



# UNIVERSIDAD TÉCNICA DE COTOPAXI

## DIRECCIÓN DE POSGRADO

### MAESTRÍA EN ELECTRICIDAD

### MODALIDAD: PROYECTO DE INVESTIGACIÓN

**Título:**

---

**Aplicación de la técnica de Machine Learning para la estimación del potencial eólica en el sector de Lasso a partir de mediciones de una estación meteorológica**

---

Trabajo de titulación previo a la obtención del título de magister en Electricidad mención sistemas eléctricos de potencia

**Autor:**

Basantes Tisalema Juan Carlos

**Tutor:**

Franklin Hernán Vásquez Teneda MSc

**LATACUNGA – ECUADOR**

**2022**

## AVAL DEL TUTOR

En mi calidad de Tutor del Trabajo de Titulación “Aplicación de la técnica de Machine Learning para la estimación del potencial eólica en el sector de Lasso a partir de mediciones de una estación meteorológica” presentado por Basantes Tisalema Juan Carlos, para optar por el título magíster en Electricidad mención sistemas eléctricos de potencia

### CERTIFICO

Que dicho trabajo de investigación ha sido revisado en todas sus partes y se considera que reúne los requisitos y méritos suficientes para ser sometido a la presentación para la valoración por parte del Tribunal de Lectores que se designe y su exposición y defensa pública.

Latacunga, agosto, 13, 2021.



.....  
Ing. Franklin Hernán Vásquez Teneda MSc  
CC.: 1710434497

## AVAL DEL TRIBUNAL

El trabajo de Titulación: “Aplicación de la técnica de Machine Learning para la estimación del potencial eólica en el sector de Lasso a partir de mediciones de una estación meteorológica”, ha sido revisado, aprobado y autorizado su impresión y empastado, previo a la obtención del título de Magíster en Electricidad mención sistemas eléctricos de potencia; el presente trabajo reúne los requisitos de fondo y forma para que el estudiante pueda presentarse a la exposición y defensa.

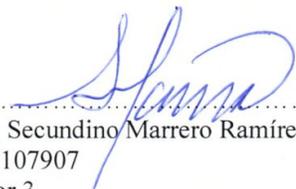
Latacunga, mayo, 05, 2022.



.....  
MSc. Edgar Roberto Salazar Achig  
0502847619  
Presidente del tribunal



.....  
MSc. José Efrén Barbosa Galarza  
0501420723  
Lector 2



.....  
PhD. Secundino Marrero Ramírez  
1757107907  
Lector 3

## DEDICATORIA

A Dios y a la Virgen María, quien me dio la fortaleza y la salud para terminar esta investigación.

A mis padres, dándome ejemplos dignos de superación y entrega, a mi madre por ser mi mejor ejemplo de vida y de lucha, por la paciencia y el cariño que me brinda, a mi padre por todo el amor, esfuerzo y apoyo incondicional.

A mis hermanos, gracias por sus palabras de aliento que me motivaron a esforzarme para alcanzar mi sueño.

Finalmente dedico esta investigación a la persona incondicional que está a mi lado, que día a día me acompaña en buenos y malos momentos y que con sus palabras de aliento y apoyo incondicional hicieron este trabajo posible.

Juan

## AGRADECIMIENTO

Mi agradecimiento fraterno a los maestros de la Universidad Técnica de Cotopaxi al impartirme sus conocimientos en la formación académica que contribuyeron a este logro sea posible. El agradecimiento a mi tutor MBA. Franklin Hernán Vásquez Teneda y en especial quien con su guía pude concluir mi investigación con éxito y perseverancia.

Juan Carlos Basantes Tisalema



UNIVERSIDAD  
TÉCNICA DE  
COTOPAXI



POSGRADO

## RESPONSABILIDAD DE AUTORÍA

Quien suscribe, declara que asume la autoría de los contenidos y los resultados obtenidos en el presente trabajo de titulación.

Latacunga, mayo, 05, 2022

Juan Carlos Basantes Tisalema  
0503621864



UNIVERSIDAD  
TÉCNICA DE  
COTOPAXI



POSGRADO

## RENUNCIA DE DERECHOS

Quien suscribe, cede los derechos de autoría intelectual total y/o parcial del presente trabajo de titulación a la Universidad Técnica de Cotopaxi.

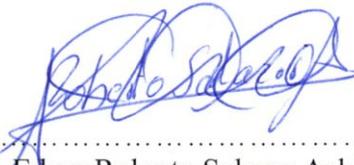
Latacunga, mayo, 05, 2022.

.....  
Juan Carlos Basantes Tisalema  
0503621864

## AVAL DEL PRESIDENTE DEL TRIBUNAL

Quien suscribe, declara que el presente Trabajo de Titulación: “Aplicación de la técnica de Machine Learning para la estimación del potencial eólica en el sector de Lasso a partir de mediciones de una estación meteorológica” contiene las correcciones a las observaciones realizadas por los lectores en sesión científica del tribunal.

Latacunga, mayo, 05, 2022



.....  
MSc. Edgar Roberto Salazar Achig  
0502847619



**UNIVERSIDAD TÉCNICA DE COTOPAXI**  
**DIRECCIÓN DE POSGRADO**

**MAESTRÍA EN ELECTRICIDAD**  
**MENCIÓN SISTEMAS ELÉCTRICOS DE POTENCIA**

**Título:** Aplicación de la técnica de Machine Learning para la estimación del potencial eólica en el sector de Lasso a partir de mediciones de una estación meteorológica.

**Autor:** Basantes Tisalema Juan Carlos

**Tutor:** Franklin Hernán Vásquez Teneda. MSc.

**RESUMEN**

La energía eólica es renovable que no contamina permite remplazar el uso de combustibles fósiles, es una alternativa para generar electricidad que puede suplir la demanda en sectores de difícil acceso de las redes eléctricas para lo cual se ha realizado el diseño del algoritmo aplicando técnicas de Machine Learning permitiendo predecir la velocidad del viento para determinar el potencial eólico que se tiene en base a las mediciones realizadas con la estación meteorológica, cabe destacar que al existir valores nulos se realizó una depuración de la data, el registro obtenido se analizó mediante el lenguaje de programación de Python, se utilizó el análisis de series temporales usando Red LSTM, que es un tipo de Red Neuronal Recurrente que permite analizar datos históricos, para su entrenamiento se utilizó el 80% de datos y el 20% para su validación, con el optimizador RMSPROP se generó mejores resultados de entrenamiento, optimizando la predicción con los valores reales además para su validación se aplicó el error porcentual absoluto medio (MAPE) obteniendo como resultado un 4% valor que se encuentra dentro del límite permitido para una correcta validación de predicción, con los valores pronosticados de la velocidad del viento se aplicó la distribución de Weibull determinando la velocidad promedio de 2.72 m/s con la cual permitió seleccionar el aerogenerador Enair 30Pro que se ha realizado atendiendo estrictamente a la Norma IEC 61400-2, con la que se determinó el potencial eólico promedio extraíble de 51.53 W y la energía producida de 522 kWh/año.

**PALABRAS CLAVE:** Machine Learning, red neuronal LSTM, predicción, velocidad del viento, potencial eólico.

**UNIVERSIDAD TECNICA DE COTOPAXI  
DIRECCION DE POSGRADO**

**MAESTRÍA EN ELECTRICIDAD  
MENCIÓN SISTEMAS ELÉCTRICOS DE POTENCIA**

**Title:** "Machine Learning technique application for the wind potential estimation into the Lasso sector from a meteorological station measurement".

**Author:** Basantes Tisalema Juan Carlos

**Tutor:** Franklin Hernán Vásquez Teneda. MSc.

**ABSTRACT**

The wind energy is renewable, which does not pollute, it allows to replace the fossil combustibles use, it is an alternative to generate electricity that can supply the demand into electrical networks difficult access sectors, which has been performed the algorithm design by applying the Machine Learning techniques, by allowing to predict the wind speed to determine the wind potential, based on the measurements made with the weather station, it should be noted that when there were null values, it was made a data purification, the got record was analyzed, through the Python programming language, it was used the time series analysis, using LSTM Network, what is a Recurrent Neural Network type, which allows analyzing historical data, its training was used data 80% and 20% for its validation, with the RMSPROP optimizer was generated better training results, optimizing the prediction with the real values, besides,, to its validation was applied the mean absolute percentage error (MAPE), getting as a result a 4% found value within the allowed limit for a correct prediction validation, with the wind speed predicted values were applied the Weibull distribution, by determining the 2.72 m/s average speed, which it can select the Enair 30Pro wind turbine, which has been performed strictly according with the IEC 61400-2 Standard that was determined the extractable average wind potential of 51.53 W and the produced energy of 522 kWh/year.

**KEYWORD:** Machine Learning, LSTM neural network, prediction, wind speed, wind potential.

Yo, Marco Paul Beltrán Semblantes con cédula de identidad número:0502666514 Magister en Lingüística Aplicada a la Enseñanza del idioma Inglés como Lengua Extranjera con número de registro de la SENESCYT: 1020-2021-2354162; **CERTIFICO** haber revisado y aprobado la traducción al idioma Inglés del resumen del trabajo de investigación con el título: "**Aplicación de la técnica de Machine Learning para la estimación del potencial eólica en el sector de Lasso a partir de mediciones de una estación meteorológica**" de: Basantes Tisalema Juan Carlos, aspirante a Magister en Electricidad, Mención Sistemas Eléctricos de Potencia.



Mg. Marco Paul Beltrán Semblantes  
0502666514



CENTRO  
DE IDIOMAS

Latacunga, mayo, 2022

## ÍNDICE DE CONTENIDOS

### CAPÍTULO I

1. FUNDAMENTACIÓN TEORICA- METODOLOGICA .....	5
1.1 Antecedentes de la investigación .....	5
1.2 Fundamentación Teórica.....	12
1.2.1 Generalidades .....	12
1.2.2 Técnicas de Machine Learning.....	12
1.2.3 Clasificación.....	14
1.2.4 Las principales ventajas que representan .....	16
1.2.5 Tipos de Redes Neuronales: .....	17
1.2.6 Velocidad del viento.....	20
1.2.7 Densidad del Potencia Eólico .....	22
1.3 Fundamentación metodológica.....	24
1.4 Conclusiones Capítulo I.....	25

### CAPÍTULO II

2 PROPUESTA .....	27
2.1 Título del proyecto.....	27
2.2 Objetivo del proyecto.....	27
2.3 Descripción de la propuesta.....	27
2.4 Metodología o procedimientos empleados para el cumplimiento de los objetivos planteados.....	29
2.4.1 Flujograma aplicado para el análisis de datos. ....	29
2.4.2 Seudocódigo aplicado para el procesamiento de datos.....	31
2.4.3 Machine Learning utilizando redes neuronales .....	32



2.5	Lenguaje de programación.....	35
2.5.1	Recolección de datos .....	35
2.5.2	Depuración de la data con la lectura de datos.....	36
2.5.3	Coeficientes de correlación lineal .....	37
2.5.4	Desarrollo del algoritmo y utilización de librerías.....	38
2.5.5	Datos de la velocidad del viento de la ciudad de Latacunga.....	40
2.5.6	Código de línea para cargar la base de datos .....	41
2.5.7	Estandarización y escalado de variables numéricas o data .....	42
2.5.8	Dividir en data de train y test.....	42
2.5.9	Estructura de la Red Neuronal LSTM .....	43
2.5.10	Optimizadores del algoritmo .....	46
2.5.11	Validación.....	48
2.5.12	Grafica de resultados .....	49
2.5.13	Análisis de comportamiento de los errores .....	49
2.5.14	Distribución de la función densidad de probabilidad de Weibull.....	50
2.5.15	Exponiendo y procesando los datos obtenidos .....	52
2.5.16	Rosa de los vientos .....	53
2.5.17	Aerogenerador .....	54
2.5.18	Potencial eólico suministrada por un aerogenerador de eje horizontal.....	54
2.5.19	Factor de planta.....	56
2.6	Conclusiones Capítulo II.....	56

### CAPÍTULO III

3	APLICACIÓN DE LA PROPUESTA .....	57
3.1	Análisis de los resultados.....	57



3.1.1	Selección de la red neuronal.....	57
3.1.2	Selección del lenguaje de programación.....	57
3.1.3	Recolección de datos.....	58
3.1.4	Datos descargados de la estación meteorológica.....	60
3.1.5	Lectura de la data y depuración de datos nulos.....	60
3.1.6	Análisis de correlación de la velocidad del viento recolectada por la estación meteorológica.....	62
3.1.7	Desarrollo del algoritmo de predicción en Python.....	63
3.1.8	Carga de datos depurados.....	64
3.1.9	Comparación de la velocidad de viento medidos por la estación meteorológica del ISTC y los datos obtenidos on-line de Latacunga.....	65
3.1.10	Selección de las variables de entrada para el desarrollo del algoritmo.....	66
3.1.11	Estandarización de variables numéricas.....	66
3.1.12	División de la data de entrenamiento y de prueba.....	67
3.1.13	Estructura de la Red Neuronal Recurrente LSTM.....	67
3.1.14	Optimizadores del algoritmo.....	68
3.1.15	Variación del número de neuronas.....	69
3.1.16	Variación del número de épocas.....	71
3.1.17	Validación y grafica de resultados.....	73
3.1.18	Análisis del comportamiento de los errores de la velocidad del viento del año 2020 para determinar la confiabilidad y viabilidad del programa realizado en Python	75
3.1.19	Distribución de Weibull.....	78
3.1.20	Rosa de los vientos.....	79
3.1.21	Velocidad del viento promedio.....	82
3.1.22	Selección del aerogenerador.....	83



3.1.23	Potencial eólico .....	85
3.1.24	Energía obtenida del aerogenerador durante un año .....	86
3.1.25	Factor de planta del aerogenerador E30PRO .....	87
3.2	Conclusiones generales .....	88
3.3	Recomendaciones .....	89
3.4	Referencias bibliográficas.....	91
ANEXOS	.....	96

## ÍNDICE DE FIGURAS

<b>Figura. 1.</b> Red Neuronal Artificial. [21].....	16
<b>Figura. 2.</b> Estructura de una Red Neuronal Perceptron Multicapa. [21] .....	17
<b>Figura. 3.</b> Método selectivo de entrenamiento de redes neuronales. [10] .....	18
<b>Figura. 4.</b> El módulo repetitivo en un LSTM contiene cuatro capas que interactúan. [22]	19
<b>Figura. 5.</b> Anemómetro. [23].....	20
<b>Figura. 6.</b> Fuerza de Coriolis. [23] .....	21
<b>Figura. 7.</b> Movimiento de variaciones del viento. [23] .....	21
<b>Figura. 8.</b> Flujograma de selección de la técnica de Machine Learning y programación del algoritmo, parte A.....	29
<b>Figura. 9.</b> Flujograma de la distribución de Weibull, parte B .....	30
<b>Figura. 10.</b> Pseudocódigo para la implementación del algoritmo LSTM.....	31
<b>Figura. 11.</b> Estación meteorológica instalada en el Instituto Superior Tecnológico Cotopaxi .....	36
<b>Figura. 12.</b> Velocidad promedio por día del mes enero 2020 de la ciudad de Latacunga. [37] .....	41
<b>Figura. 13.</b> Datos de entrenamiento, validación y prueba. [38] .....	43
<b>Figura. 14.</b> Estructura de una red LSTM. [39] .....	44
<b>Figura. 15.</b> Algoritmo de <i>backpropagation</i> para entrenamiento de redes neuronales. [41]	47
<b>Figura. 16.</b> Datos medidos vs datos reales. [38].....	48
<b>Figura. 17.</b> Datos Predichos en función de los valores reales. [38].....	49
<b>Figura. 18.</b> Variación de una distribución de Weibull al variar el factor de forma. [44] ...	52
<b>Figura. 19.</b> Distribución de Weibull con la estimación de la velocidad de viento promedio del distrito de Salaverry. [45] .....	53
<b>Figura. 20.</b> Orientación por medio de la rosa de los vientos. [47] .....	53
<b>Figura. 21.</b> Producción anual de energía [GWh], central eólica Villonaco.....	55
<b>Figura. 22.</b> Estación meteorológica instalada en el Instituto Superior Tecnológico Cotopaxi .....	59
<b>Figura. 23.</b> Página de comunicación RainwiseNet.....	59

<b>Figura. 24.</b> Data original del mes de enero 2019 – 2020 de la velocidad de viento [km/h] .....	60
<b>Figura. 25.</b> Data completa del mes de enero 2019-2020 velocidad del viento [km/h] .....	61
<b>Figura. 26.</b> Comparación de datos depurados .....	63
<b>Figura. 27.</b> Librerías utilizadas de Python .....	64
<b>Figura. 28.</b> Carga de datos depurados mediante el enlace al drive .....	64
<b>Figura. 29.</b> Comparación de datos medidos por hora del mes de enero año 2019 - 2020 ..	64
<b>Figura. 30.</b> Comparación de datos depurados .....	65
<b>Figura. 31.</b> Comparación velocidad del viento medidos por la estación meteorológica del ISTC y datos en línea de la ciudad de Latacunga .....	65
<b>Figura. 32.</b> Matriz de los valores estandarizados sin índices. ....	66
<b>Figura. 33.</b> Estructura de la red neuronal LSTM .....	68
<b>Figura. 34.</b> Comportamiento de la predicción de la velocidad del viento con diferentes optimizadores .....	69
<b>Figura. 35.</b> Comportamiento de la predicción de la velocidad del viento con diferentes números de neuronas .....	70
<b>Figura. 36.</b> Comportamiento de la predicción de la velocidad del viento con diferentes números de épocas .....	71
<b>Figura. 37.</b> Entrenamiento del algoritmo con optimizador = 'rmsprop', neuronas = 25, épocas = 25 .....	72
<b>Figura. 38.</b> Curva de aprendizaje de entrenamiento LSTM .....	73
<b>Figura. 39.</b> Curva de la velocidad del viento de predicción y reales de 7 días del mes de octubre .....	74
<b>Figura. 40.</b> Grafica de intervalos de la velocidad del viento de predicción y real del día siete de octubre 2020. ....	74
<b>Figura. 41.</b> Curva de la velocidad del viento de predicción y real del día siete de octubre de 2020. ....	75
<b>Figura. 42.</b> Comportamiento de la velocidad del viento mediante Regresión lineal .....	76
<b>Figura. 43.</b> Comportamiento de la velocidad del viento mediante Python .....	76

<b>Figura. 44.</b> Comportamiento de la velocidad del viento mediante desviación media absoluta con la RNA y regresión lineal .....	77
<b>Figura. 45.</b> Comportamiento de la velocidad del viento para su validación .....	77
<b>Figura. 46.</b> Comparación de la velocidad del viento real y pronosticada del año 2020 .....	78
<b>Figura. 47.</b> Histograma de frecuencia de la velocidad de viento.....	79
<b>Figura. 48.</b> Histograma de frecuencia acumulada de la velocidad de viento .....	79
<b>Figura. 49.</b> Lugar de emplazamiento del Instituto Superior Tecnológico Cotopaxi .....	80
<b>Figura. 50.</b> Rosa de vientos ISTC, dirección de donde proviene el viento .....	81
<b>Figura. 51.</b> Rosa de vientos ISTC, dirección hacia dónde se dirige el viento.....	81
<b>Figura. 52.</b> Distribución de Weibull del año 2020 .....	82
<b>Figura. 53.</b> Aerogenerador de minieólica E30PRO.....	84
<b>Figura. 54.</b> Curva de potencia del aerogenerador E30Pro.....	84
<b>Figura. 55.</b> Producción de energía [kWh] durante los meses del año 2020 .....	87

## ÍNDICE DE TABLAS

<b>Tabla 1.</b> Ventajas y desventajas de redes neuronales .....	34
<b>Tabla 2.</b> Tipos de aerogeneradores .....	54
<b>Tabla 3.</b> Clasificación del aerogenerador por demanda energética .....	54
<b>Tabla 4.</b> Características de las redes neuronales.....	57
<b>Tabla 5.</b> Descripción del lenguaje de programación .....	58
<b>Tabla 6.</b> Muestreo de datos de la estación meteorológica .....	60
<b>Tabla 7.</b> Resumen de datos nulos de la data .....	61
<b>Tabla 8.</b> Resumen de datos completos de la data .....	62
<b>Tabla 9.</b> Correlación de variables más relevantes en la velocidad del viento. ....	62
<b>Tabla 10.</b> Datos ordenados y seleccionados en función de la correlación.....	66
<b>Tabla 11.</b> División de la data de entrenamiento y de prueba.....	67
<b>Tabla 12.</b> Resultados obtenidos aplicando diferentes optimizadores para la predicción de la velocidad del viento.....	68
<b>Tabla 13.</b> Resultados del error MAE con la variación de neuronas para la predicción de la velocidad del viento.....	70
<b>Tabla 14.</b> Resultados derivados de la variación del número de épocas para la predicción de la velocidad del viento.....	71
<b>Tabla 15.</b> Valor del error absoluto medio entre la velocidad del viento real y la pronosticada .....	73
<b>Tabla 16.</b> Velocidad del viento promedio por mes.....	83
<b>Tabla 17.</b> Potencial eólico extraída por mes del 2020 .....	85
<b>Tabla 18.</b> Energía generada por mes y año del 2020.....	86

## INTRODUCCIÓN

La energía eólica es la energía que posee el viento y que puede ser aprovechada directamente o ser transformada a otros tipos de energía, como, por ejemplo, a energía eléctrica, la misma que debe ser analizada en base a la recolección de datos por la estación meteorológica ubicada en el Instituto Superior Tecnológico Cotopaxi, la cual permitirá aplicar la técnica Machine Learning para realizar el algoritmo que obtenga como resultado la predicción de la velocidad del viento del año, con los valores pronosticados se determinara el potencial eólico y la energía que generará durante el año 2020.

### **Antecedentes**

El tema se encuentra en de la línea de investigación de Energías alternativas y renovables correspondiente a la maestría en Electricidad, enfocándose en la sub línea de conversión y uso racional de la energía eléctrica, con el análisis, evalúo y la adecuada utilización de recursos naturales para el desarrollo de proyectos de energía renovable no convencional fortaleciendo el uso racional de la energía.

### **Planteamiento del problema**

El no disponer de modelos de algoritmos que realicen la predicción utilizando Machine Learning imposibilita monitorear sistemas de energías renovables generando el uso de combustible fósiles las cuales presentan muchas desventajas siendo unos de los factores principales para la contaminación y efectos ambientales, el desconocimiento de métodos de lenguaje de programación para la predicción de sistemas que posibiliten la búsqueda de mejores fuentes de energía renovables enfocadas al pronóstico de la velocidad del viento a través de redes neuronales que memoricen y aprendan a largo plazo son muy escasos que se apliquen es decir, aprender las dependencias de valores futuros de una secuencia en función a los valores históricos aplicados a la predicción, que realicen el aprendizaje no supervisado en función de variables que son medidas y recolectadas por la estación meteorológica que influyen directamente en la velocidad del viento, debido a esto no es posible realizar un

estudio que permita determinar o explotar el potencial eólico en varios lugares del sector, generando la progresiva deserción de las energías renovables (viento) siendo un obstáculo para el análisis de la viabilidad de instalar minicentrales eólicas que puedan satisfacer pequeñas cargas.

### **Formulación del problema**

La necesidad de pronosticar la velocidad del viento a partir de los datos recolectados por la estación meteorológica en el sector de Lasso se aplicará la técnica de Machine Learning para estimar el potencial eólico.

### **Objetivo General**

Aplicar la técnica de Machine Learning para la estimación del potencial eólico en el sector de Lasso a partir de mediciones de la velocidad del viento de una estación meteorológica

### **Objetivos Específicos**

- Recopilar información de métodos aplicados en la utilización de Machine Learning para los parámetros de la velocidad del viento mediante una investigación bibliográfica.
- Analizar los parámetros que influyen dentro de la velocidad del viento para la aplicación de Machine Learning
- Realizar un algoritmo que determine la predicción de la velocidad del viento en la aplicación de Machine Learning.
- Estimar el potencial eólico en base a la distribución de Weibull para determinar la energía que se genera en el año.

**Sistemas de tareas en relación a los objetivos específicos:** estas son actividades que se realizarán para dar cumplimiento a cada objetivo específico planteado.

Objetivos específicos	Actividad (tareas)	Resultado de la actividad	Descripción de la actividad (técnicas e instrumentos)
Recopilar información de métodos aplicados en la utilización de Machine Learning para los parámetros de la velocidad del viento mediante una investigación bibliográfica.	Recopilación de información bibliográfica	Obtener los diferentes métodos aplicados en Machine Learning	Aplicación en Machine Learning para la predicción de la velocidad del viento y determinar el potencial eólico.
Analizar los parámetros que influyen dentro de la velocidad del viento para la aplicación de Machine Learning	Recolección y determinación de los parámetros necesarios de la velocidad del viento	Seleccionar los parámetros necesarios que influyen dentro de la velocidad del viento	Seleccionar los parámetros más relevantes para la generación de la velocidad del viento los mismos que influyan dentro de la aplicación Machine Learning
Realizar un algoritmo que determine la predicción de la velocidad del viento en la aplicación de Machine Learning.	Programación del algoritmo.	Predicción de la velocidad del viento en el sector de Lasso.	Ejecutar el algoritmo de Machine Learning para determinar la velocidad del viento promedio por mes durante un año.

<p>Estimar el potencial eólico en base a la distribución de Weibull para determinar la energía que se genera en el año</p>	<p>Realizar un histograma de frecuencia para determinar el valor promedio del potencial eólico</p>	<p>Obtener el valor promedio de la velocidad del viento durante un año</p>	<p>Analizar mediante la distribución de Weibull con los histogramas de frecuencia, obtener la velocidad del viento promedio para determinar el potencial eólico y la energía por año que se obtendrá en el sector de Lasso.</p>
--	--	--	---

*Elaborado por o fuente: Juan Basantes*

### Justificación

Al aplicar el lenguaje de programación que permita realizar la predicción mejorara el monitoreo de los sistemas de energías renovables siendo un peso muy importantes en el sistema eléctrico, el cual permite generar nuevas búsquedas de lugares que sean generadores de parámetros rentables relacionados a la velocidad del viento permitiendo realizar estudios en base a la recolección de datos el cual permite aplicar el algoritmo con una gran cantidad de datos que son procesados generando resultados que influyen directamente para estimar el potencial eólico en base a la distribución de Weibull, logrando determinar si el lugar es viable para la instalación de una minicentral en base de la velocidad promedio que se genere de manera mensual siendo el proyecto viable para ser aplicado el mismo que mediante la aplicación de la técnica de machine Learning generara en la sociedad un impacto para generar estudios y mejorar en función del avance tecnológico que determine el potencial eólico existente en el lugar de estudio.

### Hipótesis

La aplicación de la técnica de Machine Learning para la predicción de la velocidad del viento permitirá la estimación del potencial eólica en el sector de Lasso a partir de mediciones de la estación meteorológica

## CAPÍTULO I. FUNDAMENTACIÓN TEORICA- METODOLOGICA

### 1. FUNDAMENTACIÓN TEORICA- METODOLOGICA

#### 1.1 Antecedentes de la investigación

El pronóstico de la velocidad de viento a múltiples pasos es una tarea compleja. Sin embargo, utilizar las estrategias y modelos correctos puede facilitar esta tarea. En este trabajo se exploraron tres modelos de Inteligencia Artificial además de sus combinaciones y tres estrategias para pronósticos de múltiples pasos. Se observó que el modelo y estrategia con mejor aptitud fueron el modelo LSTM y MIMO, respectivamente. No obstante, usar modelos y estrategias más complejas tales como la estrategia directa y los modelos combinados presenta más dificultad para afinar los modelos porque tienen una gran cantidad de parámetros. Como trabajo futuro se propone combinar el uso de técnicas para optimizar estos parámetros complementando las técnicas presentadas en este trabajo.[1]

De acuerdo a las redes neuronales en la predicción de la velocidad del viento en la implementación del modelo neuronal, mientras más datos se puedan disponer para el entrenamiento de la red, más eficiente será el aprendizaje de la red y por ende la predicción será más cercana y con horizontes más amplios. La red neuronal diseñada ha sido capaz de predecir un horizonte de 72 horas, sin conocer previamente las salidas deseadas o targets, con una exactitud del 98.4%. [2]

Los estudios de predicción en series temporales de datos meteorológicos utilizando redes neuronales recurrentes en la primera etapa, donde se analizó la evolución de los errores variando alguno de sus hiperparámetros, queda patente que la red LSTM ofrece mejores resultados que la red Feedforward pero con el inconveniente de que el tiempo de entrenamiento es mucho mayor. En definitiva, se ha demostrado de forma empírica, que las redes Feedforward y las redes LSTM, con la configuración propuesta, presentan un comportamiento muy similar al modelo de persistencia. Es decir, parece ser que ambas redes generan sus salidas en función del último valor observado. Lo cual es un comportamiento

que no se esperaba con estas redes y podría ser materia de investigación en un futuro el determinar la razón de este proceder.[3]

El crecimiento acelerado que tienen las ERNC en la generación de energía en Chile, entrega un gran potencial para realizar investigación aplicada, y así facilitar el ingreso de estas energías al Sistema Eléctrico Nacional (SEN), reduciendo los costos del ingreso de las ERNC al SEN, que se reflejará en energía más económica para los chilenos. La predicción de la potencia eólica es un tema atractivo para investigar, como la serie de la velocidad del viento e indirectamente la serie de la potencia generada, tiene características que dificultan realizar un modelo estadístico de la serie, como la no linealidad y estacionalidad presentes en la serie hacen necesario recurrir a modelos estadísticos más complejos que tomen en cuenta estas características. Se mostró que el modelo LSTM al poder aprender dependencias a más largo plazo, tiene mejores resultados a horizontes de predicción más largo; en comparación al modelo ESN que realizan predicciones comparables al modelo SARI MA cuando el horizonte de predicción es muy largo.[4]

De acuerdo con las evidencias e información presentadas y como consecuencia de la escasa infraestructura actualmente disponible, resulta necesario mejorar diferentes aspectos en la generación de energía renovable en México. Entre ellos está el marco regulatorio a largo plazo y un marco legal que considere los aspectos sociales, ambientales y económicos. Se debe dar un mayor apoyo al desarrollo tecnológico en esta área y otorgar incentivos fiscales y económicos que haga competitiva la energía renovable frente a las formas convencionales de producción de electricidad. Es necesario facilitar el diseño, construcción y operación de las formas de transmisión y distribución de energía provenientes de fuentes renovables. Aunado a lo anterior, es indispensable la difusión pública de las bondades de energía eléctrica producida por medios eólicos y establecer programas de usuarios voluntarios de energía eólica, lo cual permitirá una aceptación social y llevará a los inversionistas considerar este tipo de industria como una opción rentable de producir energía limpia y sustentable. Todo ello nos llevará a ser competitivos en el mercado de certificados verdes y bonos carbono mundial[5].

Se evidencia la gran utilidad de las Tecnologías de Información Geográfica y las Técnicas de Evaluación Multicriterio para abordar la potencialidad eólica de los territorios, herramientas que ya habían demostrado parcialmente su validez para estudios similares, el modelo desarrollado resulta de gran utilidad en la planificación de estas infraestructuras, tanto para la Administración, que contará con un instrumento sencillo, repetible y capaz de identificar fácilmente zonas incompatibles, como para los propios promotores, que, ante un proyecto eólico, podrán descartar de antemano posibles ubicaciones y centrarse en la realización de análisis más detallados en las zonas que obtienen mejor puntuación. En este sentido, la definición de dos escenarios diferentes permite la obtención de resultados supeditados al mejor y el peor escenario de potencialidad posibles a esa escala, así como la comparación de ambos[6].

El aprovechamiento del potencial eólico en el Estado de Veracruz ha sido analizado con 16 estaciones anemométricas a 50m de altura, que cubren la totalidad del Estado, se ha realizado una evaluación del recurso eólico, y un mapa con SIG para determinar las zonas con mayor potencial. Los resultados han sido convenientes, se cuenta con un promedio de 5.45 m/s en todo el territorio Veracruzano, la sumatoria del promedio de la potencia y la energía generada en la región alcanzan los 14,432 kW y 10,694 MWh/año de energía exportada a la red, respectivamente, todo ello utilizando un aerogenerador ACCIONA AW 70/1500 Class I. Con estos datos se podrían ahorrar 6,188,646 litros de gasolina y una reducción anual bruta de emisiones GEI equivalentes a 28,806.1 toneladas de CO<sub>2</sub> (tCo<sub>2</sub>) [7]. Para el caso de estudio se obtiene 522 kWh con el cual se tendrá un Factor De Emisión Margen de Operación (OM) de 0.4509 obteniendo una reducción anual bruta de emisión de Gases de Efecto Invernadero equivalentes a 0.2353 toneladas de CO<sub>2</sub> (tCo<sub>2</sub>).

En las geociencias, frecuentemente se presenta la necesidad de obtener modelos en escenarios dimensionales ( $n > 1$ ) que permitan explicar el comportamiento de una variable  $U$  en un dominio fenomenológico complejo de la realidad objetiva y, en consecuencia, pronosticar su comportamiento bajo ciertas condiciones prefijadas [8]

Usando datos experimentales caracterizamos el potencial eólico a una altura de 35 metros sobre el nivel medio del mar en las instalaciones de la FIUADY. Se presentan también los resultados de la velocidad extrapolada para alturas superiores y su correspondiente evaluación del potencial eólico. Los resultados muestran que se requiere un incremento de cerca del 200% de la altura de referencia para aumentar en un 50% el potencial eólico, lo cual resulta una desventaja desde el punto de vista económico, además, al ser una región propensa a huracanes, el mantenimiento de torres de tamaño medio 40-50 metros podría facilitar y economizar el mantenimiento ante alguna amenaza climatológica. Nuestros resultados indican que sistemas de generación eólico son factibles en nuestra región siempre que su velocidad de viento nominal esté en el rango 5-7 m/s[9].

Se ha aprovechado el conocimiento de las series temporales que se ha podido adquirir durante la fase de identificación de la metodología clásica de Box-Jenkins, para poder configurar las redes neuronales que se utilizarán en el proceso de cálculo de pronósticos. Además, existe un problema adicional derivado de la gran longitud de las series temporales con las que se está trabajando, que es el del tiempo necesario para entrenar las redes. Dado el tamaño del conjunto de ejemplos de entrenamiento de la red, el tiempo necesario para que el proceso de aprendizaje sea el adecuado, es lo suficientemente grande como para que no se puedan calcular predicciones a corto plazo. Este problema se solucionó utilizando un método de entrenamiento selectivo, en el que la red se entrena sólo con aquellos ejemplos que son verdaderamente importantes en relación con el instante para el que se desea la previsión; de esta forma, es posible obtener pronósticos en un muy pocos minutos[10].

La aplicación de esta técnica es novedosa en el campo de la minería de procesos de negocio. Al usar registros de eventos para el entrenamiento de la red neuronal LSTM con datos sintéticos y del mundo real, se demuestra que el método predictivo puede ser efectivo, presentando una tasa de exactitud aceptable, en todos los casos de estudio.[11]

Un controlador con redes neuronales dinámicas aplicado a un modelo de generador eólico funciona de forma óptima ante variaciones de viento, mostrando una rápida estabilidad en

las variables controladas. El control óptimo usando redes neuronales permite obtener un estado transitorio de control más reducido, mostrando una mayor eficiencia al aplicarlo a sistemas dinámicos, llevando los estados del sistema a los valores deseados. La señal controlada con redes neuronales dinámicas presenta un sobreimpulso significativamente pequeño en comparación con el controlador PI, presentando una respuesta esperada debida al previo entrenamiento de la red [12].

