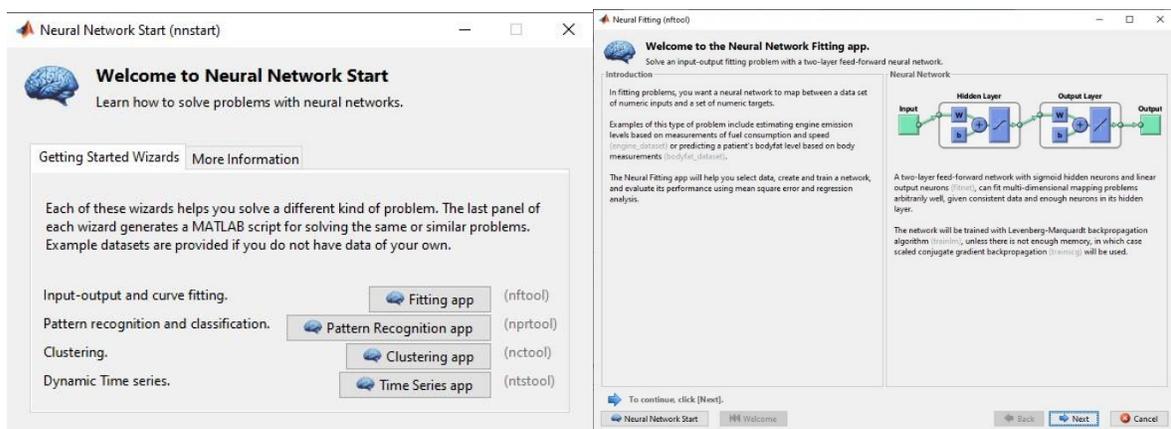


Figura 7. Inicio del GUI *nstart* y *nftool*.

Para el caso implementado se optó por *Fitting app* como se muestra en la figura 7, que permite la aplicación de una Red Neuronal Prealimentada de dos capas, facilita la selección de datos de entrada y salida, la distribución de los datos en el entrenamiento, validación y evaluación, además de la modificación de la cantidad de neuronas en la capa oculta, y su algoritmo de entrenamiento.

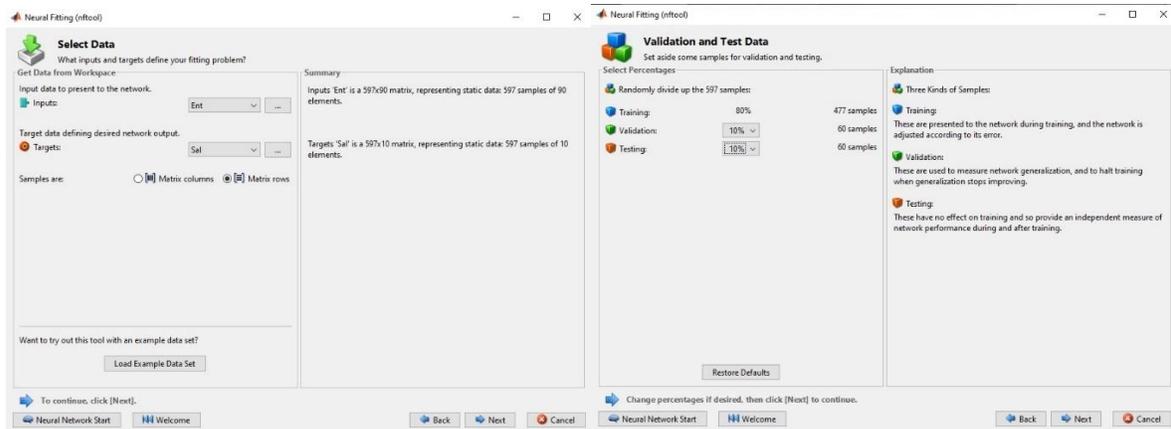


Figura 8. Selección del conjunto de variables (izquierda) y su distribución (derecha).

Para el modelo de la RNA, se tomaron las variables de entrada como el 90% de los primeros datos ordenados en el tiempo de la potencia total activa, asignada a las series de tiempo pasadas, y el otro 10% restante como las series de tiempo futuras, con el objetivo de entrenar la RNA con una proporción mayor en la cantidad de datos de entrada a comparación de la salida, así aumentando su capacidad de precisión en el pronóstico, y con el mismo objetivo la distribución de datos para el entrenamiento, validación y evaluación asignadas con el 80%,10% y 10% del total respectivamente, como se logró apreciar en la figura 8.

En la arquitectura la RNA se realizó un análisis, que consistió en la modificación de la cantidad de neuronas en la capa oculta, variando desde 30 neuronas hasta 100 neuronas, en intervalos de 10, con el propósito de encontrar el mejor modelo.

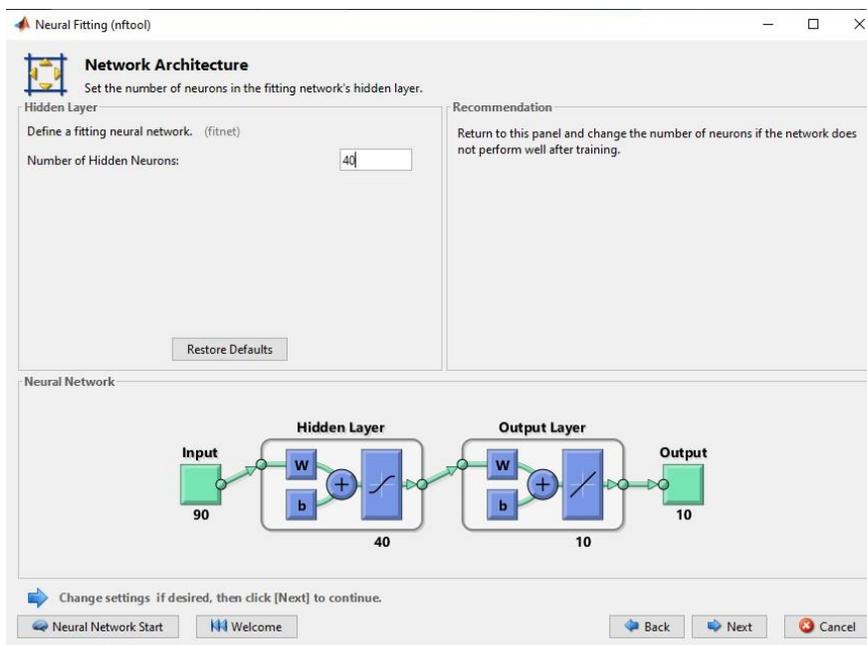


Figura 9. Ejemplo de la modificación de la cantidad de neuronas en la capa oculta.

En el entrenamiento se eligió el algoritmo de optimización de Levenberg-Marquardt evidenciado en la figura 10, y el análisis del registro de entrenamiento al finalizar el aprendizaje.

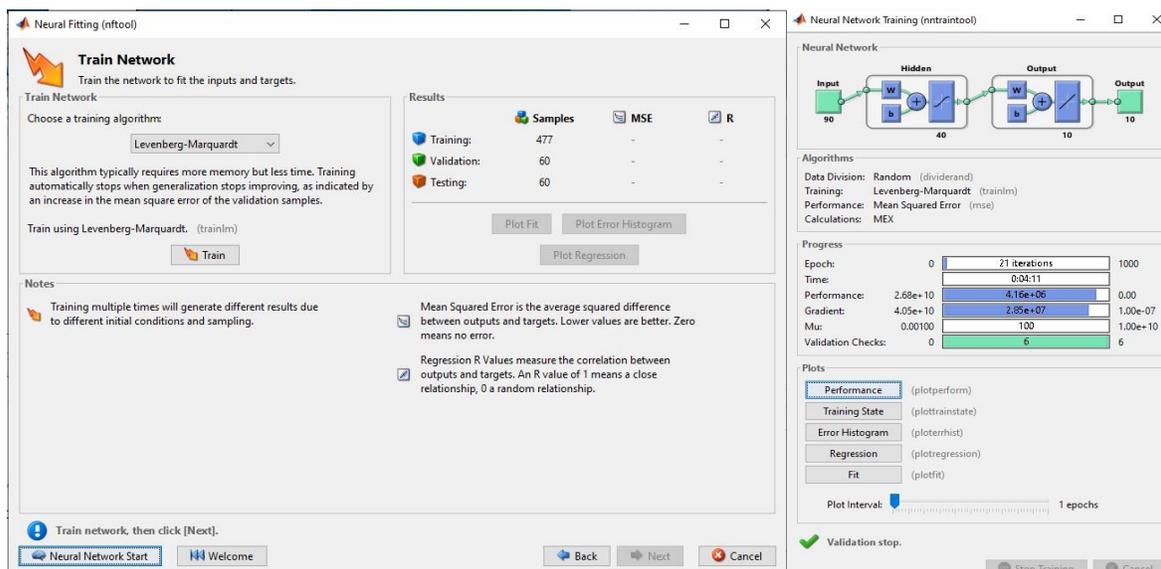


Figura 10. Selección del algoritmo de optimización de Levenberg-Marquardt (izquierda) y ejemplo del registro de entrenamiento (derecha).

Finalmente, el resumen de los parámetros de entrenamiento como: el tipo de red es perceptrón multicapa, el algoritmo de entrenamiento de Levenberg-Marquardt, la función de error con el error cuadrático medio, la cantidad máxima de épocas de entrenamiento como 1000, el máximo de épocas erradas en la validación configurada a 6 y el tiempo de entrenamiento es sin límite.

Tabla 1. Propiedades seleccionadas para la RNA en el aprendizaje automático.

<i>Artificial Intelligence Field</i>	Machine Learning
<i>Network Type</i>	Multilayer Perceptron
<i>Network Architecture</i>	Feed-Forward with two layers
<i>Error Function</i>	Mean Square Error
<i>Training Algorithm</i>	Backpropagation OF Levenberg-Marquardt
<i>Maximum Number of Epochs to Train</i>	1000
<i>Maximum Validation Failures</i>	6
<i>Maximum Time to Train</i>	Unlimited
<i>Range Of Neurons in The Hidden Layer</i>	30-100

4.3.5. Modelos de Redes Neuronales con Matlab Edit.

El modelo desarrollado en el *Matlab Edit* permite la modificación de la mayoría de aspectos y parámetros, ofreciendo mayor libertad en el entrenamiento, logrando objetivos más concretos y un progreso más ágil.

Se especificó en el código de la red neuronal el número de entradas, el número de salidas y el número de neuronas en la capa LSTM o capa oculta, y en las especificaciones del entrenamiento se establecieron el optimizados ADAM, el máximo de épocas de entrenamiento, el umbral de gradiente, la tasa de aprendizaje, y un graficador del progreso de entrenamiento, como se muestra en la figura 11.

```

numFeatures = 1;%numero de entradas
numResponses = 1;%numero de salidas
numHiddenUnits = Neuronas;%numero de capas ocultas
%creación de la red neuronal
layers = [ ...
    sequenceInputLayer(numFeatures)%Secuencias de entrada
    lstmLayer(numHiddenUnits)%Capas ocultas LSTM
    fullyConnectedLayer(numResponses)%Respuestas
    regressionLayer];
%Creacion de los parametros de entrenamiento
options = trainingOptions('adam', ... %Optimizador
    'MaxEpochs',Epochs, ...%Epocas de entrenamiento
    'GradientThreshold',1, ...%Umbral del gradiente
    'InitialLearnRate',0.005, ...%Tasa de aprendizaje
    'LearnRateSchedule','piecewise', ...%Seleccion del parametro de reducción
    'LearnRateDropPeriod',125, ...%Comiendo de tasa de reducción
    'LearnRateDropFactor',0.2, ...%Factor de multiplicacion de reducción
    'Verbose',0, ...
    'Plots','training-progress');%Graficacion del progreso

```

Figura 11. Códigos de parametrización de la RNA y del entrenamiento.

Con el objetivo de encontrar el modelo neuronal que mejor se adaptara al caso estudiado, se variaron la cantidad de neuronas en la capa oculta, comenzando con 150 neuronas y aumentando en intervalos de 50, hasta un máximo de 300 neuronas, así mismo se modificaron la cantidad de épocas máximas de entrenamiento, comenzando con 150 épocas, aumentando cada vez en 50 épocas hasta un máximo de 300 épocas.

Se entrena la RNA con los parámetros seleccionados, supervisando el progreso de entrenamiento (Figura 12), determinando si la precisión de la red está mejorando y con que rapidez, mostradas en cada interacción la precisión de cada mini lote con el error cuadrático medio, al mismo tiempo exponiendo las pérdidas de entropía cruzada en el entrenamiento.

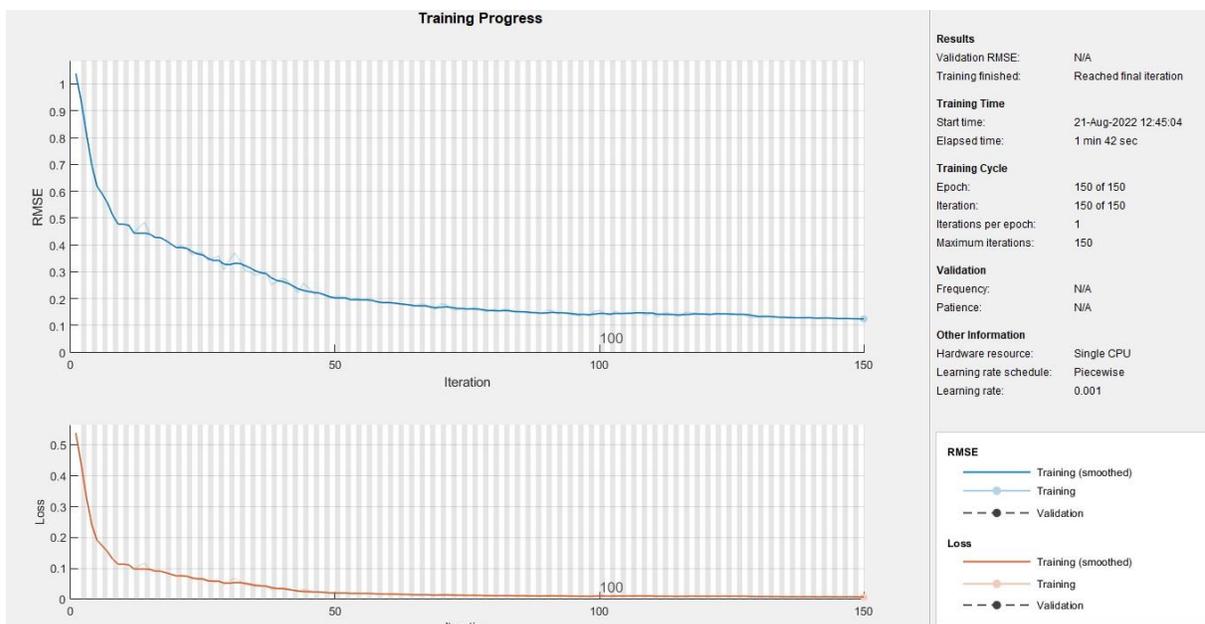


Figura 12. Ejemplo de la ventana del progreso de entrenamiento.

Después del entrenamiento de la red se inicializo la predicción sobre los datos de entrenamiento, realizando un bucle sobre la secuencia de tiempo, ingresando el último paso de tiempo de la respuesta de entrenamiento y solicitando la predicción del siguiente paso. El cálculo de las predicciones se realizaron sobre CPU, donde suelen ser más rápidas de calcular con colecciones de datos o secuencias cortas, al ser predicciones de un solo paso de tiempo, estas se registraran en una matriz separada.

```

%Para predecir los valores de múltiples pasos de tiempo en el futuro
net = predictAndUpdateState(net, InputTrain);
[net, OutputPred] = predictAndUpdateState(net, OutputTrain(end));
%Bucle de predicción
numTimeStepsTest = numel(InputTest);
for i = 2:numTimeStepsTest
    [net, OutputPred(:, i)] = predictAndUpdateState(net, OutputPred(:, i-1), 'ExecutionEnvironment', 'cpu');
    %Establezca la opción 'ExecutionEnvironment' de predictAndUpdateState como 'cpu'.
end

```

Figura 13. Código de iniciación de predicción.

Después de obtener la predicción del consumo de energía, se evaluó su precisión en comparación con los datos reales, y se reentrenó la red neuronal con los datos reales, explorando

un incremento en su precisión. Como instancia final, se recapitula la configuración de hiperparámetros de entrenamiento consistió en: el tipo de red como perceptrón multicapa, el algoritmo de entrenamiento de ADAM, la fusión de error con el cálculo del error cuadrático medio.

Tabla 2. Propiedades seleccionadas para la RNA en el aprendizaje profundo.

<i>Artificial Intelligence Field</i>	Deep Learning
<i>Network Type</i>	Multilayer Perceptron
<i>Network Architecture</i>	Long Short-Term Memorys
<i>Error Function</i>	Mean Square Error
<i>Training Algorithm</i>	Adaptive Moment Estimation
<i>Maximum Number of Epochs to Train</i>	150-300
<i>Range Of Neurons in The LSTM Layer</i>	150-300
<i>Maximum Time to Train</i>	Unlimited

4.3.6. Diseño la Interfaz Gráfica de Predicción.

La aplicación se desarrolló con el módulo *App Design*, de *Matlab*, con el objetivo de ser intuitivo y fácil para el usuario, se implementaron botones de acceso para la interacción con el aplicativo, graficas donde se exponen los resultados trasados, y tablas exponiendo los valores numéricos predichos.

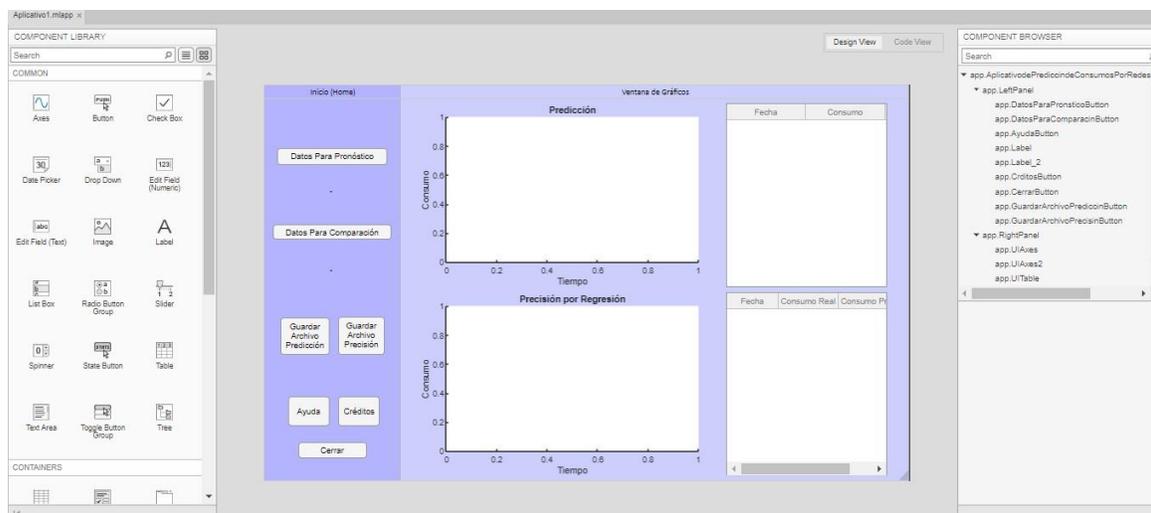


Figura 14. Desarrollo del aplicativo en el *App Design* de *Matlab*.

Para el ingreso de datos, en la columna “Inicio (Home)” se desarrollaron dos botones, el primero llamado “Datos Para Pronóstico”, asiste en la recopilación de la secuencia temporal del consumo de potencia activa, con la intención de realizar el pronóstico con la RNA, el segundo llamado “Datos Para Comparación”, facilita la recopilación de la secuencia temporal del consumo de potencia activa real, con el fin de comparar los datos con las predicciones que realizó previamente la RNA, los datos suministrados deben tener un orden específico para el correcto funcionamiento del modelo (Figura 15), y estar almacenados en archivos de Excel.

	A	B	C	D
1	21/02/2022	2:04:00 a. m.	6521	
2	21/02/2022	2:05:00 a. m.	5996	
3	21/02/2022	2:06:00 a. m.	5870	
4	21/02/2022	2:07:00 a. m.	5607	
5	21/02/2022	2:08:00 a. m.	5421	

Figura 15. Ejemplo de la distribución de datos para su inclusión en el aplicativo

Se incluyeron dos botones para la exportación de los resultados numéricos de la RNA en archivos de Excel, llamados “Guardar Archivo Predicción” y “Guardar Archivo Precisión”.

Finalmente, los botones de:

- “Ayuda” = Abre una ventana emergente con indicaciones de operación del aplicativo.
- “Créditos” = Abre una ventana emergente con la información del autor del aplicativo, director y codirector.
- “Cerrar” = Cierra el aplicativo.

En el panel de “Ventana de Gráficos” se ingresaron dos *UIAxes*, donde se grafican las predicciones del modelo neuronal, con dos trazos, primero son los datos ingresados en color azul, después la predicción de la red en color naranja, la segunda grafica ilustra la precisión de la red, comparando los datos predichos en contraste con los datos reales del mismo tiempo.

De la misma forma se ingresaron dos *UITable*, en donde se observarán las respuestas de tiempo y consumo de potencia activa en valores numéricos, la primera tabla son las predicciones y la segunda es la comparación de datos, como se logra evidenciar en la figura 16.

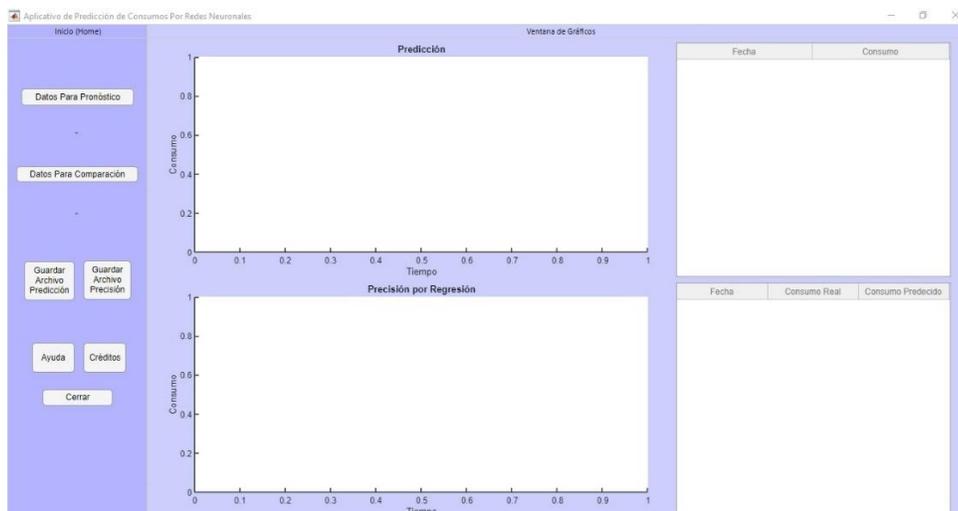


Figura 16. Diseño final del Aplicativo de Predicción de Consumos de Potencia Activa por Medio de Redes Neuronales.

4.3.7. Indicador de Calidad del Comportamiento de las Redes.

Para seleccionar el mejor modelo de red neuronal artificial se implementaron dos indicadores matemáticos, que revelan el error en el pronóstico de las redes, el principal es la regresión lineal que nos indica la precisión de la red con respecto a los datos reales, y el segundo es el error cuadrático medio que mide el promedio de los errores al cuadrado.

5. Análisis de Resultados.

5.1. Selección del Modelo Final de la RNA.

Recapitulando los experimentos realizados en el entrenamiento de las redes neuronales artificiales, se identificaron las precisiones de los modelos, al variar la cantidad de neuronas en la capa oculta con el aprendizaje profundo y además el número de épocas de entrenamiento en el caso de las redes neuronales profundas.

Con el uso de la *Neural Network Toolbox*, se incurrió en la graficación del rendimiento de las redes, como se señala en la Figura 17 con el trazado del error frente a las épocas, para los rendimientos de entrenamiento de validación y prueba del registro de entrenamiento (superior-izquierda), la graficación del histograma de los errores de la red (inferior-izquierda), la graficación de los valores de entrenamiento devueltos por la red (superior-derecha), y la graficación de la regresión lineal de las targets en relación con los outputs (inferior-derecha).

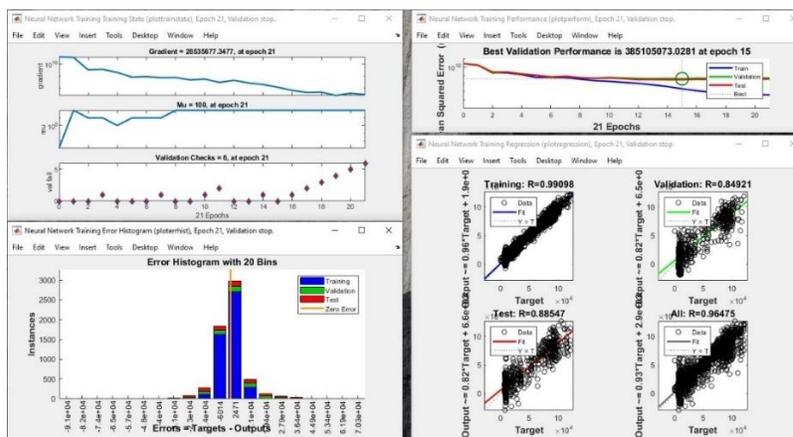


Figura 17. Graficas de los resultados devueltos por *Neural Network Toolbox* sobre la RNA.

En cambio, con *Matlab Edit*, solo se requirió la graficar de la regresión lineal de los targets de la red, y el trasado de la resta entre los valores reales y los valores predichos, además del valor RMSE.

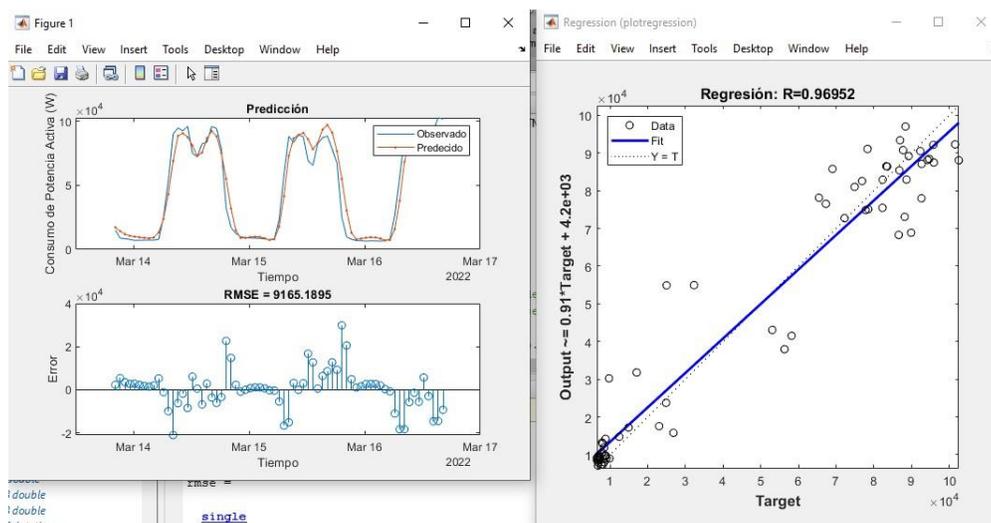


Figura 18. Graficas requerida en *Matlab Edit* sobre la RNA.

Comparando los modelos similares, el valor que tomo mayor importancia fue la progresión lineal, de los resultados de la Evaluación de la red, la observación de los resultados comprobó que las redes neuronales LSTM presentan mejores comportamientos que las Feed-Forward, En la tabla 3 se aprecia como los modelos LSTM son los más cercanos a 1, resaltando la combinación LSTM 250-200 como la red más precisa según el criterio del autor.

Tabla 3. Exposición de los resultados de las RNA diseñadas.

Arquitectura	Numero de neuronas en la capa oculta		Precisión por Regresión Lineal		Error de Raíz Cuadrático Medio	
Feed-Forward	30		0.87774		17779.8856	
	40		0.88547		19624.0941	
	50		0.83929		16715.2271	
	60		0.85688		16294.6079	
	70		0.87255		17724.0871	
	80		0.91872		12655.6351	
	90		0.93072		13842.7809	
	100		0.93671		11940.6432	
Long Short-Term Memory	Numero de neuronas en la capa oculta	Numero de Épocas máximas de entrenamiento	Precisión por Regresión Lineal		Error Cuadrático Medio	
			Red Preliminar	Red Actualizada	Red Preliminar	Red Actualizada
	150	150	0.96952	0.98779	9165.1895	5853.0193
		200	0.95755	0.98707	11372.1982	6000.2567
		250	0.85215	0.98531	20161.0625	6520.1089
		300	0.96231	0.97794	10169.7266	7870.2563
	200	150	0.98772	0.98402	7508.4761	5814.7247
		200	0.91948	0.98722	15277.6494	6033.7203
		250	0.98094	0.98733	8586.9268	5976.0328
		300	0.97549	0.98717	8670.4775	5978.1228
	250	150	0.87583	0.98852	19557.3066	5824.6083
		200	0.98286	0.98713	9681.3604	5951.5602
		250	0.95411	0.98717	11510.6641	5975.714
	300	300	0.97466	0.98368	9411.3008	6793.7277
		150	0.96177	0.98555	11598.4189	6299.1463
		200	0.94658	0.98797	13458.7305	5765.3942
250		0.96702	0.98633	9560.1143	6142.0757	
	300	0.98292	0.98484	7168.6719	6502.9665	

Se observa un aumento en la precisión de las redes Feed-Forward, con el incremento de la cantidad de neuronas de la capa oculta, pero en la práctica su tiempo de aprendizaje también aumentaba exponencialmente, revelándose en la capa oculta con 100 neuronas un transcurso de más de media hora de entrenamiento y el sistema de cómputo utilizado a punto de un bloqueo del software. Por el contrario, con las redes LSTM su tiempo de entrenamiento no superaba los 9 minutos de entrenamiento.

Al contemplar las redes LSTM se distingue que al aumentar las neuronas en la capa oculta y las épocas presenta mejoría en la precisión en los primeros casos, pero en los parámetros máximos se revela que su precisión no es la mejor en comparación con los parámetros intermedios. Dando a concebir que se pueden obtener los mejores resultados con parámetros relativamente equilibrados. Se guardo la red neuronal artificial con el código *save (net, 'nombre a elección.mat')*, en la carpeta preseleccionada con *Matlab*, el '*nombre a elección.mat*' es el mote del archivo que solicita la interfaz desarrollada.

5.2. Demostración de la Aplicación Final de la Interfaz de Predicción de Consumos de Potencia Activa por Medio de Redes Neuronales.

Para la ratificación del aplicativo se extrajo información de la base de datos principal, con el objetivo de demostrar la precisión de la red, y su funcionamiento adecuado, como se evidencia en las siguientes figuras.

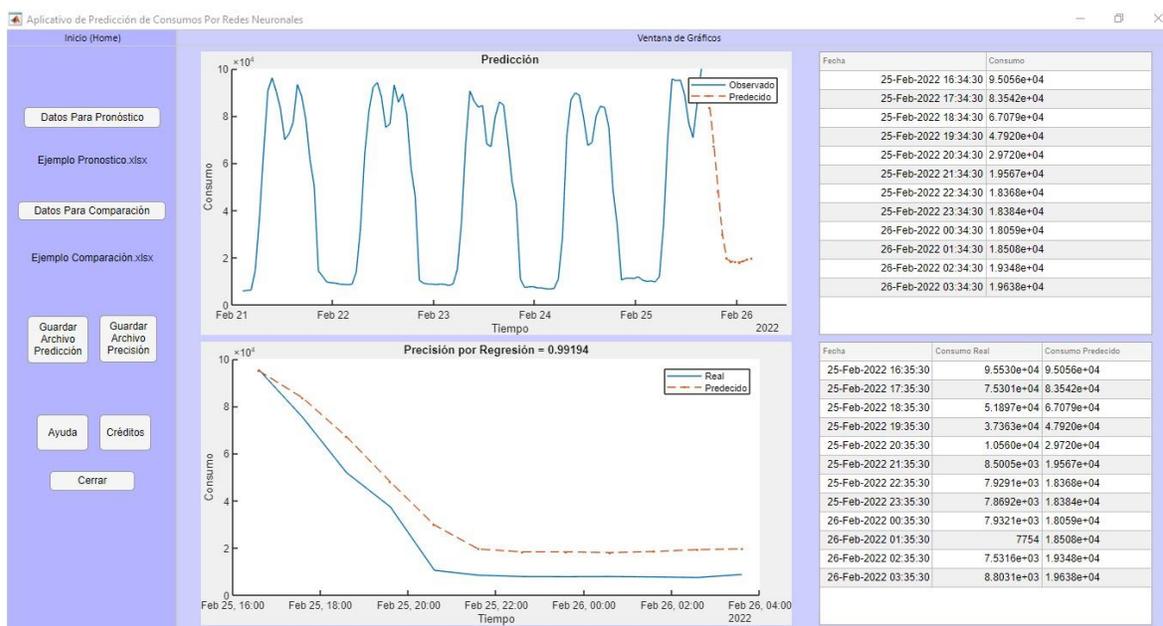


Figura 19. Demostración del Aplicativo Desarrollado.

Los resultados conseguidos por el aplicativo lograron ser muy aceptables, obteniendo una precisión del 0.99194 de regresión lineal, comparados con los datos reales, asimismo la facilidad de operación e interpretación de la información suministrada por la interfaz, y la portabilidad del ejecutable, a excepción del requerimiento de poseer el software de *Matlab*.

Conclusiones

Los resultados obtenidos demuestran que es posible realizar el pronóstico de consumo de potencia activa total del edificio administrativo-SENA, con una regresión lineal máxima de 0.99194 utilizando redes neuronales artificiales, con una primordial variable de entrada, al mismo tiempo de su implementación en un aplicativo portable, además de la posibilidad de actualizar la RNA personalizando su parametrización.

El modelo de RNA con la mayor precisión, reflejo ser de arquitectura Long Short-Term Memory, hiperparametrizada con 250 neuronas en la capa oculta y un máximo de 200 épocas de entrenamiento, además de presentar un tiempo de entrenamiento relativamente bajo con 3 minutos y 27 segundos, con una precisión con la progresión lineal de 0.98286 y 0.98713, siendo la más equilibrada y próxima a 1. Asimismo, en su implementación con el aplicativo, manifestó tener una precisión del 0.99194 con los datos ingresados, demostrando una operación sobresaliente como modelo principal.

En la comparación, los modelos neuronales LSTM resultaron ser más exactos que los modelos Feed-Forward priorizando su tiempo de entrenamiento desarrollado, especialmente el citado en el párrafo anterior, que, para la interfaz enfocada, rozaba la predicción de consumo eléctrico exacto.

La estrategia de predicción de la potencia activa total del edificio administrativo-SENA, constituye un punto de referencia, estimulando el desarrollo de planes de eficiencia energética, además de apelar a datos muy precisos con las técnicas de RNA, óptimos para la comparación con datos posteriores a la ejecución de ahorros energéticos.

El uso de la caja de herramientas *Neural Network Toolbox*, facilita la creación de una red neuronal simple, además de su interpretación, pero sacrificando la libertad de edición, como en la

modificación de la cantidad de capas ocultas, o algoritmos de entrenamiento implementados a parte de los ya instalados, conjuntamente su aprendizaje toma más tiempo, en comparación con las RNA programadas en *Matlab Edit*, que permite la modificación total de los hiperparámetros, pero su complejidad de implementación es considerable.

El diseño de la interfaz con *App Designer* refleja ser intuitivo en la selección de componentes didácticos y gráficos, pero la programación de estos llegó a ser engorrosa, debido a que cada componente presenta un código aislado, interponiéndose principalmente en la programación del botón “Datos Para Comparación”, puesto que se necesitaba los datos del primero botón “Datos Para Pronóstico”, la solución aplicada fue extraer la *Data* de la tabla de pronóstico “UITable” y esta implementarla como una variable importada.

En general, los modelos basados en Deep Learning, tuvieron un rendimiento de predicción más estable y preciso, en comparación con los modelos basados en Machine Learning. A pesar de su excelente rendimiento, el modelo LSTM está muy influenciado por los ajustes de hiperparámetros y al no poseer una guía exacta para sus elecciones, reside en el criterio y conocimiento del autor.

Recomendaciones

Este trabajo considera información del Servicio Nacional de Aprendizaje en la Ciudad de Cúcuta, específicamente el edificio administrativo, también se pueden tomar en evaluación datos de otros edificios burocráticos o de enseñanza, para la creación de modelos basados en redes neuronales artificiales similares al propuesto, ya que presentan comportamientos análogos en el consumo de potencia activa total.

Aumentar la cantidad de información en datos históricos de consumo de energía, para ensanchar la capacidad de predicción de los modelos neuronales, con metas de pronóstico a largo plazo.

El ingreso de las series de tiempo debe ser verídicas, dado que el modelo de red neuronal es tipificado a un ambiente específico, porque si se presentan datos que no contengan similitud al campo aplicado, se pone en juicio la autenticidad del modelo propuesto.

Para modelos neuronales de mayor complejidad basados en el aprendizaje automático se aconseja usar equipos de cómputo con características de procesamiento, memoria RAM, memoria interna, en óptimas condiciones y de mayor rendimiento, dado que las operaciones computacionales son el escenario de aprendizaje de las RNA, y su tiempo de ejecución es inversamente proporcional al rendimiento computacional.

Poseer conocimientos en el campo de la inteligencia artificial y programación en el lenguaje de *Matlab*, para el diseño y mejora de los modelos basados en RNA, simultáneamente contar con variables fidedignas y herramientas computacionales optimas.

Trabajos Futuros

Los siguientes estudios a futuro se deben concentrar, primero, en el mejoramiento del modelo neuronal, explorando arquitecturas distintas a las ya desarrolladas en este proyecto, o agregando variables de mayor impacto que establezca una predicción más personalizada, con alcances superiores en el tiempo.

Segundo, mejorar la capacidad de interacción del aplicativo a desarrollar, que permita la actualización de la RNA ingresando datos verídicos, exponga mayor información en el análisis del consumo de energía eléctrica, anuncie sugerencias de estrategias enfocadas en la eficiencia energética en distintos tiempos del pronóstico de consumo de energía.

Tercero, diseñar sistemas de predicción en campos diferentes, como el pronóstico de temperaturas medioambientales, predicción del nivel de humedad e iluminación en zonas de cultivo, previsión del consumo de energía eléctrica en zonas industriales, predicción del consumo de agua potable en zonas particulares, pronóstico del rendimiento o vida útil de equipos electromecánicos análogos.

Cuarto, exportar los modelos basados en RNA diseñados en Matlab a otro software como Python, o explorar la portabilidad de los aplicativos a equipos móviles más compactos.

Referencias

- Aguirre, W.A.L. (2019). Espectroscopia con infrarrojo y técnicas de Machine learning y Deep learning para la detección y clasificación de frutas para la agroindustria. Caso: arándanos - Empresa Talsa - 2018. (Tesis de Maestría, Universidad Privada Antenor Orrego). Repositorio Digital de la UPAO. <https://hdl.handle.net/20.500.12759/5106>.
- Berzal, F. (2018). Redes neuronales & deep learning: Volumen I. Independently published.
- Cabeza, L.F., Rincón, L., Vilarino, V., Pérez, G., Castell, A. (2014). Life cycle assessment (LCA) and life cycle energy analysis (LCEA) of buildings and the building sector: a review. *Renew. Sunstein. Energy Rev.* 29, 394–416.
- Cao, Z., Liu, Gang, Duan, H., Xi, F., Liu, Guiwen, Yang, W. (2019). Unravelling the mystery of Chinese building lifetime: a calibration and verification based on dynamic material flow analysis. *Appl. Energy* 38, 442–452.
- Carbajal-Hernández, J. J., Sánchez-Fernández, L. P., Hernández-Bautista, I., & Hernández-López, J. (2017). Modelo basado en redes neuronales artificiales para la evaluación de la calidad del agua en sistemas de cultivo extensivo de camarón. *Tecnología y ciencias del agua*, 8(5), 71-89. Epub 30 de julio de 2021. <https://doi.org/10.24850/j-tyca-2017-05-05>.
- Castro, A. J., & Mejía G., M. I. (2022). (Tesis de Grado, Universidad de Guayaquil). Repositorio Digital UG. <http://repositorio.ug.edu.ec/handle/redug/60210>.
- Commission E. COMMUNICATION from the commission A clean planet for all. (2018). EUR-Lex -52018DC0773 - EN - EUR-Lex. <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52018DC0773>.

- Commission E. COMMUNICATION from the commission clean energy for all Europeans. (2016). EUR-Lex -52016DC0860 - EN - EUR-Lex. <https://eur-lex.uropa.eu/legal-content/en/TXT/?uri¼CELEX:52016DC0860>.
- Condeixa, K., Haddad, A., Boer, D. (2017). Material flow analysis of the residential building stock at the city of Rio de Janeiro. *J. Clean. Prod.* 149, 1249–1267.
- Dixit, M.K. (2017). Life cycle embodied energy analysis of residential buildings: a review of literature to investigate embodied energy parameters. *Renew. Sustain. Energy Rev.* 79, 390–413.
- Donoso O., C. (2018). Phased long short term memory para clasificación de objetos variables muestreados irregularmente. (Tesis de Grado, Universidad de Concepción). Repositorio Digital UC. <http://repositorio.udec.cl/jspui/handle/11594/3340>.
- Du, M., Wang, B., Zhang, N. (2018). National research funding and energy efficiency: evidence from the national science foundation of China. *Energy Policy* 120 (December 2017), 335–346.
- Fazeli, R., Davidsdottir, B. (2017). Energy performance of dwelling stock in Iceland: system dynamics approach. *J. Clean. Prod.* 167, 1345–1353.
- Flores Claros, I.F., Herrera Calderon, J.A. (2018). Modelo Predictivo de la Demanda de Energía Eléctrica en el Campus Central de la Universidad de El Salvador. Tesis de Ingeniería Eléctrica. Universidad de El Salvador.
- Geng, S., Wang, Y., Zuo, J., Zhou, Z., Du, H., Mao, G. (2017). Building life cycle assessment research: a review by bibliometric analysis. *Renew. Sustain. Energy Rev.* 76, 176–184.

- Guo, Q., Wu, Y., Ding, Y., Feng, W., Zhu, N. (2016). Measures to enforce mandatory civil building energy efficiency codes in China. *J. Clean. Prod.* 119, 152–166.
- Hasan, A.S.M.J., Tuhin, R.A., Ullah, M., Sakib, T.H., Thollander, P., Trianni, A. (2021). A comprehensive investigation of energy management practices with in energy intensive industries in Bangladesh. *Energy*, 232, 120932.
- Isasi-Viñuela, P., y Galván-León, I.M. (2004). *Redes de neuronas artificiales. Un enfoque práctico.* Pearson Prentice Hall. Madrid.
- Jin, X.W., Cai, S.Z., Li, H., and Karniadakis, G.E. (2021). NSFnets (Navier-Stokes flow nets): Physics-informed neural networks for the incompressible Navier-Stokes equations, *J. Comput. Phys.*, 426, 109951.
- Kangas, H.L., Lazarevic, D., Kivimaa, P. (2018). Technical skills, disinterest, and non-functional regulation: barriers to building energy efficiency in Finland viewed by energy service companies. *Energy Policy* 114, 63–76.
- Koskimäki, P.L., (2012). Africa could take a leap to energy efficiency: what lessons could Sub-Saharan countries learn from European energy efficiency policy implementation *Energy Sustain. Dev.* 16 (2), 189–196.
- Malinauskaite J, Jouhara H, Ahmad L, Milani M, Montorsi L, Venturelli M. (2019). Energy efficiency in industry: EU and national policies in Italy and the UK. *Energy*. 172:255.69
- Malinauskaite, J., Jouhara, H., Egilegor, B., Al-Mansour, F., Ahmad, L., Pusnik, M. (2020). Energy efficiency in the industrial sector in the EU, Slovenia, and Spain, *Energy*, Volume 208, 118398.

Mathworks. (2022). Levenberg-Marquardt backpropagation (r2022a). Recuperado el Agosto 16, 2022 de <https://la.mathworks.com/help/deeplearning/ref/trainlm.html>.

Mathworks. (2022). Training options for Adam optimizer (r2022a). Recuperado el Agosto 16, 2022 de https://la.mathworks.com/help/deeplearning/ref/nnet.cnn.trainingoptionsadam.html?s_tid=srchtitle_adam_1.

Maziar, R., Perdikaris P. and Karniadakis, G.E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *J. Comput. Phys.*, 378, 686–707.

Mining and Energy Planning Unit [Unidad de planeación Minero Energética - UPME]. (2016). Plan de acción indicativo de eficiencia energética PAI-PROURE 2017-2022, 157.

Mining and Energy Planning Unit [Unidad de planeación Minero Energética - UPME]. (2020). Incentivos Eficiencia Energética - Normatividad.

Ministry of Housing City and Territory [Ministerio de Vivienda Ciudad y Territorio]. (2015). Resolucion Nro 0549 de 2015. Minist. Vivienda Ciudad y Territ.

Mosquera, R., Castrillón, O. D., & Parra, L. (2018). Predicción de Riesgos Psicosociales en Docentes de Colegios Públicos Colombianos utilizando Técnicas de Inteligencia Artificial. *Información tecnológica*, 29(4), 267-280. <https://dx.doi.org/10.4067/S0718-07642018000400267>.

- Mostafa, N., Ramadan, H. S. M., Elfarouk, O. (2022). Renewable energy management in smart grids by using big data analytics and machine learning, *Machine Learning with Applications*, Volume 9, 100363, <https://doi.org/10.1016/j.mlwa.2022.100363> .
- Ocampo, E. M. T., Giraldo, D. A. M., & Isaza, H. S. (2004). Pronóstico de ventas usando redes neuronales. *Scientia et technica*, 10(26), 25-30.
- Packwood, D., Nguyen, L.T. H., Cesana, P., Zhang, G., Staykov, A., Fukumoto, Y., Nguyen, D. H. (2022). Machine Learning in Materials Chemistry: An Invitation, *Machine Learning with Applications*, Volume, 100265, <https://doi.org/10.1016/j.mlwa.2022.100265>.
- Pang, G.F., D'Elia, M., Parks, M., Karniadakis G.E. (2020). nPINNs: Nonlocal physics-informed neural networks for a parametrized nonlocal universal Laplacian operator, *Algorithms and applications*, *J. Comput. Phys.*, 422, 109760.
- Pang, G.F., Lu, L., Karniadakis, G.C. (2019). fPINNs: Fractional physics-informed neural networks, *SIAM J. Sci. Comput.*, 41, A2603–A2626.
- Pérez del Barrio, A., Fernández-Miranda, P.M., Sanz B., P., Lloret I., L., Rodríguez G., D. (2022). Inteligencia artificial en Radiología: introducción a los conceptos más importantes. *Radiología*, Volume 64, Issue 3, Pages 228-236, <https://doi.org/10.1016/j.rx.2022.03.003>.
- Ren, Y.M., Alhajeri, M.S., Luo, J., Chen, J.L., Abdullah, F., Wu, Z., Christofides, P.D. (2022). A Tutorial Review of Neural Network Modeling Approaches for Model Predictive Control. *Computers & Chemical Engineering*, 30, 107956.

- Rincón, L., Castell, A., Pérez, G., Solé, C., Boer, D., Cabeza, L.F. (2013). Evaluation of the environmental impact of experimental buildings with different constructive systems using Material Flow Analysis and Life Cycle Assessment. *Appl. Energy* 109, 544–552.
- Rodríguez González, Joselyn, & Ugalde Saborio, Edgar. (2021). Impacto de la estandarización y escalado: factor para predicción de costos en proyectos a través de una red neuronal artificial. *Ingeniare. Revista chilena de ingeniería*, 29(2), 265-275. <https://dx.doi.org/10.4067/S0718-33052021000200265>
- Rodriguez O., K. A., & Monsalve E., D. E. (2019). Algoritmo de una red neuronal profunda para la predicción de riesgos de morbilidad materna extrema (MME) en la ESE Clínica de Maternidad Rafael Calvo (Tesis de Grado, Universidad del Sinú, seccional Cartagena). Repositorio Digital UniSinú. <http://repositorio.unisinucartagena.edu.co:8080/xmlui/handle/123456789/89>.
- Sahlol, D.G., Elbeltagi, E., Elzoughiby, M., Abd Elrahman, M. (2020). Sustainable building materials assessment and selection using system dynamics. *J. Build. Eng.* 35, 101978.
- Sheng, J., and Yang, C. (2021). PFNN: A penalty-free neural network method for solving a class of second-order boundary-value problems on complex geometries, *J. Comput. Phys.*, 428, 110085.
- Sinh, M.P., Singh, G. (2021). Two phase learning technique in modular neural network for pattern classification of handwritten Hindi alphabets. *Machine Learning Applications*, 6, 100174.
- Sudha, C., Suresh, D., Nagesh, A. (2022). Accurate data aggregation created by neural network and data classification processed through machine learning in wireless sensor networks,

Theoretical Computer Science, Volume 925, Pages 25-36,

<https://doi.org/10.1016/j.tcs.2022.04.020>.

Tumbaz, M.N.M., Moğulkoç, H.T. (2018). Profiling energy efficiency tendency: a case for Turkish households. *Energy Policy* 119 (January), 441–448.

<https://doi.org/10.1016/j.enpol.2018.04.064>.

Union Oj of the E. Directive (EU) 2018/2002 OF the EUROPEAN parliament and OF the council – of 11 December 2018 - amending Directive 2012/27/EU on energy efficiency; 2018.

Union Oj of the E. DIRECTIVE 2012/27/EU OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 25 October 2012 on energy efficiency, amending Directives 2009/125/EC and 2010/30/EU and repealing Directives 2004/8/EC and 2006/32/EC. EUR-Lex - 32012L0027 - EN – EUR Lex; 2012. <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=4celex%3A32012L0027>.

Urrego, C. A. C. (2022). Pronóstico de las exportaciones de productos del sector agropecuario y la industria manufacturera de Colombia basado en series de tiempo. (Tesis de Maestría, Universidad de Guanajuato). Repositorio Digital de UG.

<http://repositorio.ugto.mx/handle/20.500.12059/6484>.