El diseño de redes neuronales y su modelado en VHDL, permite definir los componentes modulares para la descripción de diversas configuraciones de RNA sobre hardware, lo que por sus características de procesamiento distribuido y paralelismo, logra un comportamiento más eficiente con respecto a su implementación en software. Otro aspecto de interés corresponde a la capacidad de reconfiguración en circuito, que facilita la actualización de la red de forma adaptativa, siendo el algoritmo de entrenamiento un aspecto de gran importancia en el modelo. El método de entrenamiento diseñado en este trabajo representa un aporte significativo con respecto a la velocidad de cálculo de los parámetros de la red. Éste resulta eficiente, gracias a la implementación de una arquitectura lógica, usando el registro de aproximaciones sucesivas para cada peso sináptico y polarización de la red, se obtiene una mayor velocidad de cómputo, con el menor uso de recursos del FPGA.[13]

Se debe ser conservador en la interpretación tanto de los datos obtenidos a raíz de mediciones realizadas en el campo por los equipos de medición, así como con las extrapolaciones que sean realizadas, ya sea en altura como en el terreno. Es conveniente estimar entre un 10% y un 20% menos en los datos obtenidos, y con esos valores realizar los cálculos y estimaciones.[14]

El objetivo de esta investigación es con base en el consumo del lugar de estudio, analizar el lugar y sus condiciones, para desarrollar un programa y poder simular la generación de energía eléctrica disminuyendo el consumo a la empresa abastecedora de energía, y determinar si es rentable o no la instalación del aerogenerador, y en cuanto tiempo retornan

la inversión al decidir instalar. El programa computacional implementado para esta investigación fue realizado en Matlab TM, incluye la evaluación de las curvas de potencia y arroja los datos de eficiencia y retorno de inversión de los cinco aerogeneradores que se tomaron para esta investigación. Además, brinda la posibilidad de hacer los cálculos a la altura que el usuario desee y también permite llevar a cabo el estudio en distintos sitios, debido a que se puede introducir el término de rugosidad para poder ofrecer los resultados deseados. Por supuesto, en este aspecto es posible apoyarse en Simulink, debido a que utiliza simulaciones más interactivas y un poco más sencillas, para un mayor entendimiento del software. Todo esto para poder tener un proyecto más acertado y eficiente. El software ayuda al usuario para poder asumir una mejor decisión del diseño de la instalación en cualquier sitio de estudio, obteniendo la máxima eficiencia [15].

Las estimaciones iniciales obtenidas para el potencial eólico, en los niveles de probabilidad del 25, 50 y 75%, son adecuadas para diversas actividades agrícolas y turísticas en la localidad bajo consideración. La ubicación de Torres demuestra la indicación de regiones favorables para el uso de energía eólica, expresada en el Mapa del Potencial Eólico de Rio Grande do Sul. La información obtenida por el trabajo constituye un subsidio para futuros estudios sobre la posibilidad de implantar una planta de energía eólica en la región [16].

A partir de la estimación del potencial eólico presente en las cuatro ubicaciones colombianas de carácter predominantemente urbano, y de acuerdo con el análisis de pre-factibilidad realizado para un proyecto de generación de mini eólica doméstico, no se lograría amortizar la inversión necesaria en ninguna de las ubicaciones, incluso considerando el caso con aplicación de incentivos legales contemplados por la ley 1715 tales como exclusión de IVA e impuestos arancelarios. Los resultados indican que las necesidades energéticas de un hogar promedio colombiano estimadas en 200 kWh mensuales, no logran ser cubiertas de manera satisfactoria[17].

El período de vientos mínimos coincide perfectamente con el período lluvioso del estado (primer semestre). Y período de vientos máximos, con el período más seco del estado

(segundo semestre). Dada la dirección predominante de los vientos, se concluye que la fuerza principal de los vientos son los vientos alisios, que se intensifican o debilitan por su acoplamiento con la circulación valle-montaña. En las regiones de São Benedito y Ubajara, los resultados muestran que existe un excelente potencial eólico allí. El factor de capacidad se considera muy bueno. Los resultados solo caracterizan los vientos a tiempo, lo que requiere más mediciones de tierra o simulaciones para caracterizar mejor las regiones estudiadas[18].

Utilizado una red neuronal LSTM la cual tiene como entrada la información histórica de los precios de las acciones, el trabajo consistió en predecir el precio de apertura de las acciones. Los resultados experimentales muestran un acercamiento a los valores reales de una acción debido a que se desarrollaron tres modelos para elegir finalmente cual es el más adecuado. Por lo tanto, el modelo tiene varias posibilidades de aplicación y puede ser competitivo con algunos modelos existentes, principalmente por su facilidad de implementación, en este sentido el trabajo futuro tiene varias direcciones ya que este trabajo ha encontrado que un LSTM basado en la economía del comportamiento tiene más aciertos para la predicción de precios que otros métodos, es decir, el impacto de los datos históricos sobre las tendencias de los precios es demasiado singular y es posible que no se pueda pronosticar completamente y con precisión el precio en un día determinado.[19]

Las redes LSTM proporcionan errores inferiores dentro de la muestra a los de otras alternativas entre las cuatro técnicas analizadas, las redes se encuentran, en todas las CCAA, entre las dos alternativas que ofrecen menores errores de estimación y en la mayoría de los casos es la técnica mejor, en definitiva, la incorporación de las herramientas que se desarrollan en el marco de la IA y específicamente de las redes neuronales LSTM abre un número de posibilidades muy importante a la vez que muestra ya unos resultados muy positivos en materia de seguimiento de la coyuntura regional.[20]

## 1.2 Fundamentación Teórica.

### 1.2.1 Generalidades

Generalidades de Machine Learning Machine learning es un conjunto de técnicas que hacen parte de la inteligencia artificial, que basadas en algoritmos buscan el aprendizaje dentro de grandes conjuntos de datos. Una característica muy importante de estos algoritmos es la predicción de nuevos casos basándose en la experiencia aprendida del conjunto de datos utilizados para su entrenamiento, a esto se le conoce en la literatura como generalización. [21].

### 1.2.2 Técnicas de Machine Learning Regresión Logística

La regresión logística es un tipo especial de regresión que se utiliza para explicar y predecir una variable categórica binaria, en función de varias variables independientes que a su vez puede ser cuantitativas o cualitativas. Por sus características, los modelos de regresión logística permiten dos finalidades[21]:

- a. Cuantificar la importancia de la relación existente entre cada una de las covariables o variables independientes y la variable dependiente, lo que lleva implícito también clarificar la existencia de interacción y confusión entre covariables respecto a la variable dependiente (es decir, conocer la “odds ratio” para cada covariable)[21].
- b. Clasificar individuos dentro de las categorías (presente/ausente) de la variable dependiente, según la probabilidad que tenga de pertenecer a una de ellas dada la presencia de determinadas covariables. No cabe duda que la regresión logística es una de las herramientas estadísticas con mejor capacidad para el análisis de datos en investigación clínica y epidemiología, de ahí su amplia utilización. El objetivo primordial que resuelve esta técnica es el de modelar cómo influye en la probabilidad de aparición de un suceso, habitualmente dicotómico, la presencia o no de diversos factores y el valor o nivel de los mismos. También puede ser usada para estimar la probabilidad de aparición de cada una de las posibilidades de un suceso con más de dos categorías (politómico)[21].

## Arboles de Decisión

Un árbol de decisión es una estructura en la cual cada nodo interno denota una prueba sobre uno o varios atributos, cada rama representa una salida de la prueba y los nodos hoja representan clases. La característica principal de los árboles de decisión es que son modelos de caja blanca en los cuales se puede ver directamente la frecuencia de aparición de cada atributo. Además, le permite al experto conocer el atributo con mayor poder de clasificación, es decir, aquel que se localice en el nodo raíz[21].

## Máquinas de Soporte Vectorial.

Las Máquinas de Soporte Vectorial (del inglés Support Vector Machine) fueron desarrolladas en 1995 por Vladimir Vapnik cuando propone un modelo matemático para la resolución de problemas de clasificación y regresión el cual llamaron Modelo MSV. Están basadas en la teoría de aprendizaje estadístico que permiten resolver problemas de clasificación y regresión de manera eficiente[21].

El éxito de las máquinas de soporte vectorial radica en tres ventajas fundamentales:

- a. Poseen una sólida fundamentación matemática[21].
- b. Se basan en el concepto de minimización del riesgo estructural, esto es, minimizar la probabilidad de una clasificación errónea sobre nuevos ejemplos, particularmente importante cuando se dispone de pocos datos de entrenamiento[21].
- c. Se disponen de potentes herramientas y algoritmos para hallar la solución de manera rápida y eficiente. Las máquinas de soporte vectorial a diferencia de las redes neuronales, abstraen el problema desde un espacio de atributos a un espacio de patrones de características con mayor dimensión, a fin de que puedan ser separadas por un hiperplano. Así, mediante una función no lineal de mapeo apropiada, que aumente la dimensión de forma adecuada, es posible separar muestras que pertenezcan a dos categorías diferentes mediante un hiperplano[21].

### **Clasificación Linealmente Separables:**

Dentro de los conjuntos de datos que abarcamos con las máquinas de soporte vectorial, encontramos aquellos que son linealmente separables, y en ellos, minimizar la función de costos resulta muy sencillo. La hipótesis de partida es que las clases son linealmente separables y por ello existen infinitos hiperplanos que separan las muestras de una clase, de la otra. Los puntos del espacio que caen dentro de cada uno de estos hiperplanos son los que satisfacen la siguiente expresión[21].

Donde  $w \in \mathbb{R}^d$ , siendo  $d$  la dimensión del espacio de entrada.

La resolución para dicho caso sería suponer que se tiene un conjunto de  $n$  datos separables linealmente  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$  donde  $x_i \in \mathbb{R}^d$  y  $y_i \in \{-1, 1\}$ . Se cumplirá, según el lado en el que se encuentren respecto del hiperplano. [21].

Hay casos de datos linealmente separables en los que puede existir ruido debido a errores en la medida de los datos o por la presencia de algún dato atípico o extremo. En dichos casos no es conveniente que la máquina de soporte vectorial se ajuste totalmente a los datos. Este punto no debería ser considerado para hallar la frontera de decisión ya que podría alterar los resultados deseados nos llevaría a clasificaciones incorrectas [21].

### **1.2.3 Clasificación**

#### **Linealmente no Separable**

Cuando los datos no son linealmente separables, existe la posibilidad de transformar los datos a un espacio de mayor dimensión utilizando una función, donde se encontrará un hiperplano que los pueda separar. La frontera de decisión resultante en el espacio de entrada ya no será lineal y vendrá dada por otro tipo de función que pueda ser polinómica de grado superior a 1, gaussiana, sigmoide, entre otras, dichas funciones se conocen como funciones núcleo o “Kernel”. Las muestras una vez proyectadas, pueden usarse como un nuevo conjunto de entrenamiento, de esta forma se buscará una frontera lineal en el espacio, dicha frontera otra

en el espacio de partidacuya forma dependerá de la función de proyección empleada y de las muestras del conjunto de entrenamiento[21].

Como se nota ya los datos en la formulación de la máquina de soporte vectorial sólo aparecen como producto entre las muestras, en este caso solo se necesita la función Kernel, tal que Teniendo esta función se puede aplicar el algoritmo de entrenamiento de la Máquina de soporte vectorial sin conocer explícitamente cual es la transformación o incluso el espacio. La única condición necesaria es que el kernel usado esté correctamente definido[21].

Las funciones más utilizadas son:

La Función Polinómica

La Función Gaussiana

La Función Sigmoide

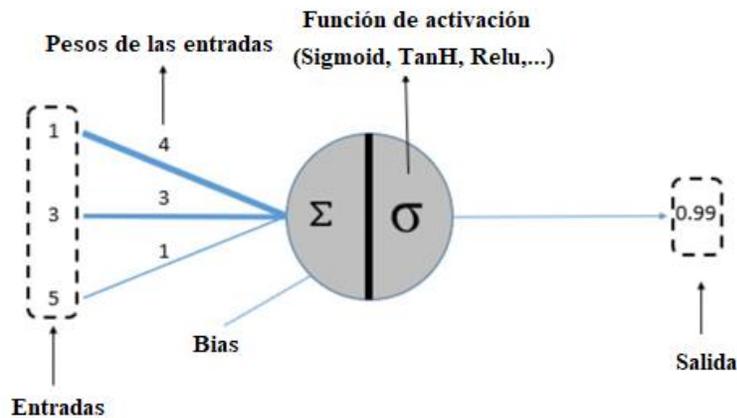
La selección del mejor kernel para una aplicación es todavía un tema de investigación. El procedimiento más común es el de seleccionar los parámetros de kernel (el grado del polinomio  $d$  para las funciones polinomiales o el ancho del kernel para la función Gaussiana) calibrando estos parámetros en conjunto con el proceso de selección del modelo (parámetro  $C$  que controla la generalización del modelo) [21].

### **Generalidades de las Redes Neuronales**

La neurona es la unidad fundamental del sistema nervioso y en particular del cerebro. Cada neurona es una simple unidad procesadora que recibe y combina señales desde y hacia otras neuronas. Si la combinación de entradas es suficientemente fuerte, la salida de la neurona se activa. El cerebro consiste en uno o varios billones de neuronas densamente interconectadas. El axón (salida) de la neurona se ramifica y está conectada a las dendritas (entradas) de otras neuronas a través de uniones llamadas sinapsis. La eficacia de la sinapsis es modificable durante el proceso de aprendizaje de la red [21].

Desde la primera mitad del siglo XX se han empezado a desarrollar modelos computacionales que han intentado emular el comportamiento del cerebro humano. Aunque se han propuesto una gran cantidad de ellos, todos usan una estructura en red en la cual los nodos o neuronas

son procesos numéricos que involucran estados de otros nodos según sus uniones. Una clase de estos modelos computacionales son las redes neuronales artificiales, las cuales se han hecho muy populares debido a la facilidad en su uso e implementación y la habilidad para aproximar cualquier función matemática. (Posee). El gráfico 1 muestra el modelo de una Red Neuronal Artificial[21].



**Figura. 1.** Red Neuronal Artificial. [21]

#### **1.2.4 Las principales ventajas que representan**

**Aprendizaje Adaptativo:** Capacidad de aprender a realizar tareas basadas en un entrenamiento o en una experiencia inicial[21].

**Auto-organización:** Una red neuronal puede crear su propia organización o representación de la información que recibe mediante una etapa de aprendizaje[21].

**Tolerancia a fallos:** La destrucción parcial de una red conduce a una degradación de su estructura; sin embargo, algunas capacidades de la red se pueden retener, incluso sufriendo un gran daño.[21]

**Operación en tiempo real:** Los cálculos neuronales pueden ser realizados en paralelo; para esto se diseñan y fabrican máquinas con hardware especial para obtener esta capacidad. [21]

**Fácil inserción dentro de la tecnología existente:** Se pueden obtener chips especializados para redes neuronales que mejoran su capacidad en ciertas tareas. Ello facilitará la integración modular en los sistemas existentes.[21]

### 1.2.5 Tipos de Redes Neuronales:

#### Perceptron Multicapa

Es responsable de la popularidad de las redes neuronales a partir de mediados de los años ochenta, como su nombre lo indica, perceptron multicapa, tiene sus neuronas organizadas por capas, su número es variable dependiendo de la funcionalidad de la red, pero en general se distinguen estos tipos de capas. [21]

- Una capa de entrada donde se alimenta a los datos de la red (capa 0).
- Capas ocultas cuyo número es variable.
- Una capa de salida donde se lee el resultado del cálculo de la red El gráfico 2 ilustra la estructura de una red Perceptron Multicapa.

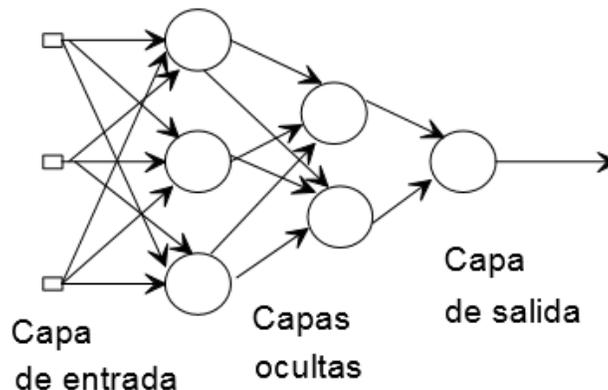


Figura. 2. Estructura de una Red Neuronal Perceptron Multicapa. [21]

#### Método de selección de los ejemplos de entrenamiento de la RN

Una de las características fundamentales de las RNAs es la velocidad de respuesta; así es posible obtener soluciones casi instantáneamente para ciertos tipos de redes, y en muy poco tiempo en la mayoría. Pero para que esto sea así, al menos en el perceptrón multicapa (que es la arquitectura que utilizaremos aquí), y suponiendo que el ordenador en el que se trabajará es un Pentium III o IV, debería darse alguna de estas condiciones[10]:

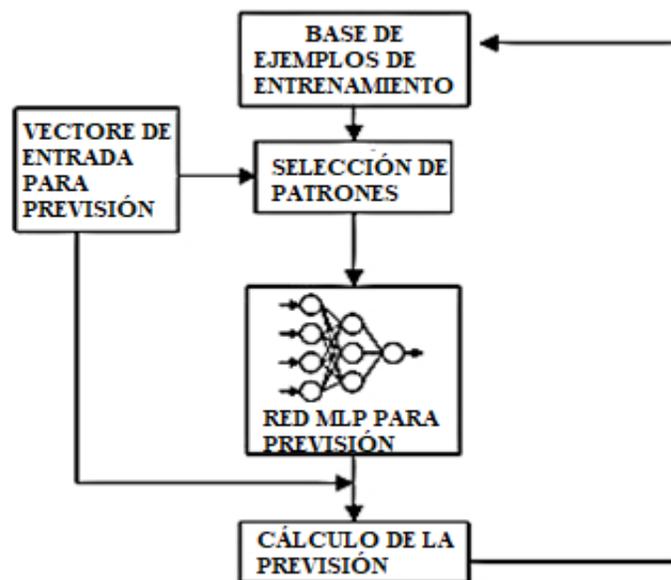
1. La configuración de la red no debe ser excesivamente grande, de forma que el número de enlaces cuyo peso es necesario calcular no sea muy elevado[10].

2. El conjunto de ejemplos de entrenamiento no debe ser demasiado grande; cuanto más pequeño sea, más veces por unidad de tiempo pasará cada ejemplo por la red y más rápido se llegará a la solución[10].

3. La composición del conjunto de ejemplos de entrenamiento deberá ser homogénea, es decir, cuanto más parecidos entre sí sean los ejemplos que la red utiliza para aprender, más rápido será el proceso de aprendizaje[10].

De acuerdo con esto, una forma de acelerar el proceso es hacer una selección previa de los patrones o ejemplos de entrenamiento, de forma que se escojan los estrictamente necesarios como para que todas las relaciones subyacentes en la serie temporal estén debidamente representadas. En este caso, el proceso de entrenamiento utilizando un conjunto de ejemplos pequeño y que además está compuesto por casos muy similares entre sí (sin llegar a ser exactamente iguales), será sensiblemente más corto, de tal manera que la RNA estará lista para calcular previsiones en un plazo de tiempo muy breve[10].

En los últimos años han aparecido trabajos en los que se incluyen algoritmos de selección de patrones incorporados a una metodología de previsión con redes neuronales. Estos algoritmos se basan en la idea del “entrenamiento selectivo y continuo”, que se ha logrado implementar gracias al incremento de la potencia de las computadoras [10].



**Figura. 3.** Método selectivo de entrenamiento de redes neuronales. [10]

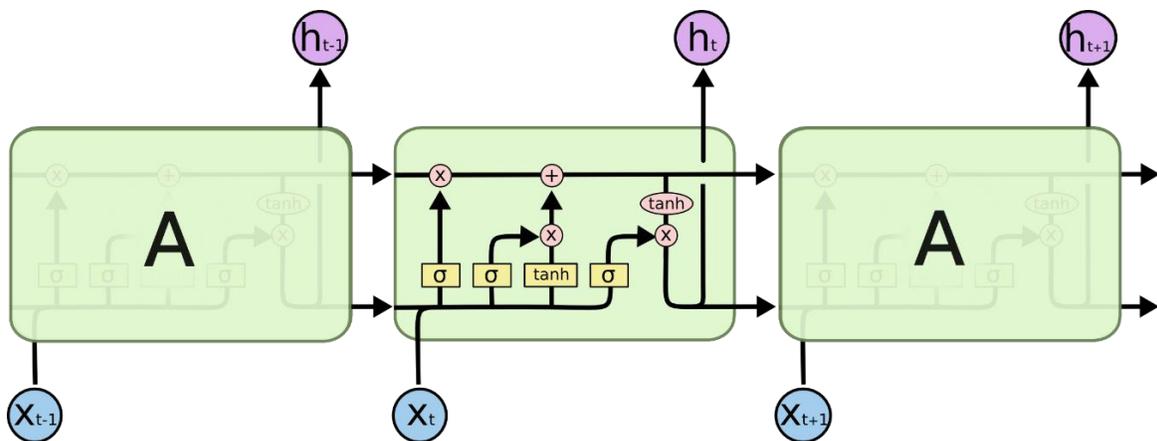
## Redes LSTM

Las redes de memoria a largo plazo, generalmente llamadas simplemente "LSTM", son un tipo especial de RNN, capaces de aprender dependencias a largo plazo. Fueron introducidos por Hochreiter & Schmidhuber (1997) , y muchas personas los refinaron y popularizaron en sus siguientes trabajos. Funcionan tremendamente bien en una gran variedad de problemas y ahora se utilizan ampliamente.[22]

Los LSTM están diseñados explícitamente para evitar el problema de la dependencia a largo plazo. Recordar información durante largos períodos de tiempo es prácticamente su comportamiento predeterminado, ¡no es algo que les cueste aprender![22]

Todas las redes neuronales recurrentes tienen la forma de una cadena de módulos repetidos de red neuronal. En RNN estándar, este módulo repetitivo tendrá una estructura muy simple, como una sola capa de tanh [22]

Los LSTM también tienen esta estructura similar a una cadena, pero el módulo repetitivo tiene una estructura diferente. En lugar de tener una sola capa de red neuronal, hay cuatro que interactúan de una manera muy especial. [22]



**Figura. 4.** El módulo repetitivo en un LSTM contiene cuatro capas que interactúan. [22]

### 1.2.6 Velocidad del viento

El viento es consecuencia del movimiento del aire en una dirección y velocidad específica. Se utilizan anemómetros para medir su velocidad, en meteorología se utilizan los de molinetes con aspas que giran por la fuerza del viento y así es posible medir la cantidad de vueltas que da en un determinado lapso de tiempo. [23]



**Figura. 5.** Anemómetro. [23]

Para determinar hacia donde sopla el viento se debe identificar las fuerzas que afectan el movimiento horizontal del aire

#### **Gradiente de Presión**

Si a una misma altura tenemos zonas de baja y alta presión, el aire se moverá de la zona de alta presión a la de baja. Cuanto mayor sea la diferencia de presión mayor será el movimiento del aire en este desplazamiento [23].

#### **Coriolis.**

Debido a la rotación de la Tierra sobre su eje, se produce una desviación inercial en los vientos hacia la izquierda en el hemisferio sur y a la derecha en el norte, lo cual hace que el viento tienda a ser paralelo a las isobaras. Se opone a la fuerza de gradiente de presión. Fue descubierto en 1835 por el ingeniero y matemático francés Gaspard-Gustave de Coriolis [23].



Figura. 6. Fuerza de Coriolis. [23]

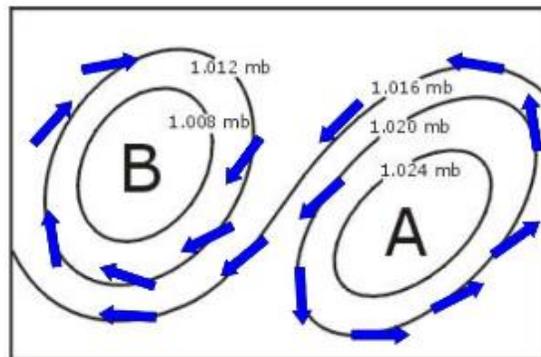


Figura. 7. Movimiento de variaciones del viento. [23]

### Centrífuga

En isobaras curvas actúa radialmente hacia el exterior, de magnitud pequeña no tiene normalmente incidencia a menos en vientos fuertes en trayectorias curvas acelerando o desacelerando según dicha curvatura [23].

### Fricción

Producida por el rozamiento del aire con la superficie de la Tierra, actúa aproximadamente hasta los 1000 metros de altura, por arriba de ese nivel puede considerarse insignificante su efecto fluyendo el viento en forma paralela a las isobaras [23].

La dirección del viento se representa según los puntos cardinales Norte (N), Sur (S), Este (E) y Oeste (O) o bien en grados acimutales del Norte (0 -360°) siendo 0° Viento Norte, 45°

viento Noreste, 90° Viento Leste, 135° Sudeste, 180° Sur, 225° Suroeste, 270° Oeste y 315° Noroeste. Las unidades de medida utilizadas comúnmente son [23] :

- Nudos (knen ISO o ktpor el inglés knot) donde 1 ktequvale a 1,852 km/h
- Kilómetros por hora (Km/h) donde 1 Km/hequvale a 0,27778 m/s
- Metros por segundo (m/s)
- Beaufort basada en el estado del mar, olas y fuerza del viento

### 1.2.7 Densidad del Potencia Eólico

Las variaciones que presenta el viento dependen de las condiciones climatológicas de la zona, por esto es necesario el uso de métodos estadísticos para evaluar el aprovechamiento del potencial eólico en la zona [25]

$$P1 = \frac{1}{2} \rho v^3$$

donde p es la densidad del aire; v-3 es el promedio de las velocidades del viento al cubo[7]

### Potencial eólico

## PARTES DE UN MAPA EÓLICO

### Datos

Los datos más frecuentemente representados sobre el mapa para una altura determinada son la velocidad media del viento (medida en m/s), la dirección del viento promedio (expresada en forma de flecha o símbolos característicos de ploteo en meteorología), la densidad media de energía (medida en W/m<sup>2</sup>), la distribución de frecuencias, las rosas de los vientos, los parámetros de Weibull (A y k), los estudios de rafagosidad de los vientos y estudios de turbulencia, entre otros. Además, los resultados deben presentar no sólo los datos promedio históricos, sino también los regímenes estacionales y los ciclos diurnos y nocturnos del recurso[14].

Existen otros datos que se utilizan como entradas para los modelos de mapa eólico, pero que puede resultar muy útil su utilización y representación como un compendio de la salida. Ellos son el mapa de rugosidad del suelo, el mapa de uso del suelo y cobertura vegetal, y mapas topográficos. Todos estos datos estarán representados a distintas alturas, siendo en la actualidad las más comunes 30 y 50 m; aunque también hay atlas que representan la información a 10, 25, 30, 50, 75, 80, 100, 125 y 200 m. En realidad, una vez hechos los cálculos y conociendo cómo se comporta el perfil de vientos para una zona, los valores pueden ser fácilmente extrapolados en altura mediante métodos como los mencionados anteriormente [14]

### **Tipos de datos necesarios**

Los datos con los que se debe contar para la confección del mapa eólico de una región son de muy variada fuente y, dependiendo del método empleado para realizar la labor, tendrán que ser de distinto tipo, teniendo cada método sus entradas de datos obligatorias bien definidas. De esta manera, al momento de desarrollar un nuevo mapa eólico debe relevarse muy bien qué datos hay disponibles y de qué manera pueden ser utilizados para emplear qué método. Sin embargo, y abstrayéndose del método a emplear, los datos necesarios para la tarea de mapeo pueden sintetizarse en el siguiente listado[14]:

- Mediciones anemométricas o mediciones de superficie.
- Datos orográficos.
- Datos topográficos.
- Datos de uso del suelo/cobertura natural.
- Imágenes satelitales.

Los datos medidos en superficie son de suma importancia, dado que pueden utilizarse tanto para obtener el mapa eólico de una región como para validar los resultados obtenidos por otros métodos que no utilicen mediciones como entradas. Por otra parte, los datos de superficie siguen siendo los más exactos a la hora de realizar un proyecto[14]

### 1.3 Fundamentación metodológica

El presente proyecto se dedica a la predicción del potencial eólico en el sector de Lasso, aplicara diferentes variables acorde a sus características y al enfoque investigativo que se requiere denominar en función a la problemática suscitada y con ello configurara el diseño de la investigación en función de su naturaleza hipotética con las variables en estudio misma que describirán un plan estratégico metodológico para realizar el estudio mediante los métodos, técnicas, estrategias y procedimientos que conlleva a la obtención de datos recolectados en la investigación de campo.

#### **Enfoque cualitativo**

Con este enfoque se realizará un sondeo de variable que serán recolectadas a base de central meteorológica, en el que se exprese las diferentes variables que se tendrá en el potencial eólico, a partir de los datos recolectados, se ingresara en el algoritmo para una predicción respecto del resultado verdadero.

#### **Tipo de investigación**

La importancia de la presente investigación es entender las particularidades de cada paradigma, sus características y las relaciones que se puede establecer entre ellos a los contextos reales de investigación, por medio de ellos podemos obtener las bases teóricas y validar los resultados obtenidos, mediante la investigación bibliográfica y la de campo.

**Investigación descriptiva.** – Se denomina a la recolección de las variables que se obtendrá de la estación meteorológica, búsqueda específica de las variables que influyen dentro del potencial eólico que se someta a un análisis de la investigación planteada, es decir, únicamente pretenden medir o recoger información de manera independiente o conjunta de las variables a las que se refieren.

**Investigación explicativa.** - Está dirigido a responder las causas de los eventos y fenómenos físicos; por ende, como su nombre lo indica, su interés se centra en explicar el método que se realizara para poder determinar la precisión del potencial eólico y en condiciones se manifiesta.

**Investigación correlacional.** – Se pretenden responder a las variables en estudio de la investigación en las que se indaga la fundamentación teórica bibliográfica de la variable dependiente e independiente, en consiguiente se requiere validar la información mediante la comprobación de hipótesis según la investigación o mediante los resultados de predicción.

### **Técnica de recolección de información**

Se utilizar la estación meteorológica la cual está instalada en el Instituto Superior Tecnológico Cotopaxi, la misma que se recolectara datos diarios a partir del año 2018 los mismo que servirán para tener datos anteriores para su procesamiento de información.

### **Procedimientos empleados para la obtención y análisis de la información.**

El plan de recolección cumple con las estrategias metodológicas requeridas para cumplir con los objetivos planeados de la información que se encuentra basado a un enfoque de investigación cualitativo por la situación actual en la cual se recolectara los datos de la estación meteorológica la misma que será ingresada en el algoritmo que se realice en Machine Learning para la predicción del potencial eólico.

## **1.4 Conclusiones Capítulo I**

- La elaboración del planteamiento del problema permite enfocarse y encontrar el problema que se presenta en la sociedad y al cual se tiene que enfocar para determinar una o varias soluciones al tema planteado el cual permitirá dar soluciones a fenómenos que se asemejen al objeto de estudio.

- Las referencias bibliográficas realizadas permiten encontrar información de temas similares las cuales son fundamentales para avanzar y tener bases principales que nos permitan guiar y determinar el procedimiento que se realizara para enfocar y lograr obtener resultados del tema.
- De acuerdo a las redes neuronales en la predicción de la velocidad del viento en la implementación del modelo neuronal, mientras más datos se puedan disponer para el entrenamiento de la red, más eficiente será el aprendizaje de la red y por ende la predicción será más cercana y con horizontes más amplios. La red neuronal diseñada ha sido capaz de predecir un horizonte de 72 horas, sin conocer previamente las salidas deseadas o targets, con una exactitud del 98.4%. [2], tomando como referencia los estudios realizados recomiendan utilizar mayor cantidad de datos para obtener un entrenamiento en la red neuronal con ello obteniendo un resultado de predicción preciso, para lo cual se recomienda utilizar la red LSTM debido a que memoriza datos a largo plazo a diferencia de otros tipos de métodos realizados y comparados

## **CAPÍTULO II. PROPUESTA**

### **2.1 Título del proyecto.**

Aplicación de la técnica de Machine Learning para la estimación del potencial eólica en el sector de Lasso a partir de mediciones de una estación meteorológica

### **2.2 Objetivo del proyecto.**

Determinar la predicción de la velocidad del viento mensual mediante la aplicación de la técnica de Machine Learning en base a la recolección de datos con la estación meteorológica ubicada en el sector de Lasso para determinar el potencial eólico.

### **2.3 Descripción de la propuesta.**

Determinar el potencial eólico en función de la predicción de la variación del viento se ha visto afectado por factores como la rugosidad y la topografía del terreno que producen perturbaciones en su movimiento. La presencia de colinas puede generar, por un lado, aumento de velocidad en el viento hasta llegar a la cima, pero, a su vez, en la parte inferior aumenta la turbulencia produciendo efectos de recirculación. Este comportamiento no permite conocer en detalle el desarrollo de los perfiles de velocidad y la energía cinética turbulenta del viento.