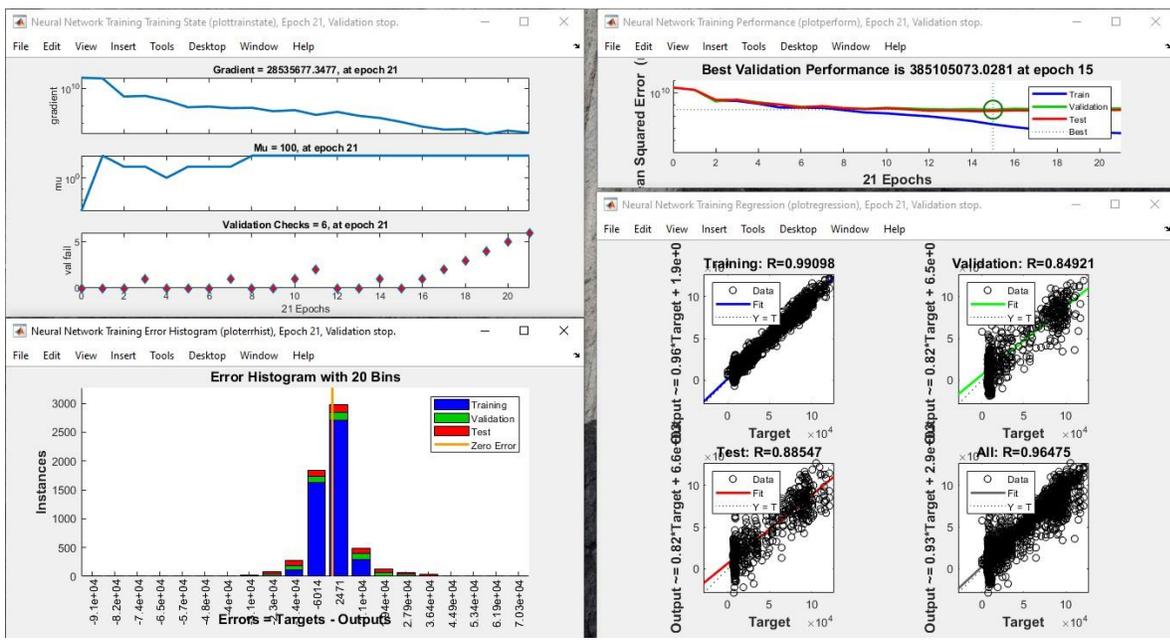
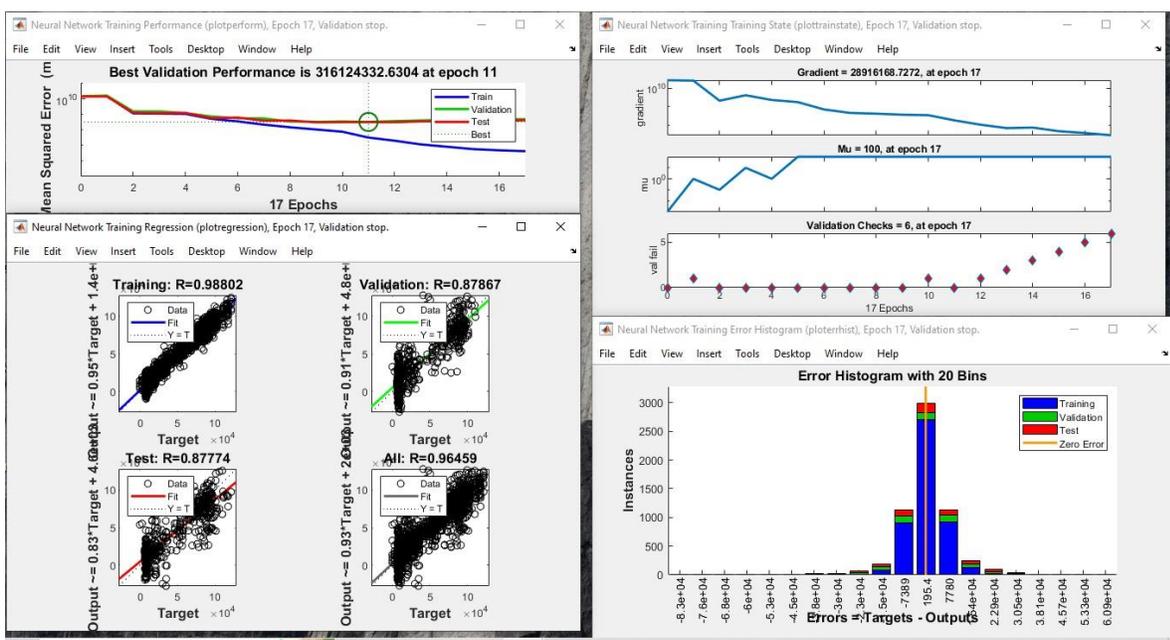
Vivas, H. Martínez, H. J., & Pérez, R. (2014). Método secante estructurado para el entrenamiento del perceptrón multicapa. *Revista de Ciencias*, 18(2), 131-150. Retrieved August 11, 2022, from

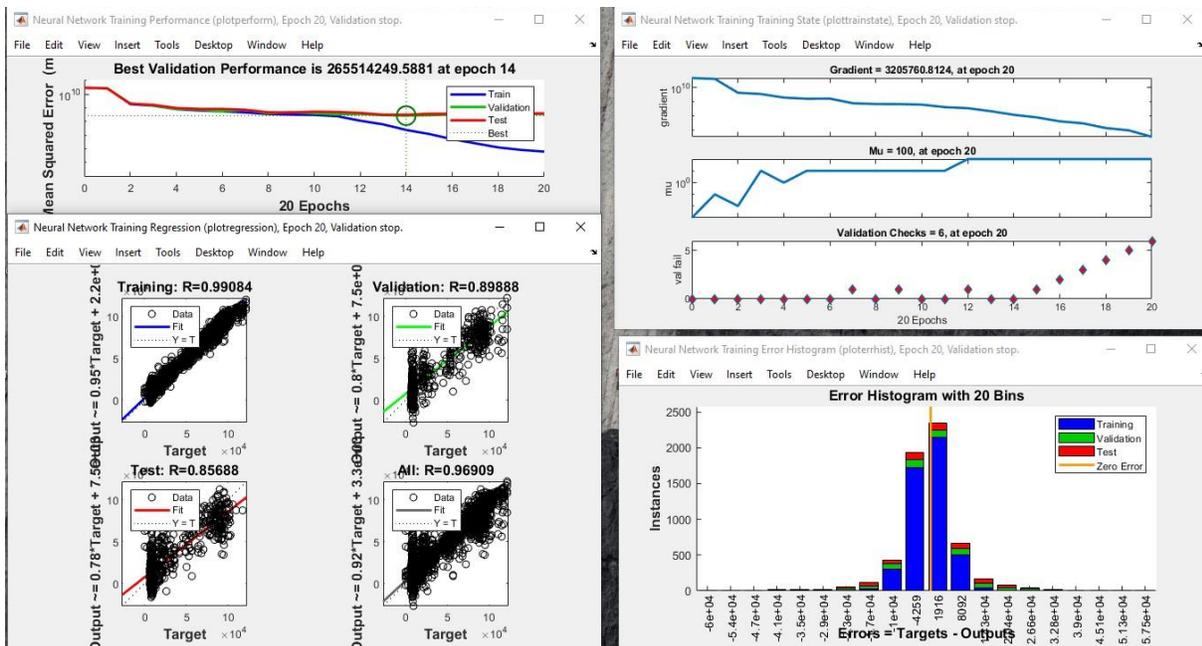
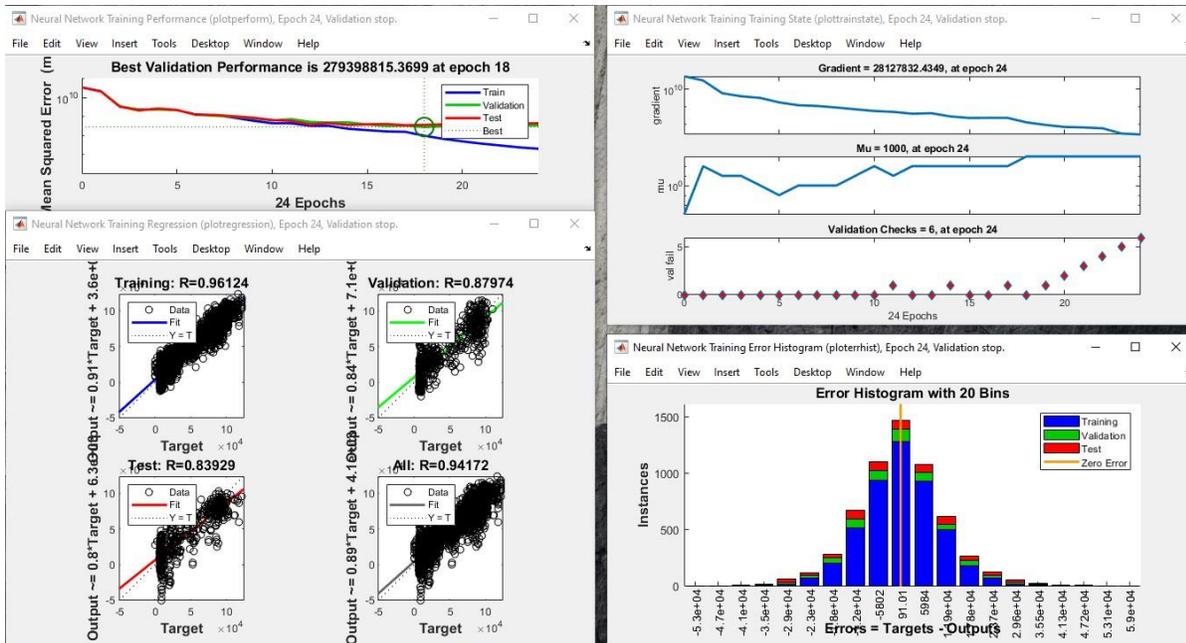
http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0121-19352014000200010&lng=en&tlng=es.

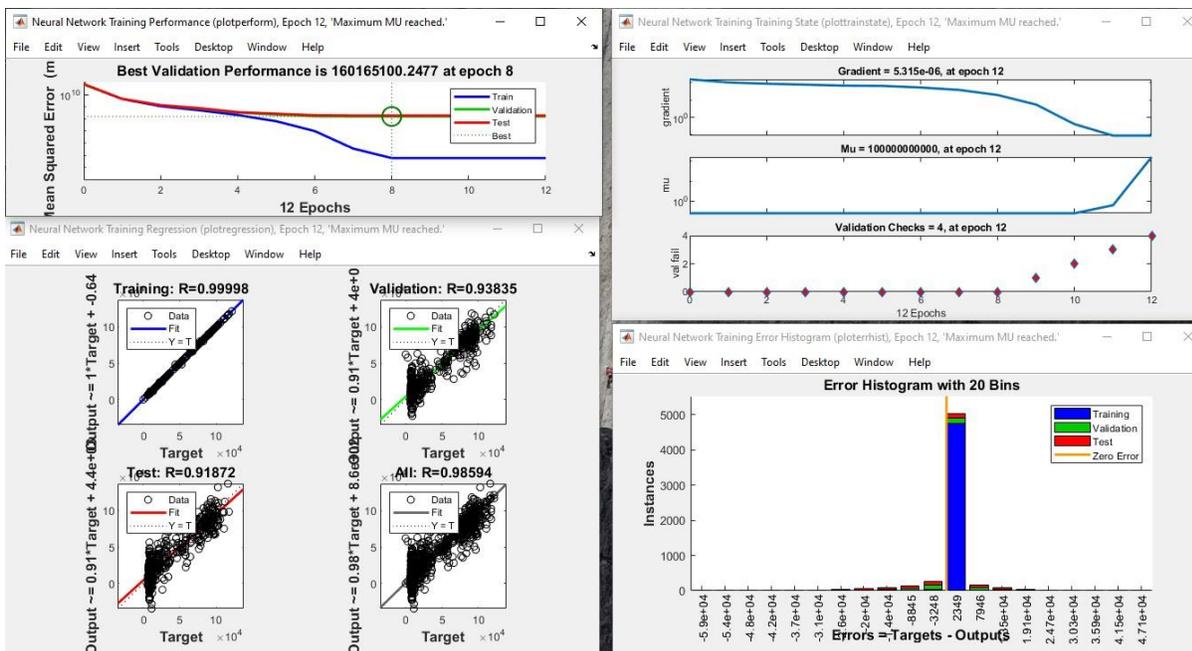
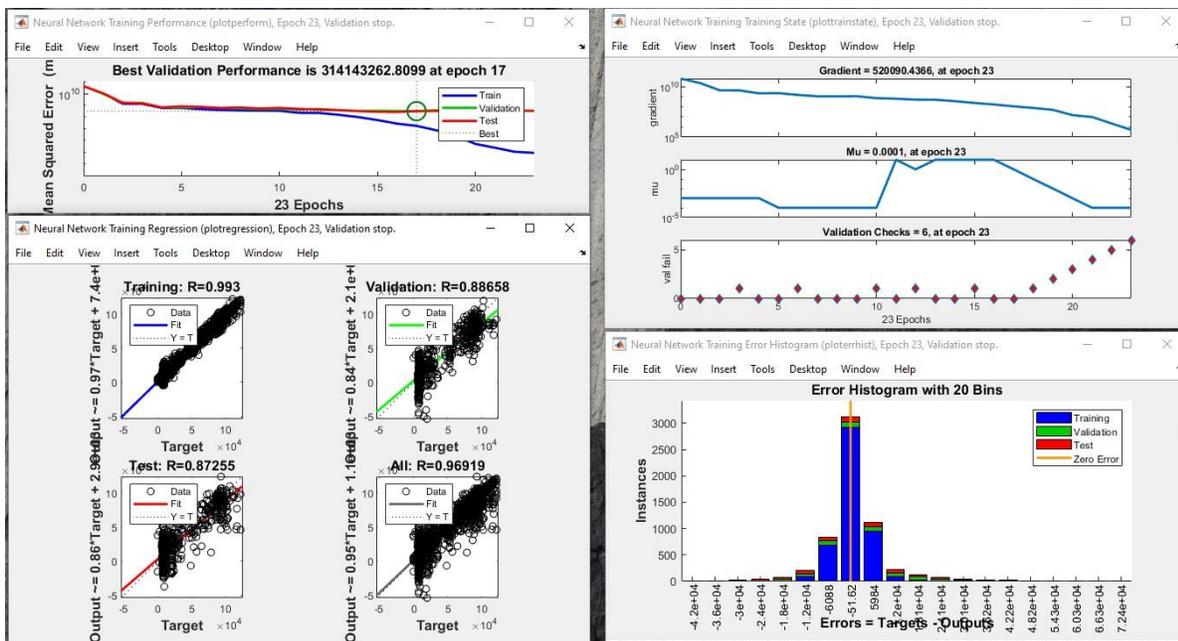
Xue, Y., Jiang, P., Neri, F., & Liang, J. (2021). A multi objective evolutionary approach based on graph-in-graph for neural architecture search of convolutional neural networks. *International Journal of Neural Systems*, 31, 2150035.

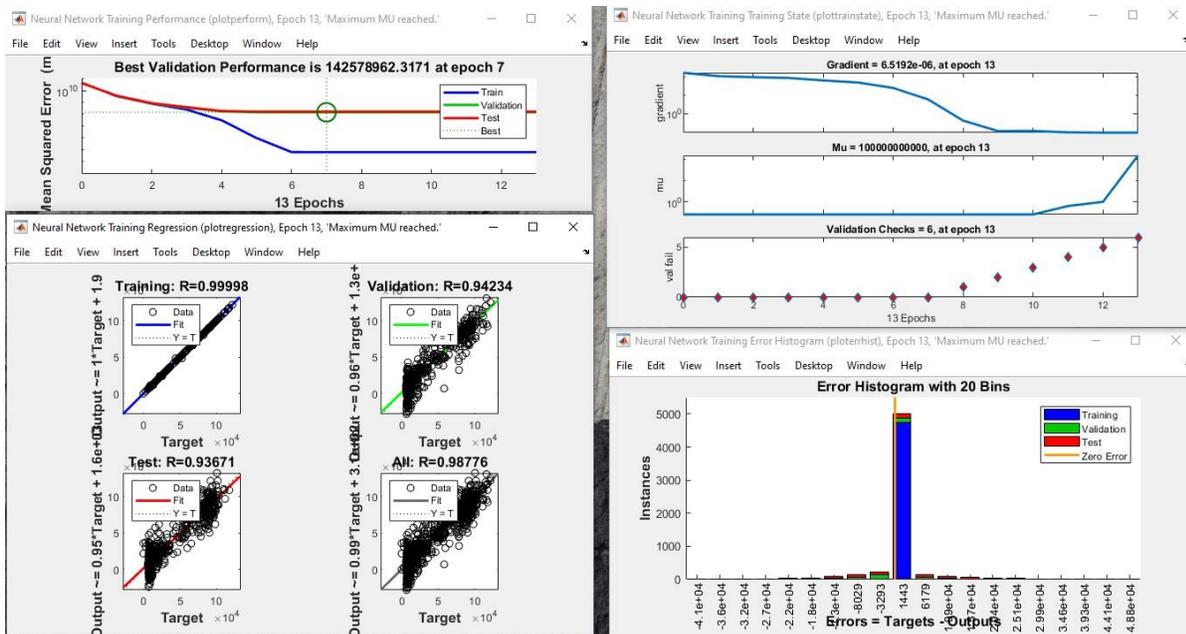
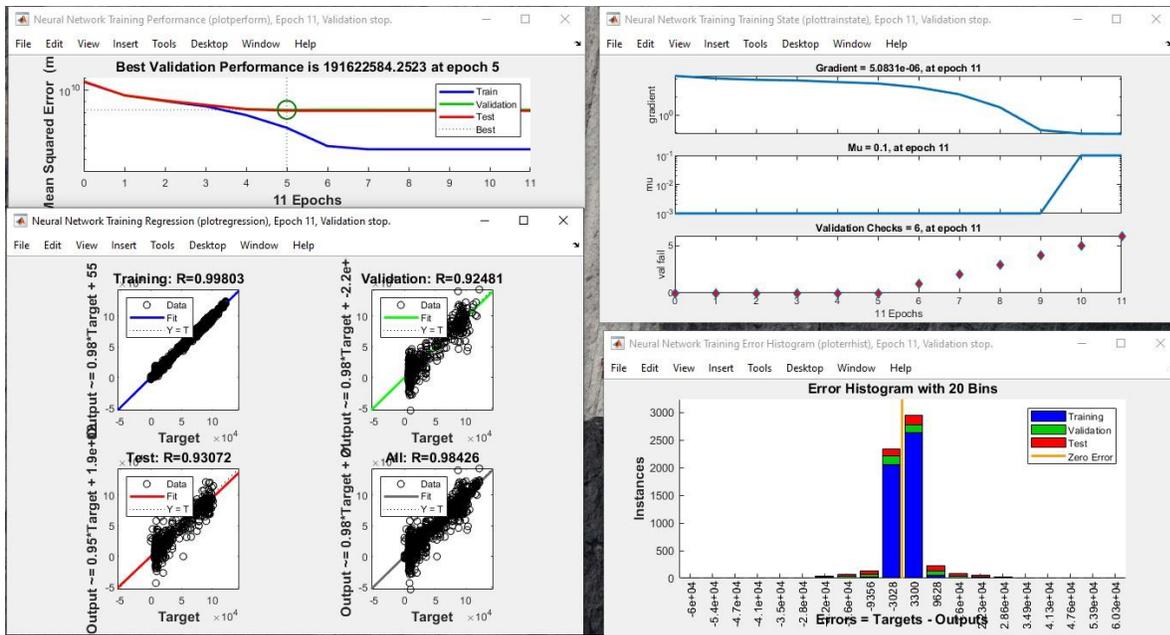
Anexos

Anexo 1. Graficas de los Resultados Obtenidos por Neural Network Toolbox Sobre la RNA de 40 a 100 Neuronas en la Capa Oculta.

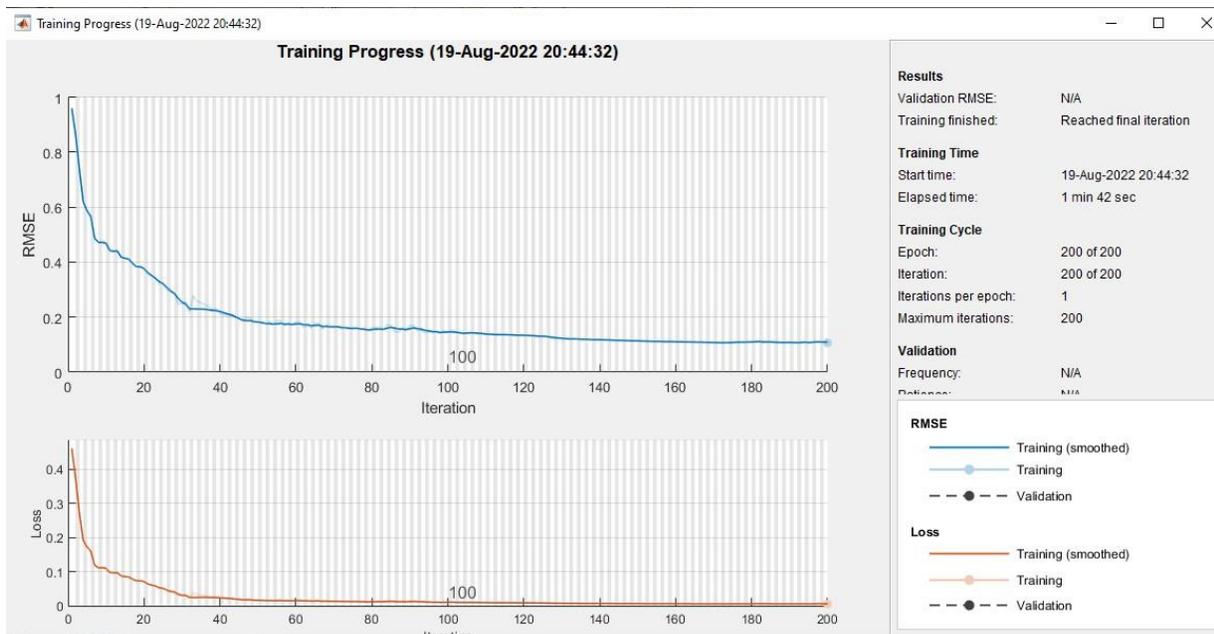
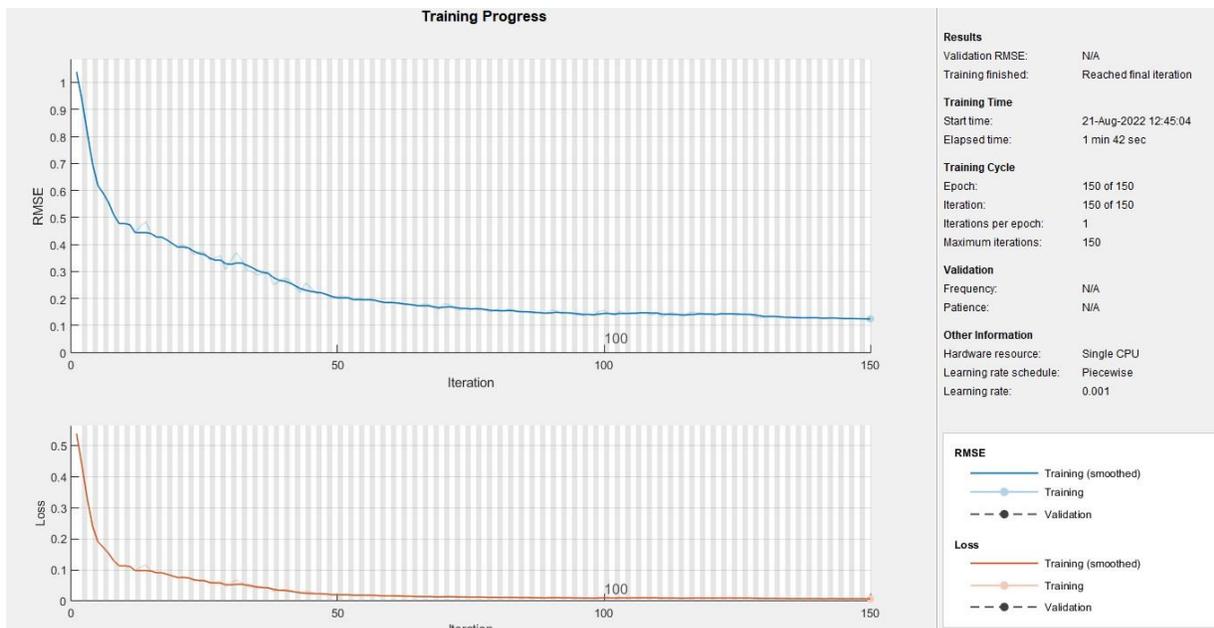


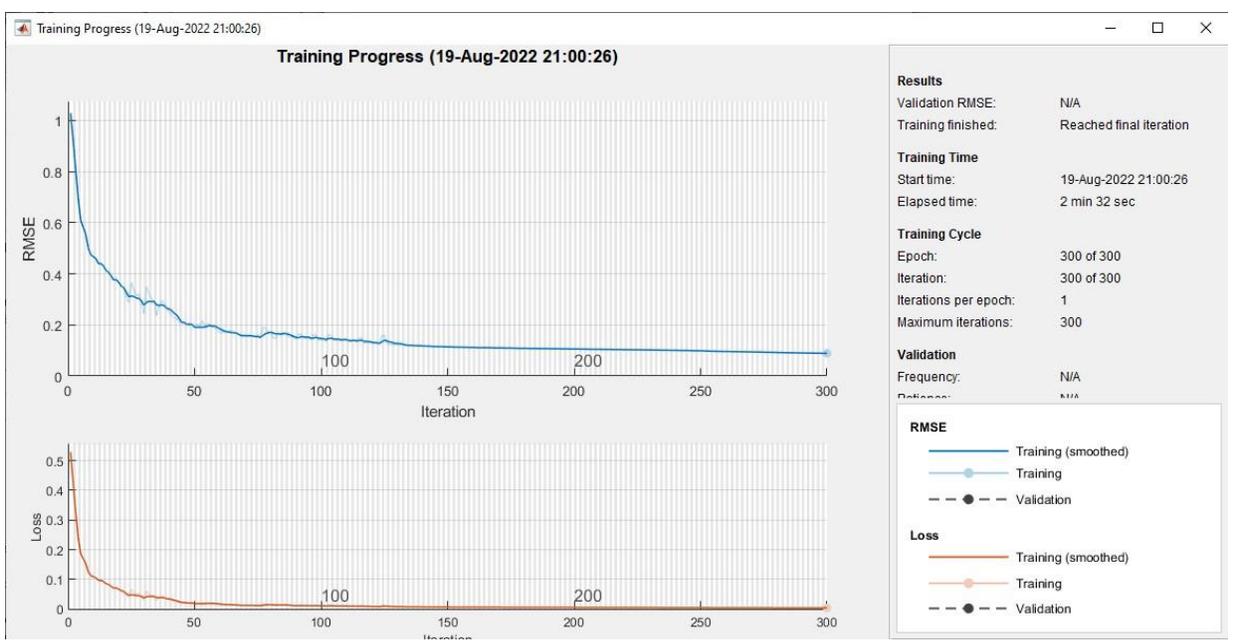
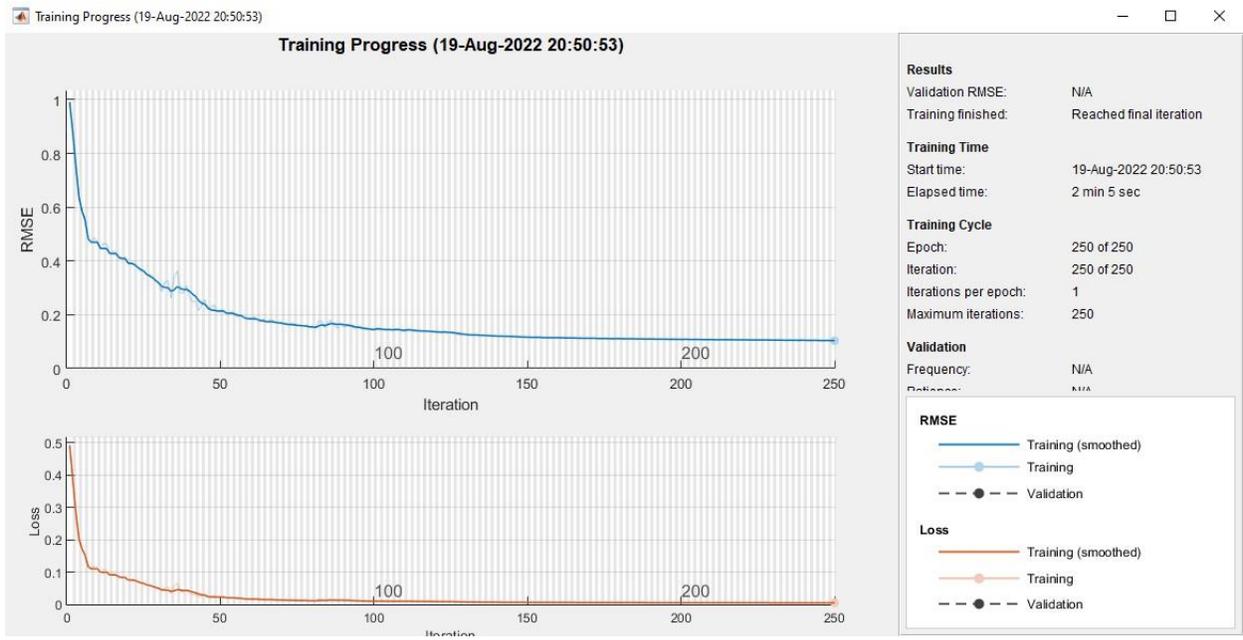




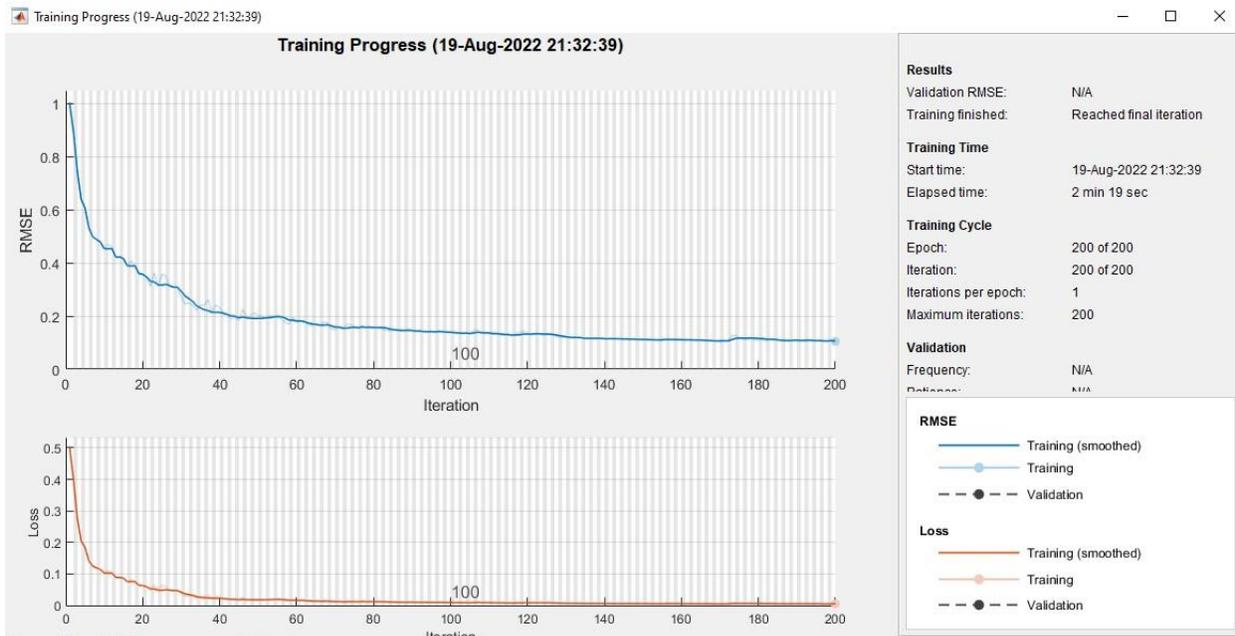
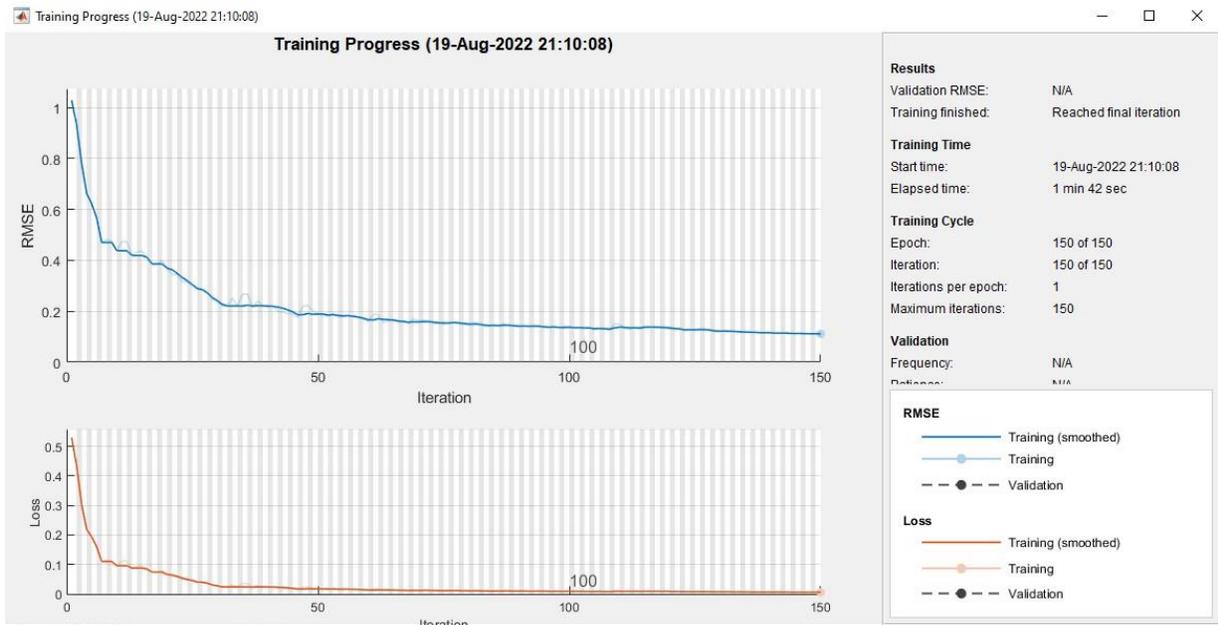


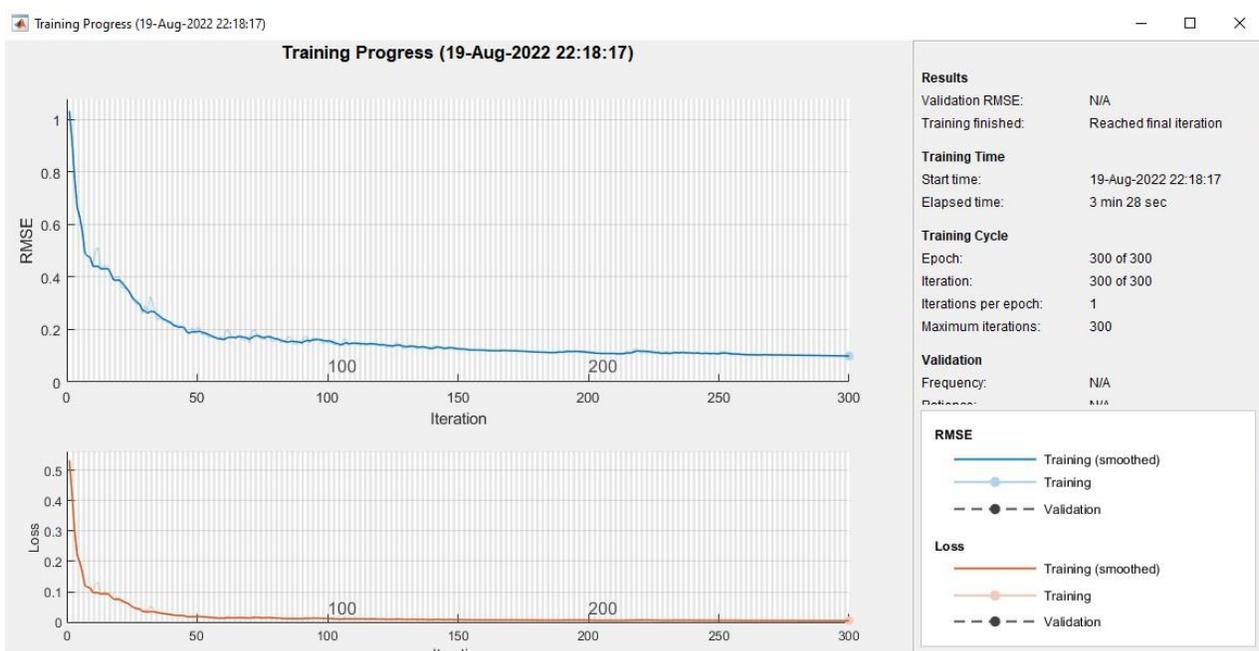
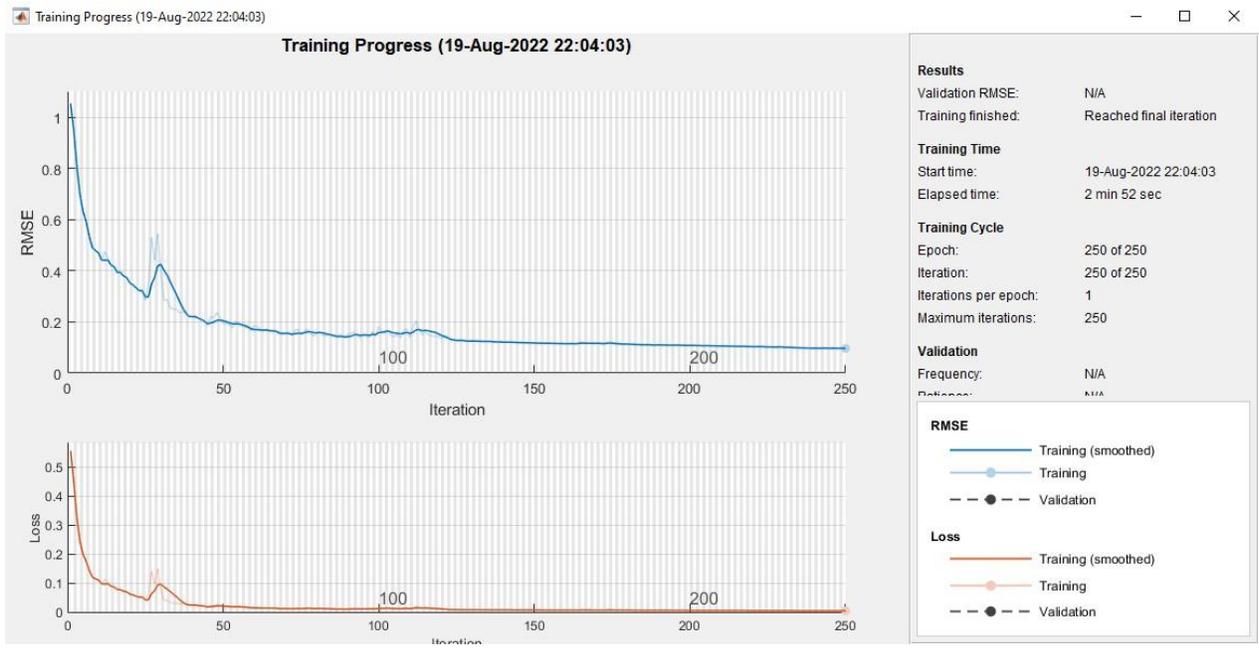
Anexo 2. Ventana del progreso de entrenamiento de la RNA con Arquitectura LSTM con 150 Neuronas Ocultas y 150 a 300 Épocas Máximas.



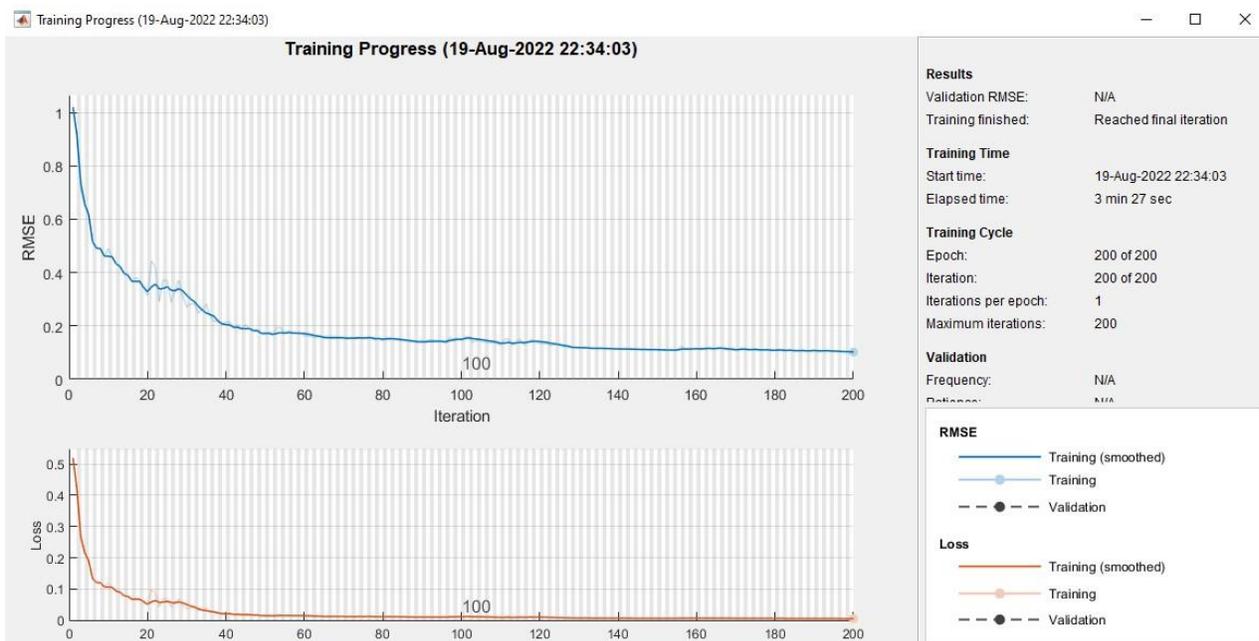
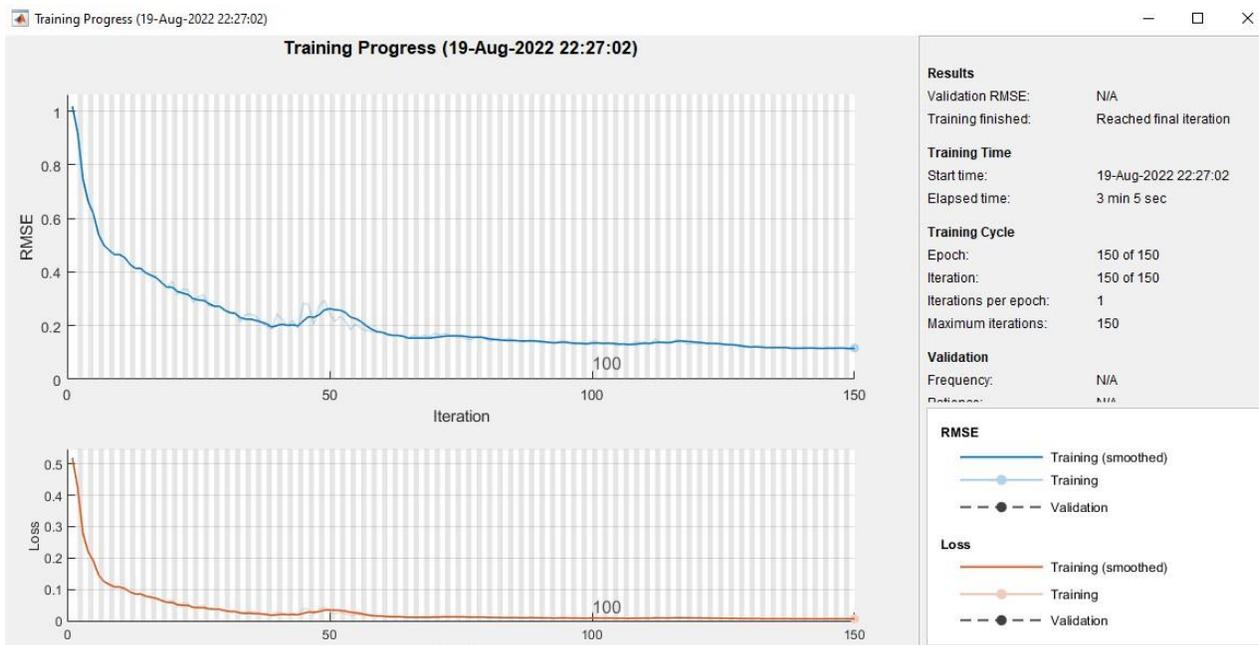


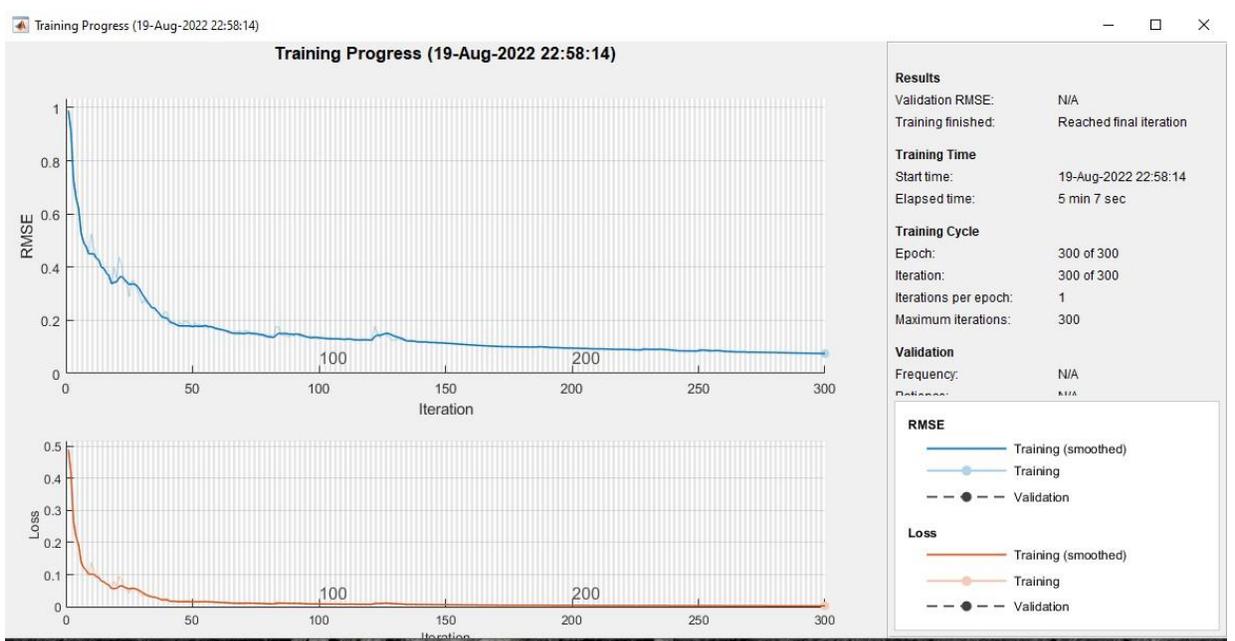
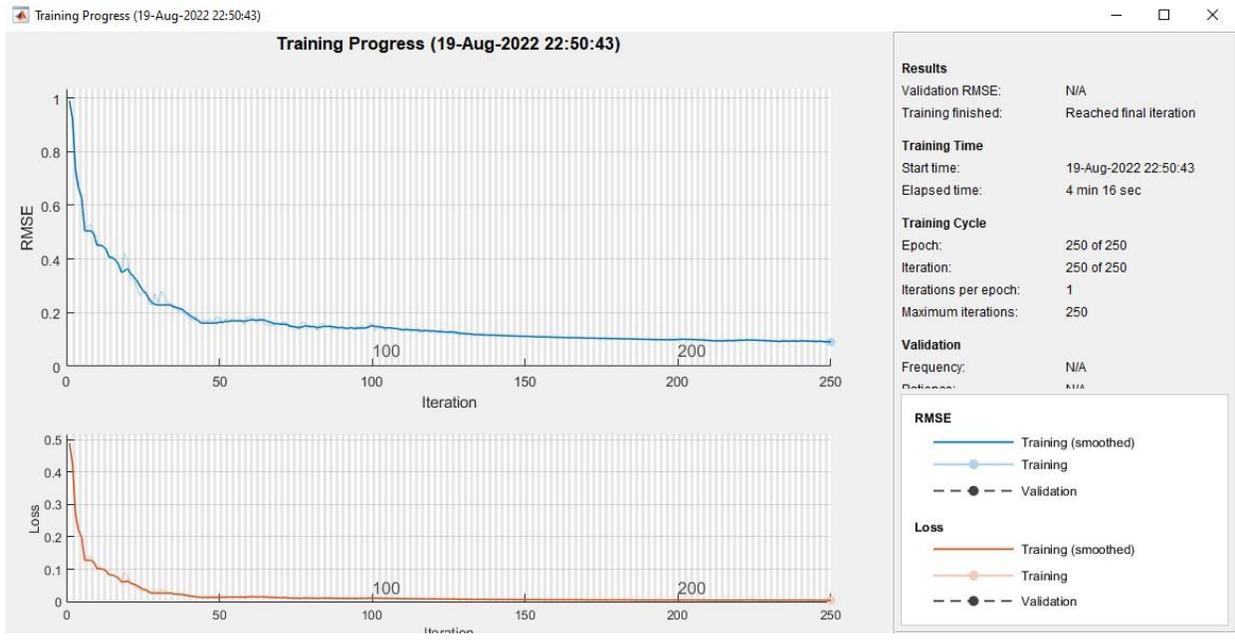
Anexo 3. Ventana del progreso de entrenamiento de la RNA con Arquitectura LSTM con 200 Neuronas Ocultas y 150 a 300 Épocas Máximas.