En este sentido, por el gran interés en el estudio para determinar el potencial eólico se aplicará técnicas de Machine Learning que se ajusten a las variables de la velocidad del viento que se recolectaron por la estación meteorológica ubicada en el sector de Lasso la misma que permitirá predecir la velocidad del viento de un mes y con ello poder determinar mediante la distribución de Weibull la velocidad de viento promedio y el potencial eólico que se genere por mes de la velocidad del viento en terrenos complejos, el presente proyecto tiene como

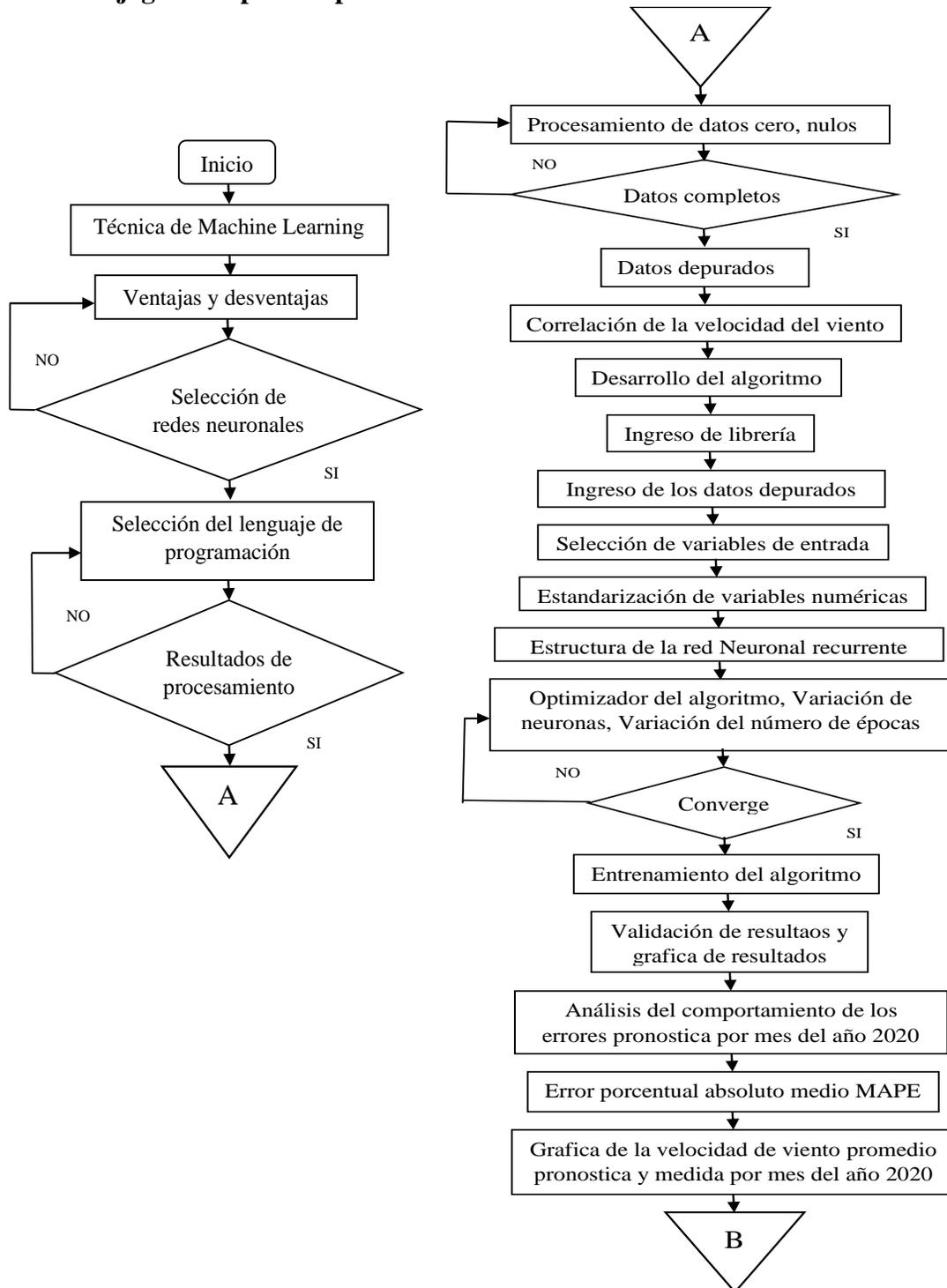
objetivo predecir la velocidad del viento en base a la recolección de datos mediante una estación meteorológica que no requiere de una conexión eléctrica ni de hilar cables, pues funciona totalmente con su sistema fotovoltaico, transfiriendo la información inalámbrica hasta 1000 m de distancia, viene con sensores de velocidad y dirección de viento, temperatura y humedad y presión barométrica, los sensores y la precisión de las mismas está direccionado al mercado semiprofesional, está compuesto de Anemómetro con un Rango: 0.4 ... 67 m/s, Precisión: +/- 5% y Resolución: 1 m/s y Veleta con un Rango: 0 ... 360°, Precisión: +/-3°, Resolución: 1° este sistema combinado tipo RM Young de forma de avión con una hélice de eje horizontal y una veleta que direcciona la hélice al viento.

Además cuenta con una página web de Rainwise que está equipado para cargar datos en tiempo real, muestra información desde cualquier navegador web, registra y archiva datos, valores mínimos y máximos el cual se utilizara para descargas de datos de la velocidad de viento que es un parámetro a ser analizado cabe mencionar que para la obtención de datos la estación meteorológica se realizó la compra de la licencia por dos años (2019 y 2020), la cual permite descargar datos con intervalos de tiempo reducidos para su análisis, el equipo se instaló en el Sector de Lasso sector la Avelina en el Instituto Superior Tecnológico Cotopaxi en el área administrativa siendo el lugar con mayor altura para su respectiva medición.

Cabe mencionar que entre más valores se recolecten mayor será el entrenamiento del algoritmo, dichos datos servirán para entrenar las redes neuronales y evaluar los resultados obtenidos con porcentajes de predicción, el algoritmo planteado permitirá realizar análisis con diferentes tipos de data que se obtengan de diferentes lugares con la cual se podrá determinar la instalación de un sistema de aerogenerador que cubra un porcentaje de la demanda he incluso la demanda completa en lugares inasumibles de la red eléctrica.

## 2.4 Metodología o procedimientos empleados para el cumplimiento de los objetivos planteados

### 2.4.1 Flujograma aplicado para el análisis de datos.



**Figura. 8.** Flujograma de selección de la técnica de Machine Learning y programación del algoritmo, parte A

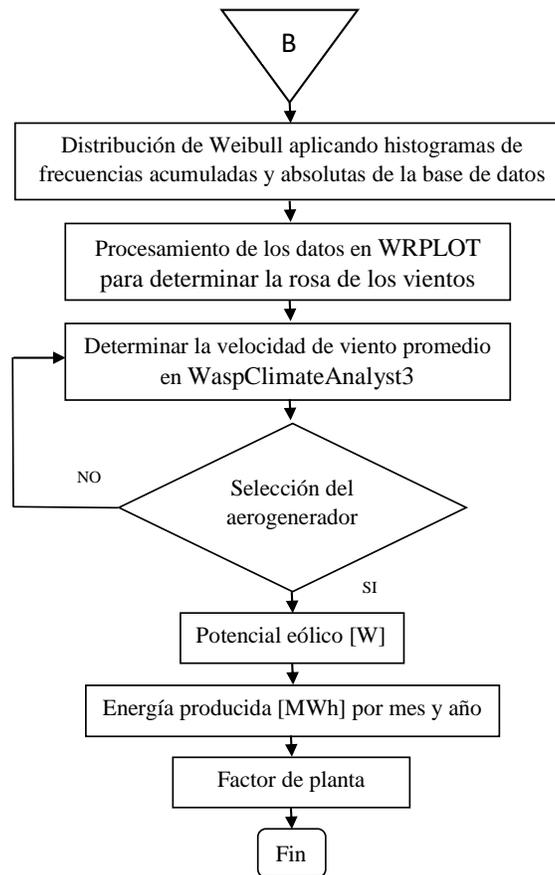


Figura. 9. Flujograma de la distribución de Weibull, parte B

## 2.4.2 Seudocódigo aplicado para el procesamiento de datos.

Inicio

```
#Importar Librería: Procesamiento, estructura, Lectura y graficas
#Eliminar números valores nulos de Las componentes:
"Windspeed", "Baro Avg", "Hum Avg", "Temp Avg"
#Realizar el análisis de correlación de Las componentes seleccionadas
"Windspeed", "Baro Avg", "Hum Avg", "Temp Avg"
Lectura de los datos: año 2019 - 2020 depurado
data =pd.read_csv('DEPURACION_FINAL.csv')
#Definir variables: data_final
"Windspeed"
#Transforma texto a datettime
    fechaInicio_real = str(data_train[-time_step:].index[0]) #2020-08-10 08:00:00
    fechaFin = datetime.strptime(fechaFin, "%Y-%m-%d %H:%M:%S")
    fechaInicio_real = datetime.strptime(fechaInicio_real, "%Y-%m-%d %H:%M:%S")
    fecha_inicio = datetime.strptime(fecha_inicio, "%Y-%m-%d %H:%M:%S")
#Escalar valores de La dataframe
from sklearn.preprocessing import MinMaxScaler
sc = MinMaxScaler(feature_range = (0,1))
#Separación de datos de entrenamiento 80% de La data
#Definición de La LSTM con 25 neuronas de entrada, 25 neuronas en capa oculta, una s
model.add(LSTM(units=num_neuronas, return_sequences = True, input_shape=dim_entrada
model.add(Dropout(0.4))
model.add(LSTM(units=num_neuronas))
model.add(Dropout(0.2))
    model.add(Dense(units = dim_salida))
#Entrenamiento del algoritmo con 25 épocas
model, history = NN_LSTM (optimizador = 'rmsprop', neuronas = 25, epocas = 25)
#Iteramos hasta que tengamos La prediccion de todas Las horas
    pos =0
    for i in range(int(total_horas)):
        X_test = []
        X_test.append(data_inicial)
        X_test = np.array(X_test)
        X_test = np.reshape(X_test, (X_test.shape[0], X_test.shape[1], 1))
        X_test.shape #(totaldatos, time_step, 1) - (1, 60, 1)
        # data_inicial.shape
        predic = model.predict(X_test)
        predic_final.append(predic)
#Gráfica Las pérdidas de entrenamiento y perdidas de prueba
    plt.plot(history.history['loss'], label='LOSS', color='brown')
    plt.plot(history.history['mean_absolute_error'], label='MAE', color='blue')
#Selección del 20% de La data no vista por el algoritmo
    size = int(0.2*data_final.shape[0])
#Grafica de Los valores pronosticados y medidos
    plt.plot(semana_estudio_test.index,
    semana_estudio_test,
    color='blue',
    label='Velocidad del Viento Real')
    label='Predicción de la Velocidad del Viento')
# Obtención de resultados pronosticados de La velocidad del viento
semana_estudio_pred.to_csv('prediccion_octubre_semana.csv')
FIN
```

Figura. 10. Pseudocódigo para la implementación del algoritmo LSTM

### 2.4.3 Machine Learning utilizando redes neuronales

En simulación, son interesante los estudios de algoritmos metaheurísticos a los que se recurren con la intención de aumentar la precisión de los pronósticos. Tal como lo reflejan en un modelo analítico de abastecimiento de medicamentos basado en redes neuronales artificiales para optimizar el inventario de los medicamentos del sector privado de salud. El estudio obtiene como resultado la afirmación que el modelo de red neuronal artificial concurrente tiene mayor precisión frente a los modelos estadísticos.[26]

#### **Red neuronal Feedforward**

Una red neuronal feedforward es un tipo de red neuronal donde las conexiones de la unidad no viajan en un bucle, sino en una única ruta dirigida. Esto difiere de una red neuronal recurrente, donde la información puede moverse hacia adelante y hacia atrás en todo el sistema. Una red neuronal feedforward es quizás el tipo más común de red neuronal, ya que es una de las más fáciles de entender y configurar. Estos tipos de redes neuronales se utilizan en la minería de datos y otras áreas de estudio donde se requiere un comportamiento predictivo.[27]

Una red neuronal es una red de inteligencia artificial diseñada para imitar libremente los procesos de «pensamiento» de un cerebro humano. Al introducir cadenas de datos en la red, la computadora tiene la oportunidad de «aprender» los patrones que fluyen a través de ella, lo que le permite identificar correctamente las respuestas y proporcionar análisis de tendencias. Se utilizan en tareas en las que se requiere cierto grado de aprendizaje y reconocimiento de patrones, como durante las operaciones de minería de datos. La minería de datos es simplemente el análisis de tendencias a partir de una colección de información, como el análisis de las tendencias de compra de los consumidores y las progresiones del mercado de valores. [27]

#### **Las redes neuronales recurrentes**

Son muy potentes para todo lo que tiene que ver con el análisis secuencias, tiene un procedimiento de aprendizaje más eficiente y alcanza un nivel de estabilidad más rápido tanto en el entrenamiento como en su funcionamiento en la predicción que la red feedforward. Las

redes recurrentes alcanzan una efectividad del 73 % , superando a la red feedforward, que presentó una efectividad del 69 %.[28]

Las redes LSTM, incorporan una serie de pasos para decidir qué información va a ser almacenada y cual, borrada, son capaces de realizar una amplia variedad de tareas computacionales incluyendo el tratamiento de secuencias, la continuación de una trayectoria la predicción no lineal y la modelación de sistemas dinámicos es capaz de “recordar” un dato relevante en la secuencia y de preservarlo por varios instantes de tiempo. Por tanto, puede tener una memoria tanto de corto plazo (como las Redes Recurrentes básicas) como también de largo plazo.

### **Redes convolucionales**

La CNN es un tipo de Red Neuronal Artificial con aprendizaje supervisado que procesa sus capas imitando al contexto visual del ojo humano para identificar distintas características en las entradas que en definitiva hacen que pueda identificar objetos y “ver”. Para ello, la CNN contiene varias capas ocultas especializadas y con una jerarquía: esto quiere decir que las primeras capas pueden detectar líneas, curvas y se van especializando hasta llegar a capas más profundas que reconocen formas complejas como un rostro o la silueta de un animal. funcionando muy bien en reconocimiento de voz y procesamiento de imágenes, funcionan mejor que cualquier otro algoritmo.[29]

### **El método de Monte Carlo**

Es una técnica de análisis numérico para simular, que se basa en el uso de una secuencia de números aleatorios, con el propósito de muestrear los valores correspondientes a las variables probabilísticas de un determinado problema. A diferencia de la simulación normal, este consiste en la combinación de conceptos estadísticos con la capacidad que tienen los ordenadores para generar números pseudo-aleatorios y automatizar cálculos. En este proyecto, se necesitó de la elección de un algoritmo metaheurístico para asegurar la precisión de los datos simulados que reflejen la demanda de ventas de los años futuros que tendrá el negocio en estudio tiene la ventaja de proporcionar resultados muy rápido y se necesitan ajustar pocos parámetros, por lo que resulta atractivo para muchas áreas de aplicación e investigación tales como la optimización. Con el uso de este algoritmo, se logra realizar una

simulación que permita cumplir el objetivo de minimizar los costos de ventas de “Fruta Bar”.  
[26]

Brinda una metodología utilizando Simulación Monte Carlo y Lógica Difusa que permita pronosticar el comportamiento de la demanda, proponiendo una solución para el reabastecimiento de inventarios, dicho estudio solicitó opinión de expertos para comprobar la veracidad de los resultados obtenidos, sin embargo tal como lo describen en el artículo, la aplicación de Lógica Difusa a los inventarios da una aproximación de la cantidad de inventario necesaria, este valor no será el más óptimo pero si el más probable. Dicho esto, se evidencia las ventajas que ofrece el algoritmo PSO, que se caracteriza por formar parte del grupo de métodos metaheurísticos utilizados para optimización, por ende, los resultados obtenidos con su aplicación contarán con mayor precisión. [26]

**Tabla 1.** Ventajas y desventajas de redes neuronales

Método	Ventajas	Desventajas
<b>Red feedforward</b>	Son los métodos más rápidos para ejecutar. Son utilizados como los aproximadores universales de función. [30]	No se puede desarrollar un algoritmo confiable y suficientemente rápido por lo que se vuelve muy lento para entrenar. [30]
<b>Red neuronal recurrente</b>	Con función de memoria, Puede procesar y predecir datos de series de tiempo. LSTM evita el problema de la dependencia a largo plazo mediante un diseño deliberado.[31]	El resultado del siguiente momento a menudo debe combinarse con la influencia de muchos valores de tiempo anteriores al procesar el lenguaje natural.[31]
<b>Red convolucional</b>	Cada neurona ya no está conectada a todas las neuronas de la capa anterior Reconocimiento de imagen Aprendizaje profundo Un grupo de conexiones puede compartir el mismo peso.[32] Este método es muy directo	Demasiados parámetros Sin uso de información de ubicación entre píxeles Límite de capa de red.[32] La simulación más compleja resulta ser muy extensa, debido al número de variables a analizar.
<b>Monte Carlo</b>	Un ordenador es el elemento principal para realizar estos cálculos, lo que nos permite una gran exactitud. Mientras más compleja es la simulación, los resultados obtenidos son más aproximados a los hechos reales. Esta simulación solo nos permite simular sin interferir los resultados reales. [33]	No nos genera soluciones globales, ya que solo es para un cierto número de eventos. No brinda una decisión a tomar, sino más bien nos ayuda a resolver el comportamiento mediante las aproximaciones. Cada simulación arroja datos diferentes, ya que son valores aleatorios. [33]

## 2.5 Lenguaje de programación

Los datos registrados por la estación meteorológica han sido recolectados durante periodos de dos años los mismos que han sido almacenados, obteniendo una gran cantidad de información, los cuales al ser procesados por diferentes herramientas informáticas se tiene problemas de saturación del procesamiento, se buscó alternativas para procesar grandes cantidades de información se encuentran licenciadas como Matlab y también software libre como Python, R studio, por tal motivo para el análisis se ha optado con la utilización del software libre de Python, al no tener costo de adquisición se ha descargado desde la página oficial la distribución de Anaconda que contiene una colección de paquetes de código abierto para análisis de datos, la aplicación a utilizar es Jupyter Notebook disponibles para los sistemas operativos Windows, MacOS y Linux, para no tener incompatibilidades solo es necesario instalar la versión actualizada en Python, una alternativa si el equipo a utilizar no tiene buenas características para el procesamiento de datos se puede utilizar el intérprete Google Colab que es procesamiento en la nube basado en Jupyter Notebooks, el código de programación se inicia en el servidor del equipo mediante el navegador de preferencia obteniendo los mismos resultados de procesamiento de manera virtual sin saturar su procesamiento por la gran cantidad de datos.

### 2.5.1 Recolección de datos

Para la ejecución del proyecto se utilizó la estación meteorológica, consta de sensores de velocidad, temperatura, humedad, presión barométrica, los sensores y la precisión está direccionado al mercado semiprofesional, este sistema combinado tipo RM Young de forma de avión con una hélice de eje horizontal y una veleta que direcciona la hélice al viento como se observa en la figura 8

El observador meteorológico siempre busca que los datos que registra su estación sean lo máximo fiables y representativos posibles. Para eso, la Organización Meteorológica Mundial (OMM) ha establecido una normativa universal a seguir para la instalación de los

observatorios meteorológicos, tanto sean manuales cómo analógicos. Estas normas están recopiladas en este documento oficial.[34]

Tiene que estar situado en un terreno plano, de 25 x 25m o 10m x 7m), libre de obstáculos y sin pendientes. El suelo tiene que estar, preferentemente, cubierto de vegetación típica de la zona o césped de menos de 25cm de altura. El instrumental tiene que estar alejado de cualquier fuente de calor natural o artificial (edificios, paredes, hormigón, etc.) y también a más de 10m de una extensión de agua. [34]

Es muy importante mantenerla alejada de fuentes de calor cómo chimeneas, muros y paredes. Para la medición correcta del viento, la estación debe estar en un lugar alto y despejado, recomendable a unos 10m de altura. Ubicación libre de obstáculos, a 10m de la superficie del suelo o bien a  $10 + 2m$  por encima del obstáculo más cercano, el cual tiene 2m de altura. [35]



**Figura. 11.** Estación meteorológica instalada en el Instituto Superior Tecnológico Cotopaxi

## **2.5.2 Depuración de la data con la lectura de datos**

Los datos obtenidos por la estación meteorológica en formato csv. se procede con la revisión de la data se observa que existen valores nulos y vacíos los mismos que deben ser revisados y graficados para detectar sus errores, cabe mencionar que los datos que se encuentran con parámetros que influyen al momento de ser procesados dando como resultados valores de

predicción no cercanos a los deseados, para lo cual es indispensable realizar como primer paso el tratamiento de la información.

### 2.5.3 Coeficientes de correlación lineal

Los coeficientes de correlación lineal son estadísticos que cuantifican la asociación lineal entre dos variables numéricas. Existen diferentes tipos, de entre los que destacan permitiendo seleccionar los valores a ser influenciados dentro de la velocidad del viento, el valor está comprendido en el rango  $[+1, -1]$ . Siendo +1 una correlación positiva perfecta y -1 una correlación negativa perfecta, se emplean como medida de la fuerza de asociación entre dos variables (tamaño del efecto):

- 0: asociación nula.
- 0.1: asociación pequeña.
- 0.3: asociación mediana.
- 0.5: asociación moderada.
- 0.7: asociación alta.
- 0.9: asociación muy alta

Como métodos estadísticos que poseen una caja negra están los análisis de series temporales. una ventaja de estos métodos es la capacidad de modelar en una sola etapa la predicción meteoenérgica que se produce en la etapa de adaptación/entrenamiento. [25]

Con la carga de datos, se considera las variables más relevantes en la cual el viento vendrá en sus unidades de medida  $df = df[["Windspeed", "Baro Avg", "Hum Avg", "Temp Avg"]]$

Windspeed: Velocidad del viento [km/h]

Baro Avg: Presión Barométrica es la presión atmosférica que está relacionada con la (temperatura atmosférica, humedad y vientos) su unidad de medida es [mbar]

Hum avg: Humedad relativa promedio es la cantidad de vapor de agua presente en el aire la unidad de medida se define como el porcentaje de la cantidad de agua presente en 1 m<sup>3</sup> de aire en una temperatura dada [%]

Temp Avg: Temperatura promedio determina las condiciones climáticas que se presenta en el medio determina las sensaciones de calor y frío, es la cantidad de energía calorífica que posee el aire en un momento determinado, su unidad de medida [°C]

Son los factores más relevantes para determinar o predecir la velocidad del viento se ejecutó los comandos, colocando de forma ordenada para presentar los parámetros mencionados además se configuro para que se presente en formato de horas, con esto se genera un formato que se pueda entender el algoritmo y visualice la matriz de forma ordenada obteniendo la siguiente.

Para predecir la velocidad del viento y con ella determinar la velocidad del viento se utilizara el análisis de series temporales usando Red LSTM es un tipo de Red Neuronal Recurrente que permite analizar secuencias o datos histórico y que además tiene una memoria de largo plazo, para el análisis se tomara los parámetros de la velocidad del viento, temperatura, presión barométrica y humedad la cual determinara a qué hora del día ocurren las incrementos y disminuciones velocidad, determinando la predicción de la hora en que se obtendrá la mayor y menor cantidad de energía la cual será útil para determinar si es o no factible instalar un aerogenerador.

La data recolectada durante los dos años por la estación meteorología permitirá entrenar el algoritmo para predecir la velocidad de viento por cada hora para la cual se debe seguir una serie de pasos consecutivos con la cual se realizará el modelamiento del algoritmo la misma que se comparará el valor precedido con los valores medidos para determinar el margen de error que se tenga en el algoritmo dicho valor influenciara para determinar el potencial eólico que se tenga en el lapso de un día completo.

#### **2.5.4 Desarrollo del algoritmo y utilización de librerías.**

Para que la programación comprenda las actividades que debe realizar con la data que se va a ingresar se utilizara varias librerías con las cuales interpretara los datos y procesara para obtener resultados acordes que se deseen obtener.

Aplicando estos principios se implementa el algoritmo de vecinos próximos basado en elementos utilizando el lenguaje de programación Python y la librería es un algoritmo de

aprendizaje automático de tipo supervisado. Se utiliza para clasificar valores buscando puntos de datos. [36]

- Cadenas
- Estructura de datos
- Funciones numéricas y matemáticas
- Compresión de datos
- Formatos de archivo
- Criptografía
- Servicios de los Sistemas Operativos
- Comunicación entre Procesos
- Manejo de datos de Internet
- Servicios multimedia
- Manejo de excepciones

### **NumPy as np**

NumPy significa “Numerical Python”, especializada en el cálculo numérico y el análisis de datos, especialmente para un gran volumen de datos, con esta librería se convertirá los datos obtenidos en matrices proporciona potentes estructuras de datos ya sea en filas o en columnas que se necesite implementando matrices y matrices multidimensionales. `import numpy as np`

### **Matplotlib.pyplot as plt**

Con esta librería se comprenderá de mejor manera el análisis de datos obtenidos y se interpretará por medio graficas a presentar en función de la data de predicción teniendo una mejor visualización de los resultados. `import matplotlib.pyplot as plt`

### **Tensorflow as tf**

Tensor Flow es una librería de código abierto para cálculo numérico, usando como forma de programación somos capaces, entre otras operaciones, de construir y realizar entrenamiento de redes neuronales en base a la data que se ingrese para detectar correlaciones y descifrar

patrones, análogos al aprendizaje y razonamiento usados por los humanos.  
`import tensorflow as tf`

### **Datetime as dt**

Es la librería con la cual el algoritmo pueda leer en formato de fechas (año, mes, día, hora, minutos, segundos y microsegundos) y no de números de forma normal esto permite operaciones aritméticas con fechas y horas, su principal objetivo es poder extraer campos de forma eficiente para su posterior manipulación o formateo. `import datetime as dt`

### **Tensorflow.keras as keras**

Es una biblioteca de redes neuronales artificiales de código abierto diseñado para ir construyendo por bloques la arquitectura de cada red neuronal que se necesite generar.  
`import tensorflow.keras as keras`

### **Sklearn.preprocessing.MinMaxScaler**

Con esto proporciona varias funciones y convertidores que se usan comúnmente en el preprocesamiento de datos, que se utilizan para cambiar el vector de características original a una forma más adecuada para los modelos de aprendizaje automático este estimador escala y traduce cada característica individualmente de manera que se encuentre en el rango dado en el conjunto de entrenamiento de cero y uno el cual se aplicará:

```
from sklearn.preprocessing import MinMaxScaler
```

## **2.5.5 Datos de la velocidad del viento de la ciudad de Latacunga**

Para realizar la comparación de los datos medidos se tomó la velocidad del viento [km/h] como referencia los datos medidos de una estación meteorológica de Latacunga de la página web de <https://www.woespana.es/weather/maps/city> se observa que la velocidad del viento varía en función del tiempo, además se grafica los valores promedios durante el mes de enero del 2020 como se observan en las imágenes

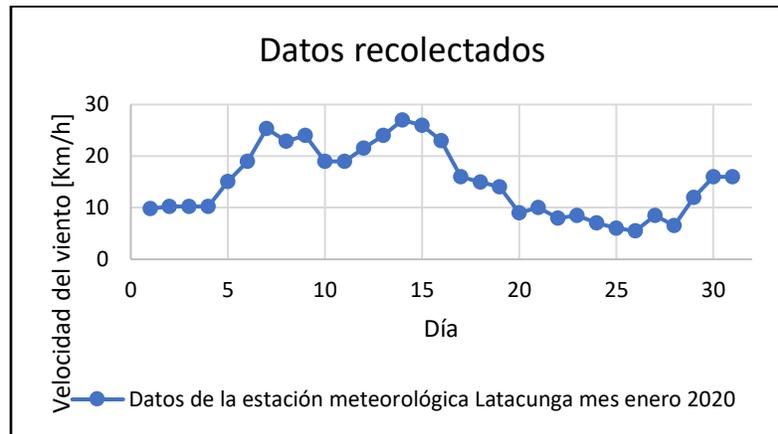


Figura. 12. Velocidad promedio por día del mes enero 2020 de la ciudad de Latacunga. [37]

### 2.5.6 Código de línea para cargar la base de datos

A partir de la matriz de datos obtenida se pueden realizar varios análisis de datos uno de los más empleados para describir el comportamiento de las variables es el análisis estadístico evaluación del potencial eólico y predicción de la velocidad de viento con minería de datos.[25]

El modelamiento del algoritmo se lo realiza la función de la base de datos medidos en los 2 años (2019 - 2020) para lo cual se utilizará diferentes librerías para la programación que ayudara a la hora de implementar el código del proyecto, con las que se beneficiara para mejorar el tiempo de desarrollo en este caso se puede utilizar el modelo LSTM el cual sirve para aprender y memorizar, la librería para importar datos se utilizara pandas que permite leer archivos he importar tablas DataFrame en la cual se ingresara la variable de la velocidad del viento que estará recolectada en varios intervalos de tiempos la cual importara y analizará con el comando `pd.read_csv` (“ ”) de una manera fácil y flexible que hace la manipulación de datos para el lenguaje de programación Python, dicho documento se lo puede cargar en formato csv para importar el archivo a Python Notebooks realizando el trabajo, los documentos contendrán la medición de cada mes de los 2 años mencionados, la línea de código es la siguiente.

```
data=pd.read_csv('nombre del archivo').
```

A partir de la matriz de datos obtenida se pueden realizar varios análisis de datos uno de los más empleados para describir el comportamiento de las variables es el análisis estadístico evaluación del potencial eólico y predicción de la velocidad de viento con minería de datos.[25]

### **2.5.7 Estandarización y escalado de variables numéricas o data**

Cuando los predictores son numéricos, la escala en la que se miden, así como la magnitud de su varianza pueden influir en gran medida en el modelo. Muchos algoritmos de Machine Learning (SVM, redes neuronales, lasso...) son sensibles a esto, de forma que, si no se igualan de alguna forma los predictores, aquellos que se midan en una escala mayor o que tengan más varianza dominarán el modelo, aunque no sean los que más relación tienen con la variable respuesta. Existen principalmente 2 estrategias para evitarlo:[38]

Se utilizara la Normalización (StandardScaler) con la cual a los datos ingresados se pondrán en una escala esto implica transformar o convertir el conjunto de datos en una distribución normal debido que una cantidad no es igual a la otra, con este parámetro se asegurara que, aunque algunas características son grandes no se usarán como predictor principal estableciendo dentro del rango de [0:1], para lo cual se utilizara la librería from sklearn.preprocessing import StandardScaler con esto se realiza el procesamiento de la normalización de datos aplicando el siguiente código quedando dentro del rango.

### **2.5.8 Dividir en data de train y test**

Evaluar la capacidad predictiva de un modelo consiste en comprobar cómo de próximas son sus predicciones a los verdaderos valores de la variable respuesta. Para poder cuantificarlo de forma correcta, se necesita disponer de un conjunto de observaciones, de las que se conozca la variable respuesta, pero que el modelo no haya "visto", es decir, que no hayan participado en su ajuste. Con esta finalidad, se dividen los datos disponibles en un conjunto de entrenamiento y un conjunto de test. El tamaño adecuado de las particiones depende en gran medida de la cantidad de datos disponibles y la seguridad que se necesite en la

estimación del error, 80% - 20% suele dar buenos resultados. El reparto debe hacerse de forma aleatoria o aleatoria-estratificada.[38]

Para entrenar el algoritmo de la data total se dividirá en un conjunto de datos inicial en 2 partes: conjunto de entrenamiento (train) y conjunto de Pruebas (test). Por lo cual se dividirá en “80% para el entrenamiento y 20% para realizar el test teniendo un total del 100% de la data general, se utilizara la librería que divide los datos de entrenamiento con los de test: `from sklearn.model_selection import train_test_split` que me permite dividir un dataset en dos bloques, típicamente bloques destinados al entrenamiento y validación del modelo

```
X_train = [80%] # Variables de entrenamiento
```

```
y_train = [20%] # Variables de validación
```

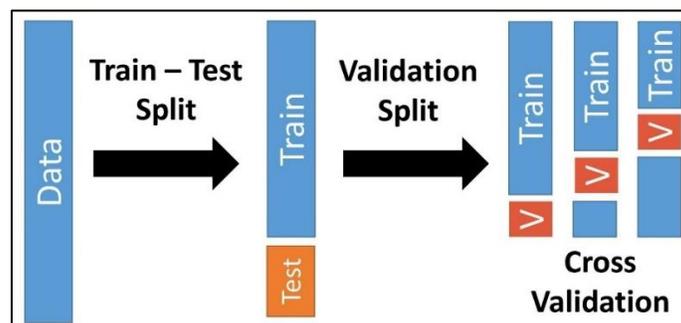


Figura. 13. Datos de entrenamiento, validación y prueba. [38]

### 2.5.9 Estructura de la Red Neuronal LSTM

Una red neuronal recurrente LSTM tiene como objetivo aprender dependencias a largo plazo; es decir, aprender las dependencias de valores futuros de una secuencia en función a los valores anteriores. En [20] introducen el concepto de una red LSTM, la cual busca eliminar el problema que nace al tratar de integrar t-n valores para predecir el valor t + 1 de una secuencia. Como toda red neuronal recurrente, una LSTM tiene la forma de una cadena con unidades (o celdas) repetidas, pero en vez de tener una estructura simple dentro de cada unidad, en una LSTM se tiene 4 componentes que interactúan entre sí (Figura 14). [39]

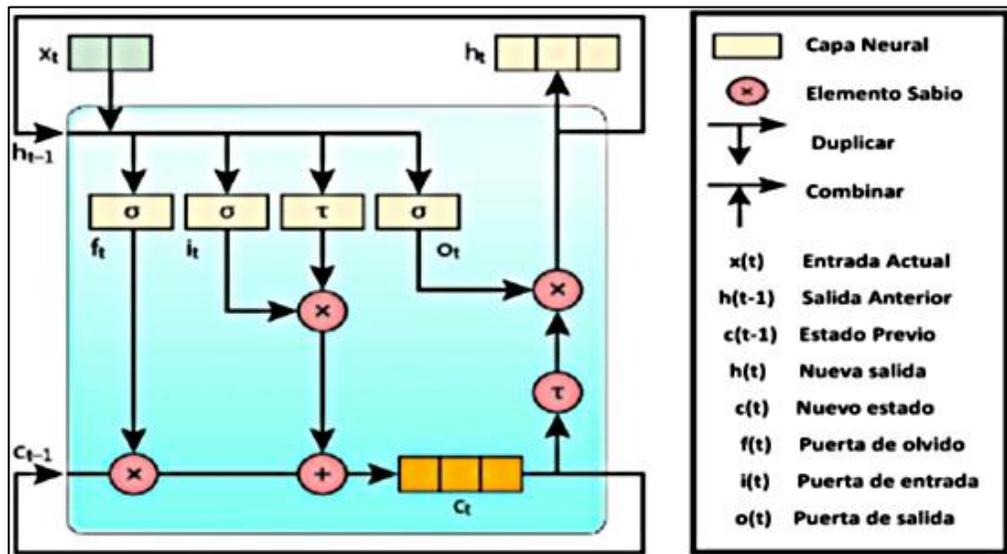


Figura. 14. Estructura de una red LSTM. [39]

La unidad de memoria LSTM contiene tres puertas que controlan el modo en que la información fluye dentro o fuera de la unidad.

- Puerta de entrada controla cuando la información nueva puede entrar en la memoria.
- Puerta del olvido controla cuando se olvida una parte de la información, lo que permite a la celda discriminar entre datos importantes y superfluos, dejando así sitio para nuevos datos.
- Puerta de salida controla cuando se utiliza en el resultado de los recuerdos almacenados en la celda.