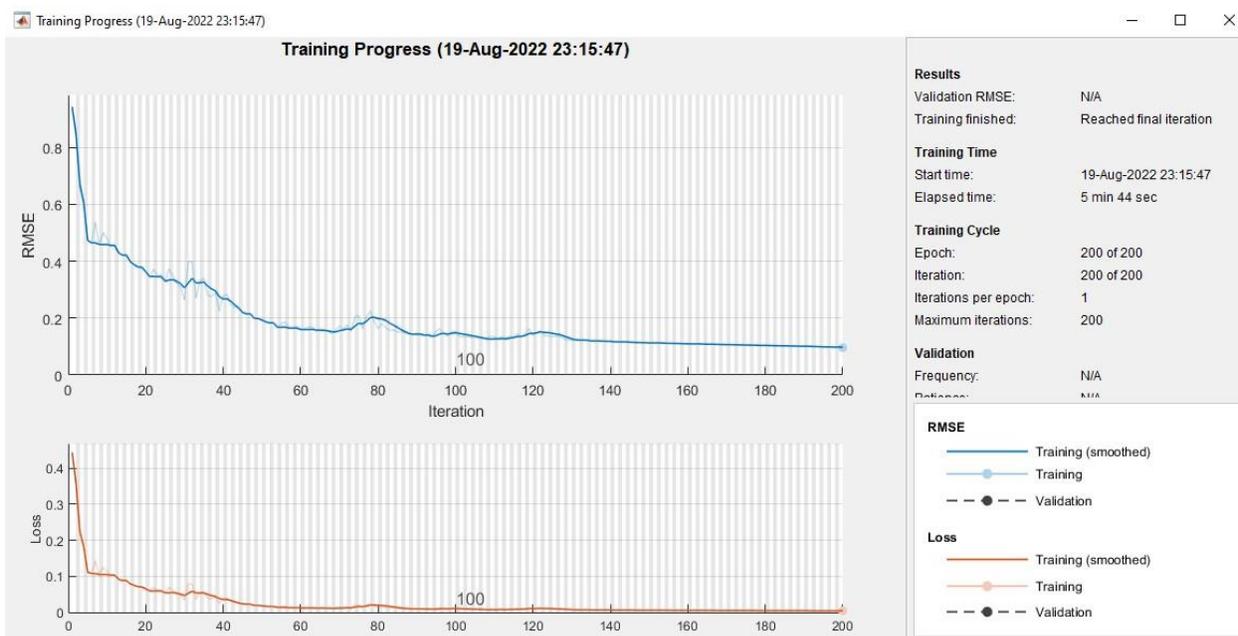
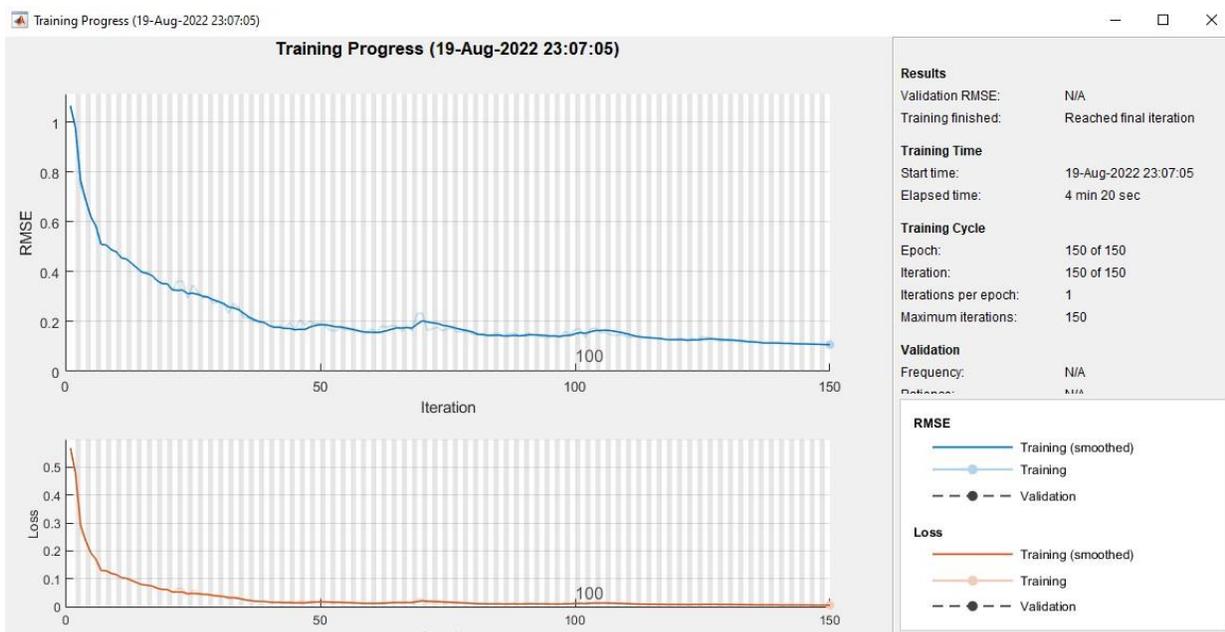


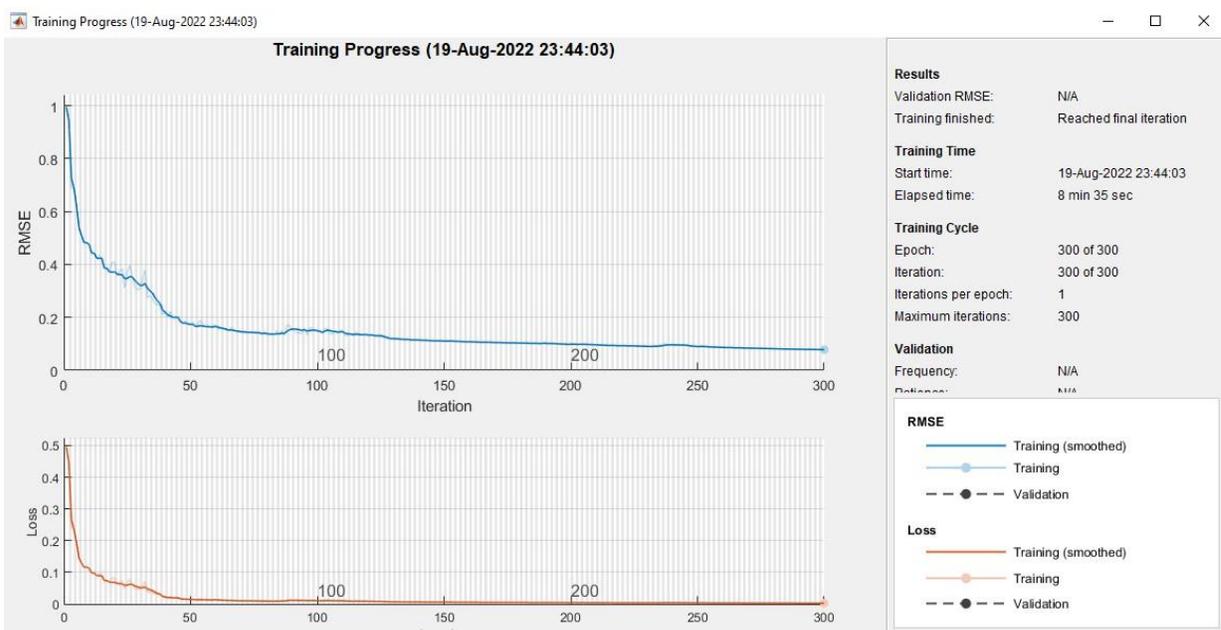
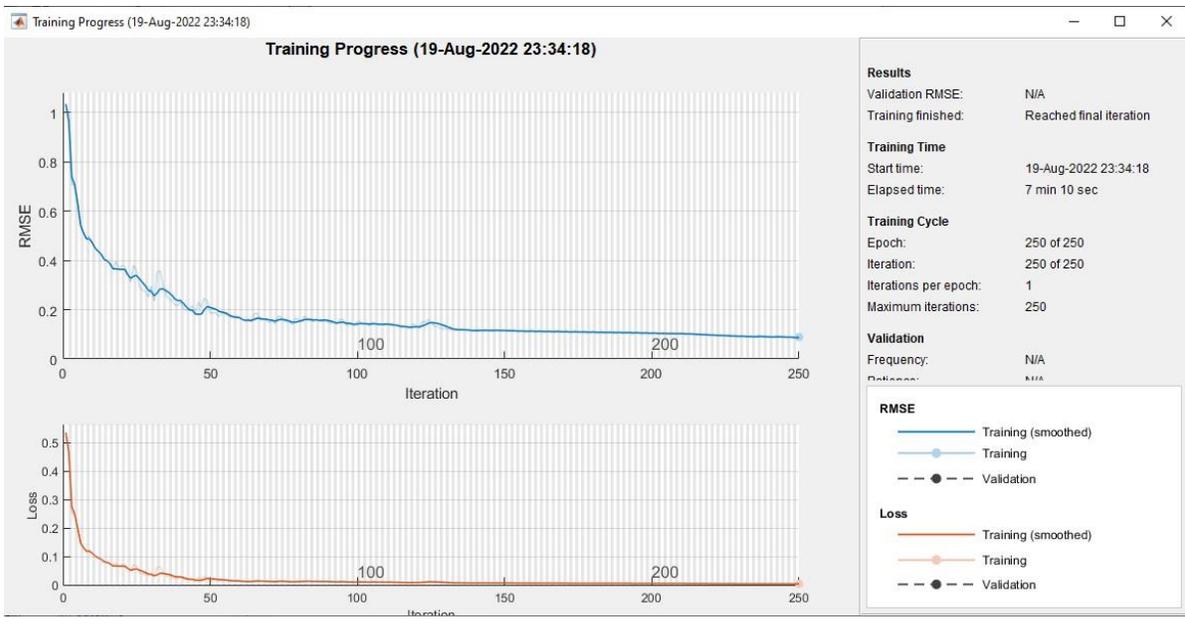
Anexo 4. Ventana del progreso de entrenamiento de la RNA con Arquitectura LSTM con 250 Neuronas Ocultas y 150 a 300 Épocas Máximas.



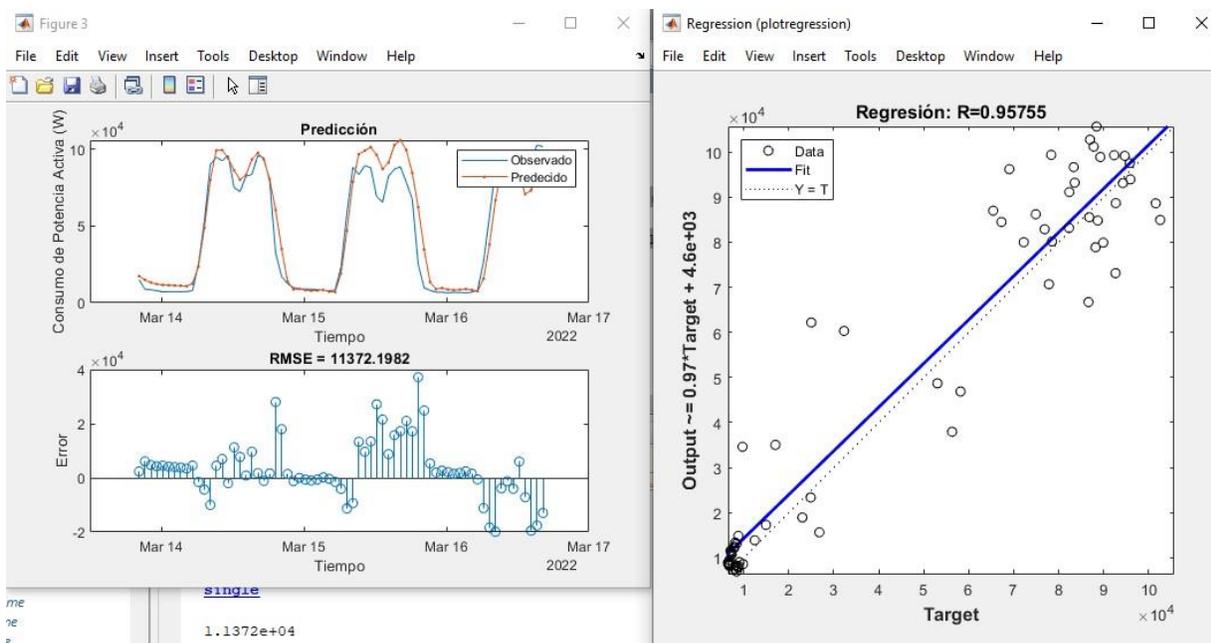
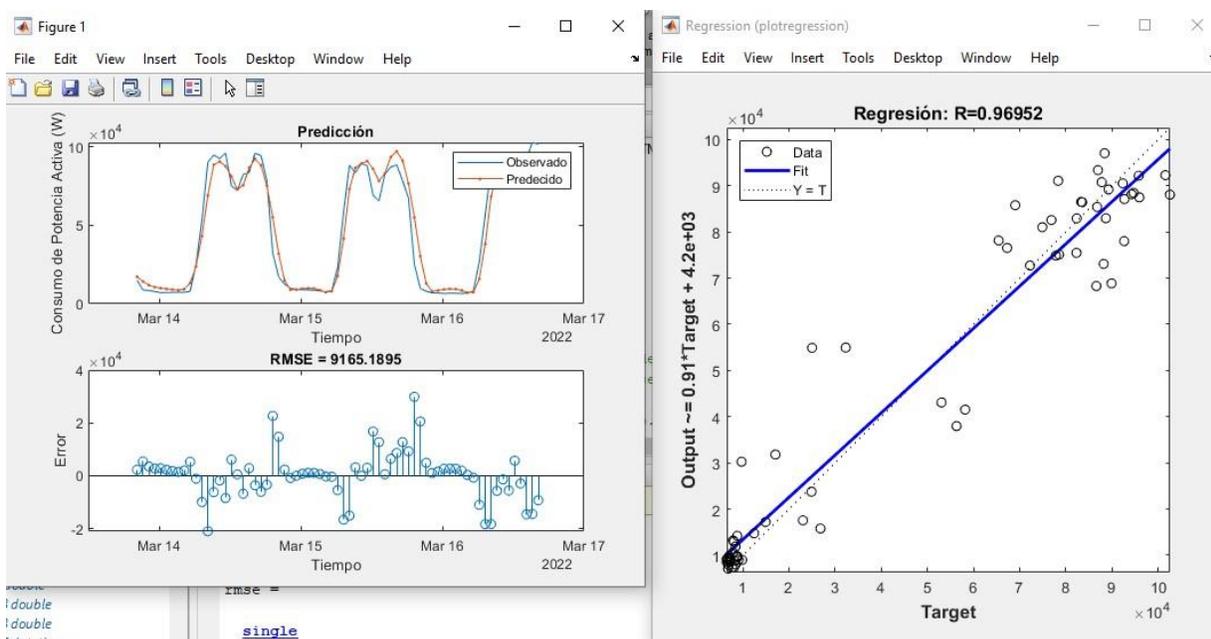


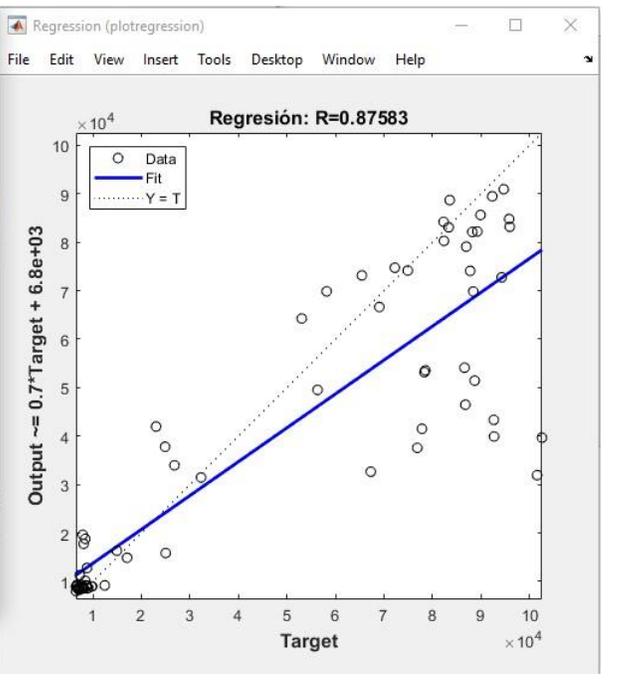
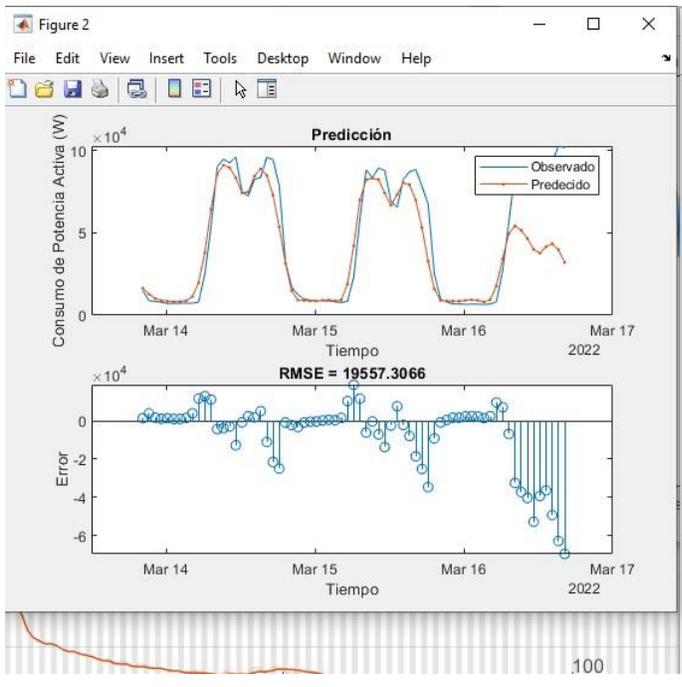
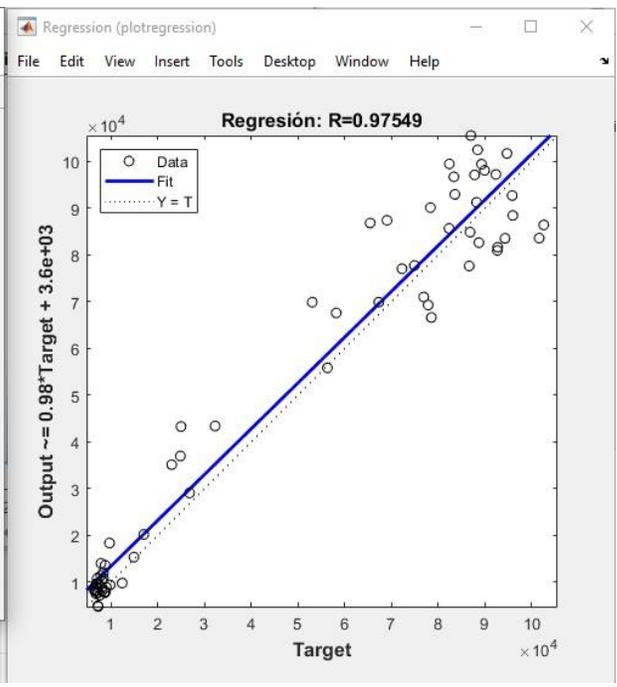
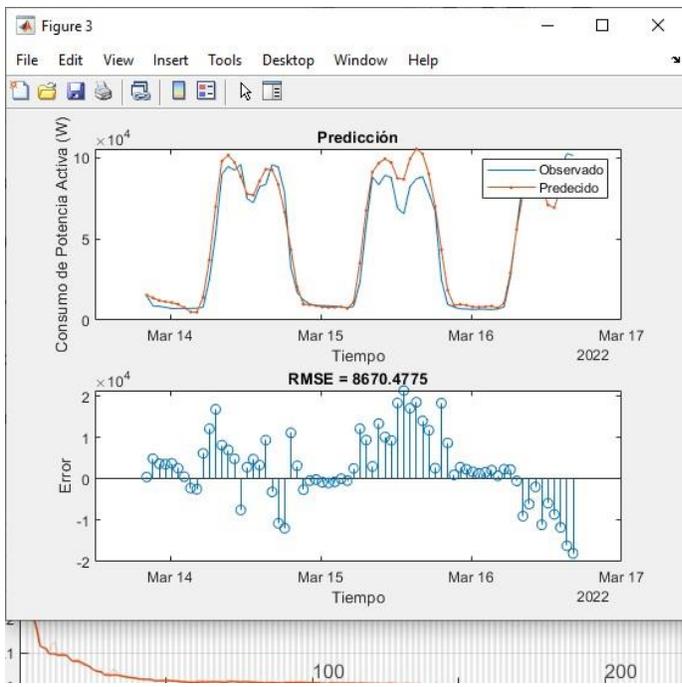
Anexo 5. Ventana del progreso de entrenamiento de la RNA con Arquitectura LSTM con 300 Neuronas Ocultas y 150 a 300 Épocas Máximas.



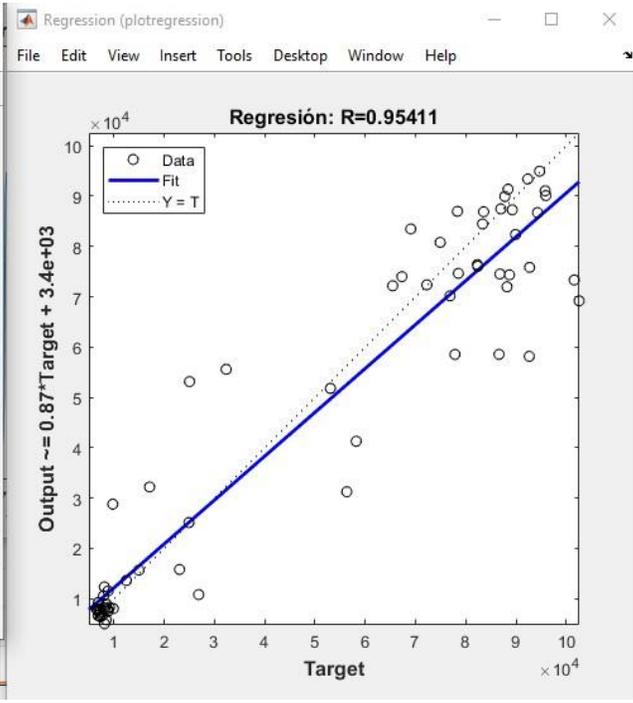
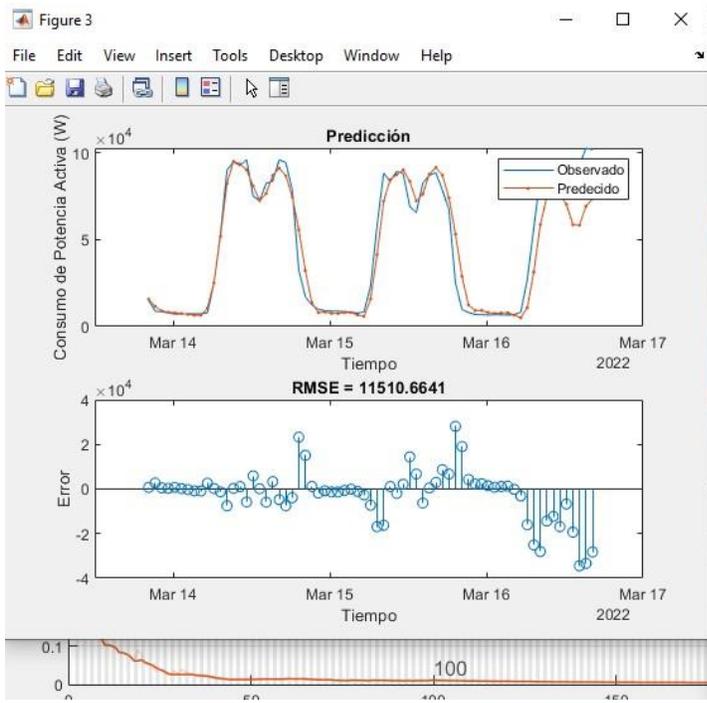
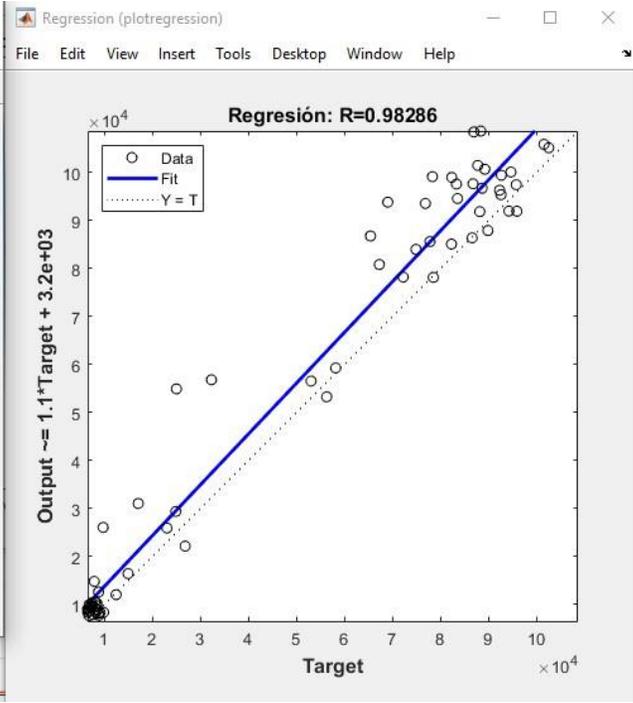
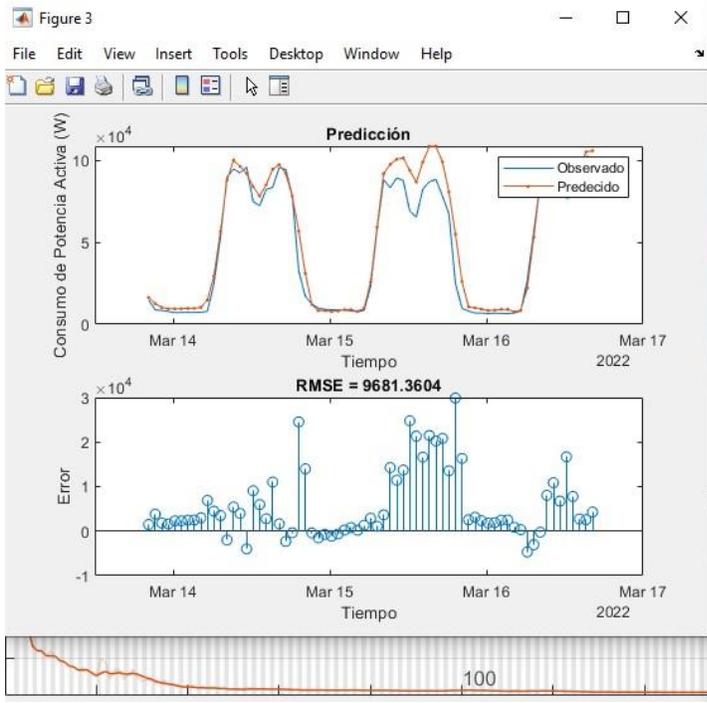


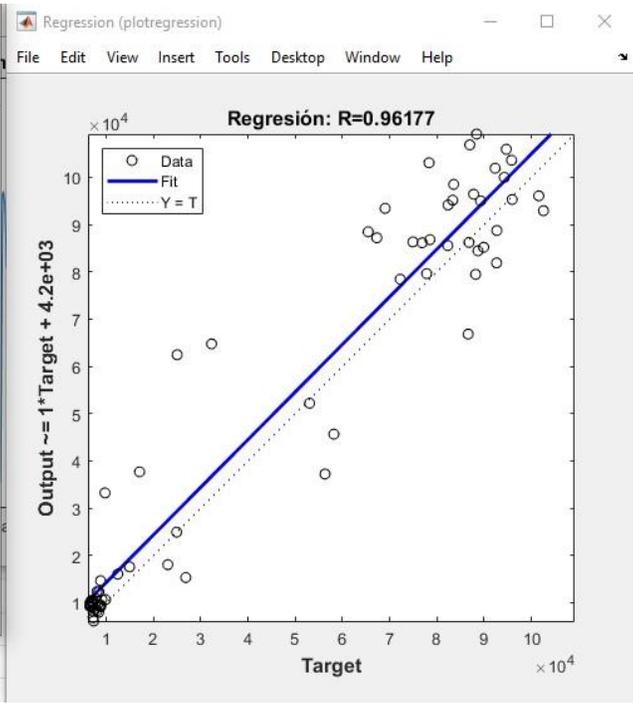
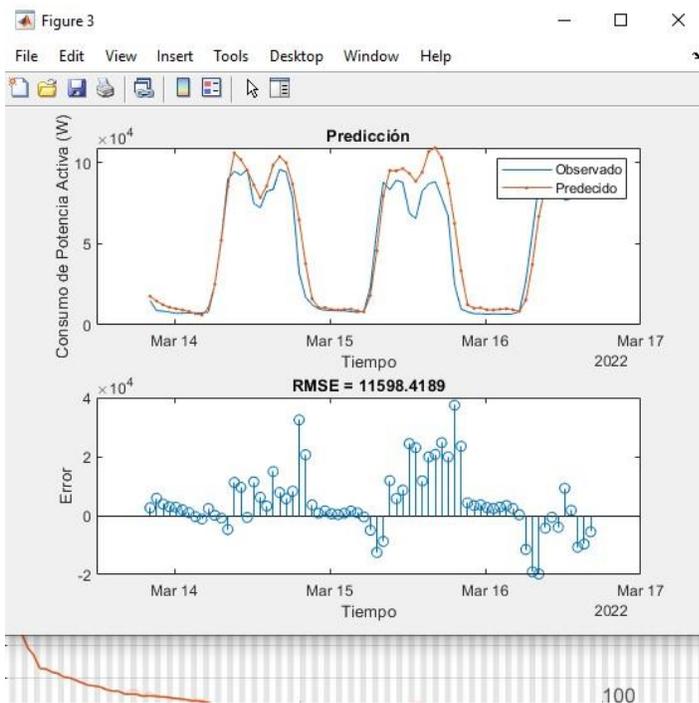
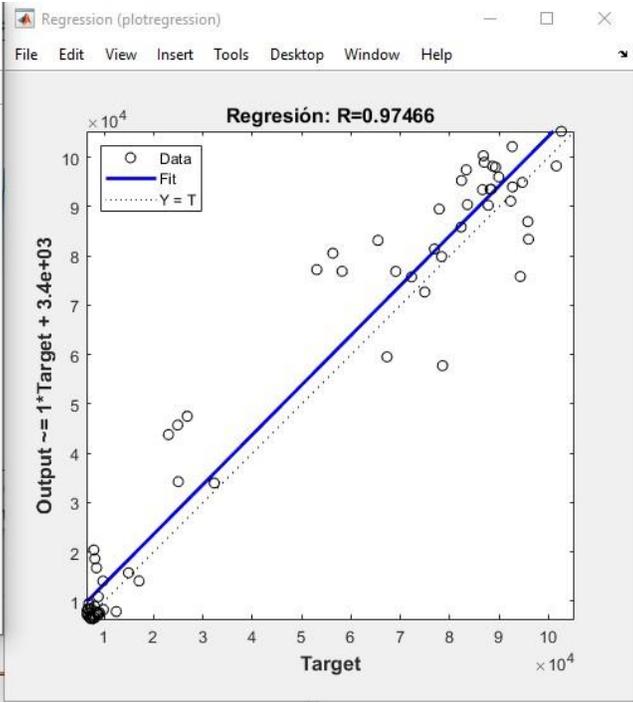
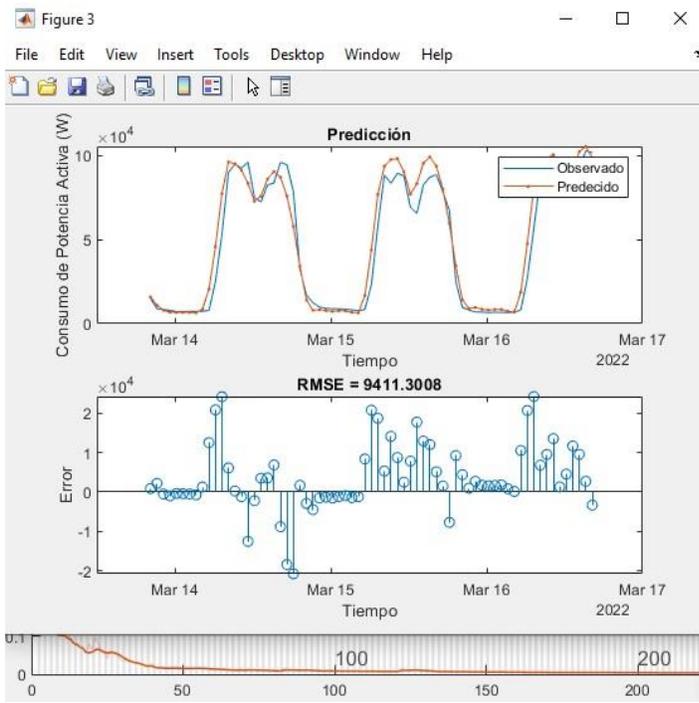
Anexo 6. Graficas de Resultados de la RNA con Arquitectura LSTM con 150 Neuronas Ocultas y 150 a 300 Épocas Máximas.



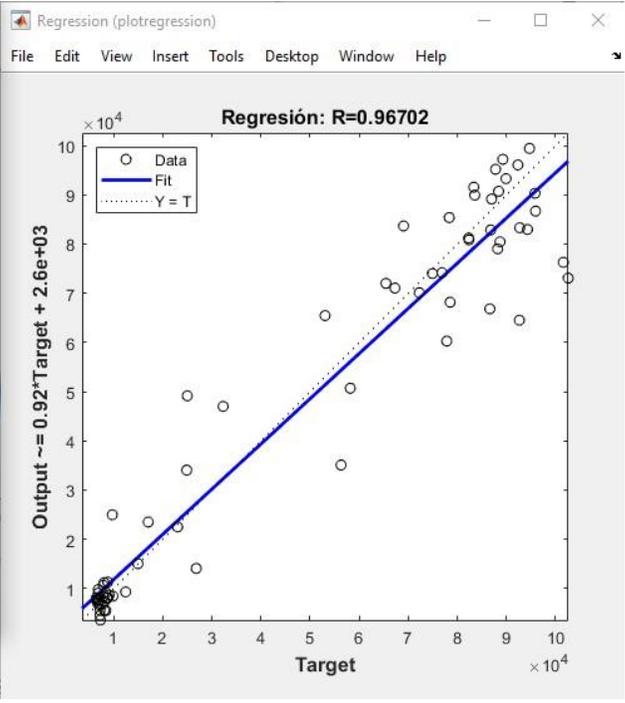
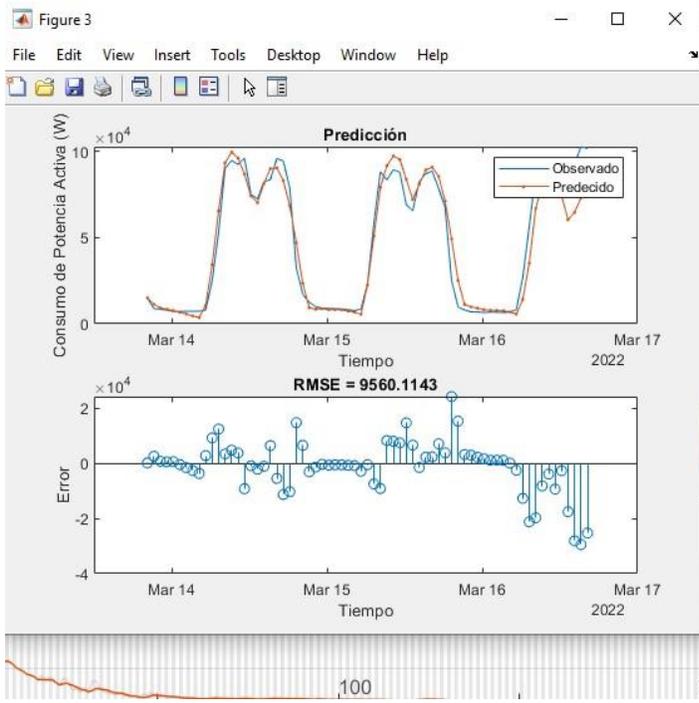
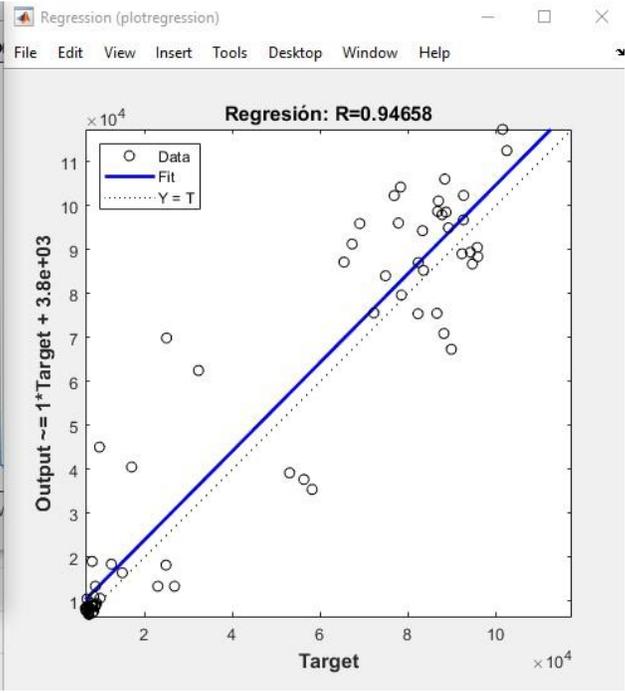
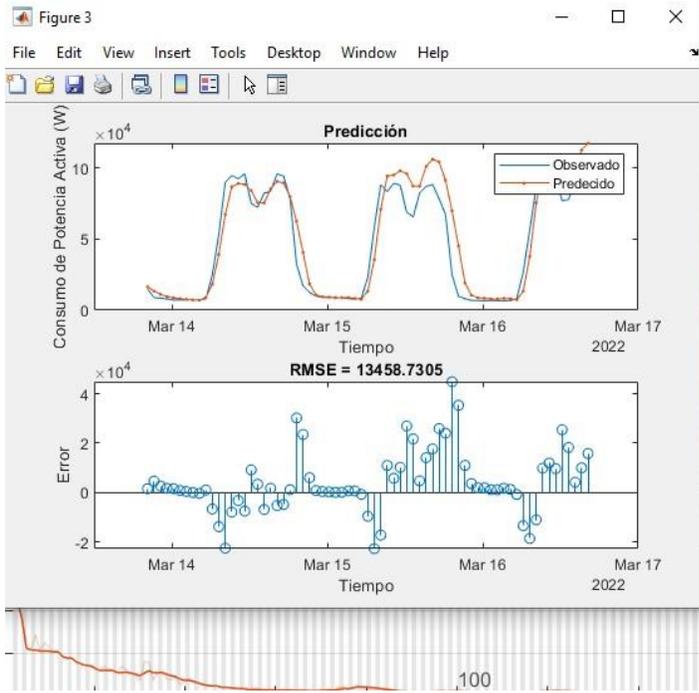


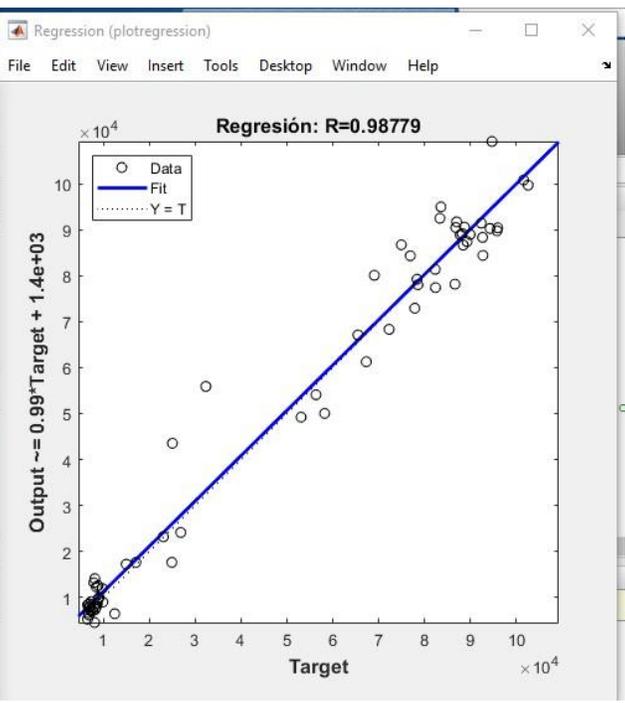
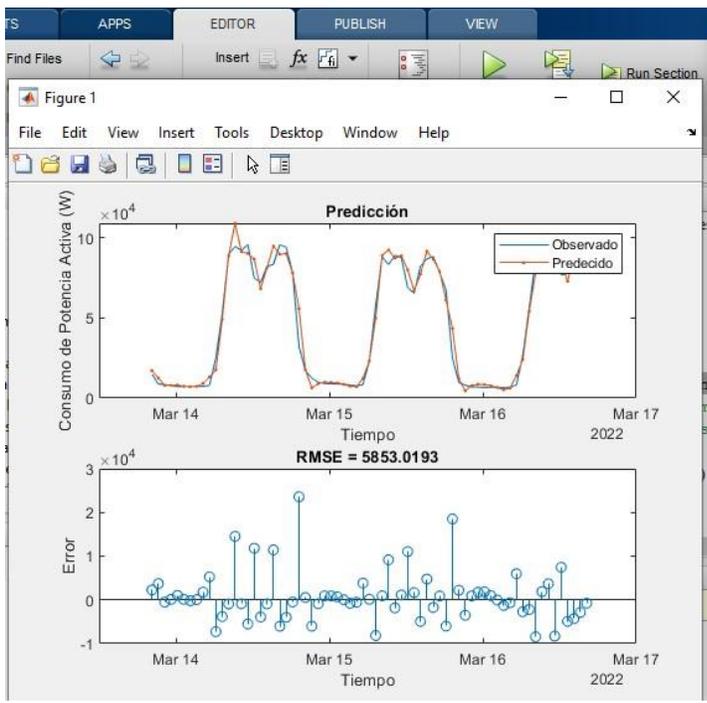
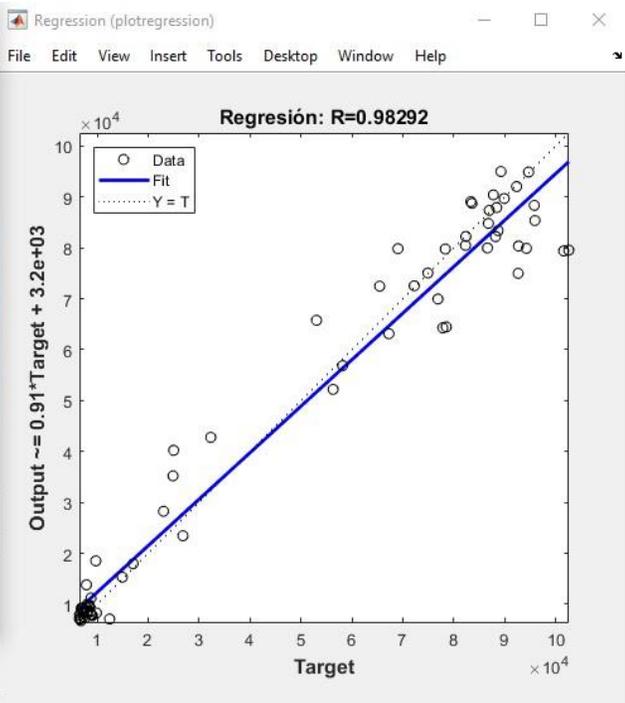
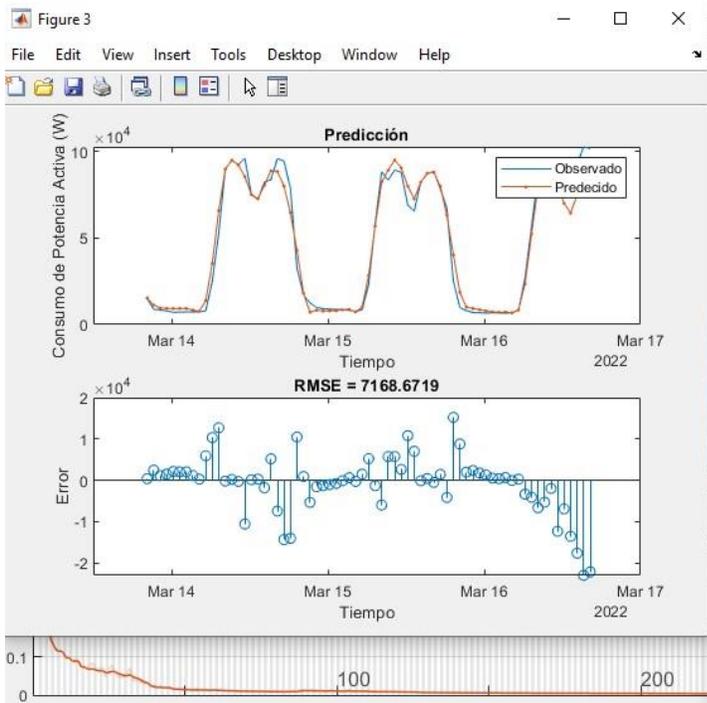
Anexo 7. Graficas de Resultados de la RNA con Arquitectura LSTM con 200 Neuronas Ocultas y 150 a 300 Épocas Máximas.



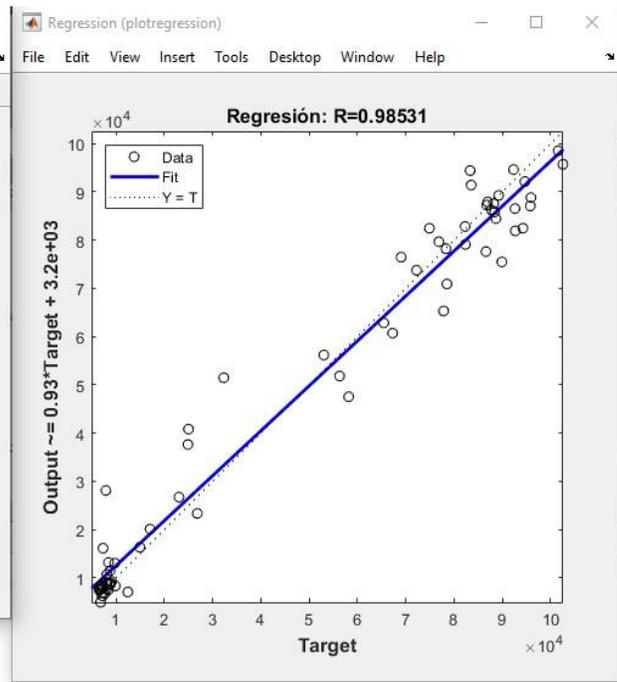
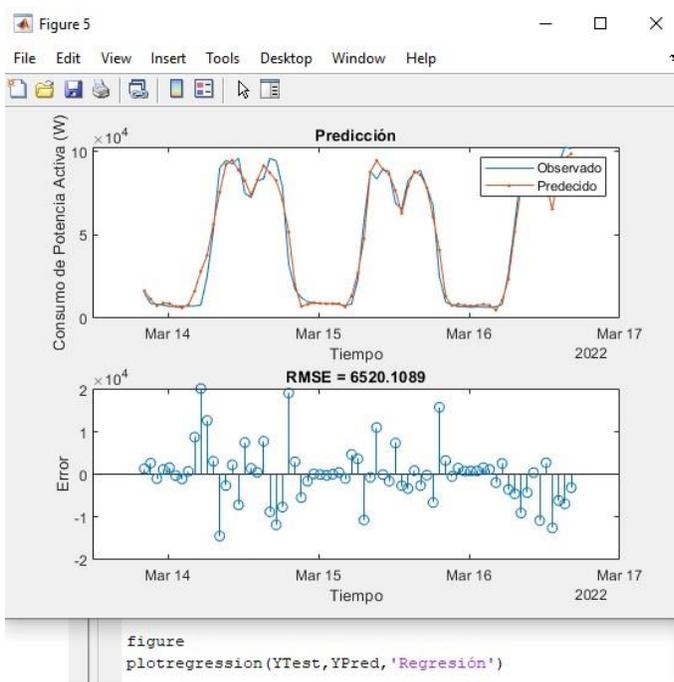
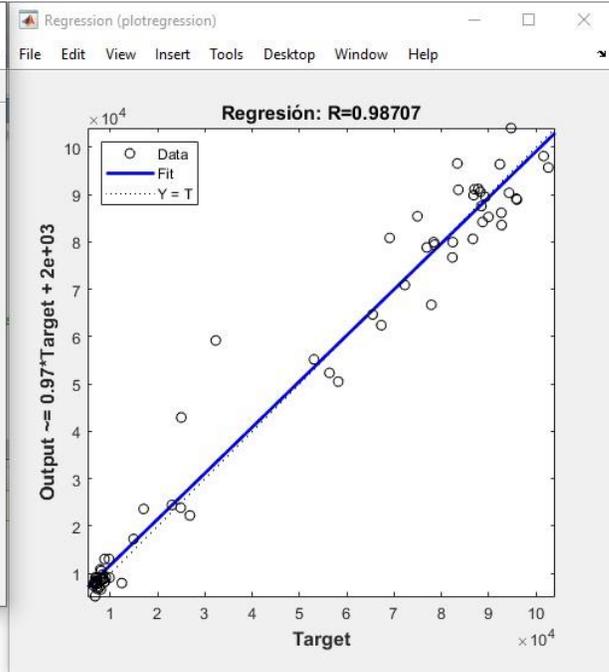
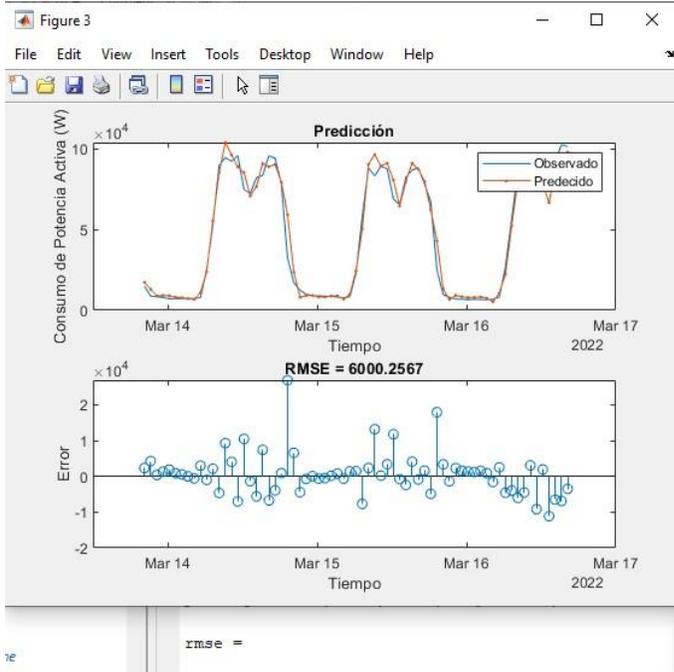


Anexo 8. Graficas de Resultados de la RNA con Arquitectura LSTM con 250 Neuronas Ocultas y 150 a 300 Épocas Máximas.