El *core* de una red LSTM se basa en el estado de una celda. La red LSTM modifica el estado de la celda al añadir o quitar información del estado de la celda anterior  $c_{t-1}$  para producir un nuevo estado  $c_t$ . El primer paso en una LSTM consiste en decidir qué información va a permanecer en el estado de la celda. Este procedimiento se realiza con una capa de activación sigmoideal, la cual toma la salida  $h_{t-1}$  y la entrada  $x_t$ , y representa con valores en el intervalo  $[0,1]$  la necesidad de información (un valor más cercano a uno es más necesario que uno cercano a cero) para cada valor del estado  $c_{t-1}$ . Matemáticamente, este procedimiento se detalla en la ecuación (1), donde  $W_f$  representa la matriz de pesos y  $b_f$  el *bias* para el “forget gate”. [39]

$$f_t = \sigma(W_f * [X_t, h_t - 1] + b_f)$$

El siguiente paso es decidir qué información retener en el estado de la celda. Para esto se utilizan dos capas, una sigmoide con la que se decide qué valores se actualizarán y una tangente hiperbólica que creará un nuevo vector de valores candidatos a ser añadidos al estado de la celda (ecuaciones (2) y (3)). Estos dos valores serán combinados más adelante para la actualización del estado de la celda. [39]

$$i_t = \sigma(W_i * [X_t, h_t - 1] + b_i)$$

$$\check{C}_t = \tanh(W_c * [X_t, h_t - 1] + b_c)$$

Al terminar estos cálculos, el nuevo estado  $C_t$  se obtiene al multiplicar  $C_{t-1}$  por  $f_t$ , olvidando la información innecesaria y añadiendo la nueva información al multiplicar  $i_t$  por la matriz de candidatos  $\alpha$  (ecuación (4)). Estos candidatos son escalados por  $i_t$ , ya que este representa cuánto se decidió actualizar cada valor del estado anterior. [39]

$$(4) C_t = f_t * C_{t-1} + i_t * \check{C}_t$$

Finalmente, se necesita calcular la salida. Esta se debe basar en el estado actual, pero no toda la información es parte de la salida. Para realizar este filtrado, se aplica una capa sigmoide para decidir qué valores del estado actual formarán parte de la salida. Luego, se aplica una tangente hiperbólica (para tener valores entre  $-1$  y  $1$ ) y se multiplica estos valores por la activación de la capa sigmoide para solo tener como output los valores seleccionados [39]

$$O_t = \sigma(W_o * [X_t, h_t - 1] + b_o)$$

$$h_t = O_t * \tanh(h_t)$$

$$(1) f_t = \sigma(W_f X_t + U_f h_{t-1} + b_f)$$

$$(2) i_t = \sigma(W_i X_t + U_i h_{t-1} + b_i)$$

$$(3) o_t = \sigma(W_o X_t + U_o h_{t-1} + b_o)$$

$$(4) c_t = f_t * c_{t-1} + i_t * \tau(W_c X_t + U_c h_{t-1} + b_c)$$

$$(5) h_t = o_t * \tau(c_t)$$

Donde:

$n$ : tamaño de entrada  $m$ : tamaño del estado de la celda y salida

$x_t$ : vector de entrada, tiempo  $t$ , tamaño  $n \times 1$

$f_t$ : vector de puerta de olvido, tamaño  $m \times 1$

$i_t$ : vector de puerta de entrada, tamaño  $m \times 1$

*ot*: vector de puerta de salida, tamaño  $mx1$

*ht*: vector de salida, tamaño  $mx1$

*Ct*: vector de celda de estado, tamaño  $mx1$

*Wf, Wi, Wo, Wc*: matrices de peso de la puerta de entrada, tamaño  $mxn$

*Uf, Ui, Uo, Uc*: matrices de peso de la puerta de salida, tamaño  $mxm$

*bf, bi, bo, bc*: vectores de sesgo, tamaño  $mx1$

$\sigma$ : función de activación logística sigmoide

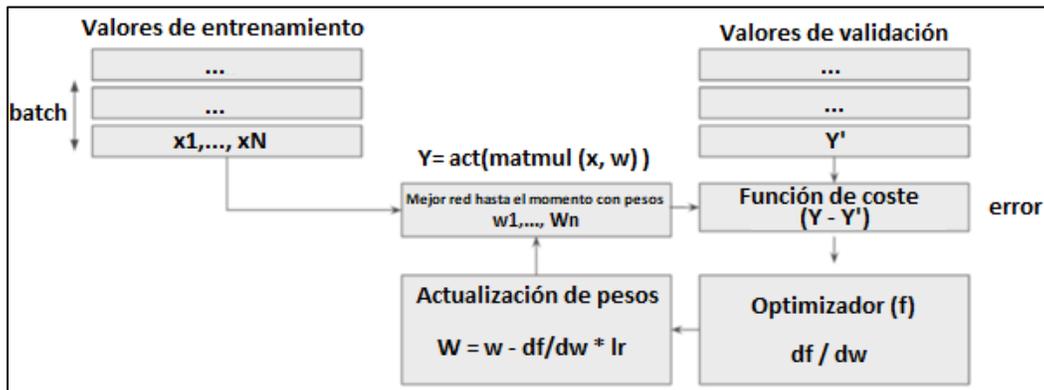
$\tau$ : función de activación tangente hiperbólica

Las ecuaciones (1), (2) y (3) definen tres puertas: una puerta de olvido, una de entrada y una de salida. Cada puerta es un vector de valores entre 0,0 y 1,0, que se usan para determinar cuánta información se olvida (o, equivalentemente, se recuerda) en cada ciclo de entrada-salida. La ecuación (4) calcula el nuevo estado de celda, mientras que la ecuación (5) calcula el nuevo resultado.

La función sigmoide es precisamente que da a la compuerta el comportamiento de válvula, funcionando como número binario de 0 y 1 que permite pasar o anular los valores en la entrada [40] es el comportamiento que permite aplicar la memoria a largo plazo.

### **2.5.10 Optimizadores del algoritmo**

En esencia, el objetivo del entrenamiento de redes neuronales es minimizar la función de coste encontrando los pesos adecuados para las aristas de la red (asegurando, eso sí, una buena generalización). El descubrimiento de estos pesos se lleva a cabo mediante un algoritmo numérico llamado *backpropagation* que puede resumirse así:



**Figura. 15.** Algoritmo de *backpropagation* para entrenamiento de redes neuronales. [41]

El elemento en cuestión de este estudio, el optimizador, es el encargado de generar pesos cada vez mejores: su importancia es crucial. Su funcionamiento esencial se basa en calcular el gradiente de la función de coste (derivada parcial) por cada peso (parámetro/dimensión) de la red. Como queremos minimizar el error, modificaremos cada peso en la dirección (negativa) del gradiente: [41], Cálculo del peso en el paso t+1

$$W_{t+1} = W_t - \frac{df(\text{coste})}{dw * lr}$$

De cara a agilizar la convergencia de la función de coste hacia su mínimo, multiplicamos el vector de gradiente por un factor llamado factor de entrenamiento (lr). El conjunto de métodos iterativos de reducción de la función de error (búsqueda de un mínimo local), son conocidos como los métodos de optimización basados en el gradiente descendente. [41]

### **Adadelta**

Es una variación de AdaGrad en la que en vez de calcular el escalado del factor de entrenamiento de cada dimensión teniendo en cuenta el gradiente acumulado desde el principio de la ejecución, se restringe a una ventana de tamaño fijo de los últimos n gradientes.[41]

### **RMSprop (Root Mean Square Propagation)**

Mantiene un factor de entrenamiento diferente para cada dimensión, pero en este caso el escalado del factor de entrenamiento se realiza dividiéndolo por la media del declive exponencial del cuadrado de los gradientes [41]

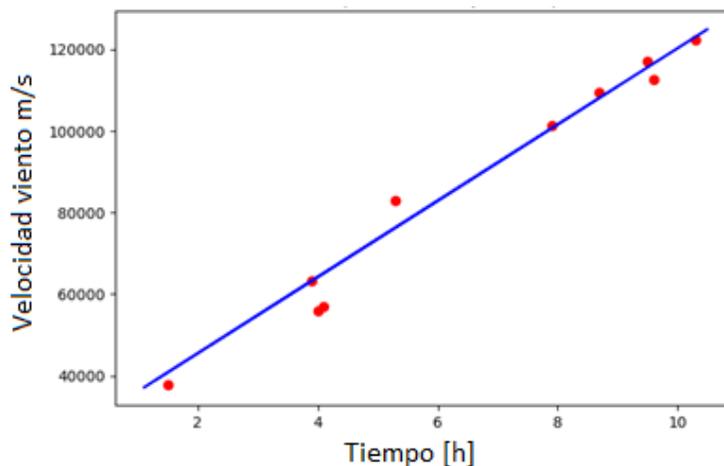
### **Adam (Adaptive moment estimation)**

El algoritmo Adam combina las bondades de AdaGrad y RMSProp. Se mantiene un factor de entrenamiento por parámetro y además de calcular RMSProp, cada factor de entrenamiento también se ve afectado por la media del momentum del gradiente. [41]

#### **2.5.11 Validación**

La finalidad última de un modelo es predecir la variable respuesta en observaciones futuras o en observaciones que el modelo no ha "visto" antes. El error mostrado por defecto tras entrenar un modelo suele ser el error de entrenamiento, el error que comete el modelo al predecir las observaciones que ya ha "visto". Si bien estos errores son útiles para entender cómo está aprendiendo el modelo (estudio de residuos), no es una estimación realista de cómo se comporta el modelo ante nuevas observaciones (el error de entrenamiento suele ser demasiado optimista). Para conseguir una estimación más certera, y antes de recurrir al conjunto de test, se pueden emplear estrategias. [38]

Mediante los métodos de validación se consiguen buenas estimaciones del error que tiene un modelo al predecir los sucesos dependiendo la data a ser analizada, la mejor forma de evaluar un modelo al final es prediciendo un conjunto test, es decir, un conjunto de observaciones que mantienen al margen del proceso de entrenamiento y optimización con esto podemos verificar si la predicción se asemeja a la realidad. [38]



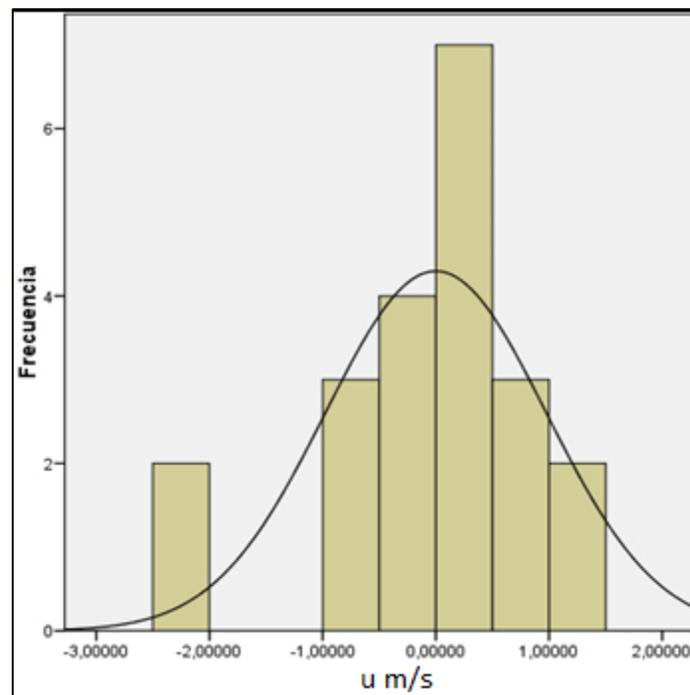
**Figura. 16.** Datos medidos vs datos reales. [38]

### 2.5.12 Grafica de resultados

Se presentará los resultados en la que se observe los datos de predicción y los datos de comparación para observar el comportamiento de forma visual para lo cual se empleara las siguientes líneas de código en las que se estimara sus dimensiones de cuadrícula y los colores con los que se identificaran las curvas.

```
plt.plot(data_test.values[60:321], color='orange', label='Velocidad del Viento (Real)')
```

```
plt.plot(prediccion[:261], color='blue', label='Velocidad del Viento (Predicción)')
```



**Figura. 17.** Datos Predichos en función de los valores reales. [38]

### 2.5.13 Análisis de comportamiento de los errores

Para la evaluación de resultados se realizará la predicción promedio de cada mes con los datos no ingresados en el algoritmo comparando con los valores medidos en el año, en el cual los resultados medidos se compararán con los datos predichos por el algoritmo con este parámetro se determinará el porcentaje de error que se presenta, cabe mencionar que al ser un método predictivo el algoritmo es entrenado y determina según los datos que se ingrese.

El error porcentual absoluto medio (MAPE) expresa la exactitud como un porcentaje del error. Debido a que el MAPE es un porcentaje, puede ser más fácil de entender que otros estadísticos de medición de exactitud. Por ejemplo, si el MAPE es 5, en promedio, el pronóstico está errado en un 5%. [42]

Sin embargo, es posible que algunas veces observe un valor de MAPE muy elevado, aunque el modelo parezca ajustarse a los datos adecuadamente. Examine la gráfica para ver si los valores de los datos se aproximan a 0. Debido a que MAPE divide el error absoluto entre los datos reales, los valores que se aproximan a 0 pueden aumentar significativamente el MAPE.

#### **2.5.14 Distribución de la función densidad de probabilidad de Weibull.**

Para realizar la distribución de frecuencias de la velocidad del viento se aplicó la regla de *Sturges* (González, 2005), para lo cual es necesario conocer el rango de datos, el número y el ancho del intervalo; esta distribución se observa en la tabla 4. En esta se encuentran los límites de velocidad del viento inferior ( $L_i$ ), superior ( $L_s$ ), la frecuencia relativa ( $fr$ ) y su sumatoria acumulada ( $Fr$ ), la frecuencia absoluta ( $fa$ ) y su sumatoria acumulada ( $Fa$ ). [25]

$$Fi = \frac{Ni}{N}$$

Siendo ( $X_1, X_2, \dots, X_n$ ) el conjunto de datos y  $N_i$  el total de datos igual o menor a  $X_i$   
La frecuencia relativa es la relación de Frecuencia absoluta  $F_i$  sobre Numero de datos totales.  
[25]

$$Fr = \frac{Fi}{N}$$

Las variaciones que presenta el viento dependen de las condiciones climatológicas de la zona, por esto es necesario el uso de métodos estadísticos para evaluar el aprovechamiento del potencial eólico en la zona. Dada la variabilidad y asimetría del histograma representado en la figura 2, se puede normalizar la función de la probabilidad de distribución de la velocidad

del viento a través del análisis de Weibull. Los parámetros para dicho análisis se obtuvieron mediante el análisis de regresión lineal entre los valores de la variable y su probabilidad acumulativa, utilizando la transformada logarítmica y el método de mínimos cuadrados.

La función de distribución de acumulación se presenta en la ecuación.

$$W(v) = e^{-\left(\frac{v^\alpha}{\beta}\right)}$$

En donde:

W(v): función de distribución de probabilidad Weibull para la velocidad del viento.

v: velocidad del viento.

$\alpha$ : parámetro de forma.

$\beta$ : parámetro de escala.

Para la ecuación, la pendiente de la recta representa el parámetro de forma ( ). En esta Y se representa por la ecuación y X por la ecuación

Con una transformación de doble logaritmo la ecuación

$$\text{Ln} \left( \text{Ln} \left( \frac{1}{1 - W(v)} \right) \right) = \alpha \text{Ln} (v) - \alpha \text{Ln} (\beta)$$

De esta manera se representa la ecuación 13. en una ecuación lineal descrita en la ecuación

$$y = \alpha x - b$$

$$y = \text{Ln} \left( \text{Ln} \left( \frac{1}{1 - W(v)} \right) \right)$$

$$x = \text{Ln} (v)$$

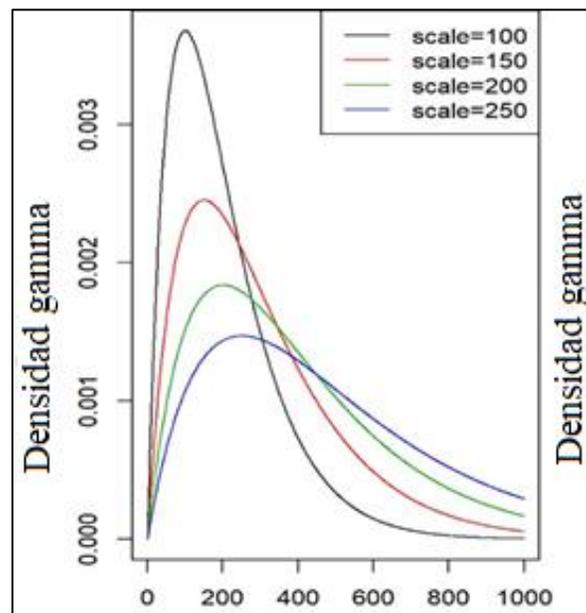
Para predecir el comportamiento de la velocidad del viento en un determinado tiempo se necesita modelar la distribución de la velocidad por medio de expresiones matemáticas. Estudios realizados han comprobado que la velocidad del viento se aproxima a la distribución de la función densidad de la probabilidad de Weibull.[43]

$$\rho(v) = \frac{k}{A} * \left(\frac{V}{A}\right)^{k-1} * e^{-\left(\frac{V}{A}\right)^k}$$

Donde  $\rho(V)$  es la función densidad de probabilidad de Weibull,  $V$  es la velocidad del viento,  $K$  es el factor de forma y  $A$  es el factor de escala.

Los parámetros  $A$  y  $K$  definen a la función Weibull, donde  $A$  toma siempre valores próximos a la velocidad promedio, mientras que  $K$  es un factor que caracteriza la asimetría de la función de probabilidad.

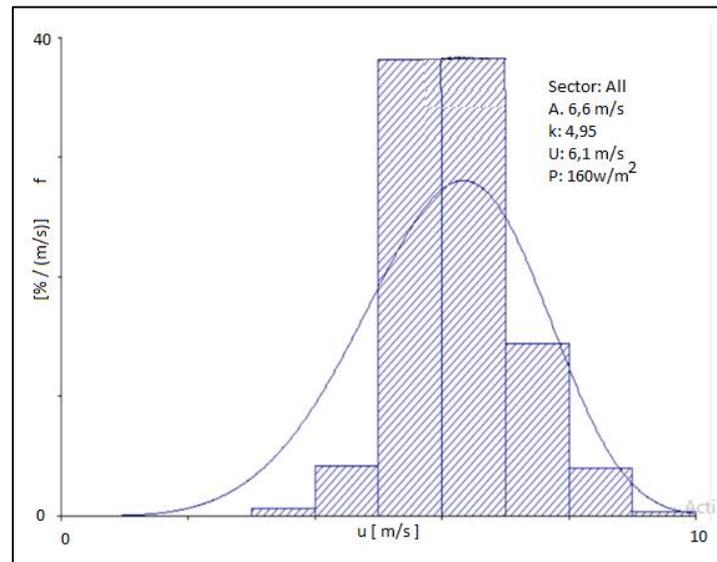
La distribución, es función de la velocidad de viento y queda definida por los dos parámetros y  $k$ : Factor de forma ( $k$ ), adimensional: Caracteriza la forma de la curva, es decir, la asimetría.[44]



**Figura. 18.** Variación de una distribución de Weibull al variar el factor de forma. [44]

### 2.5.15 Exponiendo y procesando los datos obtenidos

Ya recolectados los 300 datos de la velocidad, temperatura y hora del viento, los cuales están anexados al final del presente informe, fueron evaluados mediante el software de procesamiento de datos eólicos llamado WAsP, el cual se encargó de evaluar y generar su propio pronóstico acerca de la velocidad del viento. Los datos obtenidos fueron los siguientes.[45]



**Figura. 19.** Distribución de Weibull con la estimación de la velocidad de viento promedio del distrito de Salaverry. [45]

### 2.5.16 Rosa de los vientos

Es importante conocer de dónde provienen los vientos más frecuentes y más fuertes en el área donde se quiere instalar. En la medida de lo posible, esa dirección debe de estar libre de obstrucciones. Para conocer estos datos suele recurrirse a estudios realizados con un anemómetro y una veleta que miden la dirección y la velocidad del viento en un periodo de tiempo determinado. Así se configuran lo que se llaman “rosas de vientos.”[46]

Los datos los podemos representar en dieciséis direcciones, a ocho de ellas se les llama primarias, a las otras ocho, secundarias.[47] como se presenta en la siguiente figura 20

<b>Primarias</b>	<b>Secundarias</b>
N (norte)	NNE (nor noreste)
S (sur)	ENE (este noreste)
E (este)	ESE (este sudeste)
O (oeste)	SSE (sud sudeste)
NE (nordeste)	SSO (sud sudoeste)
NO (noroeste)	OSO (oeste sudoeste)
SE (sudeste)	ONO (oeste noroeste)
SO (sudoeste)	NNO (nor noroeste)

**Figura. 20.** Orientación por medio de la rosa de los vientos. [47]

### 2.5.17 Aerogenerador

Los nuevos diseños de la serie PRO se han realizado atendiendo estrictamente a la Norma IEC 61400-2, usando técnicas innovadoras como la Dinámica Computacional de Fluidos. Estas técnicas propias de la industria aeroespacial incrementan la eficiencia en generación consiguiendo una disminución del ruido. Esto junto a la gran robustez del aerogenerador lo convierten en una máquina profesional y de alto rendimiento.[48]

**Tabla 2.** Tipos de aerogeneradores

Tipo de Turbina	Potencia nominal del Generador [W]	Clase de viento IEC	Rango de generación eficiente Viento [m/s]	
			Arranque	Parada
Enair 30Pro	3000	61400-2	2	60
Enair 70Pro	4000	61400-2	2	60
Enair 200 L	10140	61400-2	1.85	60

**Tabla 3.** Clasificación del aerogenerador por demanda energética

Tipo de Turbina	Demanda energética	Producción de energía	Viento promedio
Enair 30Pro	Para vivienda habitual, con viento bajo[48]	30kWh/día	8 y 12m/s
Enair 70Pro	Para cualquier vivienda habitual, gasolinera, granja o pequeño comercio, con muy bajo viento.[49]	80kWh/día	8 y 12m/s
Enair 200 L	Para pequeñas industrias tales como granjas, invernaderos, bodegas, etc.[50]	150-200kWh/día	9m/s

### 2.5.18 Potencial eólico suministrada por un aerogenerador de eje horizontal

La energía almacenada en el viento es muy elevada, pero esta no se aprovecha en su totalidad. Albert Betz demostró en 1919 que la extracción de la energía del viento hace que este reduzca su velocidad. Según el teorema de Betz, la energía máxima que podrá captarse del viento para que el rendimiento sea el máximo es el 59% de la energía cinética del viento; esto representa la primera pérdida del rendimiento final de la turbina (Villanueva y Marcelo, 2016). La potencia que se puede extraer del viento con un aerogenerador está determinada por la ecuación[25]

La determinación del potencial eólico se aplicará en función de la velocidad del viento pronosticada en lapsos de una hora determinando en base a la siguiente ecuación.

$$\frac{P}{A} = \frac{1}{2} \rho v^3$$

Donde:

$$\frac{P}{A} = \text{densidad de potencia} \left( \frac{W}{m^2} \right)$$

$$\rho = \text{densidad del aire} \left( \frac{Kg}{m^3} \right)$$

$$V = \text{velocidad del viento} \left( \frac{m}{s} \right)$$

La potencia suministrada por un aerogenerador de eje horizontal depende de la superficie barrida por la hélice, es decir, de la longitud de las palas y es independiente del número de las mismas. Si el área barrida transversalmente por el viento es circular, como ocurre en todas las máquinas eólicas de eje horizontal Siendo D el diámetro; por lo que la potencia disponible se observa en la ecuación.[25]

$$P = \frac{\pi}{8} * \phi * D^2 * V^3$$

La energía producida se determina en función de las horas útiles en las que la velocidad del viento es necesaria para poder generar la potencia nominal.

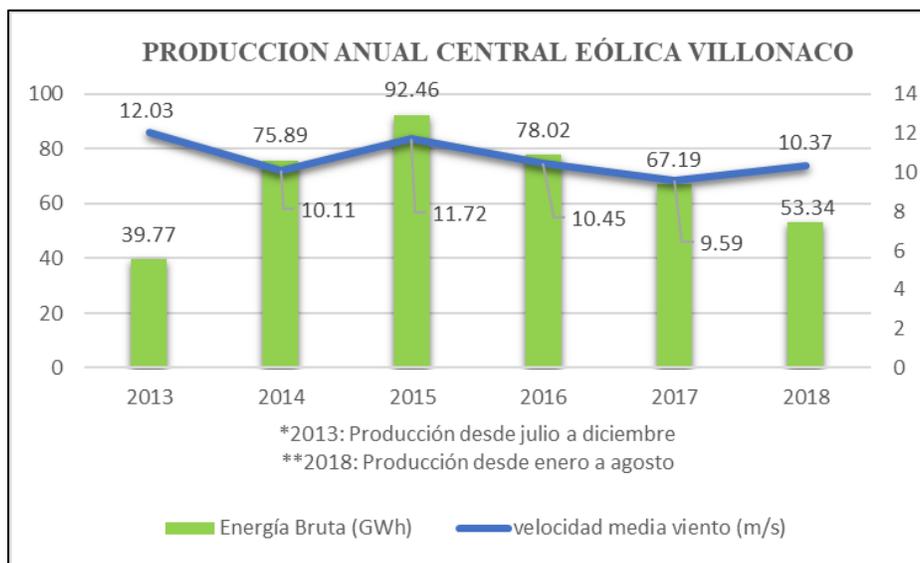


Figura. 21. Producción anual de energía [GWh], central eólica Villonaco

### 2.5.19 Factor de planta

El factor de planta representa el porcentaje de tiempo en el que la planta se encuentra generando energía.[51]

$$fp = \frac{MW * Tiempo. util}{MW * Tiempo. total}$$

## 2.6 Conclusiones Capítulo II

- El análisis de la selección de la técnica de machine Learning permitirá el modelamiento del algoritmo logrando determinar la predicción del viento que se obtendrá en dicho lugar, en base a los datos recolectados y procesados, los resultados obtenidos se compararan con datos que el algoritmo desconoce comparando los valores obtenidos y medidos de un mes con días promedio.
- Los valores recolectados para el entrenamiento del algoritmo deben ser de tiempo más prolongados para obtener un índice de error bajo, cabe mencionar que los datos recolectados son de dos años los cuales cumplen con lo requerido para poder realizar el algoritmo considerando que del 100% de la información el 80% se tomara para datos de entrenamiento y el 20% como datos de test
- En base a la distribución de Weibull se determinará con el histograma de frecuencias la velocidad promedio que se tiene en el año con ello se determinara el potencial eólico que se obtendrá en cada mes del año, el potencial que genere en el año permitirá determinar el factor de planta que se obtendrá con un aerogenerador de eje horizontal, en función al parámetro mencionado se podrá determinar si es o no viable la instalación de un aerogenerador en el sector de Lasso.

## CAPÍTULO III. APLICACIÓN DE LA PROPUESTA

### 3.1 Análisis de los resultados

#### 3.1.1 Selección de la red neuronal.

En la tabla 4 se observa las características de las redes neuronales por tal motivo para el objeto de estudio se seleccionará la red neuronal recurrente, permite trabajar con una serie de datos históricos, memorizando los datos más representativos que se obtengan, con ello se tendrá una solución óptima como resultado de la predicción de la velocidad del viento.

**Tabla 4.** Características de las redes neuronales

Método	Características
Red feedforward	Conexiones de la unidad no viajan en un bucle sino en una única ruta dirigida.
Red neuronal recurrente	Análisis secuencias, tiene un procedimiento de aprendizaje más eficiente y alcanza un nivel de estabilidad más rápido tanto en el entrenamiento como en su funcionamiento en la predicción
Red convolucional	Aprendizaje supervisado que procesa sus capas imitando al contexto visual del ojo humano para identificar distintas características en las entradas
Monte Carlo	Método probabilístico no siempre tiene porqué encontrar la solución óptima

#### 3.1.2 Selección del lenguaje de programación

Para el procesamiento de los datos se seleccionó Jupyter Notebook debido que no se necesita utilizar internet para ejecutar el programa, basta con instalar sus librerías directamente en la PC para que funcione en cualquier lugar a diferencia de su interprete Google Colab que necesita disponer de conexión a internet para poder ejecutar el programa, presentando un limitante de utilización, por tal motivo se seleccionó Jupyter Netebook en la tabla 5 se observa la descripción de algunos lenguajes de programación, en la que menciona que la

velocidad general de ejecución es la más rápida con el procesamiento de datos con el lenguaje de programación seleccionado.

**Tabla 5.** Descripción del lenguaje de programación

<b>Lenguaje de programación</b>	<b>Descripción</b>
R	Velocidad de desarrollo rápida (60% menos código que Python)
Python	La velocidad general es la más rápida.
MATLAB	Licencia alrededor de US \$ 1,000, paquete adicional requiere un pago adicional de US \$ 50 +

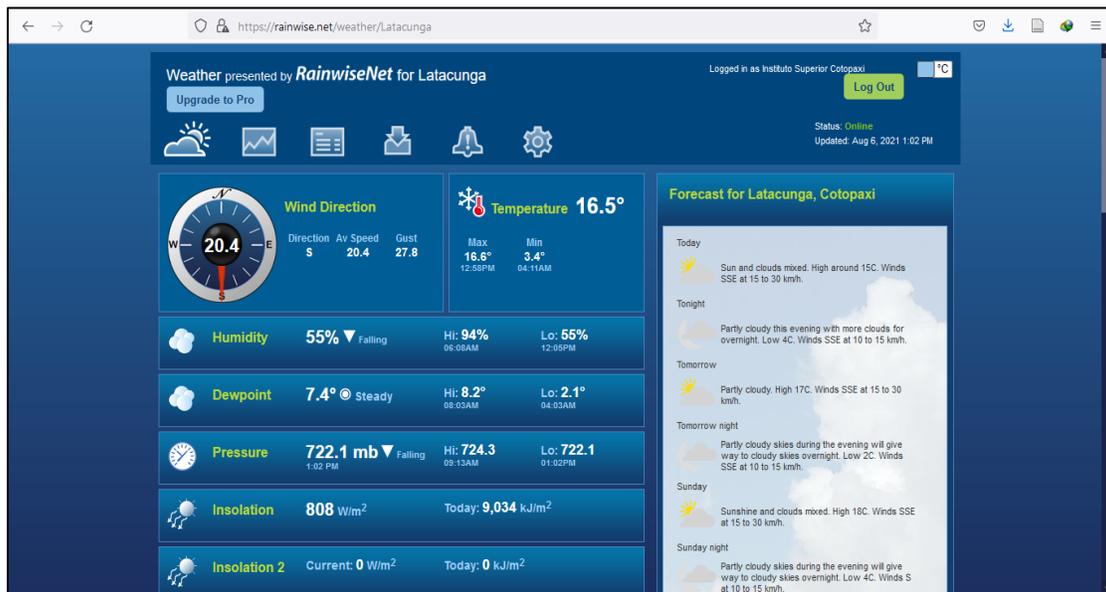
### 3.1.3 Recolección de datos

Para la recolección de datos en el sector de Lasso, se utilizó la estación meteorológica instalada a 10 metros de altura en la parte administrativa del Instituto Superior Tecnológico Cotopaxi, encontrándose libre de obstáculos y sin pendientes, el suelo está cubierto de césped de menos de 25 cm de altura, se encuentra alejado de cualquier fuente de calor natural o artificial de edificios, paredes y a más de 10m de una extensión de agua cumpliendo con los parámetros de instalación según la Organización Meteorológica Mundial (OMM)[34], la estación consta de sensores de velocidad del viento, temperatura, humedad, presión barométrica, los sensores y la precisión de las mismas está direccionado al mercado semiprofesional, está compuesto de Anemómetro con un Rango: 0.4 ... 67 m/s, Precisión: +/- 5% y Resolución: 1 m/s y Veleta con un Rango: 0 ... 360°, Precisión: +/-3°, Resolución: 1° este sistema combinado tipo RM Young de forma de avión con una hélice de eje horizontal y una veleta que direcciona la hélice al viento como se observa en la figura 22 con el equipo se recolecto los datos necesarios para el objeto de estudio.



**Figura. 22.** Estación meteorológica instalada en el Instituto Superior Tecnológico Cotopaxi

Mediante la página en línea <https://rainwise.net> se obtuvo los datos de la velocidad del viento en intervalos de 1 hora, del año 2019 – 2020 en la página en línea dispone de la opción para descargar los datos recolectados, cabe mencionar el tiempo de duración de la licencia es de un año, la página en línea permite recolectar información y visualizar los parámetros a utilizar en tiempo real como se observa en la figura 23.



**Figura. 23.** Página de comunicación RainwiseNet

### 3.1.4 Datos descargados de la estación meteorológica

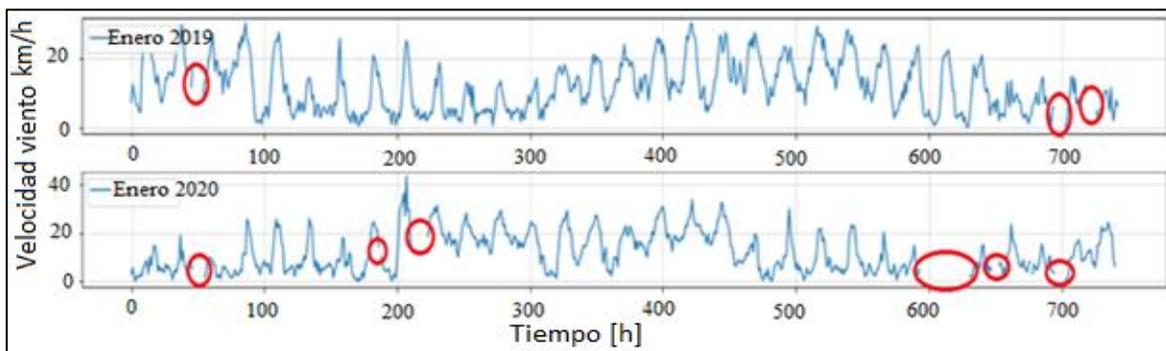
Los datos recolectados por la estación meteorológica se descargan en formato csv, la información descargada es de 24 meses de los años 2019 – 2020 en intervalos de una hora teniendo un total de 17544 datos, la data será utilizada para ingresar en el programa y posterior entrenamiento del algoritmo, se presenta un muestreo de información en la tabla 6.

**Tabla 6.** Muestreo de datos de la estación meteorológica

Time	Temp [°C]	Heat Index[°C]	Hum [%]	Baro [mb]	Windspeed [km/h]
1/2/2020 8: 00	96	96	93	7216	35
1/2/2020 9: 00	96	96	93	7217	26
1/2/2020 10:00	96	96	93	7217	29
1/2/2020 11:00	95	95	93	7217	31
1/2/2020 12:00	95	95	93	7218	13
1/2/2020 13:00	95	95	93	7218	39
1/2/2020 14:00	96	96	92	7218	51
1/2/2020 15:00	96	96	92	7218	51
1/2/2020 16:00	96	96	92	7218	51
1/2/2020 17:00	96	96	92	7218	51
1/2/2020 18:00	96	96	92	7218	51
1/2/2020 19:00	98	98	92	7219	72

### 3.1.5 Lectura de la data y depuración de datos nulos

Se procede con la lectura de los archivos descargados de la página de la estación meteorológica en formato csv, con la lectura de la data original del año 2019 y 2020, se revisa el estado de la data, determinando mediante la figura 24 que las mediciones realizadas existen espacios nulos los mismos que son valores que afectan en el comportamiento de los datos y posteriores usos que se deseen realizar.



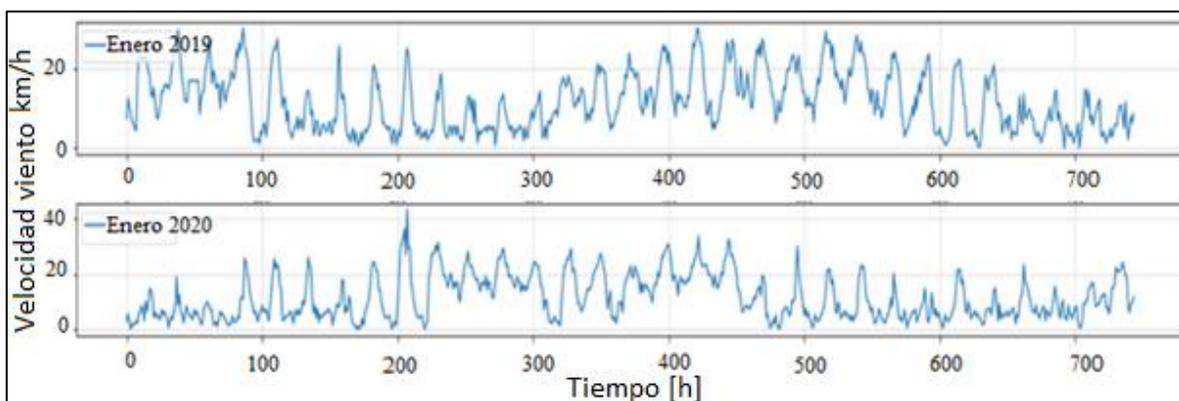
**Figura. 24.** Data original del mes de enero 2019 – 2020 de la velocidad de viento [km/h]

En la tabla 7 se presenta el resumen de valores nulos que se obtuvo al realizar el análisis de la data en general de los dos años, con esto se observa que se deben corregir 2509 valores para tener una data completa que permita proceder al paso de la programación.