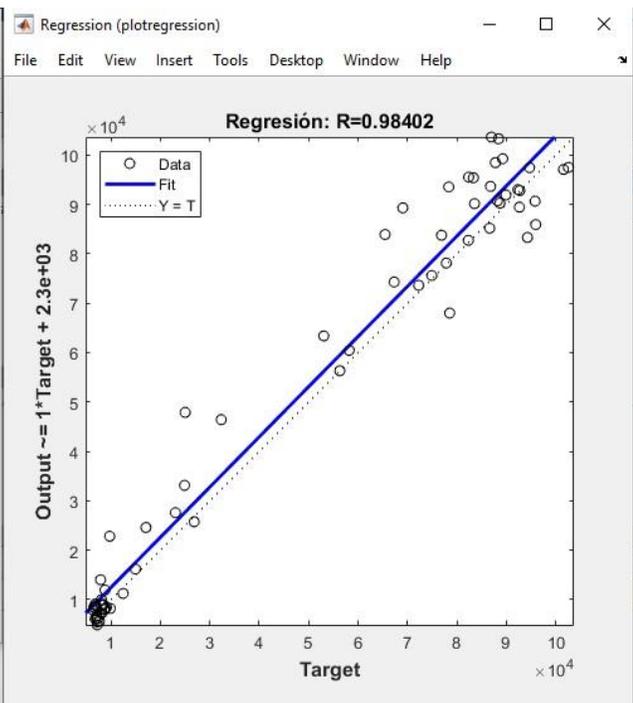
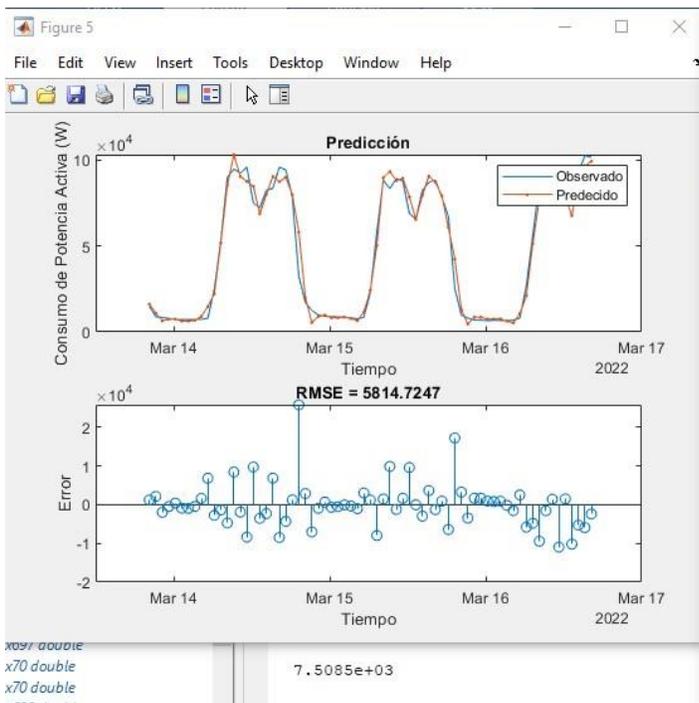
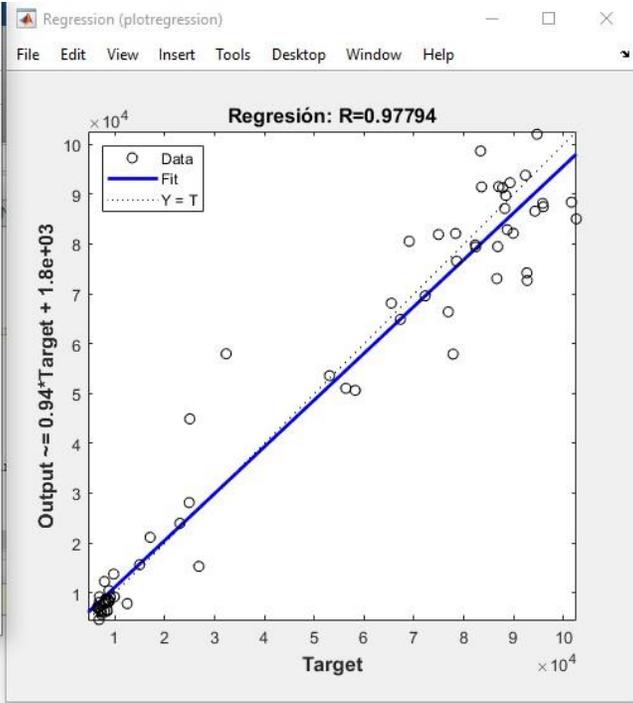
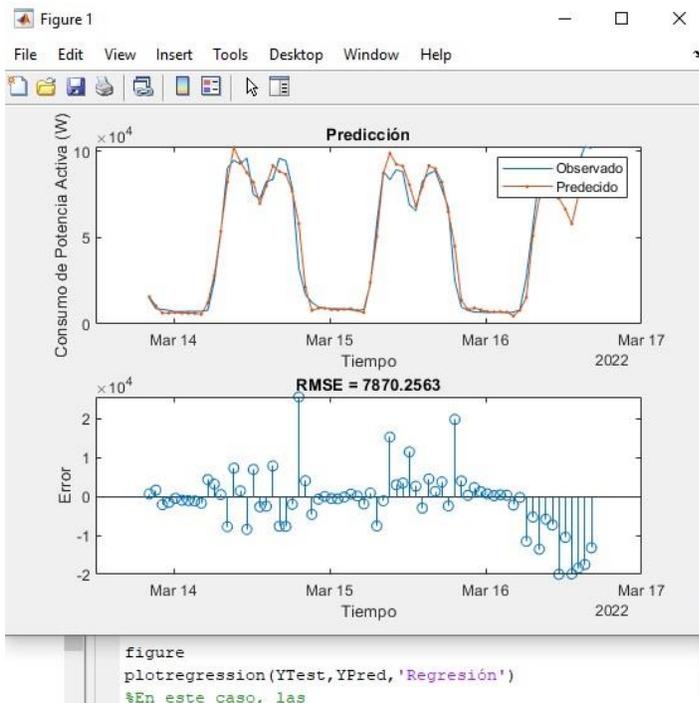




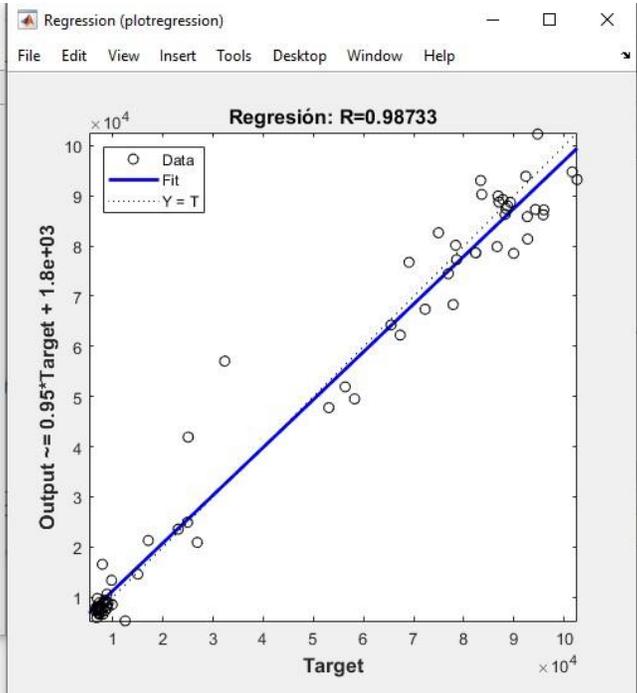
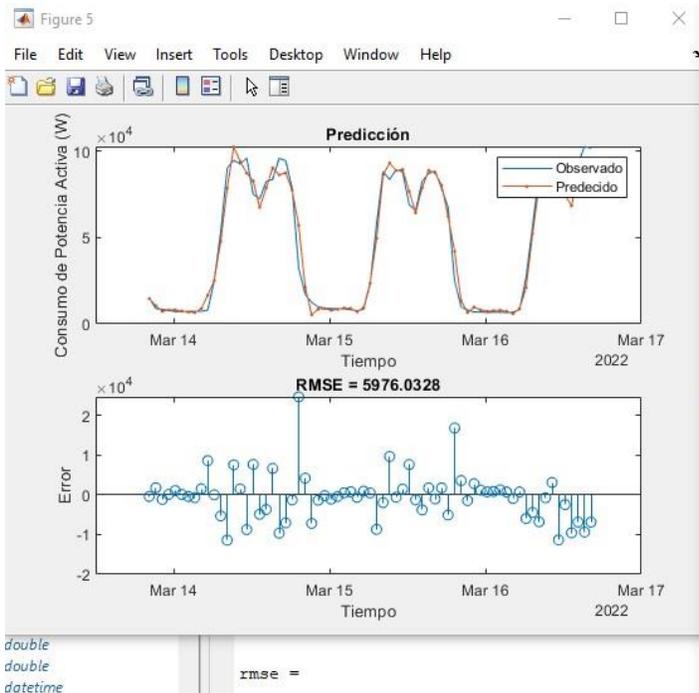
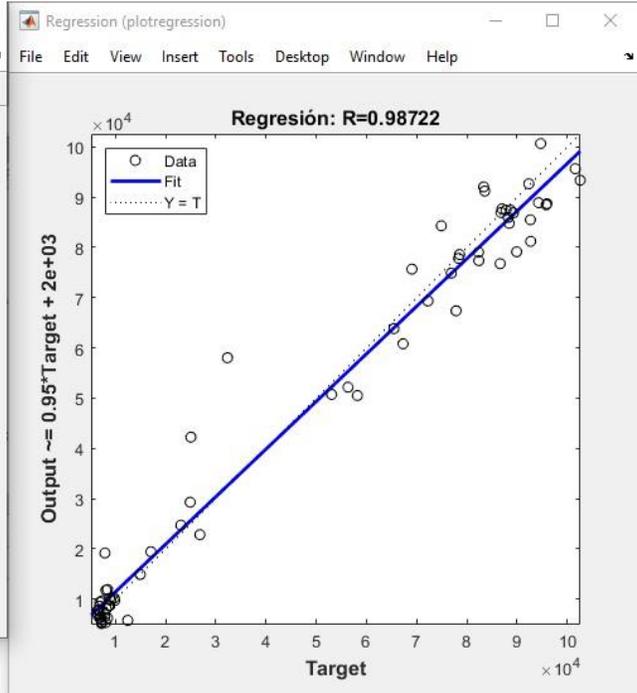
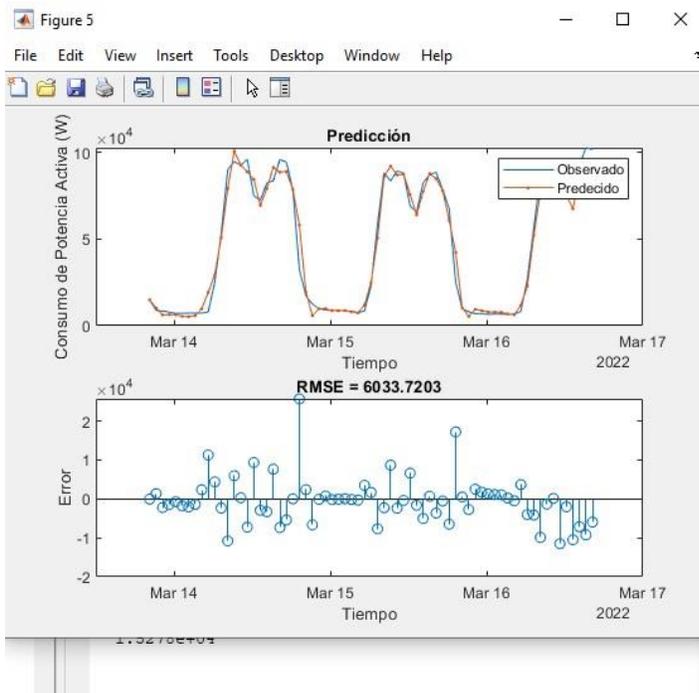
Anexo 9. Graficas de Resultados de la RNA con Arquitectura LSTM con 300 Neuronas Ocultas y 150 a 300 Épocas Máximas.



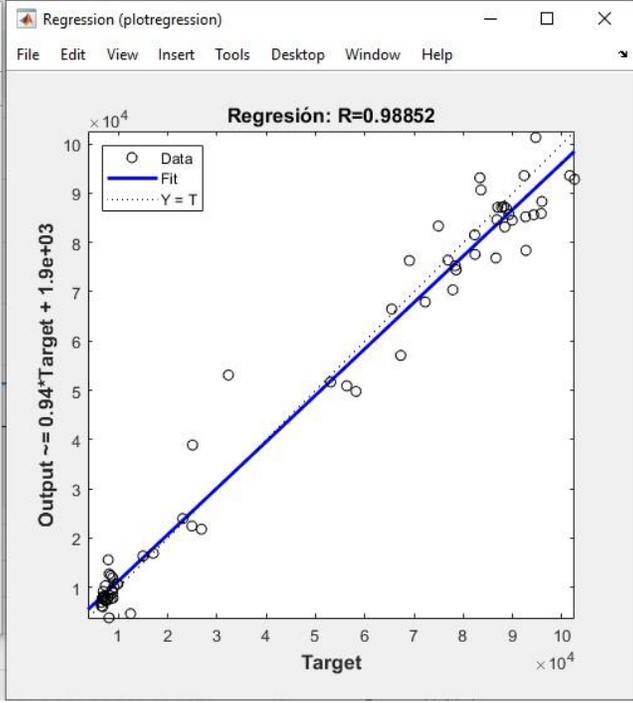
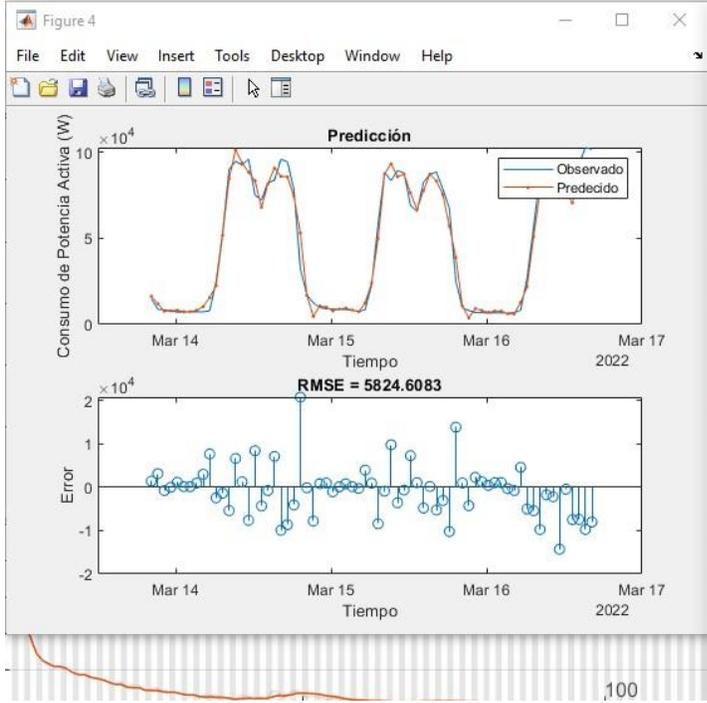
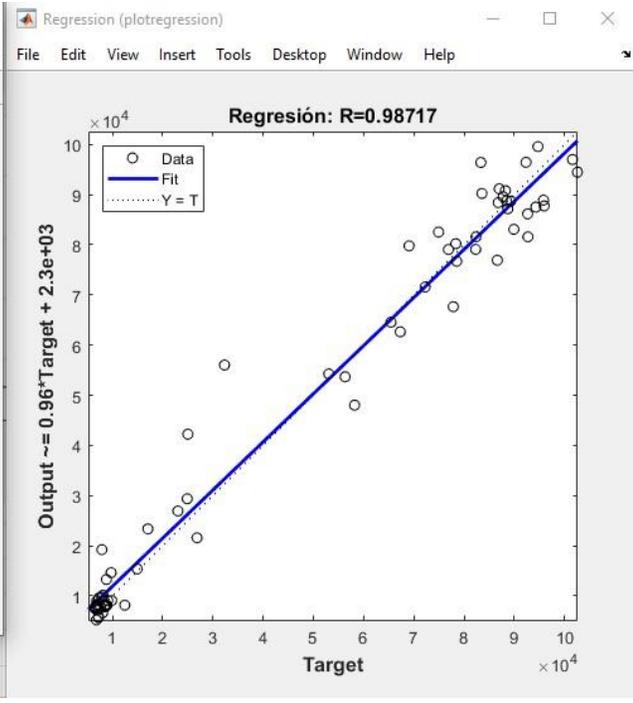
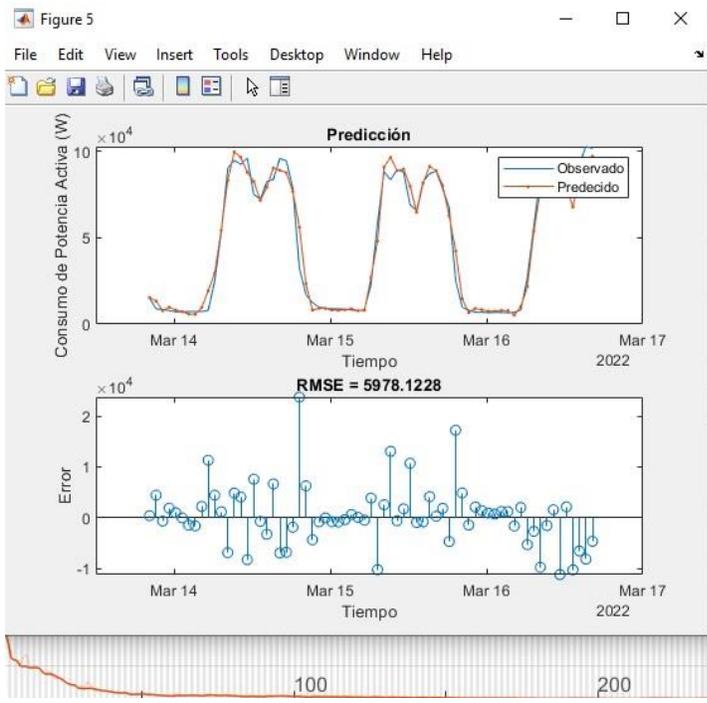
```
figure  
plotregression(YTest,YPred,'Regresión')
```



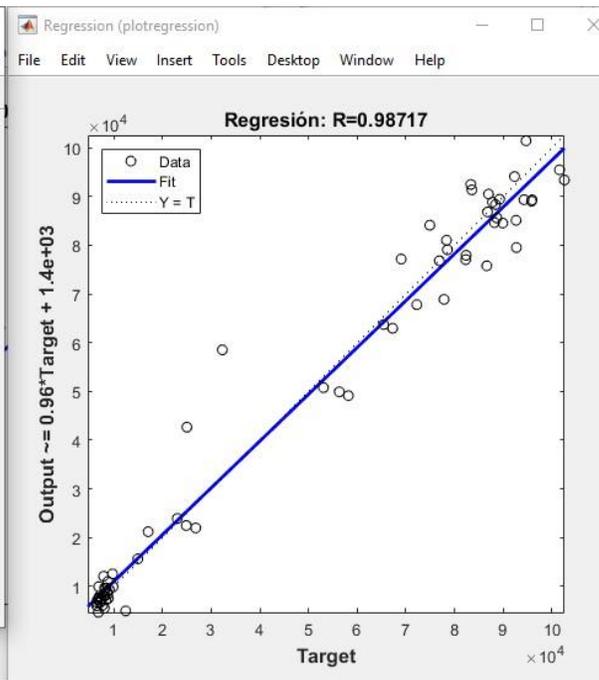
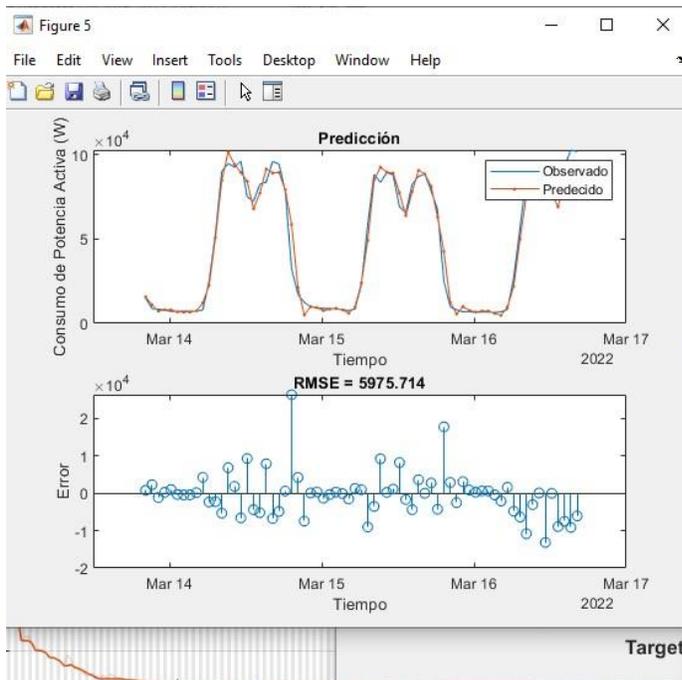
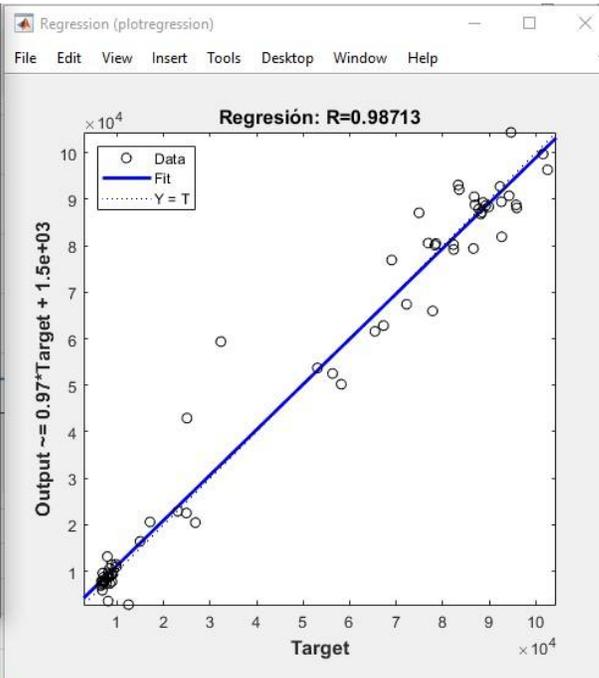
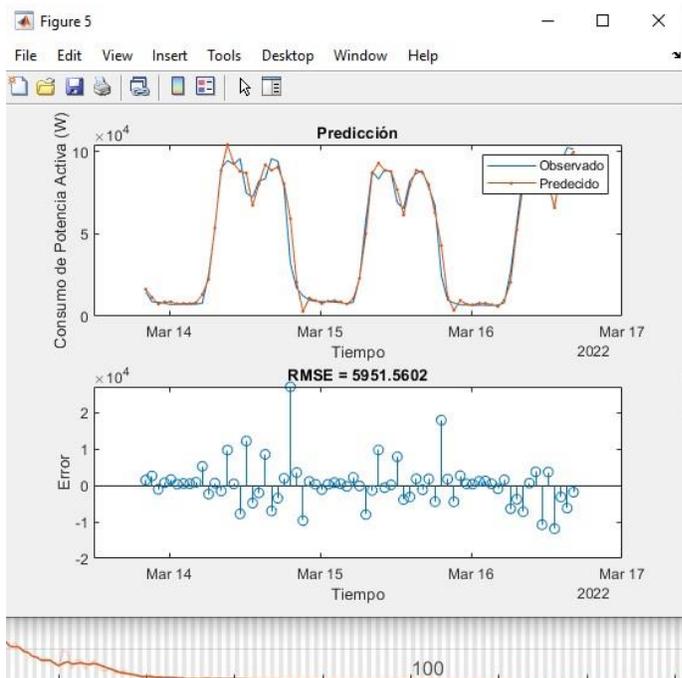
Anexo 10. Graficas de Resultados de la RNA con Arquitectura LSTM Actualizada con los Datos de la Salida Anterior usando 150 Neuronas Ocultas y 150 a 300 Épocas Máximas.