**Tabla 7.** Resumen de datos nulos de la data

Resumen del dataframe				Resumen valores nulos	
0	Temp Avg	17544	non-null float64	Temp Avg	2509
1	Heat Index	17544	non-null float65	Heat Index	2509
2	Wind Chill	17544	non-null float66	Wind Chill	2509
3	Hum AVG	17544	non-null float67	Hum AVG	2509
4	Baro Avg	17544	non-null float68	Baro Avg	2509
5	Windspeed	17544	non-null float69	Windspeed	2509
6	Wind Direction	17544	non-null float70	Wind Direction	2509

Para completar los datos nulos se aplicó la media aritmética para cada uno de los casos completando la información faltante, cabe mencionar que al realizar este proceso la data no se altera los valores y antes de iniciar con el algoritmo se debe tratar como primer paso la data, una vez depurada por completo los dos años la data estará procesada para ser ingresada en el algoritmo, en la figura 25 se observa los datos completos en su totalidad.



**Figura. 25.** Data completa del mes de enero 2019-2020 velocidad del viento [km/h]

En la tabla 8 se observa los valores depurados en su totalidad presentando en el resumen de la data valores nulos de cero, esto permite que se continúe con el paso necesario para la programación.

**Tabla 8.** Resumen de datos completos de la data

Resumen del dataframe				Resumen valores nulos	
0	Temp Avg	17544	non-null float64	Temp Avg	0
1	Heat Index	17544	non-null float65	Heat Index	0
2	Wind Chill	17544	non-null float66	Wind Chill	0
3	Hum AVG	17544	non-null float67	Hum AVG	0
4	Baro Avg	17544	non-null float68	Baro Avg	0
5	Windspeed	17544	non-null float69	Windspeed	0
6	Wind Direction	17544	non-null float70	Wind Direction	0

### 3.1.6 Análisis de correlación de la velocidad del viento recolectada por la estación meteorológica.

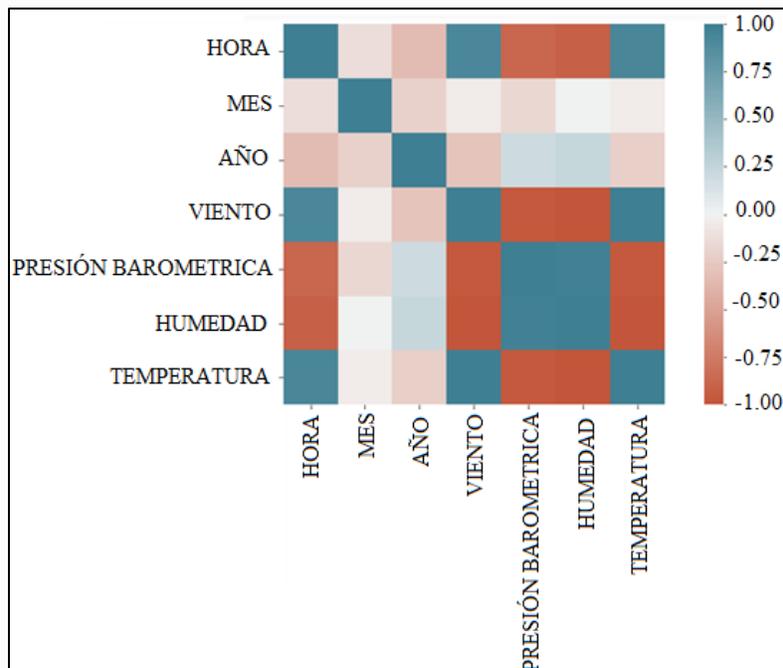
En base a estudios realizados [25] y a la correlación realizada se estima que los datos se encuentran en el rango mayor a 0,5 siendo una asociación moderada, encontrándose dentro de los parámetro que mayor influencia dispone para determinar la velocidad del viento como se observa en la tabla 9, las variables de año, mes, hora y viento se encuentran correlacionadas.

**Tabla 9.** Correlación de variables más relevantes en la velocidad del viento.

	HORA	MES	AÑO	VIENTO [km/h]	PRESION BAROMETRICA [mb]	HUMEDAD [%]	TEMPERATURA [°C]
HORA	1.000	-0.120	-0.332	0.914	-0.879	-0.928	0.931
MES	-0.120	1.000	-0.204	-0.044	-0.169	0.007	-0.043
AÑO	-0.332	-0.204	1.000	-0.297	0.188	0.230	-0.224
VIENTO	0.914	-0.044	-0.297	1.000	-0.955	-0.992	0.992
PRESION BAROMETRICA	-0.879	-0.169	0.188	-0.955	1.000	0.974	-0.968
HUMEDAD	-0.928	0.007	0.230	-0.992	0.974	1.000	-0.999
TEMPERATURA	0.931	-0.043	-0.224	0.992	-0.968	-0.999	1.000

En la figura 26 el mapa de calor muestra las relaciones entre múltiples variables, las áreas que están marcadas de color azul son las que están relacionadas correctamente, con el mapa de calor se observa que el viento está relacionado con la hora y temperatura, la velocidad

del viento va a presentar cambios directamente con la variación de la temperatura que se presente en la sector de Lasso, al ser clima frío debido a la altura que se encuentra de 2770 metros sobre el nivel del mar, con una temperatura promedio que ronda los 12 °C, esto provoca que el aire se eleve al tener menor densidad, esta es una de las razones principales por las cuales se va generando corrientes de viento por tal motivo se seleccionó como la variable de estudio que ayudara a realizar la predicción de la velocidad del viento.



**Figura. 26.** Comparación de datos depurados

### 3.1.7 Desarrollo del algoritmo de predicción en Python

Para el desarrollo del algoritmo se inicia con el ingreso de las librerías que se necesitaran para el objeto de estudio que se encargaran del procesamiento de la data las cuales permiten interpretar los datos que se desean analizar permitiendo automatizar el proceso de manipular datos con Python, modificar la data para obtener los datos necesarios en las tablas creando una matriz que se pueda tener ordenado de mejor manera, especializada en el cálculo numérico, especialmente para un gran volumen de datos, por medio de graficas se presentara los resultados que se deseen obtener, en la figura 27 se observa las librerías utilizadas para la programación.

```

import numpy as np # Realiza ecuaciones matemáticas
import pandas as pd # Lectura y procesamiento del Dataframe
import matplotlib.pyplot as plt # Permite realizar graficas
from sklearn.preprocessing import LabelEncoder # Codifica valores
from sklearn.model_selection import train_test_split # Escalar valores
import keras # Directamente todas las librerías de keras y tensorflow funcionan para crear la red LSTM
from keras.models import Sequential # Usar datos secuenciales
from keras.layers import Dense, Activation, Dropout # Procesamiento de varias capas
from sklearn.metrics import mean_squared_error # Pérdida de regresión del error cuadrático medio.
import datetime # Trabaja con fechas
    
```

**Figura. 27.** Librerías utilizadas de Python

### 3.1.8 Carga de datos depurados

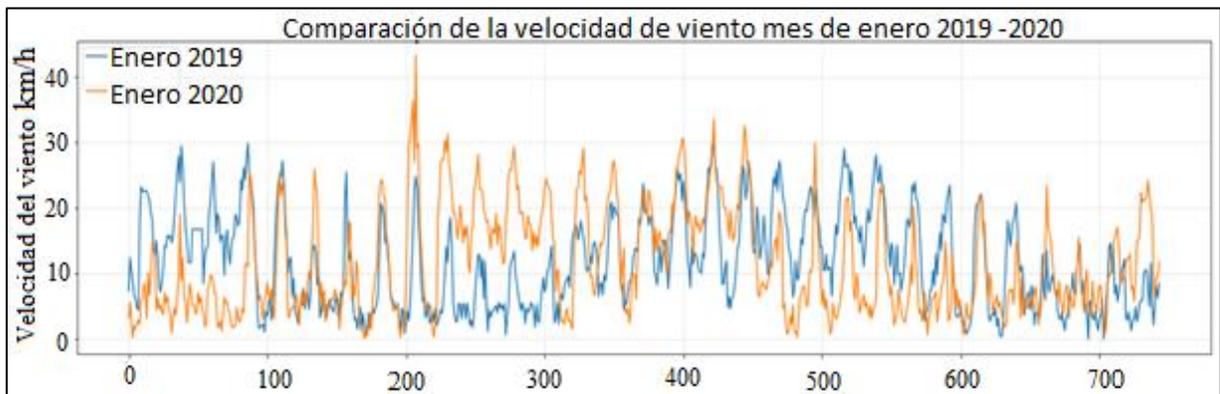
Los datos que fueron analizados y completados con anterioridad en un preprocesamiento de información se enlazan mediante la línea de código en Jupyter, permitiendo tener lectura del documento en formato csv, archivo en el que está almacenada información de dos años 2019 – 2020 en intervalos de una hora, en la figura 28 se observa el código ingresado.

```

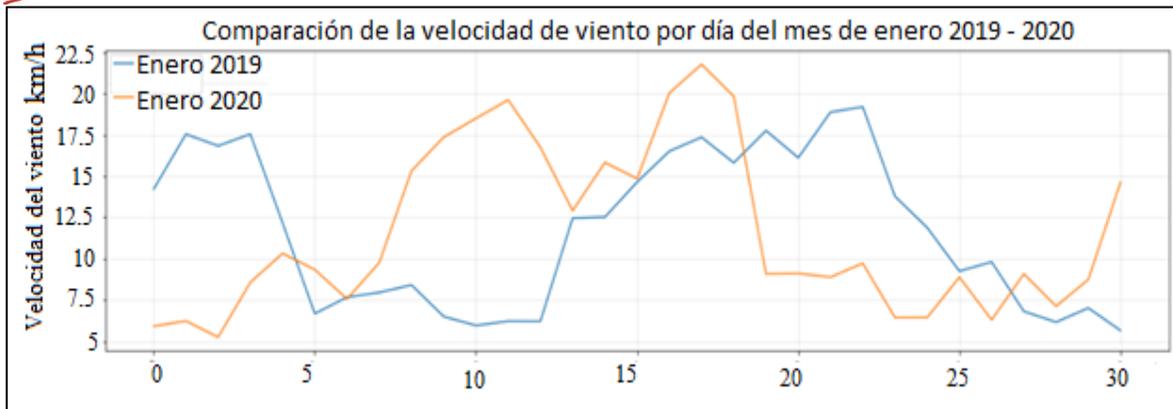
#Lectura de datos
data = pd.read_csv('DEPURACION_FINAL.csv')
data
    
```

**Figura. 28.** Carga de datos depurados mediante el enlace al drive

Con los datos depurados en su totalidad se realizó la comparación de dos maneras la primera se realizó del año 2019 – 2020 se observa en la figura 29, la segunda se tomó como muestreo la comparación de los datos medidos del mes de enero de cada año recolectados, la cual se observa en la figura 30, con la revisión se verificó que el comportamiento del viento tiene variaciones en relación a los datos comparados por mes y años.



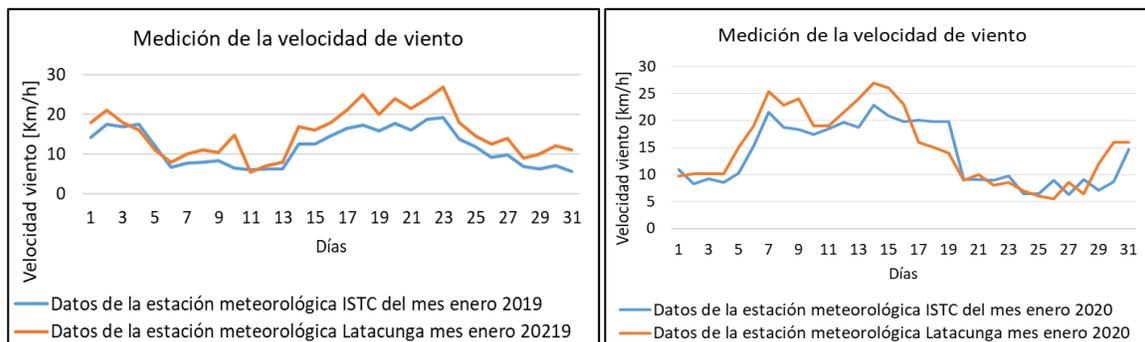
**Figura. 29.** Comparación de datos medidos por hora del mes de enero año 2019 - 2020



**Figura. 30.** Comparación de datos depurados

### 3.1.9 Comparación de la velocidad de viento medidos por la estación meteorológica del ISTC y los datos obtenidos on-line de Latacunga

Con la información depurada se realizó una comparación con los datos de la estación meteorológica ubicada en el sector de Lasso en el Instituto Superior Tecnológico Cotopaxi con los datos obtenidos en línea de la ciudad de Latacunga, se toma como referencia debido a ser el lugar más cercano al del estudio, se comparó la velocidad del viento [km/h] del mes de enero del 2019 y 2020, se observa en los dos casos que las curvas tienden a ser similares verificando que los datos recolectados no tienen una gran diferencia encontrándose en una secuencia similar de medición, cabe destacar que las mediciones de comparación no son registradas en el mismo sitio como se observa en la figura 31.



**Figura. 31.** Comparación velocidad del viento medidos por la estación meteorológica del ISTC y datos en línea de la ciudad de Latacunga.

### 3.1.10 Selección de las variables de entrada para el desarrollo del algoritmo

Una vez realizado la correlación de las variables como se presentó en la tabla 10 y en el mapa de calor de la figura 26, se procede con la selección de las variables de forma ordenada para presentar los parámetros en formato de año, mes, día, fecha, hora y velocidad del viento, tomando como referencia un muestreo del total como se observa en la tabla 10, con esto se genera un formato que puede entender el algoritmo y visualice la matriz de forma ordenada.

**Tabla 10.** Datos ordenados y seleccionados en función de la correlación

	AÑO	MES	DIA	FECHA	HORA	VELOCIDAD DEL VIENTO [km/h]
<b>0</b>	2019	1	2	1	0	7
<b>1</b>	2019	1	2	1	1	12
<b>2</b>	2019	1	2	1	2	11
<b>3</b>	2019	1	2	1	3	9
<b>4</b>	2019	1	2	1	4	7
...	...	...	...	...	...	...
<b>17539</b>	2020	12	4	31	19	7
<b>17540</b>	2020	12	4	31	20	3
<b>17541</b>	2020	12	4	31	21	2
<b>17542</b>	2020	12	4	31	22	0
<b>17543</b>	2020	12	4	31	23	2

### 3.1.11 Estandarización de variables numéricas

Para poder trabajar el algoritmo se transforma en una matriz en la cual se presenta valores sin la parte del índice quedando modificada en la que presenta en este caso filas y una columna que se trabajara dentro de la matriz como se observa en la figura 32, se presenta un muestreo de los valores obtenidos en la matriz.

```
array([[0.1372549 ],
       [0.23529412],
       [0.21568627],
       ...,
       [0.49019608],
       [0.39215686],
       [0.35294118]])
```

**Figura. 32.** Matriz de los valores estandarizados sin índices.

En función de la matriz obtenida y al tener una cantidad de valores diferentes se procede a escalar la data dentro del rango de 0 y 1 generando todos los valores dentro del rango establecido, con la cual el algoritmo trabajara.

### 3.1.12 División de la data de entrenamiento y de prueba

Para la validación de los datos se dividió los 17544 datos recolectados en periodos de una hora durante los años 2019 -2020, Valiéndose del principio de la regla 80/20, se aplicó el diagrama de Pareto que permite asignar un orden de prioridades y facilita de este modo el estudio, el 80% (14046 datos) para el entrenamiento y un 20 % (3508 datos) para la validación de los datos que no ha sido visto, en la tabla 11 se detalla las fechas con las que se consideró cada porcentaje para el estudio.

**Tabla 11.** División de la data de entrenamiento y de prueba

<b>Datos recolectados por la estación meteorológica</b>				
Totales	17544	100%	1/1/2019	31/12/2020
Entrenamiento	14036	80%	1/1/2019	7/8/2020
Validación	3508	20%	7/8/2020	31/12/2020

### 3.1.13 Estructura de la Red Neuronal Recurrente LSTM

Los valores ya escalados dentro del parámetro mencionado se proceden a empezar con el entrenamiento ingresando un valor tanto de entrenamiento como de testeo con esto se podrá obtener los datos que se necesite obtener en función de días pasados que servirá de entrenamiento para el algoritmo, para esto se aplicó un modelo secuencial de tipo LSTM como se presenta en la figura 33

- Capa de entrada: Compuesta por 25 entradas de datos que corresponde a los meses que serán tomados como base para el desarrollo del pronóstico.
- Capas ocultas: El modelo desarrollado tendrá una capa oculta con 25 neuronas respectivamente en cada capa.

- Capa de salida: el modelo posee una salida, por lo que, sería una red neuronal recurrente con varias entradas y una salida

Model: "sequential"		
Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 60, 25)	2700
dropout (Dropout)	(None, 60, 25)	0
lstm_1 (LSTM)	(None, 25)	5100
dropout_1 (Dropout)	(None, 25)	0
dense (Dense)	(None, 1)	26
Total, params: 7,826		
Trainable params: 7,826		
Non-trainable params: 0		
None		

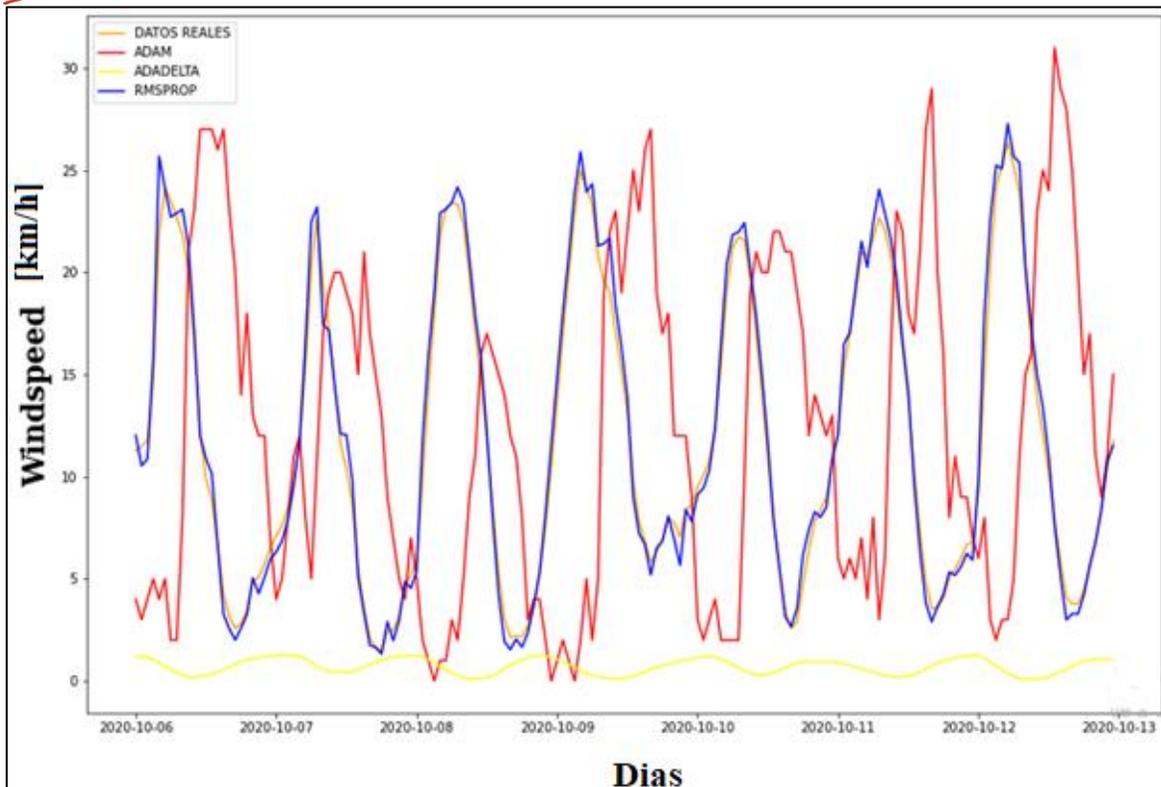
Figura. 33. Estructura de la red neuronal LSTM

### 3.1.14 Optimizadores del algoritmo

Se utilizó 3 optimizadores ADAM, ADADELTA y RMSPROP considerando 10 neuronas y 15 épocas para cada uno de los casos, los resultados se presentan en la tabla 12 con los rendimientos de cada optimizador obtenidos luego del entrenamiento del algoritmo, además se puede apreciar la figura 34 el comportamiento de la velocidad del viento con cada optimizador.

**Tabla 12.** Resultados obtenidos aplicando diferentes optimizadores para la predicción de la velocidad del viento

Algoritmos	
Optimizador	Error MAE
ADAM	0.056
ADADELTA	0.191
RMSPROP	0.052



**Figura. 34.** Comportamiento de la predicción de la velocidad del viento con diferentes optimizadores

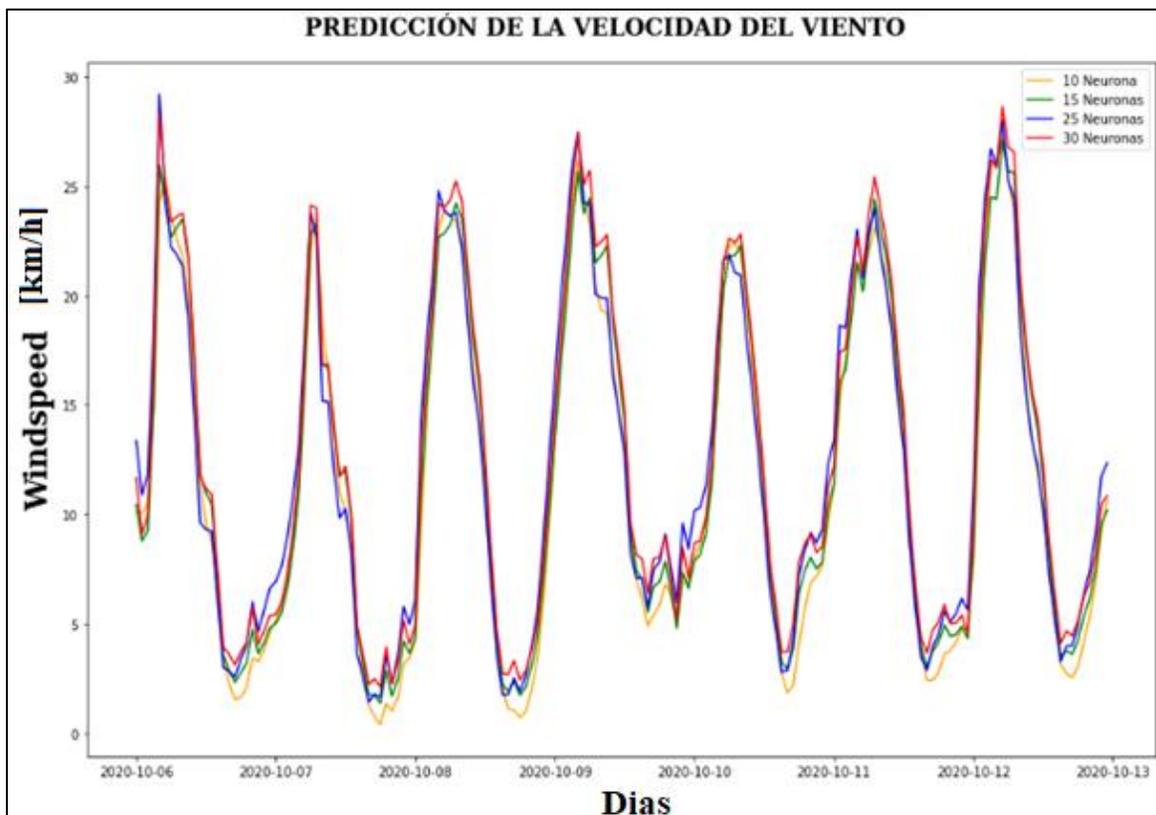
En la figura 34 se presenta las variaciones que se tiene por cada optimizador de entrenamiento, los resultados obtenidos permiten seleccionar el mejor resultado para la predicción de la velocidad del viento optando por el RMSPROP que tiene un valor 0.052 de error absoluto medio (MAE) el cual me permite acercarse de mejor manera a los valores reales a diferencia de los dos restantes que presentan un error MAE elevado.

### 3.1.15 Variación del número de neuronas

Para el análisis se consideró diferente cantidad de neuronas, se seleccionó el optimizador de entrenamiento con menor error siendo el RMSPROP y 15 épocas para todos los casos, se muestra los resultados del rendimiento en la tabla 13 luego del entrenamiento, además se puede apreciar la figura 35 el comportamiento de la velocidad del viento con diferente neurona.

**Tabla 13.** Resultados del error MAE con la variación de neuronas para la predicción de la velocidad del viento

Número de Neuronas	
Cantidad	Error MAE
10	0.054
15	0.050
25	0.048
30	0.0478



**Figura. 35.** Comportamiento de la predicción de la velocidad del viento con diferentes números de neuronas

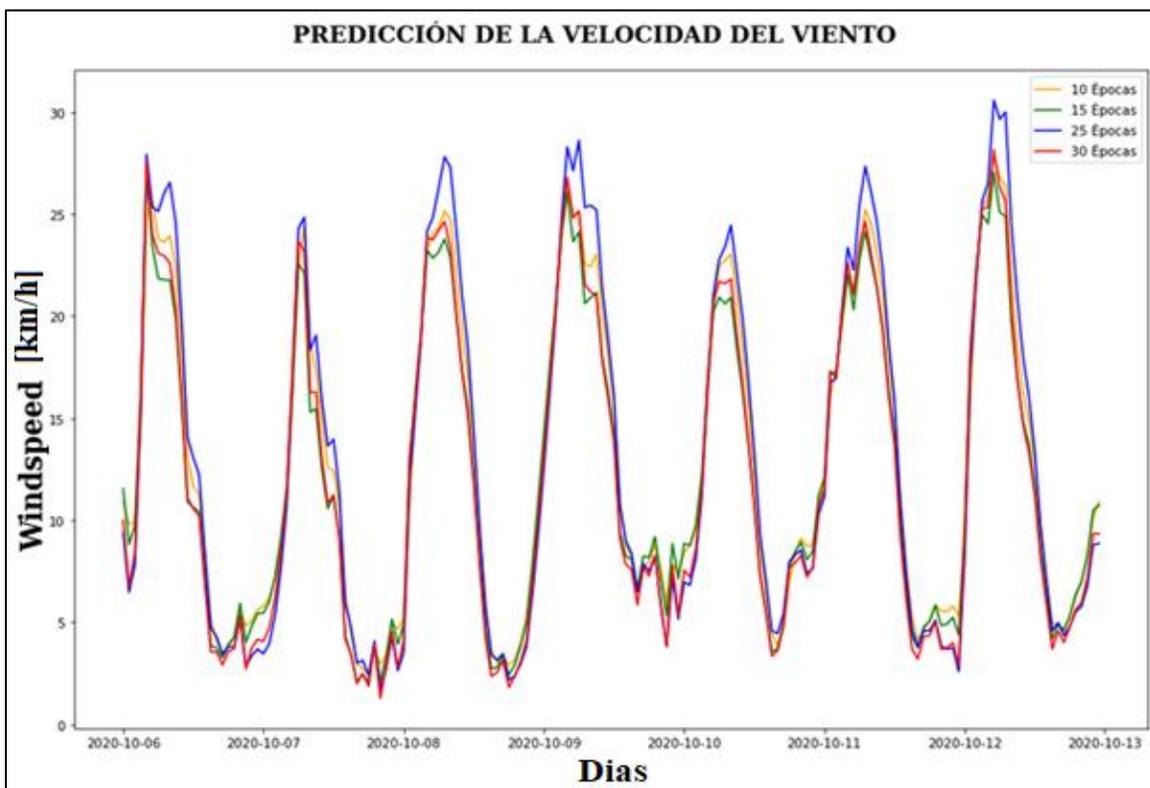
Con el análisis de los resultados se obtiene que con 25 neuronas se tiene una buena predicción de la velocidad del viento, cabe mencionar que si se utiliza una cantidad de neuronas superior a la seleccionada se tiene un menor error, pero con overfitting que hace referencia al fallo de nuestro modelo al generalizar el conocimiento que pretendemos que adquiera, en la figura 35 se observa el comportamiento para diferentes números de neuronas.

### 3.1.16 Variación del número de épocas

Para el análisis se consideró diferentes épocas o número de entrenamientos (una época es el número de veces que se ejecuta el algoritmo de la red neuronal), se seleccionó el optimizador de entrenamiento RMSPROP y las 25 neuronas que son parámetros obtenidos en los análisis anteriores en la tabla 14 se muestra los resultados del rendimiento, además se puede apreciar la figura 36 el comportamiento de la velocidad del viento con diferentes épocas.

**Tabla 14.** Resultados derivados de la variación del número de épocas para la predicción de la velocidad del viento

Número de Épocas	
Cantidad	Error MAE
10	0.050
15	0.048
25	0.046
30	0.0464



**Figura. 36.** Comportamiento de la predicción de la velocidad del viento con diferentes números de épocas

Con el análisis de los resultados se obtiene que con 25 épocas se tiene una buena predicción de la velocidad del viento cabe mencionar que, si se utiliza una cantidad de épocas superior a la seleccionada con la que se obtuvo un sobre entrenamiento, un overfitting que hace referencia al fallo de nuestro modelo al generalizar el conocimiento que pretendemos que adquiera, en la figura 36 se observa el comportamiento de la predicción con diferente número de épocas.

El algoritmo tiene un buen rendimiento utilizando el optimizador RMSPROP, 25 neuronas y 25 épocas, se ha determinado la eficiencia del algoritmo en base a los análisis realizados anteriormente obteniendo un error absoluto medio (MAE) de 0.0472, el proceso de entrenamiento se observa en la figura 37 con los parámetros mejor optimización.

```

model, history=NN_LSTM(optimizador='rmsprop', neuronas=25, épocas=25)

Epoch 1/25
110/110 [=] - 23s 122ms/step - loss: 0.0166 - mean_absolute_error: 0.1018
Epoch 2/25
110/110 [=] - 13s 117ms/step - loss: 0.0077 - mean_absolute_error: 0.0686
Epoch 3/25
110/110 [=] - 13s 118ms/step - loss: 0.0068 - mean_absolute_error: 0.0638
Epoch 4/25
110/110 [=] - 13s 120ms/step - loss: 0.0062 - mean_absolute_error: 0.0611
Epoch 5/25
110/110 [=] - 13s 119ms/step - loss: 0.0058 - mean_absolute_error: 0.0590
Epoch 6/25
110/110 [=] - 12s 105ms/step - loss: 0.0055 - mean_absolute_error: 0.0574
Epoch 7/25
110/110 [=] - 11s 97ms/step - loss: 0.0051 - mean_absolute_error: 0.0552
Epoch 18/25
110/110 [=] - 13s 117ms/step - loss: 0.0039 - mean_absolute_error: 0.0477
Epoch 19/25
110/110 [=] - 13s 117ms/step - loss: 0.0039 - mean_absolute_error: 0.0477
Epoch 20/25
110/110 [=] - 13s 118ms/step - loss: 0.0038 - mean_absolute_error: 0.0473
Epoch 21/25
110/110 [=] - 13s 117ms/step - loss: 0.0038 - mean_absolute_error: 0.0473
Epoch 22/25
110/110 [=] - 13s 118ms/step - loss: 0.0038 - mean_absolute_error: 0.0473
Epoch 23/25
110/110 [=] - 13s 117ms/step - loss: 0.0038 - mean_absolute_error: 0.0473
Epoch 24/25
110/110 [=] - 13s 116ms/step - loss: 0.0038 - mean_absolute_error: 0.0473
Epoch 25/25
110/110 [=] - 13s 115ms/step - loss: 0.0038 - mean_absolute_error: 0.0472

```

Figura. 37. Entrenamiento del algoritmo con optimizador = 'rmsprop', neuronas = 25, épocas = 25

Para el entrenamiento con el análisis realizado anteriormente se seleccionó 25 épocas, en la figura 38 se observa el comportamiento del aprendizaje de la red LSTM siendo de manera descendente hasta llegar a valores bajos de error del MAE de 0.0472 y Loss de 0.0038, la predicción al tener un error bajo se asemejan a los valores reales,

### COMPORTAMIENTO DEL APRENDIZAJE DE LA RED LSTM

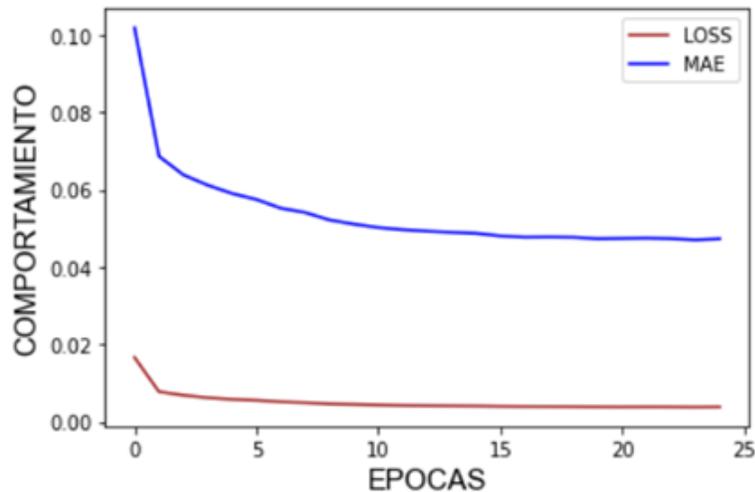


Figura. 38. Curva de aprendizaje de entrenamiento LSTM

#### 3.1.17 Validación y grafica de resultados

Para la validación y evaluación de los datos se realizó la predicción de 7 días del mes de octubre que representa al 20% de información que no fue vista por el algoritmo como se especificó en la tabla 15, el Error absoluto medio que presenta es de 0.0472 como se observa en la figura 37 de esta manera se comprueba la precisión de la predicción con los datos reales de la velocidad de viento en la figura 39 se presenta la gráfica de resultado, además en el anexo III se adjunta la programación completa

**Tabla 15.** Valor del error absoluto medio entre la velocidad del viento real y la pronosticada

ERROR MAE	
Valor del error	0.0472

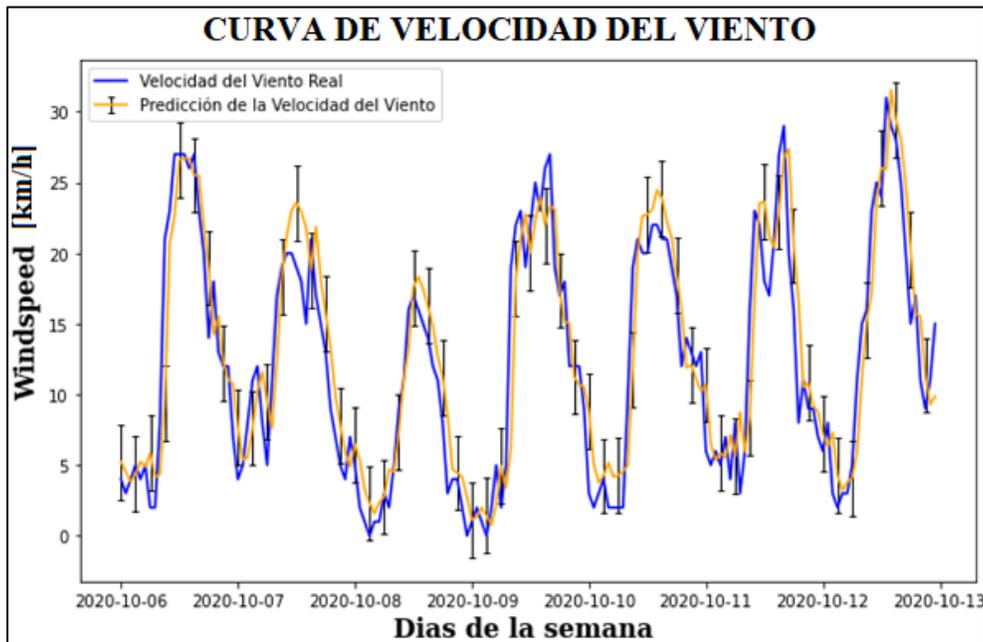


Figura. 39. Curva de la velocidad del viento de predicción y reales de 7 días del mes de octubre.

Adicionalmente en la figura 40 se presentan los intervalos de la velocidad del viento comparados con los datos reales y los pronosticados en intervalos de 1 hora durante un día, donde se verifica el comportamiento de las curvas presentándose una similitud.

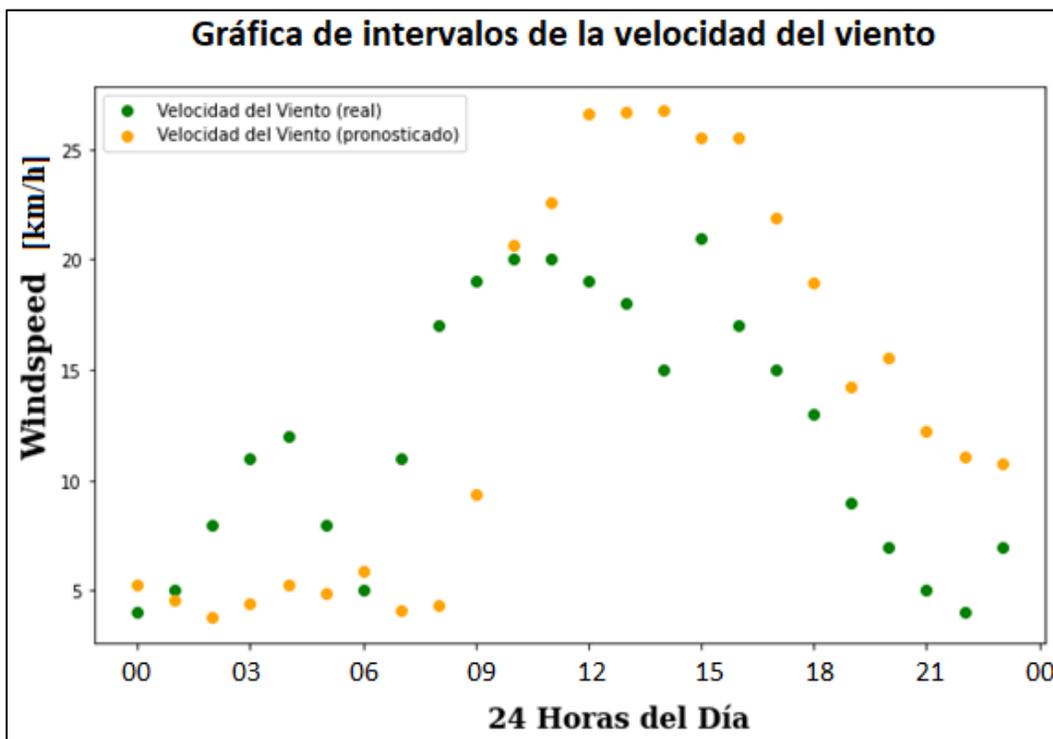
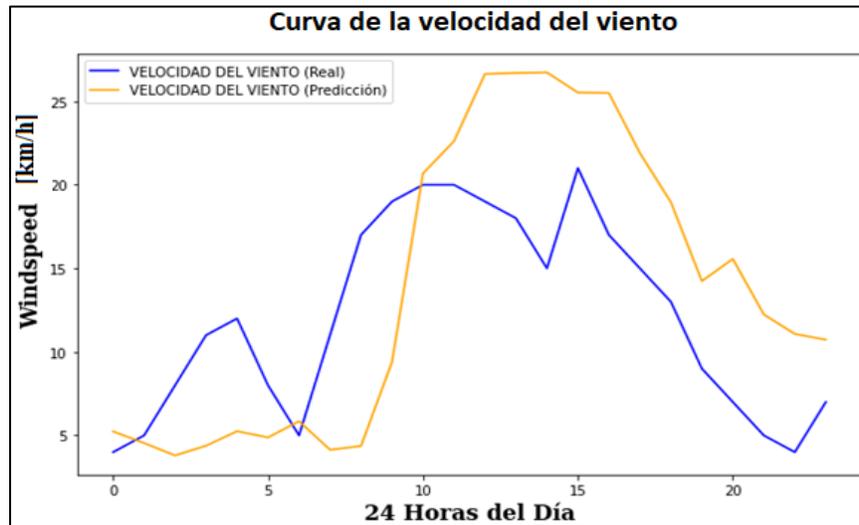


Figura. 40. Grafica de intervalos de la velocidad del viento de predicción y real del día siete de octubre 2020.

En la figura 41 se visualiza el comportamiento de la velocidad del viento pronosticado y real tienden a ser similares, en la 00:00 horas parte de una velocidad de 5 km/h, entre las 6:00 hasta las 16:00 la velocidad de viento comienza a incrementarse, a partir de las 16:00 hasta las 00:00 comienza a descender la velocidad de viento teniendo un comportamiento similar en la pronosticada y real.



**Figura. 41.** Curva de la velocidad del viento de predicción y real del día siete de octubre de 2020.

### 3.1.18 Análisis del comportamiento de los errores de la velocidad del viento del año 2020 para determinar la confiabilidad y viabilidad del programa realizado en Python

Para el criterio de validación de los datos pronosticados se debe cumplir que el error promedio de la RNA sea de menor que el error promedio aplicando el método de regresión lineal para determinar la precisión de la medición, la proporción del error con respecto al valor exacto de la medición y teniendo en cuenta que una medida es buena cuando no supera el 5%. [52]

Se realizó el método de regresión lineal para determinar el error que se tendrá con la velocidad del viento del año 2020, donde se observa que el error porcentual absoluto medio (MAPE) es de 13% valor que se encuentra fuera del porcentaje admitido. Como se observa en la figura 42, además en la figura 46 se presenta la ecuación y un  $R^2 = 0.1008$  valor bajo para realizar la predicción con el método mencionado.

PRONÓSTICO DE LA VELOCIDAD DE VIENTO MEDIANTE REGRESIÓN LINEAL										
n	Mes	Velocidad Viento Real [km/h]	Velocidad Pronosticada [km/h]	Error Abs.	Error % Abs.	$\Sigma$ Error Abs.	MAD	Error Nor.	$\Sigma$ Error Nor.	TS
1	Enero	11.62	9.09	2.534	<b>21.80%</b>	2.53	<b>2.50</b>	2.53	2.53	<b>1.00</b>
2	Febrero	9.30	9.38	0.086	<b>0.93%</b>	2.62	<b>1.30</b>	-0.09	2.45	<b>1.90</b>
3	Marzo	8.54	9.48	0.931	<b>10.90%</b>	3.55	<b>1.20</b>	-0.93	1.52	<b>1.30</b>
4	Abril	8.47	9.49	1.011	<b>11.93%</b>	4.56	<b>1.10</b>	-1.01	0.51	<b>0.50</b>
5	Mayo	12.05	9.04	3.008	<b>24.97%</b>	7.57	<b>1.50</b>	3.01	3.51	<b>2.30</b>
6	Junio	9.72	9.33	0.397	<b>4.08%</b>	7.97	<b>1.30</b>	0.40	3.91	<b>3.00</b>
7	Julio	9.44	9.36	0.070	<b>0.75%</b>	8.04	<b>1.10</b>	0.07	3.98	<b>3.60</b>
8	Agosto	11.22	9.14	2.084	<b>18.57%</b>	10.12	<b>1.30</b>	2.08	6.06	<b>4.70</b>
9	Septiembre	10.30	9.26	1.047	<b>10.16%</b>	11.17	<b>1.20</b>	1.05	7.11	<b>5.90</b>
10	Octubre	10.23	9.26	0.970	<b>9.48%</b>	12.14	<b>1.20</b>	0.97	8.08	<b>6.70</b>
11	Noviembre	8.57	9.47	0.900	<b>10.50%</b>	13.04	<b>1.20</b>	-0.90	7.18	<b>6.00</b>
12	Diciembre	7.30	9.63	2.327	<b>31.87%</b>	15.37	<b>1.30</b>	-2.33	4.85	<b>3.70</b>
SUMA %					155.93%					
n					12					
MAPE					<b>13%</b>					

Figura. 42. Comportamiento de la velocidad del viento mediante Regresión lineal

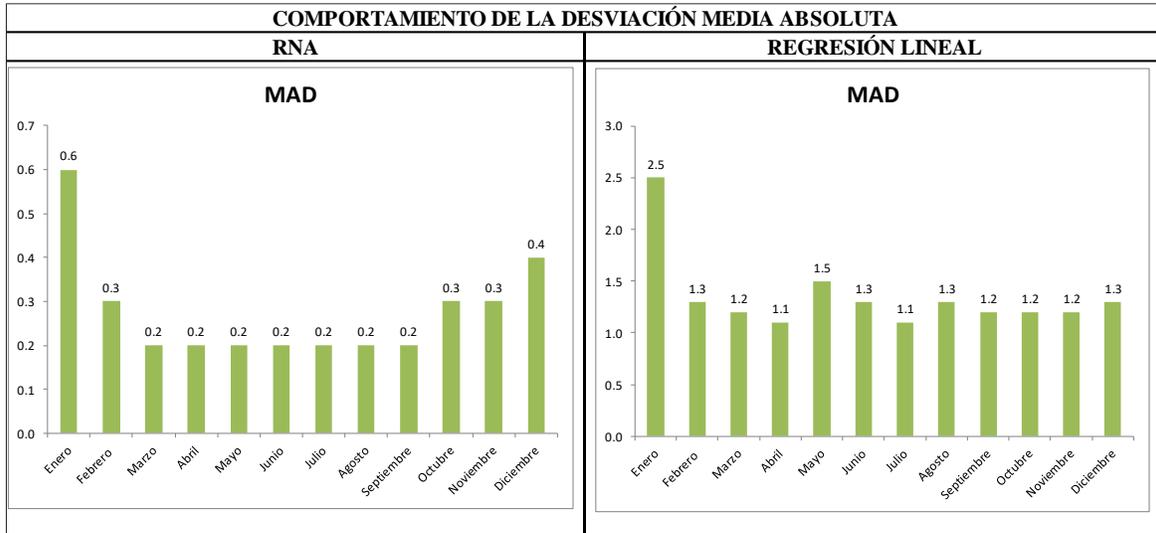
La predicción mediante redes neuronales de cada mes se determinó el error de cada uno para conocer el comportamiento de la velocidad de viento del año 2020, donde se visualiza que el valor de MPAE es del 4% ubicándose dentro del rango determinando para obtener una buena precisión con respecto al valor real como se observa en la figura 43.

PRONÓSTICO DE LA VELOCIDAD DE VIENTO MEDIANTE REDES NEURONALES ARTIFICIALES (PYTHON)										
n	Mes	Velocidad Viento Real [km/h]	Velocidad Pronosticada [km/h]	Error Abs.	Error % Abs.	$\Sigma$ Error Abs.	MAD	Error Nor.	$\Sigma$ Error Nor.	TS
1	Enero	11.62	11.02	0.6083	<b>5.23%</b>	0.6083	<b>0.6</b>	0.6083	0.6083	<b>1</b>
2	Febrero	9.30	9.28	0.0131	<b>0.14%</b>	0.6214	<b>0.3</b>	0.0131	0.6214	<b>2.1</b>
3	Marzo	8.54	8.60	0.0543	<b>0.64%</b>	0.6756	<b>0.2</b>	-0.0543	0.5671	<b>2.8</b>
4	Abril	8.47	8.57	0.0982	<b>1.16%</b>	0.7739	<b>0.2</b>	-0.0982	0.4689	<b>2.3</b>
5	Mayo	12.05	12.03	0.0156	<b>0.13%</b>	0.7895	<b>0.2</b>	0.0156	0.4845	<b>2.4</b>
6	Junio	9.72	9.36	0.3666	<b>3.77%</b>	1.1561	<b>0.2</b>	0.3666	0.8512	<b>4.3</b>
7	Julio	9.44	9.16	0.2713	<b>2.88%</b>	1.4274	<b>0.2</b>	0.2713	1.1225	<b>5.6</b>
8	Agosto	11.22	11.24	0.0118	<b>0.10%</b>	1.4392	<b>0.2</b>	-0.0118	1.1107	<b>5.6</b>
9	Septiembre	10.30	10.90	0.5961	<b>5.79%</b>	2.0353	<b>0.2</b>	-0.5961	0.5146	<b>2.6</b>
10	Octubre	10.23	9.40	0.8384	<b>8.19%</b>	2.8737	<b>0.3</b>	0.8384	1.3530	<b>4.5</b>
11	Noviembre	8.57	9.04	0.4675	<b>5.45%</b>	3.3413	<b>0.3</b>	-0.4675	0.8855	<b>3</b>
12	Diciembre	7.30	8.49	1.1884	<b>16.27%</b>	4.5297	<b>0.4</b>	-1.1884	-0.3029	<b>-0.8</b>
SUMA %					49.75%					
n					12					
MAPE					<b>4%</b>					

Figura. 43. Comportamiento de la velocidad del viento mediante Python

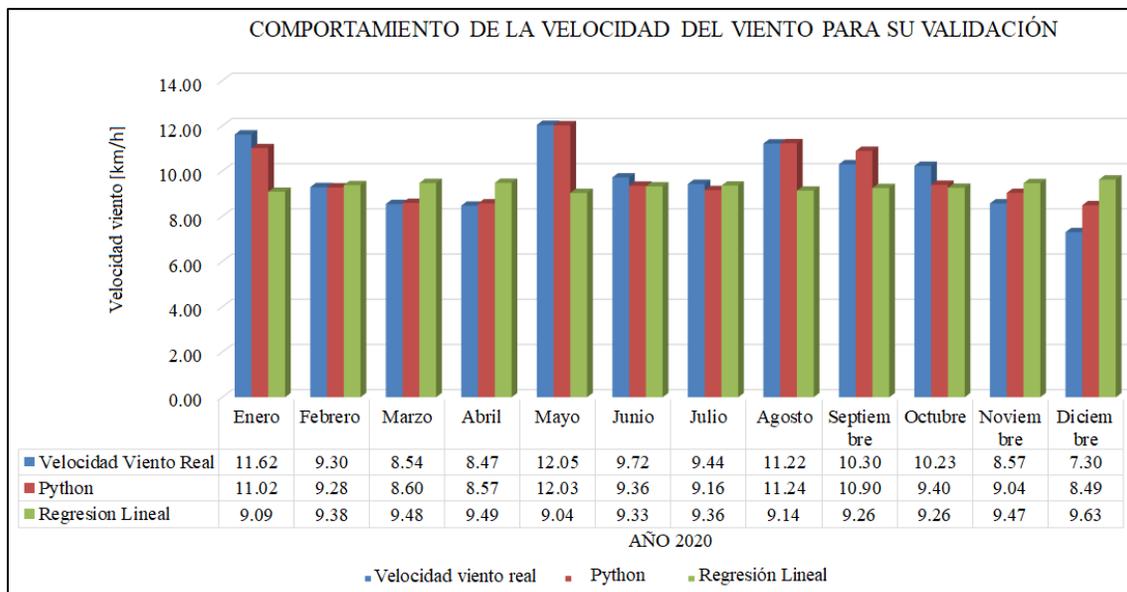
Con los resultados obtenidos de los dos métodos se determinan la desviación media absoluta (MAD) que se define como la distancia media entre cada valor y la media, se observa el comportamiento de cada uno de las mediciones de los errores promedios de los

pronosticados, los valores MAD en las redes neuronales presentan un error bajo a diferencia del método de regresión lineal que los valores son altos para cada mes del año 2020.



**Figura. 44.** Comportamiento de la velocidad del viento mediante desviación media absoluta con la RNA y regresión lineal

Con los análisis realizados en las gráficas de barras se observan los dos métodos utilizados con los datos reales de la velocidad del viento se observa que la red neuronal artificial tiene mayor similitud con un error del MAPE de 4 % a diferencia de los de regresión lineal que presento un error del 13%, se realiza la validación de los resultados verificando que el de mayor precisión es el de las RNA como se observa en la figura 45.



**Figura. 45.** Comportamiento de la velocidad del viento para su validación

Con los resultados obtenidos de la predicción de la velocidad del viento con la RNA se realiza la gráfica con los valores reales promedio de cada mes, donde se observa que la curva es similar en los diferentes meses del año 2020 como se observa en la figura 46.

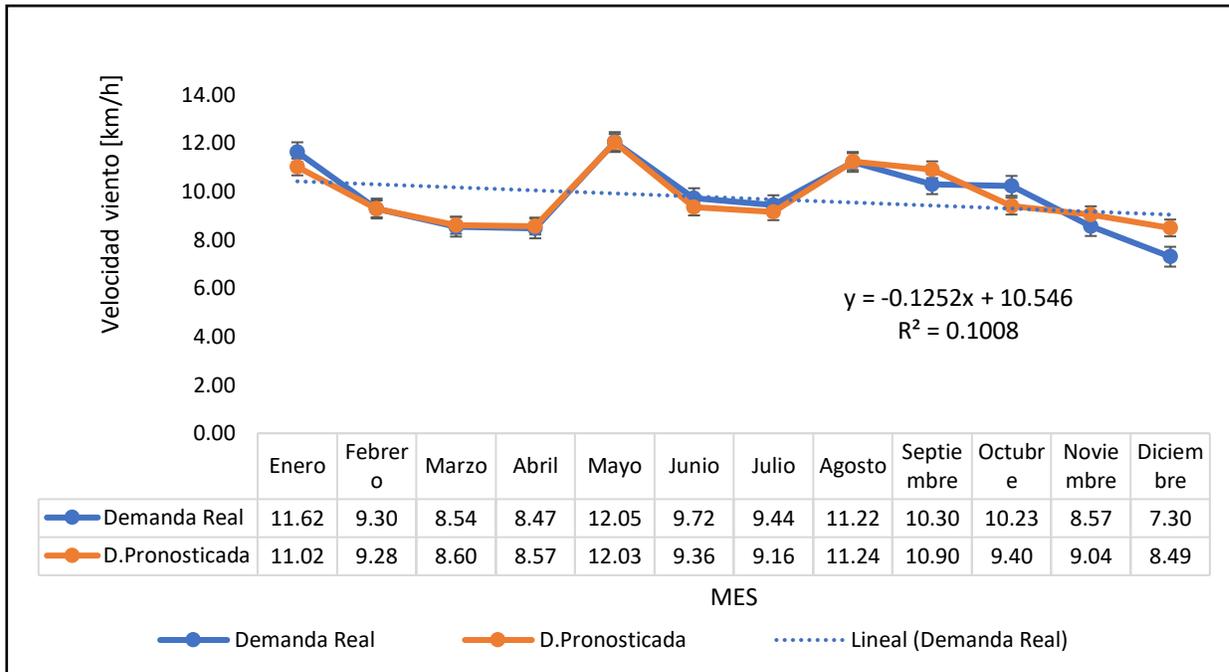
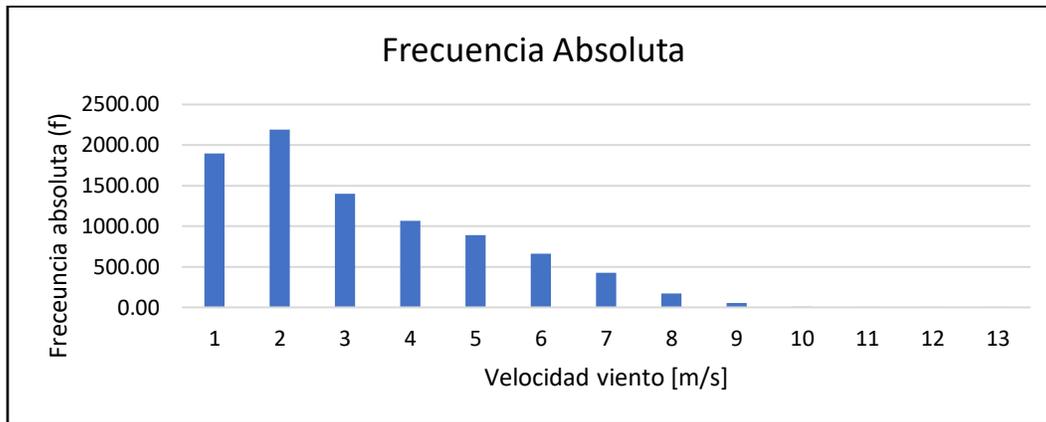


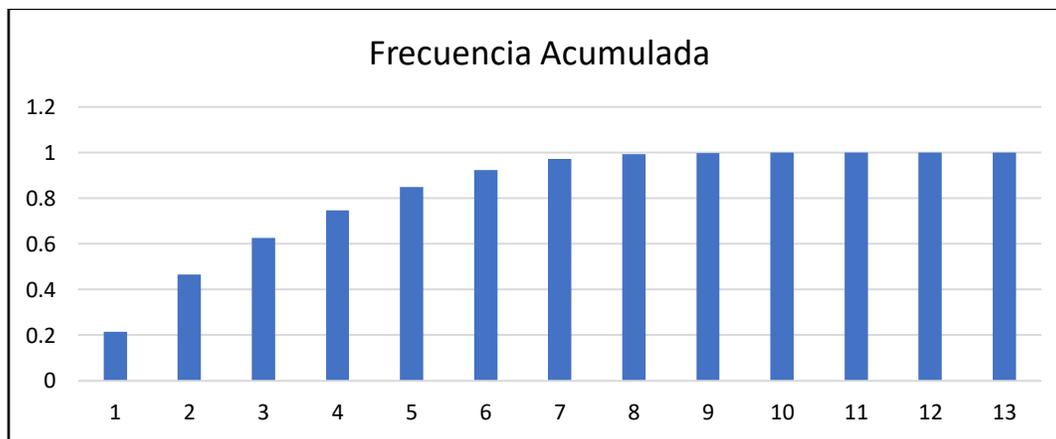
Figura. 46. Comparación de la velocidad del viento real y pronosticada del año 2020

### 3.1.19 Distribución de Weibull

Para analizar el comportamiento de la velocidad del viento se tomó los datos pronosticados con la red neuronal artificial, se modeló la distribución de la velocidad por medio de la regla de Sturges en la cual interviene los datos del año 2020 a ser procesados para determinar la función de probabilidad de Weibull, la distribución se observa en el anexo VIII observando los rangos de velocidad, frecuencias relativas, frecuencias absolutas y la sumatoria de frecuencias acumuladas como se observa en la figuras 48, interpretando que la mayor incidencia que se presenta es el rango de 2 m/s con 2189 repeticiones, con la frecuencia acumulada se obtendrá el tiempo de aprovechamiento que se tendrá con el aerogenerador a partir del rango que determine la ficha técnica, para realizar el procedimiento se realizó la conversión de los datos pronosticados de km/h a m/s.



**Figura. 47.** Histograma de frecuencia de la velocidad de viento

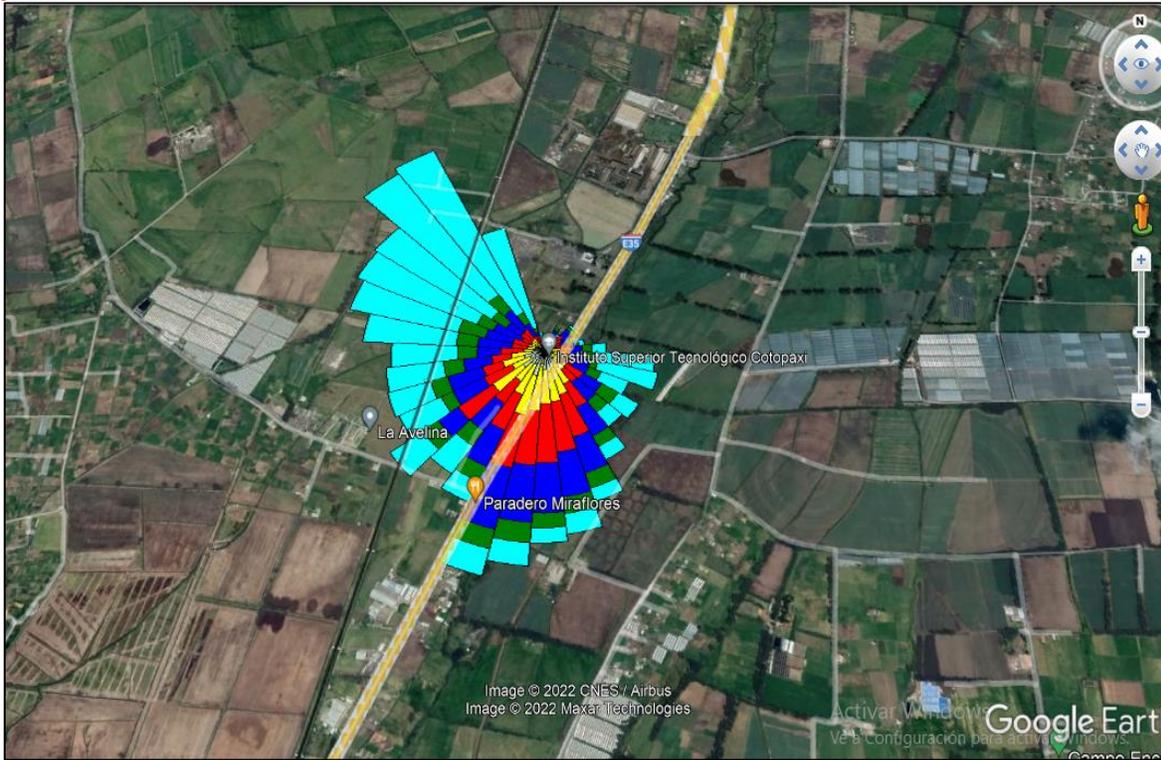


**Figura. 48.** Histograma de frecuencia acumulada de la velocidad de viento

### 3.1.20 Rosa de los vientos

Para el análisis del objeto de estudio es importante conocer de dónde provienen los vientos más frecuentes y más fuertes en el área donde se quiere instalar un aerogenerador, para conocer estos datos se recurre a la estación meteorológica que miden la dirección y la velocidad del viento en un periodo de tiempo determinado del año 2020,

Los datos ingresados fueron en periodos de una hora basados en el MANUAL PARA EL MANEJO DE SOFTWARE ESPECIALIZADO WRPLOT [53] para obtener la rosa de vientos del Instituto Superior Tecnológico Cotopaxi, las coordenadas del lugar son  $0^{\circ}, 47' 57.63''$  S y  $78^{\circ} 37' 08.31''$  O, en la figura 49 se observa el lugar de emplazamiento que se genera en Google Earth



**Figura. 49.** Lugar de emplazamiento del Instituto Superior Tecnológico Cotopaxi

En las figuras 50 - 51 se observa la rapidez y dirección del viento en el área determinada, se ha utilizado las dieciséis direcciones para el análisis que son las recomendadas 8 rumbos colaterales (es decir, 16 rumbos de dirección), los anillos forman la longitud de radio de la rosa de los vientos indicando la frecuencia acumulada en la zona con esto se puede identificar qué el color mostrado en la rosa de vientos tiene diferentes rangos de velocidad, como se observa, el viento ingresa por el Noroeste y es la de mayor incidencia saliendo por el Sureste, debido a la dirección analizada se conoce la posición del aerogenerador en la que se colocara.

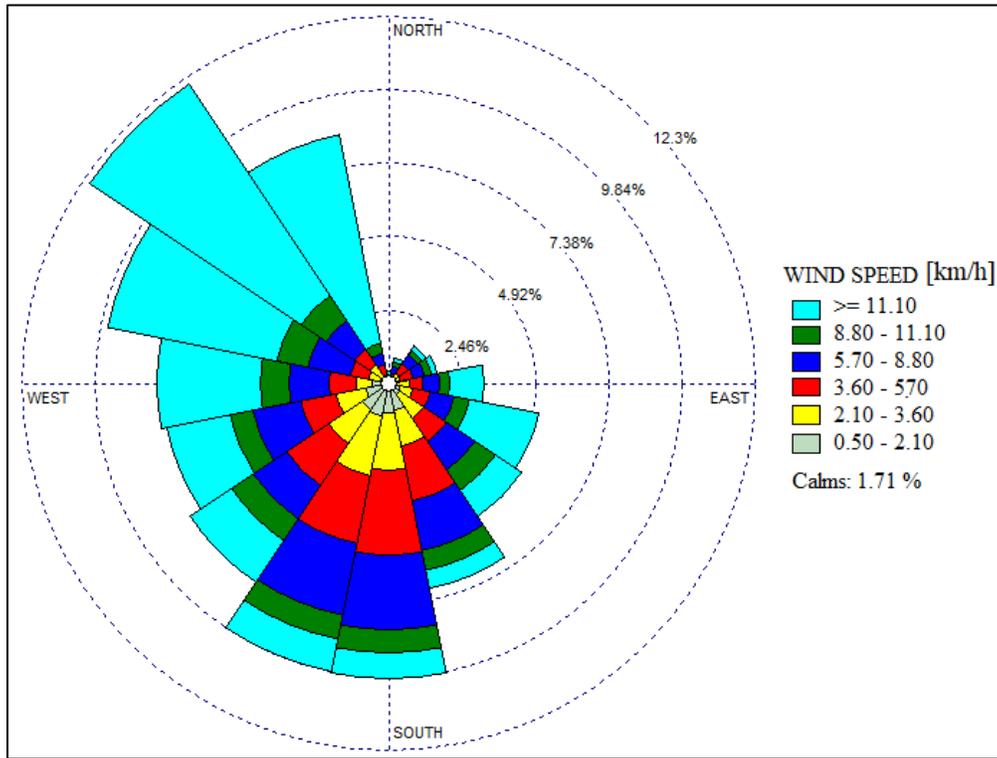


Figura. 50. Rosa de vientos ISTC, dirección de donde proviene el viento

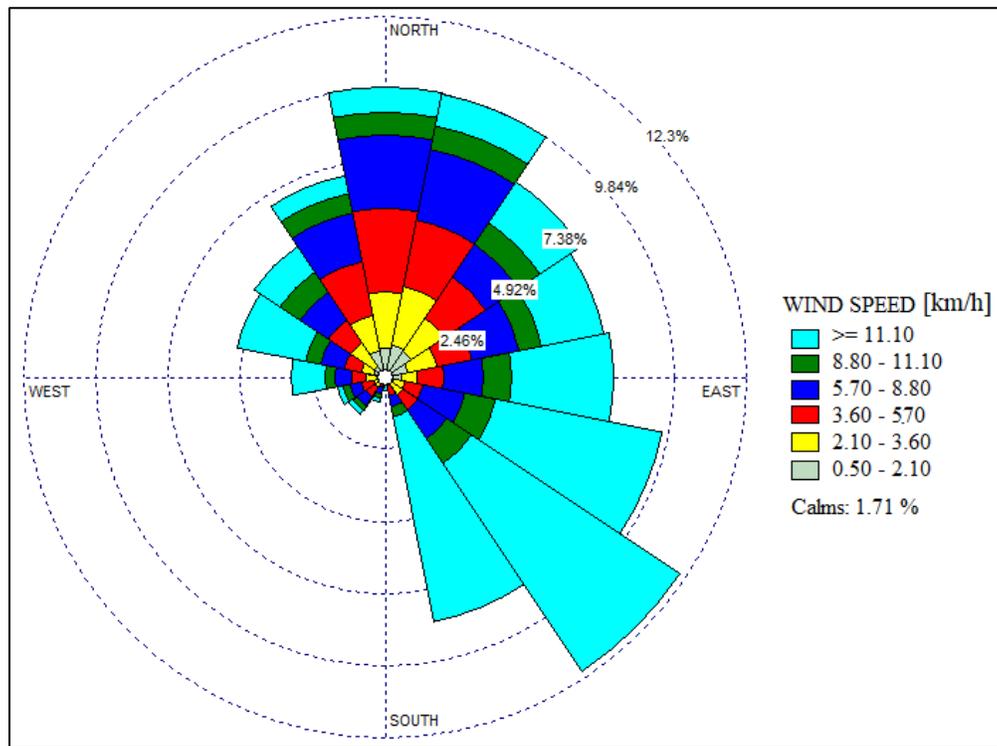


Figura. 51. Rosa de vientos ISTC, dirección hacia dónde se dirige el viento.

### 3.1.21 Velocidad del viento promedio

Para determinar el valor de la velocidad promedio se ha utilizado el software WaspClimateAnalyst3 se ingresó los datos pronosticados para determinar la velocidad promedio que se obtendrá durante los diferentes meses del año del 2020, en la figura 52 se presenta las gráficas del año 2020 y los resultados que presenta el factor de escala (A), el factor de forma (K) la velocidad promedio (U) y la densidad de potencia (P), en la Sin embargo los meses de manera individual se adjuntan en el anexo V.

La velocidad promedio del sector de Lasso es de 2.72 m/s que se obtuvo como resultado en el software WaspClimateAnalyst3, valor que fue comparado en la página de Enair (<https://www.enair.es/es/app>) en la que se ingresó las coordenadas del lugar de estudio obteniendo una velocidad promedio de 3.3 m/s, la información parte del origen de la NASA, la complejidad de los cálculos pueden existir márgenes de error en las medias de viento que pueden variar del 1 al 15%, también es importante considerar que no se aplican pérdidas por turbulencias de objetos cercanos, la información se adjunta en el anexo VI.

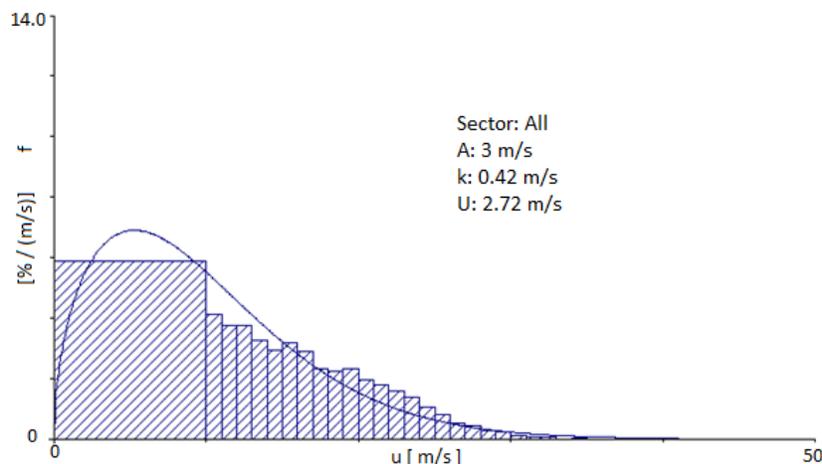


Figura. 52. Distribución de Weibull del año 2020

Los datos obtenidos por cada mes se presentan en la tabla 16, se observa el mes de mayo tendrá una velocidad promedio de 3.61 m/s siendo la de mayor rango y la de menor el mes de diciembre siendo de 2,06 m/s ingresando en los parámetros mínimos, además se obtuvo el valor promedio de viento con 2.72 m/s por año, el factor de forma que muestra a qué tipo de distribución de probabilidad se aproxima y el factor de escala o vida característica.