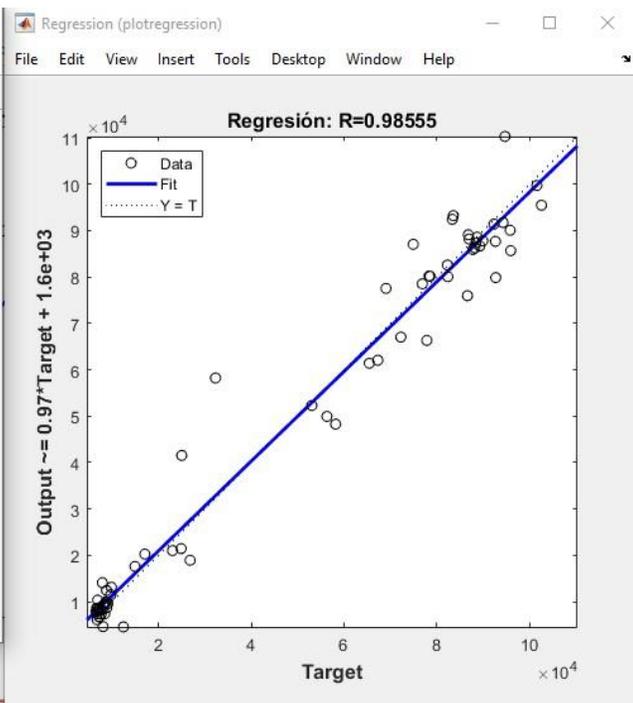
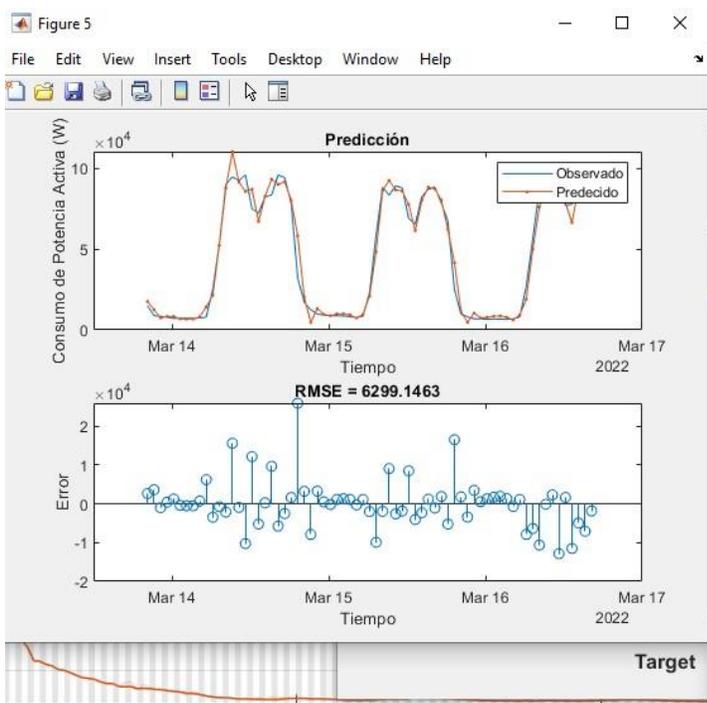
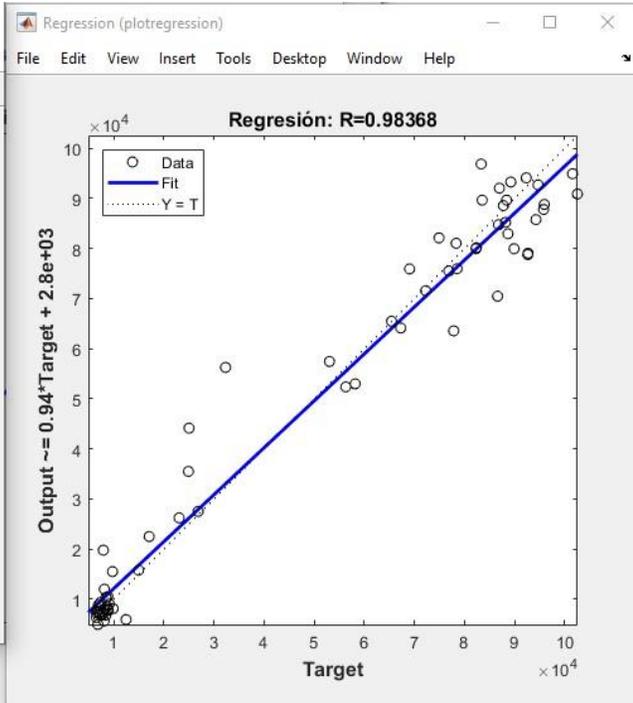
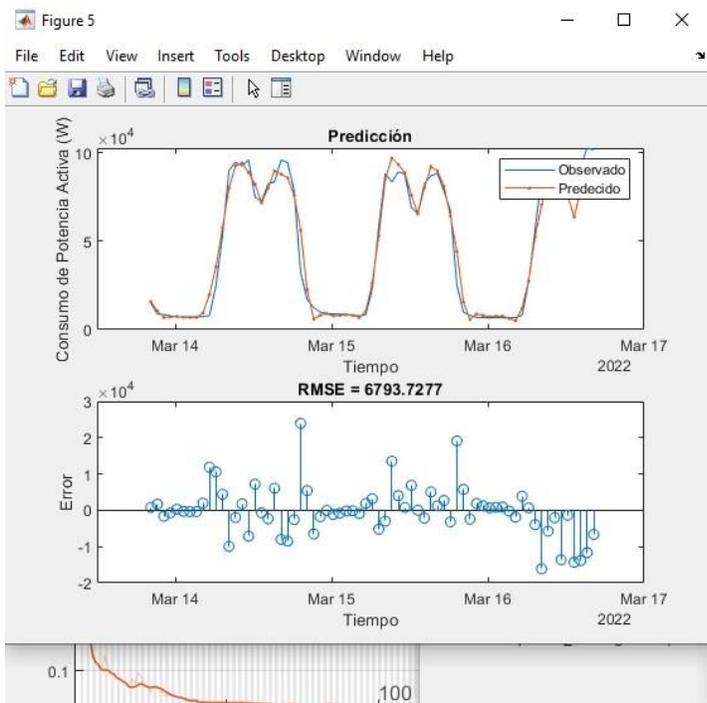


```
double  
double  
datetime  
  
rmse =
```

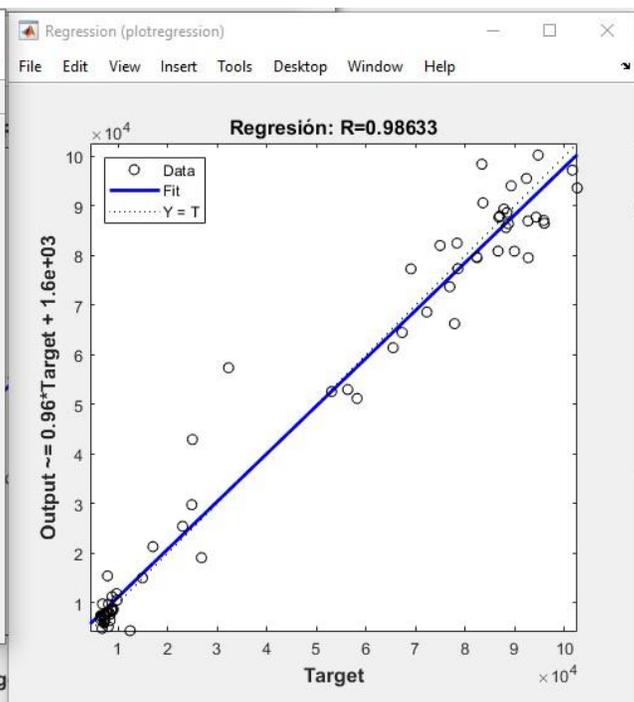
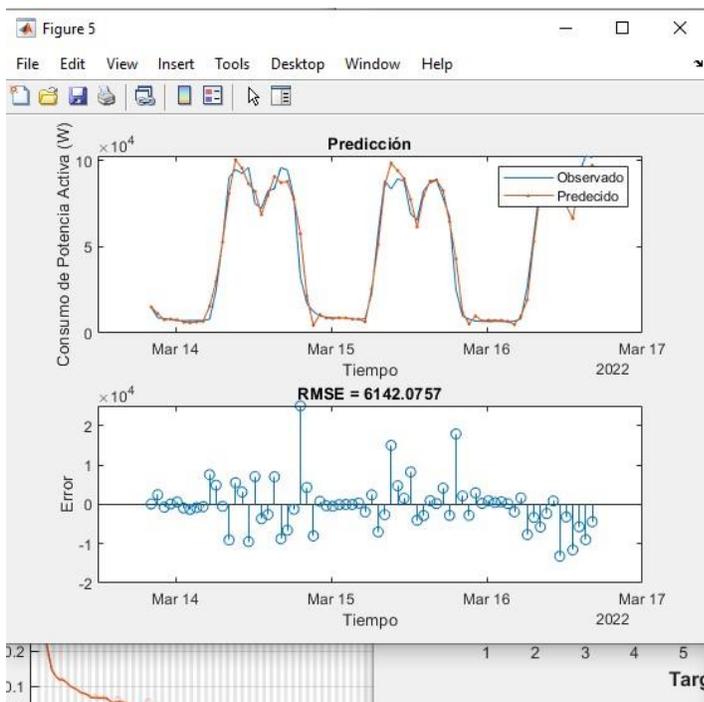
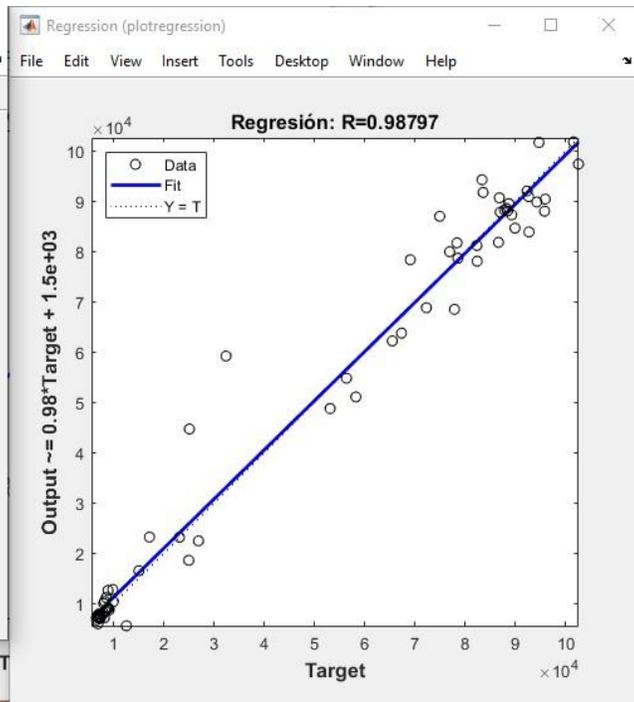
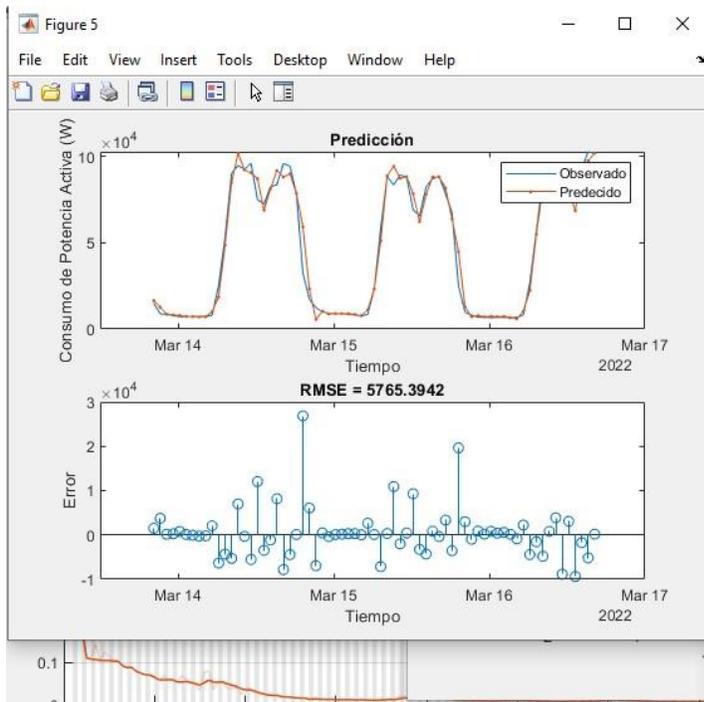


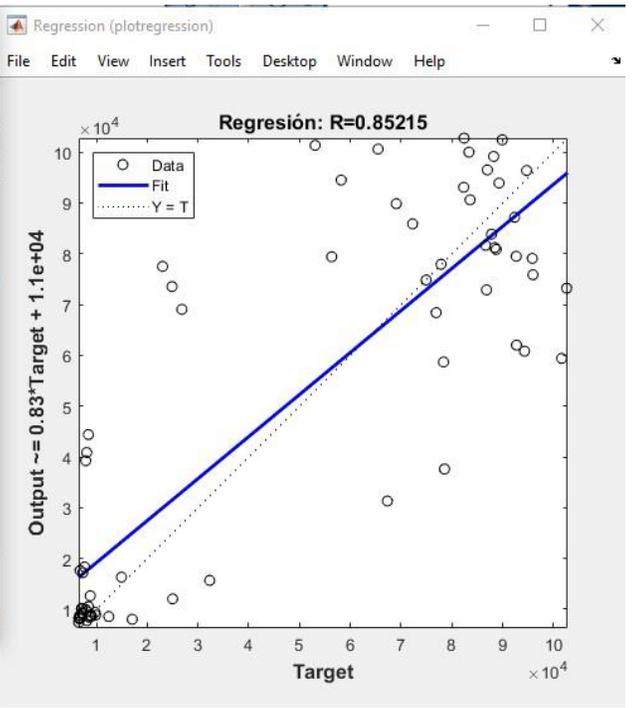
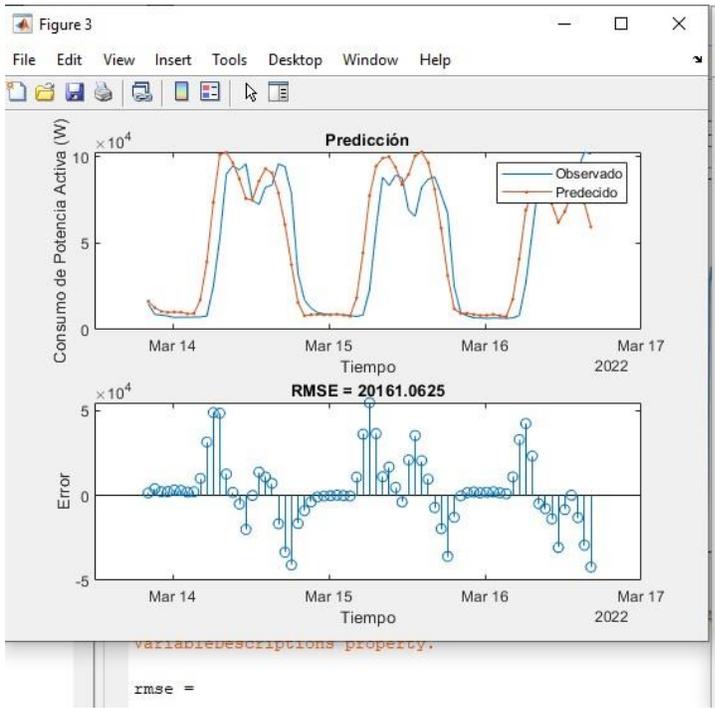
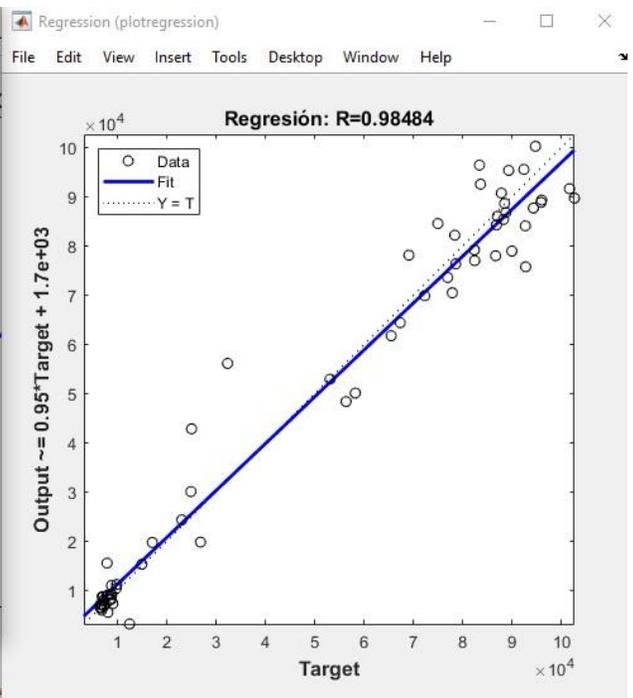
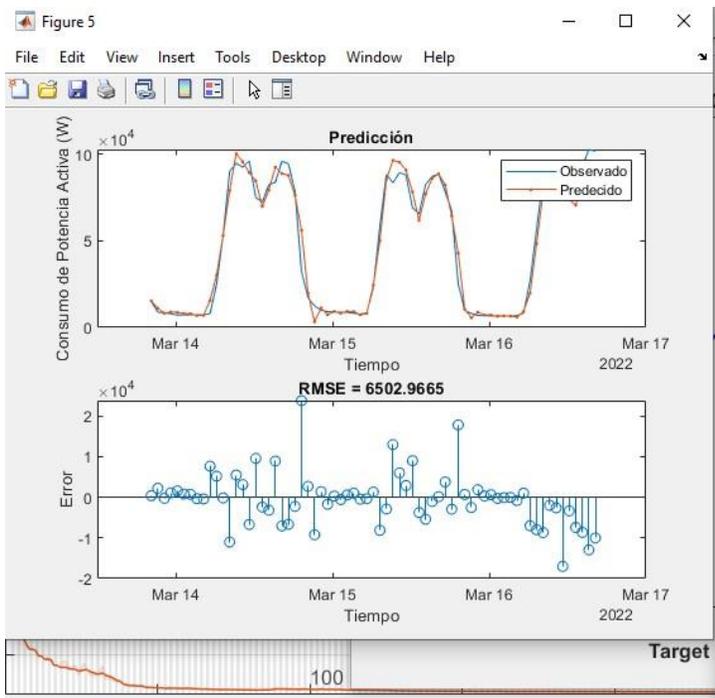
Anexo 11. Graficas de Resultados de la RNA con Arquitectura LSTM Actualizada con los Datos de la Salida Anterior con 200 Neuronas Ocultas y 150 a 300 Épocas Máximas.



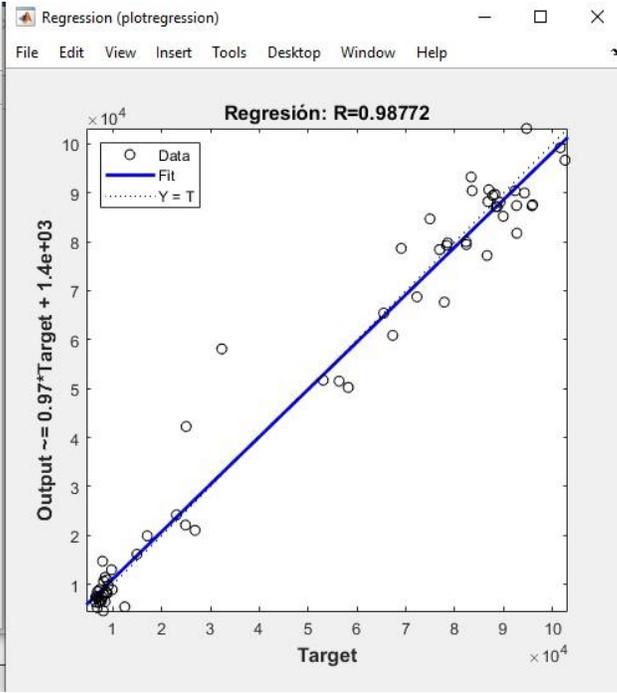
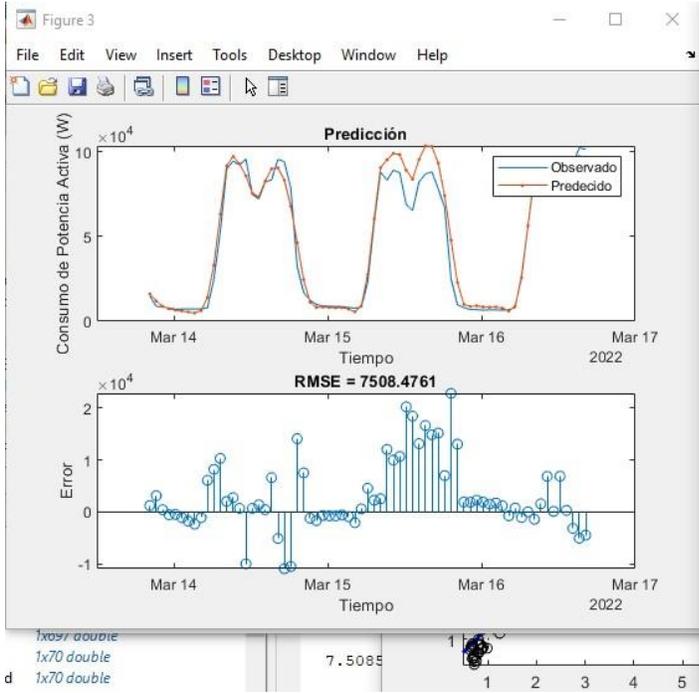
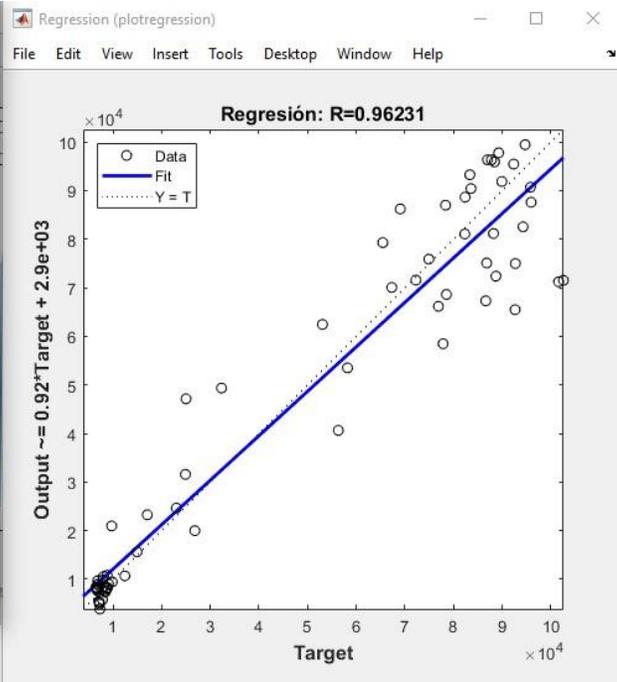
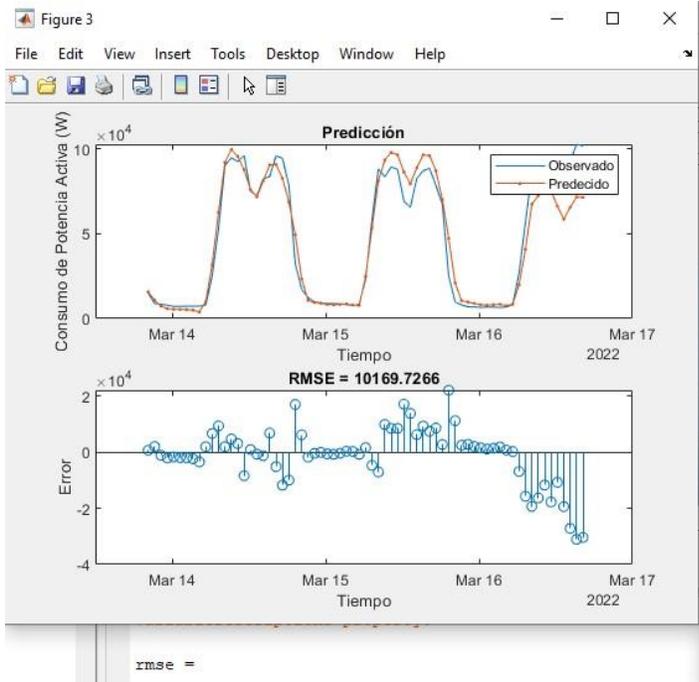


Anexo 12. Graficas de Resultados de la RNA con Arquitectura LSTM Actualizada con los Datos de la Salida Anterior con 250 Neuronas Ocultas y 150 a 300 Épocas Máximas.





Anexo 13. Graficas de Resultados de la RNA con Arquitectura LSTM Actualizada con los Datos de la Salida Anterior con 300 Neuronas Ocultas y 150 a 300 Épocas Máximas.



1x57 double
1x70 double
d 1x70 double
7.5085