**Tabla 16.** Velocidad del viento promedio por mes

Mes 2020	Velocidad Promedio [km/h]	Velocidad Promedio [m/s]	Factor K [m/s]	Factor de escala A [m/s]
Enero	10,5	2.92	0.39	3.17
Febrero	8,8	2.44	0.38	2.67
Marzo	8,9	2.47	0.43	2.75
Abril	8,8	2.44	0.41	2.72
Mayo	13	3.61	0.53	4.08
Junio	10,4	2.89	0.51	3.25
Julio	10	2.78	0.46	3.11
Agosto	11,2	3.11	0.44	3.47
Septiembre	10,8	3.00	0.48	3.36
Octubre	10	2.78	0.41	3.06
Noviembre	9,1	2.53	0.38	2.75
Diciembre	7,4	2.06	0.46	2.31
Anual 2020	9,8	2.72	0.42	3.00

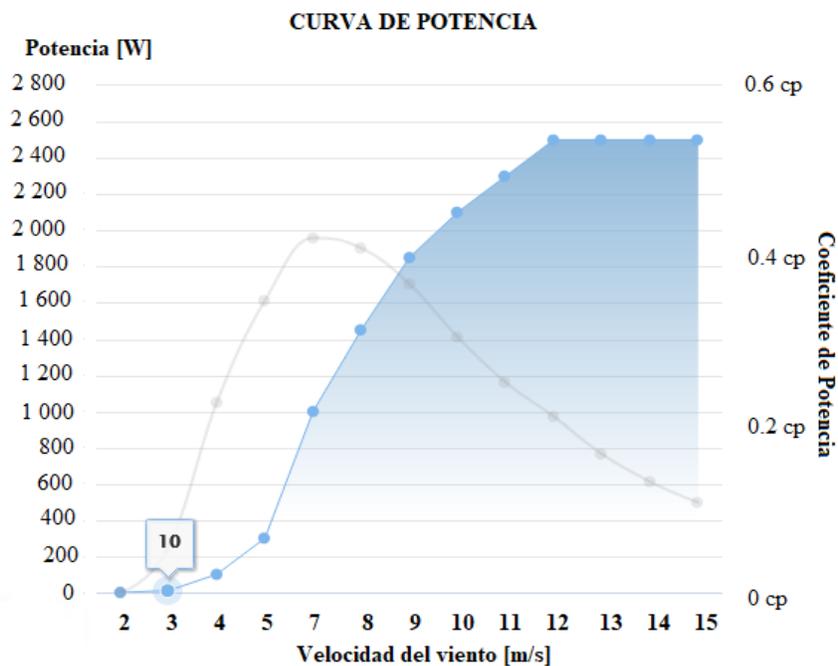
### 3.1.22 Selección del aerogenerador

Se selecciono el aerogenerador de mini eólica E30PRO basado a la Norma IEC 61400-2, está diseñado para cubrir la demanda energética de cualquier vivienda habitual. Su eficiencia de generación y diseño permite generar energía con muy bajo viento y a su vez soportar de forma eficiente la generación de fuertes vientos. Su producción se sitúa a partir de un viento de 2m/s, en relación a los demás aerogeneradores que cumplen características similares y de mayor producción sin embargo al utilizar un aerogenerador de mayor generación de potencia su costo es elevado y al no tener que cubrir una carga muy grande estará sobredimensionado, la ficha técnica del aerogenerador se adjunta en el anexo VII.



**Figura. 53.** Aerogenerador de minieólica E30PRO

Para realizar un cálculo de producción eólico, es necesario considerar múltiples factores como los valores Weibull, la densidad del aire, el factor de rugosidad del terreno, la temperatura, etc dada la complejidad de los cálculos recomendamos utilizar esta sencilla herramienta, de acuerdo a los valores representados según la IEC-61400-2[48], para determinar el valor de la potencia se tomó como referencia el valor promedio de la velocidad del viento  $2.72 \text{ m/s} \approx 3 \text{ m/s}$  basándose en la figura 54 de la curva de potencia entregada por el distribuidor se obtendrá una potencia de 10 W



**Figura. 54.** Curva de potencia del aerogenerador E30Pro

### 3.1.23 Potencial eólico

Para determinar el potencial eólico en función de la velocidad promedio obtenida por mes se generó en función del aerogenerador E30PRO, con los parámetros de viento obtenidos se aplicó para el modelo del aerogenerador de eje horizontal con un diámetro de rueda eólica de 3.8 m las características se adjunta en el anexo VII. El aerogenerador consta de una entrada de velocidad del viento de arranque de 2 m/s, su producción se sitúa en más de 30kWh/día, con un viento promedio entre 8 y 12m/s, su velocidad de supervivencia a los 60 m/s el mismo que me permite con el cálculo obtener la potencia disponible en diferentes meses, además se aplica la ley de Betz en la que menciona que la potencia obtenida se debe aplicar un coeficiente de Betz del 59% con el que se determinó el potencial eólico extraíble promedio al año de un aerogenerador de 51.53 W.

**Tabla 17.** Potencial eólico extraída por mes del 2020

Mes 2020	Velocidad Promedio [m/s]	Potencia Disponible [W]	Potencial eólico extraída [W]
Enero	2.92	107.42	63.38
Febrero	2.44	63.24	37.31
Marzo	2.47	65.42	38.60
Abril	2.44	63.24	37.31
Mayo	3.61	203.87	120.29
Junio	2.89	104.38	61.59
Julio	2.78	92.80	54.75
Agosto	3.11	130.37	76.92
Septiembre	3.00	116.90	68.97
Octubre	2.78	92.80	54.75
Noviembre	2.53	69.93	41.26
Diciembre	2.06	37.60	22.19
<b>Año 2020</b>	<b>2.72</b>	<b>87.34</b>	<b>51.53</b>

### 3.1.24 Energía obtenida del aerogenerador durante un año

Se determinó la energía que se obtendrá con la velocidad del viento promedio de cada mes del año 2020, la velocidad del viento media anual de 2.72 m/s, obteniendo un total 522 kWh/año, considerando el aerogenerador seleccionado E30PRO.

**Tabla 18.** Energía generada por mes y año del 2020

Datos	Velocidad Promedio [m/s]	Energía generada mes [kWh/ mes]
Enero	2.92	57
Febrero	2.44	28
Marzo	2.47	30
Abril	2.44	28
Mayo	3.61	118
Junio	2.89	55
Julio	2.78	47
Agosto	3.11	72
Septiembre	3.00	63
Octubre	2.78	47
Noviembre	2.53	33
Diciembre	2.06	13
<b>Energía / Anual 2020</b>	<b>2.72</b>	<b>522</b>

En la figura 55 se presenta la producción mensual que se tendrá en el año 2020, en el mes de mayo se generara una energía de 118 kWh siendo la de mayor producción y la de menor el mes de diciembre de 13 kWh, cabe mencionar que cada mes tiene un comportamiento diferente que compensan a meses que tienen menor producción debido a las horas útiles que presenta cada mes, logrando obtener de manera anual una energía de 522 kWh.

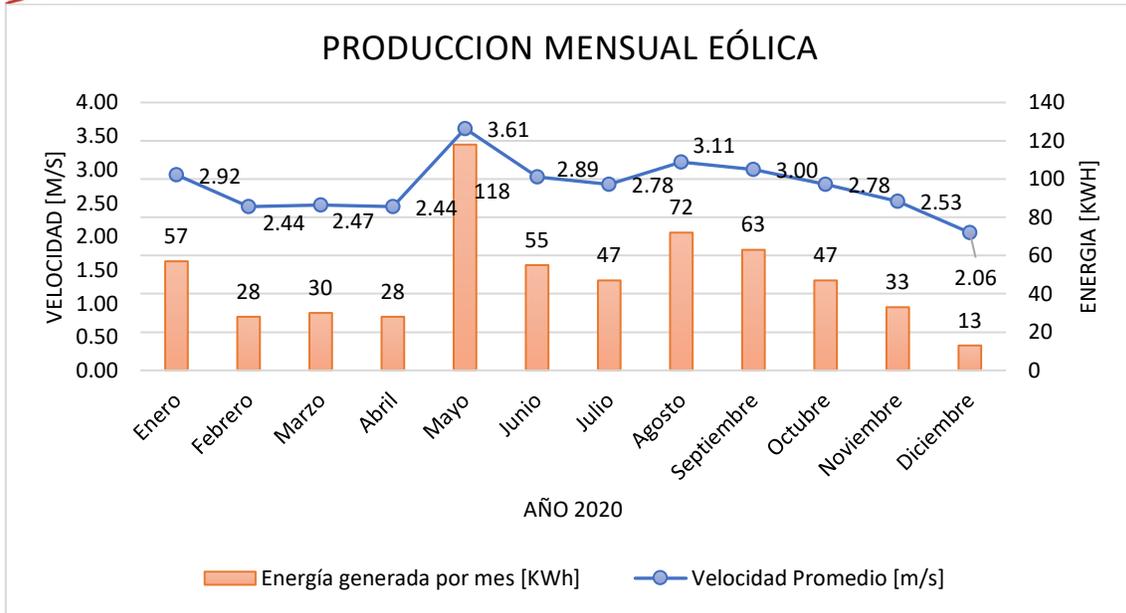


Figura. 55. Producción de energía [kWh] durante los meses del año 2020

### 3.1.25 Factor de planta del aerogenerador E30PRO

Para determinar el factor de planta se consideró los valores obtenidos de la potencia extraíble por año y en función de las 2224 horas útiles que se encuentra a partir de la velocidad del aerogenerador con lo cual se obtuvo el factor de planta de 0.253.

$$fp = \frac{MW * Tiempo.util}{MW * Tiempo.total}$$

$$fp = \frac{0.000052 MW * 2224 horas}{0.000052 MW * 8784 horas}$$

$$fp = 0,253$$

### 3.2 Conclusiones generales

- Mediante citas bibliográficas se identificó que existen diferentes métodos de redes neuronales y métodos estadísticos los cuales son utilizadas dependiendo el proceso de información que se vaya a realizar, para el objeto de estudio se utilizó la redes neuronales recurrentes que sirven para analizar datos históricos y tiene un buen proceso de optimización de resultados referente a las predicciones además se investigo cuales son los parámetros que influyen directamente en la predicción de la velocidad del viento siendo la presión barométrica, temperatura y humedad son parámetros más comunes para realizar los análisis en el cual se realizan algoritmo basados en series temporales para una predicción.
- Como primer paso se realizó la depuración de los datos completando valores nulos que se presenta en la data origina con ello se realizó la correlación de las variables velocidad del viento, presión barométrica, temperatura y humedad en la que se determinó mediante el mapa de calor las variables más representativas, procediendo con la selección del año, mes, hora y velocidad del viento, se encuentran correlacionadas a diferencia de las demás variables que presentaron valores pobres de aportación, por tal motivo la variable de estudio seleccionada fue la velocidad de viento, dichos parámetros permitieron ordenar de mejor manera la data para proceder con entrenamiento del algoritmo.
- Para el diseño del algoritmo se utilizó el análisis de series temporales usando Red LSTM es un tipo de Red Neuronal Recurrente que permite analizar secuencias o datos histórico con una memoria de largo plazo, además se utilizó diferentes optimizadores que permitieron obtener menor error siendo el RMSPROP el que presenta mayor optimización de precisión con los datos reales, con 25 neuronas y 25 épocas, de los datos totales de 17544 que representa a los dos años, para su entrenamiento se tomó el 80% que equivale a 14036 datos, para el test de predicción se utilizó el 20% restante, para su validación se aplicó dos métodos de regresión lineal obteniendo un

porcentual absoluto medio (MAPE) de 13% y la predicción por la red neuronal un error porcentual absoluto medio (MAPE) de 4% con esto se determinó que el porcentaje de predicción se encuentra dentro del rango de 5% establecido en base a estudios realizados de predicción.

- Con la predicción obtenida de la velocidad del viento se aplicó la distribución de Weibull para estimar el potencial eólico en la cual la data fue procesada para determinar la velocidad promedio por mes y año del 2020, las horas útiles para determinar la energía que se obtendrá, en base a los resultados se seleccionó el aerogenerador E30PRO basado a la Norma IEC 61400-2 que genera a partir de una velocidad de 2 m/s, obtenido una energía de 118 kWh siendo la de mayor producción en el mes de mayo y la de menor el mes de diciembre de 13 kWh, obteniendo de manera anual una energía de 522 kWh/año, determinado el factor de planta de 0.253 en base a las 2224 horas útiles que se tendrá durante el año.

### 3.3 Recomendaciones

- Existen varios métodos para realizar el análisis de datos por lo cual se recomienda utilizar el método de series temporales usando Red LSTM es un tipo de Red Neuronal Recurrente para este caso de predicción debió a que no todos los métodos se dedican obtener unos óptimos resultados de predicción, en el caso de los métodos estadísticos realizan la predicción, pero los resultados de error son muy elevados.
- Se recomienda realizar una correlación para determinar los parámetros a utilizar en el algoritmo con los cuales sean datos más relevantes y se pueda aplicar para el uso y entrenamiento del algoritmo además realizar la gráfica de mapa de calor que es una manera más visible para identificar con exactitud los valores de mayor y menor correlación.

- Se recomienda, para la obtención de la velocidad del viento la data descargada como primer paso debe ser analizada verificando valores nulos que se presenten, con el preprocesamiento de la data, con la información depurada se procede a ingresar en el algoritmo para su entrenamiento, cabe mencionar que entre más data sea recolectada para el entrenamiento del algoritmo mayor será su precisión de predicción.
- Se recomienda estimar el potencial eólico en base a datos que se recolectan en los lugares de estudio debido a que al realizar el proceso de recolección de datos cada sitio presentan diferentes condiciones geográficas esto influye directamente para la estimación del potencial, además la energía se puede determinar de manera directa en función de la curva de potencia del aerogenerador, cabe mencionar que el valor estimado son utilizados para condiciones de manera general por tal motivo se recomienda considerar para el cálculo los parámetros y condiciones geográficas reales del lugar.

### 3.4 Referencias bibliográficas

- [1] J. R. V. Barraza, H. Rodríguez R., V. González Huitrón, C. A. Lara Alvarez, y J. J. Flores, «Pronóstico de la velocidad del viento mediante técnicas de Inteligencia Artificial», *RCS*, vol. 148, n.º 7, pp. 181-195, dic. 2019, doi: 10.13053/rcs-148-7-14.
- [2] P. F. Bolaños-Logroño, J. L. Yaulema-Castañeda, O. D. Gavilánez-Álvarez, y B. L. Ayavaca-Vallejo, «Redes neuronales en la predicción de la velocidad del viento Neural networks in predicting wind speed Redes neurais na previsão da velocidade do vento», vol. 5, n.º 05, p. 37, 2020.
- [3] B. M. Santana y C. G. Artal, «ESTUDIOS DE PREDICCIÓN EN SERIES TEMPORALES DE DATOS METEOROLÓGICOS UTILIZANDO REDES NEURONALES RECURRENTES», p. 50.
- [4] C. V. Carrasco, «Aplicación de Redes Recurrentes para la predicción de potencia en plantas eólicas Chilenas», p. 65.
- [5] M. E. González-Ávila y L. F. Beltrán-Morales, «POTENCIAL DE APROVECHAMIENTO DE LA ENERGÍA EÓLICA PARA LA GENERACIÓN DE ENERGÍA ELÉCTRICA EN ZONAS RURALES DE MÉXICO», vol. 31, p. 7, 2006.
- [6] P. Díaz Cuevas, M. F. Pita López, A. Fernández Tabales, y N. Limones Rodríguez, «Energía eólica y territorio en Andalucía: diseño y aplicación de un modelo de potencialidad para la implantación de parques eólicos», *Ingeo*, n.º 67, p. 9, jun. 2017, doi: 10.14198/INGEO2017.67.01.
- [7] Q. Hernández-Escobedo, «EVALUACIÓN DEL POTENCIAL EÓLICO PARA LA GENERACIÓN DE ENERGÍA ELÉCTRICA EN EL ESTADO DE VERACRUZ, MÉXICO», p. 10.
- [8] A. A. Legrá-Lobaina y E. Terrero-Matos, «Modelación de variables eólicas mediante estimadores (A,U,Θ)», p. 16.
- [9] C. Muñoz *et al.*, «Predicción de velocidades y potencial eólico para alturas superiores: estudio en Mérida, Yucatán, México», p. 7, 2015.
- [10] R. P. Diez, J. P. Fernández, P. Priore, y A. G. Gómez, «Pronóstico de la Velocidad y Dirección del Viento mediante Redes Neuronales Artificiales», p. 10.

- [11] U. M. R. Alcocer, E. Tello-Leal, y A. B. R. Alvarado, «MODELO BASADO EN REDES NEURONALES RECURRENTE LSTM PARA LA PREDICCIÓN DE LA SIGUIENTE ACTIVIDAD EN PROCESOS DE NEGOCIO», vol. 40, n.º 130, p. 13, 2018.
- [12] J. A. R. Martínez y C. G. A. Alvarez, «DISEÑO DE UN CONTROLADOR ÓPTIMO USANDO REDES NEURONALES DINÁMICAS PARA UN SISTEMA DE GENERACIÓN EÓLICA», p. 61, 2015.
- [13] C. Sandoval-Ruiz, «MODELO NEURO-ADAPTATIVO EN VHDL, BASADO EN CIRCUITOS NLFSR, PARA CONTROL DE UN SISTEMA INTELIGENTE DE TECNOLOGÍA SOSTENIBLE», vol. 21, p. 10, 2017.
- [14] I. L. M. Faiella y I. A. J. Gesino, «GESTIÓN DE VARIABLES METEOROLÓGICAS Y MAPEO EÓLICO», p. 7.
- [15] C. Ventura, Y. Guadalupe, I. de Jesús, T. Loeza, M. Martínez, y R. Salvador, «ANÁLISIS COMPARATIVO DE AEROGENERADORES UTILIZANDO DATOS DE VIENTO: CASO DE ESTUDIO», vol. 32, p. 7.
- [16] J. BAPTISTA DA SILVA y J. ANDRIONI, «ESTIMATIVA DO POTENCIAL EÓLICO DE TORRES, RS, USANDO PROBABILIDADES PENTADAIS», dic. 2019, doi: 10.5281/ZENODO.3592977.
- [17] F. León-Vargas, E. Krejci, y M. García-Jaramillo, «Rentabilidad prevista de un aerogenerador de minieólica en cuatro ciudades de Colombia acorde a los incentivos de Ley 1715», p. 11, 2013.
- [18] «Estudo Preliminar sobre o Potencial Eólico das Regiões Serranas do Ceará», p. 6.
- [19] «Trabajo de grado.pdf». Accedido: 21 de octubre de 2021. [En línea]. Disponible en: <https://expeditiorepositorio.utadeo.edu.co/bitstream/handle/20.500.12010/13673/Trabajo%20de%20grado.pdf?sequence=1&isAllowed=y>
- [20] J. de Lucio, «Estimación adelantada del crecimiento regional mediante redes neuronales LSTM», p. 20.
- [21] G. I. G. Gazabón, «Modelo de Machine Learning para la Clasificación de pacientes en términos del nivel», p. 103.

- [22] «Understanding LSTM Networks -- colah's blog». <http://colah.github.io/posts/2015-08-Understanding-LSTMs/> (accedido 11 de marzo de 2022).
- [23] N. Gnoza y M. Barberena, «Estudio de factibilidad del uso de Machine Learning con múltiples fuentes de datos en el pronóstico del tiempo.», p. 191.
- [24] G. I. G. Gazabón, «Modelo de Machine Learning para la Clasificación de pacientes en términos del nivel», p. 103.
- [25] F. E. Checa y E. E. Rosero, «Metodología para la determinación de características del viento y evaluación del potencial de energía eólica en Túquerres-Nariño», *Revista Científica*, vol. 31, n.º 1, Art. n.º 1, ene. 2018, doi: 10.14483/23448350.12304.
- [26] D. Borbor-Murillo, G. Fajardo Jácome, L. Sánchez-Holguín, y L. Cevallos-Torres, «Modelo de Simulación de Inventario basado en Algoritmo Optimización por Enjambre de Partículas (PSO) para minimizar costos por venta de productos perecederos», *esj*, vol. 4, n.º 1, pp. 37-44, mar. 2020, doi: 10.46480/esj.4.1.40.
- [27] por, «¿Qué es una red neuronal Feedforward? - Spiegato», 25 de julio de 2021. <https://spiegato.com/es/que-es-una-red-neuronal-feedforward> (accedido 10 de abril de 2022).
- [28] «t1892mma.pdf». Accedido: 10 de abril de 2022. [En línea]. Disponible en: <https://repositorio.uta.edu.ec/bitstream/123456789/33778/1/t1892mma.pdf>
- [29] «Convolutional Neural Networks: La Teoría explicada en Español | Aprende Machine Learning». <https://www.aprendemachinelearning.com/como-funcionan-las-convolutional-neural-networks-vision-por-ordenador/> (accedido 10 de abril de 2022).
- [30] «UBA - Redes Neuronales - Redes Feedforward by Alex Hernández - Issuu». [https://issuu.com/alexhernandez38/docs/redes\\_neuronales\\_revista](https://issuu.com/alexhernandez38/docs/redes_neuronales_revista) (accedido 15 de abril de 2022).
- [31] «RNN y LSTM - programador clic». <https://programmerclick.com/article/11911637462/> (accedido 15 de abril de 2022).
- [32] «Ventajas de las redes neuronales convolucionales - programador clic». <https://programmerclick.com/article/7174589545/> (accedido 15 de abril de 2022).

- [33] «Ventajas - Ventajas y desventajas del método Montecarlo». <https://1library.co/article/ventajas-ventajas-y-desventajas-del-m%C3%A9todo-montecarlo.q05evovy> (accedido 15 de abril de 2022).
- [34] S. Organizaci??n Meteorol??gica Mundial (Ginebra, *Gu??a del sistema mundial de observaci??n*. Ginebra: Secretar??a de la Organizaci??n Meteorol??gica Mundial, 1977.
- [35] «Cómo instalar correctamente su estación meteorológica profesional», *Raig*. <https://www.raig.com/blog/como-instalar-correctamente-su-estacion-meteorologica-profesional-58/> (accedido 13 de abril de 2022).
- [36] «RTCA-1146.pdf». Accedido: 9 de agosto de 2021. [En línea]. Disponible en: <https://rtca.unca.edu.mx/articles/RTCA-1146.pdf>
- [37] «previsión del tiempo Europa - woespana». <https://www.woespana.es/weather/maps/city> (accedido 5 de mayo de 2022).
- [38] «Machine Learning con Python y Scikitlearn». [https://www.cienciadedatos.net/documentos/py06\\_machine\\_learning\\_python\\_scikitlearn.html](https://www.cienciadedatos.net/documentos/py06_machine_learning_python_scikitlearn.html) (accedido 9 de agosto de 2021).
- [39] A. Regal *et al.*, «Proyección del precio de criptomonedas basado en Tweets empleando LSTM», *Ingeniare. Rev. chil. ing.*, vol. 27, n.º 4, pp. 696-706, dic. 2019, doi: 10.4067/S0718-33052019000400696.
- [40] «¿Qué son las Redes LSTM?», 20 de julio de 2019. <https://www.codificandobits.com/blog/redes- lstm/> (accedido 10 de julio de 2021).
- [41] L. Velasco, «Optimizadores en redes neuronales profundas: un enfoque práctico», *Medium*, 26 de abril de 2020. <https://velascoluis.medium.com/optimizadores-en-redes-neuronales-profundas-un-enfoque-pr%C3%A1ctico-819b39a3eb5> (accedido 30 de abril de 2022).
- [42] «Interpretar todos los estadísticos y gráficas para Promedio móvil». <https://support.minitab.com/es-mx/minitab/18/help-and-how-to/modeling-statistics/time-series/how-to/moving-average/interpret-the-results/all-statistics-and-graphs/> (accedido 12 de abril de 2022).
- [43] «13-Articulovientofinal NO.pdf».

- [44] M. A. García y L. B. M. López, «Estudio del Potencial Eólico y Viabilidad de Parque Eólico en Andalucía.», p. 131.
- [45] J. G. E. Ramos, «Estimación del viento - cálculo del potencial eólico del distrito de Salaverri mediante el uso de software computacionales y la estadística», p. 25.
- [46] «MANUAL\_E30PROX2\_es.pdf». Accedido: 17 de abril de 2022. [En línea]. Disponible en: [https://www.enair.es/descargas/Manual/MANUAL\\_E30PROX2\\_es.pdf](https://www.enair.es/descargas/Manual/MANUAL_E30PROX2_es.pdf)
- [47] «Rosa de los vientos», p. 5, 2014.
- [48] «Aerogenerador E30PRO - La última tecnología». <https://www.enair.es/es/aerogeneradores/e30pro> (accedido 16 de abril de 2022).
- [49] «Aerogenerador E70PRO - La última tecnología». <https://www.enair.es/es/aerogeneradores/e70pro> (accedido 16 de abril de 2022).
- [50] «Aerogenerador E200L - La última tecnología». <https://www.enair.es/es/aerogeneradores/e200l> (accedido 16 de abril de 2022).
- [51] D. F. Manotas Duque, «Evaluación de proyectos de generación eléctrica bajo incertidumbre en política climática», *Entramado*, vol. 9, n.º 1, pp. 102-117, ene. 2013.
- [52] ekuatio, «Error absoluto y error relativo: Qué son y cómo se calculan», *Clases de Matemáticas Online*, 25 de octubre de 2018. <https://ekuatío.com/error-absolutos-y-error-relativos-que-son-y-como-se-calculan/> (accedido 26 de abril de 2022).
- [53] M. S. R. Herrera y M. E. M. Villegas, «MANUAL PARA EL MANEJO DE SOFTWARE ESPECIALIZADO», p. 30.



ANEXOS I. Base de datos original

Time	Temp Avg		Temp High		Temp Low		Temp (Day) High		Temp (Day) Low		Hum Avg		Hum High		Hum Low		Dew Point		Windspeed		Wind Direction		Gust Direction		Gust Max	
	Temp	Avg	High	Low	Temp	Avg	High	Low	Temp	Avg	High	Low	Temp	Avg	High	Low	Temp	Avg	High	Low	Temp	Direction	Temp	Direction	Temp	Direction
2019-01-01 00:00:00	7.5	7.5	7.5	7.5	7.3	16.7	91.0	92.0	92.0	6.1	47.0	96.0	720.0	719.9	719.9	717.1	720.8	1.9	67.0	3.5	270.0	48.4				
2019-01-01 00:01:00	7.5	7.5	7.5	7.5	7.5	7.5	92.0	92.0	92.0	6.3	92.0	92.0	719.9	719.9	719.9	719.9	719.9	5.3	67.0	7.9	67.0	7.9				
2019-01-01 00:02:00	7.5	7.5	7.5	7.5	7.5	7.5	91.0	92.0	92.0	6.1	92.0	92.0	719.9	719.9	719.9	719.9	719.9	6.6	67.0	7.9	67.0	7.9				
2019-01-01 00:03:00	7.5	7.5	7.5	7.5	7.5	7.5	92.0	92.0	92.0	6.3	92.0	92.0	719.8	719.9	719.7	719.7	719.9	4.0	67.0	5.6	67.0	7.9				
2019-01-01 00:04:00	7.5	7.5	7.5	7.5	7.5	7.5	91.0	92.0	92.0	6.1	92.0	92.0	719.8	719.9	719.7	719.7	719.9	3.2	67.0	5.1	67.0	7.9				
2019-01-01 00:05:00	7.4	7.4	7.5	7.4	7.4	7.5	92.0	92.0	92.0	6.2	92.0	92.0	719.9	719.9	719.7	719.7	719.9	3.5	54.0	5.1	67.0	7.9				
2019-01-01 00:06:00	7.4	7.4	7.4	7.4	7.4	7.5	92.0	92.0	92.0	6.2	92.0	92.0	719.8	719.9	719.7	719.7	719.9	3.7	45.0	4.8	67.0	7.9				
2019-01-01 00:07:00	7.4	7.4	7.4	7.4	7.4	7.5	91.0	92.0	92.0	6.0	92.0	92.0	719.7	719.7	719.7	719.9	719.9	1.4	45.0	3.4	67.0	7.9				
2019-01-01 00:08:00	7.4	7.4	7.4	7.4	7.4	7.5	91.0	92.0	92.0	6.0	92.0	92.0	719.7	719.9	719.7	719.7	719.9	2.6	45.0	4.2	67.0	7.9				
2019-01-01 00:09:00	7.4	7.4	7.4	7.4	7.4	7.5	92.0	92.0	92.0	6.2	92.0	92.0	719.8	719.9	719.7	719.7	719.9	1.3	45.0	2.9	67.0	7.9				
2019-01-01 00:10:00	7.4	7.3	7.4	7.4	7.4	7.5	91.0	92.0	92.0	6.0	92.0	92.0	719.8	719.9	719.7	719.7	719.9	0.0	45.0	2.1	67.0	7.9				
2019-01-01 00:11:00	7.3	7.3	7.3	7.3	7.3	7.5	92.0	92.0	92.0	6.1	92.0	92.0	719.8	719.9	719.7	719.7	719.9	0.0	45.0	0.0	67.0	7.9				
2019-01-01	7.3	7.3	7.3	7.3	7.3	7.5	91.0	92.0	92.0	5.9	92.0	92.0	719.7	719.7	719.7	719.9	719.9	0.0	45.0	0.0	67.0	7.9				



**ANEXO II. Base de datos depurados**

Time	Windspeed	Baro Avg	Hum Avg	Temp Avg
2019-12-01	11.383056	719.934653	87.461806	12.439444
2019-12-02	13.950486	720.538333	84.055556	12.542222
2019-12-03	7.417918	720.311689	74.656531	13.210033
2019-12-04	6.665061	719.907501	80.444239	13.072255
2019-12-05	6.182550	720.427103	81.037029	12.565164
2019-12-06	5.279671	721.154885	85.353202	11.783835
2019-12-07	7.033044	720.378584	80.982903	12.740679
2019-12-08	7.177009	719.457675	80.797941	13.266087
2019-12-09	7.018463	719.499240	78.855307	13.262969
2019-12-10	7.177009	719.457675	80.797941	13.266087
2019-12-11	8.271658	719.620559	77.383084	13.204983
2019-12-12	7.780417	719.113958	77.941667	12.983542
2019-12-13	9.575278	718.719306	70.475694	12.529306
2019-12-14	6.745423	718.639637	72.684764	13.797706
2019-12-15	9.692153	719.387292	83.446528	12.504583
2019-12-16	11.179994	719.043149	75.027241	14.174986
2019-12-17	8.676237	718.250253	70.668837	13.740632
2019-12-18	6.601777	718.540289	72.526507	13.824155
2019-12-19	9.692153	719.387292	83.446528	12.504583
2019-12-20	10.496323	718.957771	81.071185	12.988726
2019-12-21	10.755208	719.455903	82.502083	12.877639



### ANEXO III. Algoritmo desarrollado

```
import numpy as np # Realiza ecuaciones matemáticas
import pandas as pd # Lectura y procesamiento del Dataframe
import matplotlib.pyplot as plt # Permite realizar graficas
from sklearn.preprocessing import LabelEncoder # Codifica valores
from sklearn.model_selection import train_test_split # Escalar valores
import keras # Directamente todo las librerías de keras y tensorflow funcionan para crear la red LSTM
from keras.models import Sequential # Usar datos secuenciales
from keras.layers import Dense, Activation, Dropout # Procesamiento de varias capas
from sklearn.metrics import mean_squared_error # Pérdida de regresión del error cuadrático medio.
import datetime # Trabaja con fechas
```

```
#Lectura de datos
data = pd.read_csv('DEPURACION_FINAL.csv')
data
```

	AÑO	MES	DIA	FECHA	HORA	VELOCIDAD DEL VIENTO
0	2019	1	martes	1	0	7
1	2019	1	martes	1	1	12
2	2019	1	martes	1	2	11
3	2019	1	martes	1	3	9
4	2019	1	martes	1	4	7
...	...	...	...	...	...	...
17539	2020	12	jueves	31	19	7
17540	2020	12	jueves	31	20	3
17541	2020	12	jueves	31	21	2
17542	2020	12	jueves	31	22	0
17543	2020	12	jueves	31	23	2

17544 rows × 6 columns

```
#Verificación de nans
data.isnull().values.any()
data.isnull().sum().sum()
```

```
data.isnull().sum()
```

```
AÑO          0
MES          0
DIA          0
FECHA        0
HORA         0
VELOCIDAD DEL VIENTO  0
dtype: int64
```

```
#Verificacion de nans
data.isnull().values.any()
data.isnull().sum().sum()

data.isnull().sum()
```

```
AÑO          0
MES          0
DIA          0
FECHA        0
HORA         0
VELOCIDAD DEL VIENTO  0
dtype: int64
```

```
data.tail(17544)
#data.to_csv('data_total.csv')
```

	AÑO	MES	DIA	FECHA	HORA	VELOCIDAD DEL VIENTO
0	2019	1	2	1	0	7
1	2019	1	2	1	1	12
2	2019	1	2	1	2	11
3	2019	1	2	1	3	9
4	2019	1	2	1	4	7
...	...	...	...	...	...	...
17539	2020	12	4	31	19	7
17540	2020	12	4	31	20	3
17541	2020	12	4	31	21	2
17542	2020	12	4	31	22	0
17543	2020	12	4	31	23	2

```
#Llevamos la hora y la fecha para los indices porque lo que nos interesa es el VIENTO
from datetime import datetime
data_final = pd.DataFrame()
fechas_dt = []
viento_dt = []
for registro in data.values:
    fecha = "{}-{}-{} {}".format(registro[0], registro[1], registro[3], registro[4])
    fecha_dt = datetime.strptime(fecha, '%Y-%m-%d %H')
    fechas_dt.append(fecha_dt)
    viento_dt.append(registro[5])
#Creamos un nuevo dataframe con el formato de fecha correcto y el valor del VIENTO
data_final['FECHA'] = fechas_dt
data_final['VELOCIDAD DEL VIENTO'] = viento_dt
data_final
```



	FECHA	VELOCIDAD DEL VIENTO
0	2019-01-01 00:00:00	7
1	2019-01-01 01:00:00	12
2	2019-01-01 02:00:00	11
3	2019-01-01 03:00:00	9
4	2019-01-01 04:00:00	7
...	...	...
17539	2020-12-31 19:00:00	7
17540	2020-12-31 20:00:00	3
17541	2020-12-31 21:00:00	2
17542	2020-12-31 22:00:00	0
17543	2020-12-31 23:00:00	2

17544 rows × 2 columns

```
#Dividir los datos de entrenamiento y de test  
# No se puede utilizar train test split, debemos dividir en el ultimo segmento  
#Tomamos los datos para validacion el 20%, pero del final de los datos (todo 2020 menos enero)  
size = int(0.2*data_final.shape[0])  
data_test = data_final[-size:]  
data_train = data_final[:-size]  
data_train
```

	FECHA	VELOCIDAD DEL VIENTO
	2019-01-01 00:00:00	7
	2019-01-01 01:00:00	12
	2019-01-01 02:00:00	11
	2019-01-01 03:00:00	9
	2019-01-01 04:00:00	7
	...	...
	2020-08-07 15:00:00	33
	2020-08-07 16:00:00	29
	2020-08-07 17:00:00	25
	2020-08-07 18:00:00	20
	2020-08-07 19:00:00	18

14036 rows × 1 columns

```
#Normalizacion de Los datos
#Para LSTM es obligatoria La normalización, en este caso lo hacemos de 0 a 1
from sklearn.preprocessing import MinMaxScaler
sc = MinMaxScaler(feature_range = (0,1))
data_train_scaled = sc.fit_transform(data_train)
print(data_train_scaled.shape)
data_train_scaled
```

```
array([[0.1372549 ],
       [0.23529412],
       [0.21568627],
       ...,
       [0.49019608],
       [0.39215686],
       [0.35294118]])
```

```
#Tenemos que entrenar la red por bloques de datos
#Time_step es el tamaño del bloque
#Cada n datos genera una salida
#Tomamos bloques de 60 datos consecutivos
time_step = 60
X_train = [] #Lista de conjuntos de 60 datos
Y_train = []
m = len(data_train_scaled)
for i in range(time_step, m):
    X_train.append(data_train_scaled[i-time_step:i, 0]) #Guardamos 60 valores
    Y_train.append(data_train_scaled[i, 0]) #Buscamos predecir el dato time_step + 1, no la var VIENTO
X_train, Y_train = np.array(X_train), np.array(Y_train) #Usamos np por reshape y optimización
```

```
X_train.shape, Y_train.shape
```

```
((13976, 60), (13976,))
```

```
#pd.DataFrame(X_train).to_csv('x_train.csv')
#pd.DataFrame(Y_train).to_csv('y_train.csv')
```

```
#Obligatorio el reshape de los datos, es obligatorio por Keras, nos pide ingresar un array de 3 dim
#Llevamos de dos a 3 dimensiones nuestro arreglo
X_train = np.reshape(X_train, (X_train.shape[0], X_train.shape[1], 1)) #Agregamos una nueva dimension
X_train.shape, Y_train.shape
```

## MODELO RNA LSTM

```
#MODELO
```

```
from keras.layers import LSTM
from sklearn.metrics import mean_squared_error
```

```
def NN_LSTM(optimizador, neuronas, epocas):
    dim_entrada = (X_train.shape[1], X_train.shape[2])
    dim_salida = 1
    num_neuronas = neuronas

    model = Sequential()

    model.add(LSTM(units=num_neuronas, return_sequences = True, input_shape=dim_entrada))
    model.add(Dropout(0.4))
    model.add(LSTM(units=num_neuronas))
    model.add(Dropout(0.2))
    model.add(Dense(units = dim_salida))
    # model.add(activation = 'linear')
    model.compile(optimizer= optimizador, loss='mse', metrics = ['mean_absolute_error']) #mean_squared_error
```

```
#Entrenamiento
history = model.fit(X_train, Y_train, epochs = epocas, batch_size = 128)
return model, history

def predDataTest(modelo):
    x_test = data_test.values
    x_test = sc.transform(x_test)
    # # x_test
    X_test = []
    for i in range(time_step, len(x_test)):
        X_test.append(x_test[i-time_step: i, 0])
    X_test = np.array(X_test)
    X_test = np.reshape(X_test, (X_test.shape[0], X_test.shape[1], 1))
    X_test.shape
    prediccion = modelo.predict(X_test)
    prediccion = sc.inverse_transform(prediccion)
    return prediccion
```

```
#rmsprop LOS MEJORES RESULTADOS
model, history = NN_LSTM(optimizador = 'rmsprop', neuronas = 25, epocas = 25)

Epoch 1/25
110/110 [=====] - 23s 122ms/step - loss: 0.0166 - mean_absolute_error: 0.1018
Epoch 2/25
110/110 [=====] - 13s 117ms/step - loss: 0.0077 - mean_absolute_error: 0.0686
Epoch 3/25
110/110 [=====] - 13s 118ms/step - loss: 0.0068 - mean_absolute_error: 0.0638
Epoch 4/25
110/110 [=====] - 13s 120ms/step - loss: 0.0062 - mean_absolute_error: 0.0611
Epoch 5/25
110/110 [=====] - 13s 119ms/step - loss: 0.0058 - mean_absolute_error: 0.0590
Epoch 6/25
110/110 [=====] - 12s 105ms/step - loss: 0.0055 - mean_absolute_error: 0.0574
Epoch 7/25
110/110 [=====] - 11s 97ms/step - loss: 0.0051 - mean_absolute_error: 0.0552
Epoch 8/25
110/110 [=====] - 12s 110ms/step - loss: 0.0049 - mean_absolute_error: 0.0541
Epoch 9/25
110/110 [=====] - 13s 114ms/step - loss: 0.0046 - mean_absolute_error: 0.0521
Epoch 10/25
110/110 [=====] - 13s 117ms/step - loss: 0.0045 - mean_absolute_error: 0.0511
Epoch 11/25
```

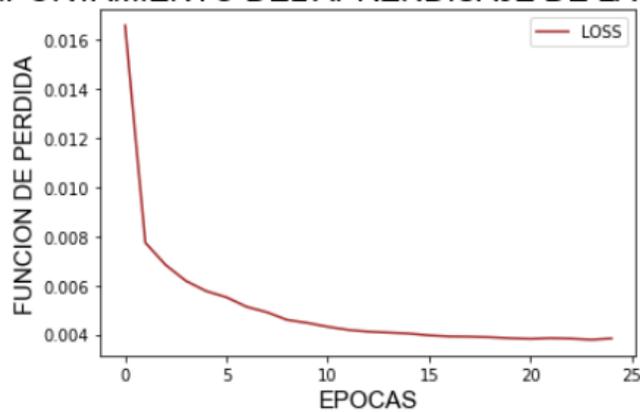
```
Epoch 15/25
110/110 [=====] - 13s 117ms/step - loss: 0.0040 - mean_absolute_error: 0.0487
Epoch 16/25
110/110 [=====] - 13s 117ms/step - loss: 0.0040 - mean_absolute_error: 0.0480
Epoch 17/25
110/110 [=====] - 13s 118ms/step - loss: 0.0039 - mean_absolute_error: 0.0477
Epoch 18/25
110/110 [=====] - 13s 117ms/step - loss: 0.0039 - mean_absolute_error: 0.0477
Epoch 19/25
110/110 [=====] - 13s 117ms/step - loss: 0.0039 - mean_absolute_error: 0.0477
Epoch 20/25
110/110 [=====] - 13s 118ms/step - loss: 0.0038 - mean_absolute_error: 0.0473
Epoch 21/25
110/110 [=====] - 13s 117ms/step - loss: 0.0038 - mean_absolute_error: 0.0474
Epoch 22/25
110/110 [=====] - 13s 118ms/step - loss: 0.0038 - mean_absolute_error: 0.0475
Epoch 23/25
110/110 [=====] - 13s 117ms/step - loss: 0.0038 - mean_absolute_error: 0.0473
Epoch 24/25
110/110 [=====] - 13s 116ms/step - loss: 0.0038 - mean_absolute_error: 0.0470
Epoch 25/25
110/110 [=====] - 13s 115ms/step - loss: 0.0038 - mean_absolute_error: 0.0473
```

```
print('El valor de la pérdida obtenido es: {}'.format(history.history['loss'][-1]))  
print('El valor del error medio absoluto es: {}'.format(history.history['mean_absolute_error'][-1]))
```

El valor de la pérdida obtenido es: 0.0038345782086253166  
El valor del error medio absoluto es: 0.04726637527346611

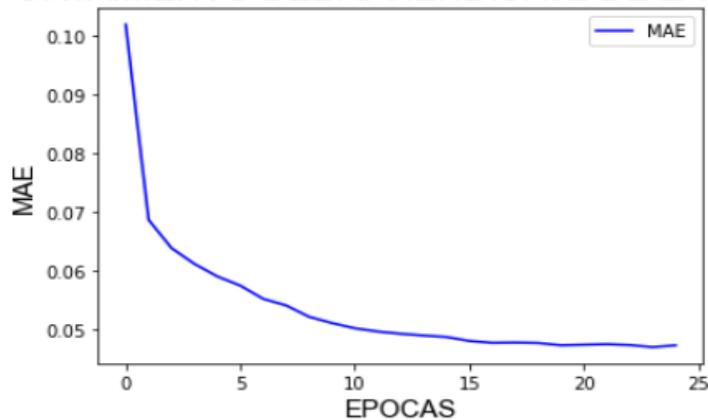
```
#plt.legend(shadow=True)  
plt.title('COMPORTAMIENTO DEL APRENDISAJE DE LA RED LSTM', family='Arial', fontsize=20)  
plt.xlabel('EPOCAS', family='Arial', fontsize=15) # Estilo eje X  
plt.ylabel('FUNCION DE PERDIDA', family='Arial', fontsize=15) # Estilo eje Y  
plt.plot(history.history['loss'], label='LOSS', color='brown')  
plt.legend()  
plt.show()
```

### COMPORTAMIENTO DEL APRENDISAJE DE LA RED LSTM



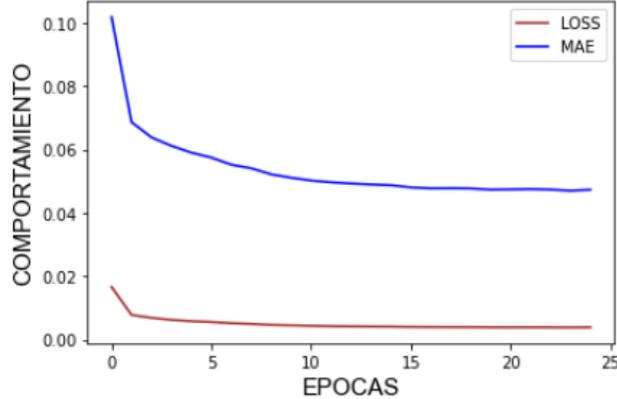
```
plt.title('COMPORTAMIENTO DEL APRENDISAJE DE LA RED LSTM', family='Arial', fontsize=20)  
plt.xlabel('EPOCAS', family='Arial', fontsize=15) # Estilo eje X  
plt.ylabel('MAE', family='Arial', fontsize=15) # Estilo eje Y  
plt.plot(history.history['mean_absolute_error'], label='MAE', color='blue')  
plt.legend()  
plt.show()
```

### COMPORTAMIENTO DEL APRENDISAJE DE LA RED LSTM



```
#plt.legend(shadow=True)
plt.title('COMPORTAMIENTO DEL APRENDISAJE DE LA RED LSTM', family='Arial', fontsize=20)
plt.xlabel('EPOCAS', family='Arial', fontsize=15) # Estilo eje X
plt.ylabel('COMPORTAMIENTO', family='Arial', fontsize=15) # Estilo eje Y
plt.plot(history.history['loss'], label='LOSS', color='brown')
plt.plot(history.history['mean_absolute_error'], label='MAE', color='blue')
plt.legend()
plt.show()
```

### COMPORTAMIENTO DEL APRENDISAJE DE LA RED LSTM



```
#Guardamos el modelo para no tener que entrenar otra vez para probar
model.save('modelos/model.h5')
```

```
#Ver la arquitectura del modelo
print(model.summary())
```

Model: "sequential"

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 60, 25)	2700
dropout (Dropout)	(None, 60, 25)	0
lstm_1 (LSTM)	(None, 25)	5100
dropout_1 (Dropout)	(None, 25)	0
dense (Dense)	(None, 1)	26

```
-----
Total params: 7,826
Trainable params: 7,826
Non-trainable params: 0
```

None

```

model = keras.models.load_model('modelos/model.h5')

prediccion = predDataTest(model)

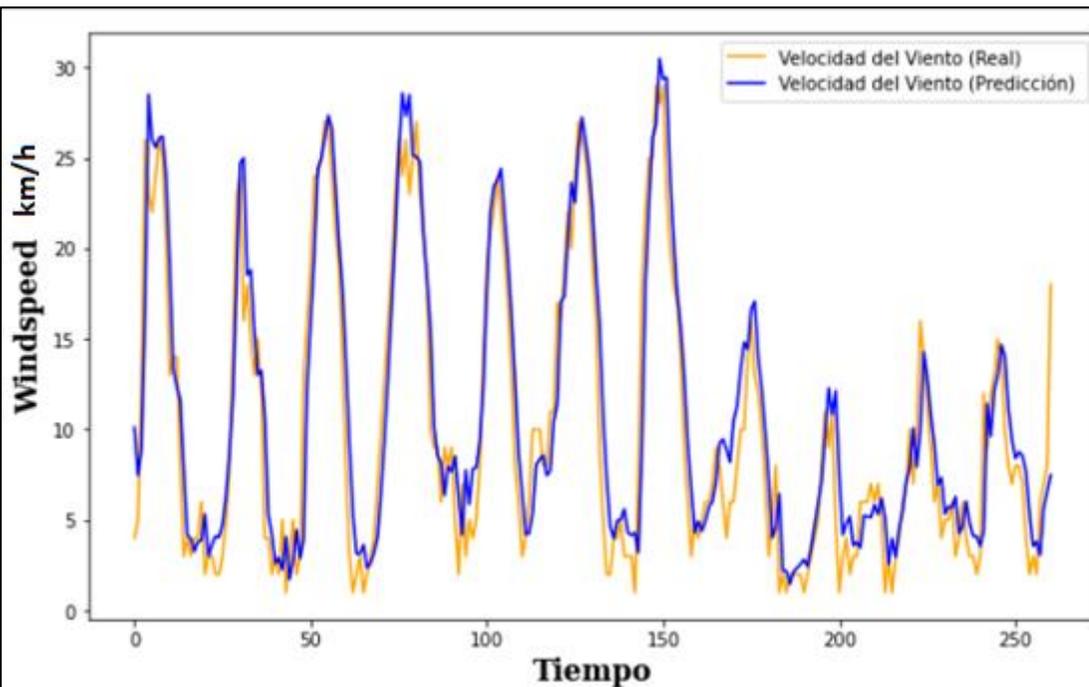
x_test = data_test.values
x_test = sc.transform(x_test)
## x_test
X_test = []
for i in range(time_step, len(x_test)):
    X_test.append(x_test[i-time_step: i, 0])
X_test = np.array(X_test)
X_test = np.reshape(X_test, (X_test.shape[0], X_test.shape[1], 1))
X_test.shape
prediccion = model.predict(X_test)
prediccion = sc.inverse_transform(prediccion)
len(prediccion)

```

```

#Grafica de valores
# Y_test #Aun no defnido
plt.figure(figsize=(10,6))
plt.plot(data_test.values[60:321], color='orange', label='Velocidad del Viento (Real)')
plt.plot(prediccion[:261], color='blue', label='Velocidad del Viento (Predicción)')
plt.xlabel('Tiempo',
           fontsize = 16,
           fontweight = 'bold',
           fontfamily='serif')
plt.ylabel('Windspeed',
           fontsize = 16,
           fontweight = 'bold',
           fontfamily='serif')
plt.legend()
plt.show()

```



```
#Union de datos
#LSTM nos predice el dato time_step + 1 por tanto debemos unir a partir del dato 61 de nuestro data
data_test_slic = data_test[60:].copy()
data_test_pred = pd.concat([data_test_slic,
                             pd.DataFrame(prediccion, |
                                           columns=['PREDICCIÓN'],
                                           index = data_test_slic.index)],
                             axis =1)
# print(data_test_slic.shape)
# print(prediccion.shape)
# print(data_test_slic)
# print(prediccion)
data_test_pred.head(17300)
```

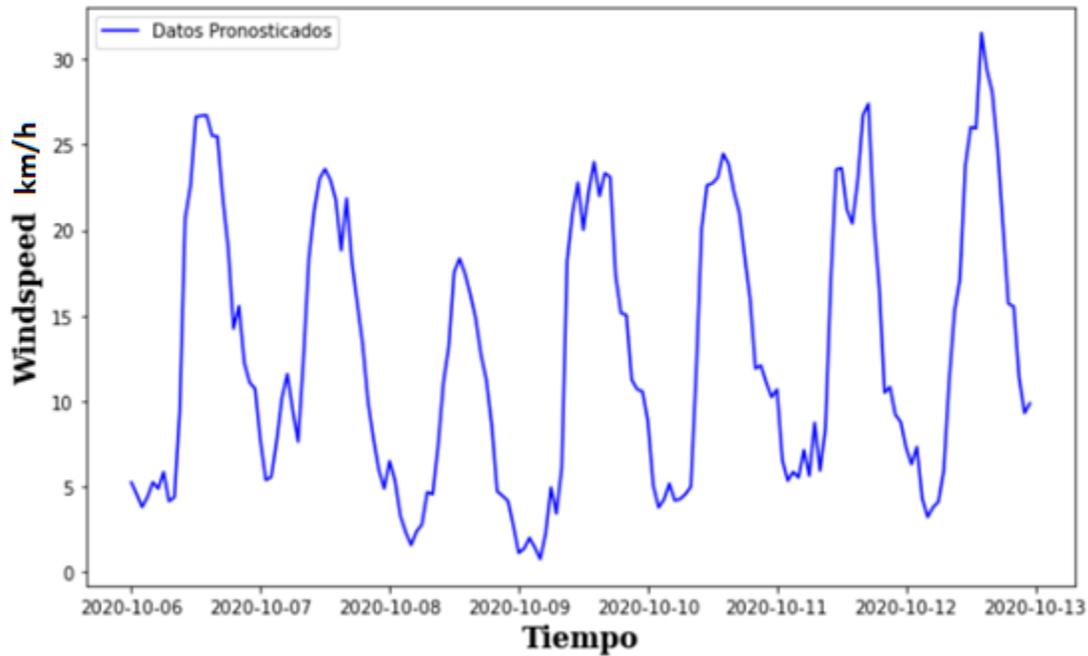
FECHA	VELOCIDAD DEL VIENTO	PREDICCIÓN
2020-08-10 08:00:00	4.0	10.107659
2020-08-10 09:00:00	5.0	7.471512
2020-08-10 10:00:00	13.0	8.977430
2020-08-10 11:00:00	26.0	16.511379
2020-08-10 12:00:00	23.0	28.506355
...	...	...
2020-12-31 19:00:00	7.0	6.847646
2020-12-31 20:00:00	3.0	6.744150
2020-12-31 21:00:00	2.0	3.976468
2020-12-31 22:00:00	0.0	2.860680
2020-12-31 23:00:00	2.0	1.290362

3448 rows × 2 columns

```
# Curva de La Velocidad del viento pronosticada
#Los primeros 8 días(completos) de La prediccion (06 de OCTUBRE)
semana_estudio_test_pred = data_test_pred[
    data_test_pred.index.get_loc('2020-10-06 00:00:00'):
    data_test_pred.index.get_loc('2020-10-13 00:00:00')]

semana_estudio_test = semana_estudio_test_pred['VELOCIDAD DEL VIENTO']
semana_estudio_pred = semana_estudio_test_pred['PREDICCIÓN']
# semana_estudio_pred
plt.figure(figsize=(10,6))
plt.plot(semana_estudio_pred,
         color='blue',
         label='Datos Pronosticados')
plt.suptitle('CURVA DEL VIENTO PRONOSTICADO',
            fontsize = 18,
            fontweight = 'bold',
            fontfamily='serif')
plt.xlabel('Tiempo',
          fontsize = 16,
          fontweight = 'bold',
          fontfamily='serif')
plt.ylabel('Windspeed',
          fontsize = 16,
          fontweight = 'bold',
          fontfamily='serif')
plt.legend()
plt.show()
```

## CURVA DEL VIENTO PRONOSTICADO



```
# Análisis descriptiva con datos reales de La Velocidad del Viento
data_pred_febr = semana_estudio_pred
maximo = np.max(data_pred_febr)
print('Máximo = {}'.format(maximo))
minimo = np.min(data_pred_febr)
print('Mínimo = {}'.format(minimo))
# q75, q25 = np.percentile(data_estudio['VELOCIDAD DEL VIENTO'], [75, 25])
# rango = q75 - q25
rango = maximo - minimo
print('Rango = {}'.format(rango))
media = np.mean(data_pred_febr)
print('Media = {}'.format(media))
mediana = data_pred_febr.median()
print('Mediana = {}'.format(mediana))
varianza = data_pred_febr.var()
print('Varianza = {}'.format(varianza))
std = np.std(data_pred_febr)
print('Desviación Estándar = {}'.format(std))
coef_var = np.std(data_pred_febr) / np.mean(data_pred_febr) * 100
print('Coeficiente de variación = {} %'.format(coef_var))
# lim_max, lim_min = np.percentile((data_pred_febr), [75, 25])
lim_max = media + 1 * std
lim_min = media - 1 * std
print('Límite máximo = {}'.format(lim_max))
print('Límite mínimo = {}'.format(lim_min))

data_pred_febr.describe()
```

```
# Desviación estándar con datos de La Velocidad del Viento
```

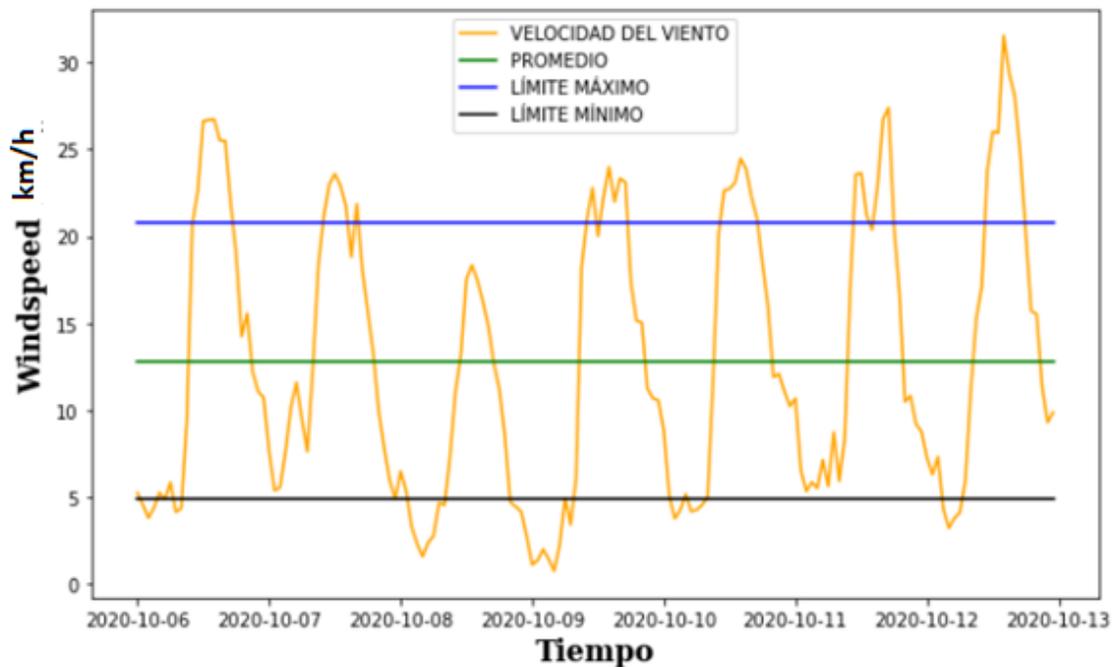
```
datos_viento = semana_estudio_pred

plt.figure(figsize=(10,6))
plt.plot(datos_viento.index,
         datos_viento,
         color = 'orange',
         label = 'VELOCIDAD DEL VIENTO')
plt.plot(datos_viento.index,
         [funcionConstante(x, media) for x in datos_viento],
         color = 'green',
         label = 'PROMEDIO')

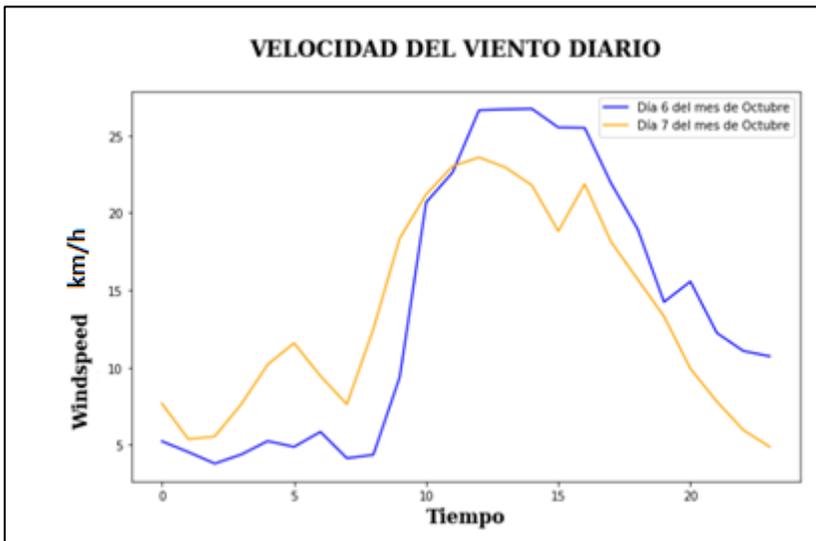
plt.plot(datos_viento.index,
         [funcionConstante(x, lim_max) for x in datos_viento],
         color = 'blue',
         label = 'LÍMITE MÁXIMO')

plt.plot(datos_viento.index,
         [funcionConstante(x, lim_min) for x in datos_viento],
         color = 'black',
         label = 'LÍMITE MÍNIMO')
```

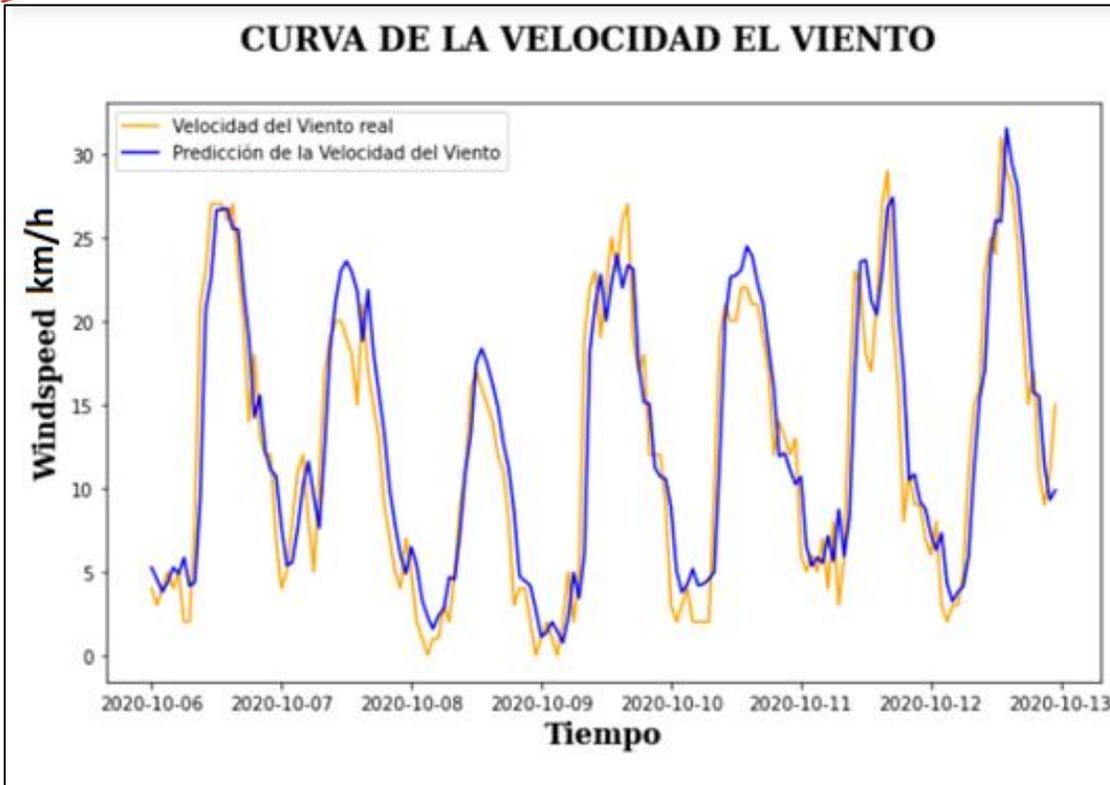
## DESVIACIÓN ESTÁNDAR



```
plt.suptitle('VELOCIDAD DEL VIENTO DIARIO',
            fontsize = 18,
            fontweight = 'bold',
            fontfamily='serif')
plt.xlabel('Tiempo',
          fontsize = 16,
          fontweight = 'bold',
          fontfamily='serif')
plt.ylabel('Windspeed',
          fontsize = 16,
          fontweight = 'bold',
          fontfamily='serif')
plt.legend()
```



```
# Curva del Viento real vs pronosticado
#Pronostico para los dias en estudio
plt.figure(figsize=(10,6))
plt.plot(semana_estudio_test,
        color='orange',
        label='Velocidad del Viento real')
plt.plot(semana_estudio_pred,
        color='blue',
        label='Predicción de la Velocidad del Viento')
plt.suptitle('CURVA DE LA VELOCIDAD EL VIENTO',
            fontsize = 18,
            fontweight = 'bold',
            fontfamily='serif')
plt.xlabel('Tiempo',
          fontsize = 16,
          fontweight = 'bold',
          fontfamily='serif')
plt.ylabel('Windspeed',
          fontsize = 16,
          fontweight = 'bold',
          fontfamily='serif')
plt.legend()
plt.show()
```



*# Gráfica de error entre curva del viento real y pronosticado (semanal)*

*# ERROR\_PROM = 0.0472 (Con 25 epocas y 25 neuronas)*

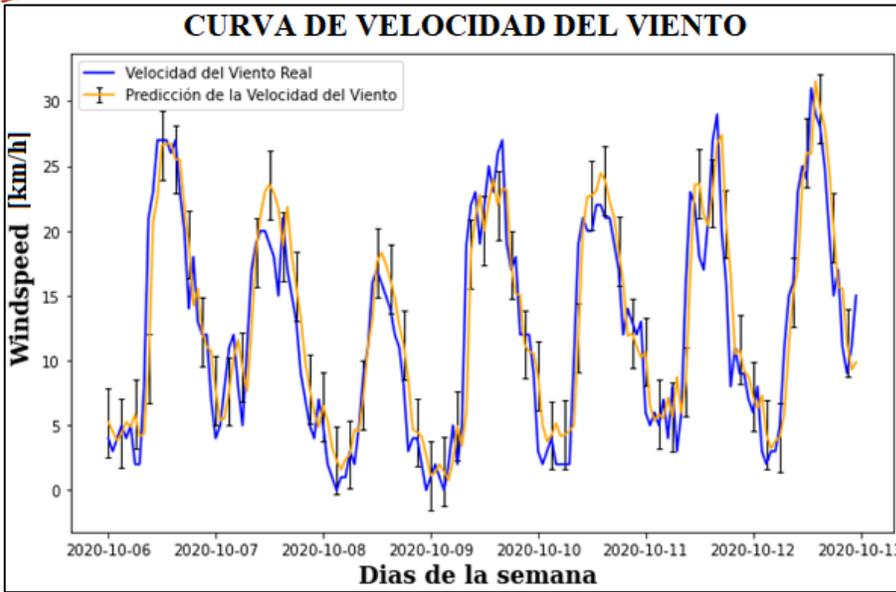
```
x = semana_estudio_test.index
y = semana_estudio_test
```

```
x_pred = x
y_pred = semana_estudio_pred
```

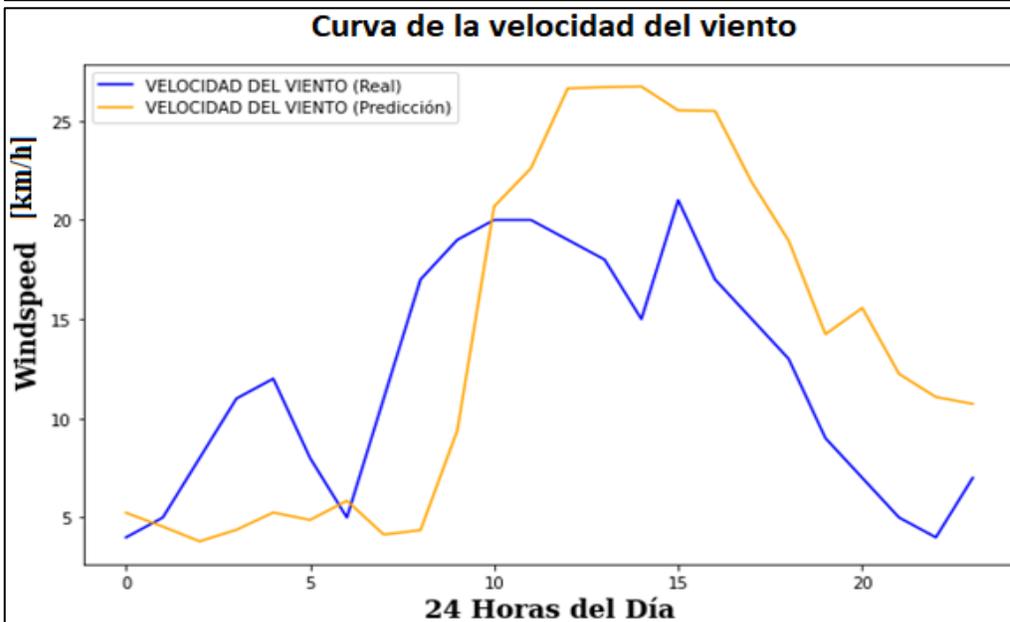
```
me = (abs(y - y_pred)).mean(axis=0) #Error medio de la semana
print(me)
```

```
yerror = me
plt.figure(figsize=(10,6))
plt.plot(semana_estudio_test.index,
         semana_estudio_test,
         color='blue',
         label='Velocidad del Viento Real')
```

```
plt.errorbar(x_pred,
            y_pred,
            yerr = yerror,
            ecolor = 'black',
            elinewidth = 1,
            capsize = 2,
            errorevery = 3,
            color='orange',
            label='Predicción de la Velocidad del Viento')
```



```
plt.suptitle('CURVA DE LA VELOCIDAD DEL VIENTO / DIARIO',
            fontsize = 18,
            fontweight = 'bold',
            fontfamily='serif')
plt.xlabel('24 Horas del Día',
          fontsize = 16,
          fontweight = 'bold',
          fontfamily='serif')
plt.ylabel('Windspeed',
          fontsize = 16,
          fontweight = 'bold',
          fontfamily='serif')
plt.legend()
```



```
# Grafica de error entre LA curva del viento real y pronosticado (Diaria)
x = dia_oct_test.index
y = dia_oct_test

x_pred = x
y_pred = dia_oct_pred

me = (abs(y- y_pred)).mean(axis=0) #Error medio de la semana
print(me)
yerror = me
plt.figure(figsize=(10,6))
plt.plot(x,
         y,
         color='yellow',
         label='Velocidad del Viento (real)')
plt.errorbar(x_pred,
             y_pred,
             yerr = yerror,
             ecolour = 'blue',
             elinewidth = 0.5,
             capsize = 2,
             errorevery = 3,
             color='red',
             label='Predicción de la Velocidad del Viento')
```

```
plt.suptitle('CURVA DE LA VELOCIDAD DEL VIENTO',
            fontsize = 18,
            fontweight = 'bold',
            fontfamily='serif')
plt.xlabel('Tiempo',
          fontsize = 18,
          fontweight = 'bold',
          fontfamily='serif')
plt.ylabel('Windspeed',
          fontsize = 18,
          fontweight = 'bold',
          fontfamily='serif')
plt.legend()
plt.show()
```



```
# Intervalos de la curva del viento (Datos reales y de pronóstico - 1dia)

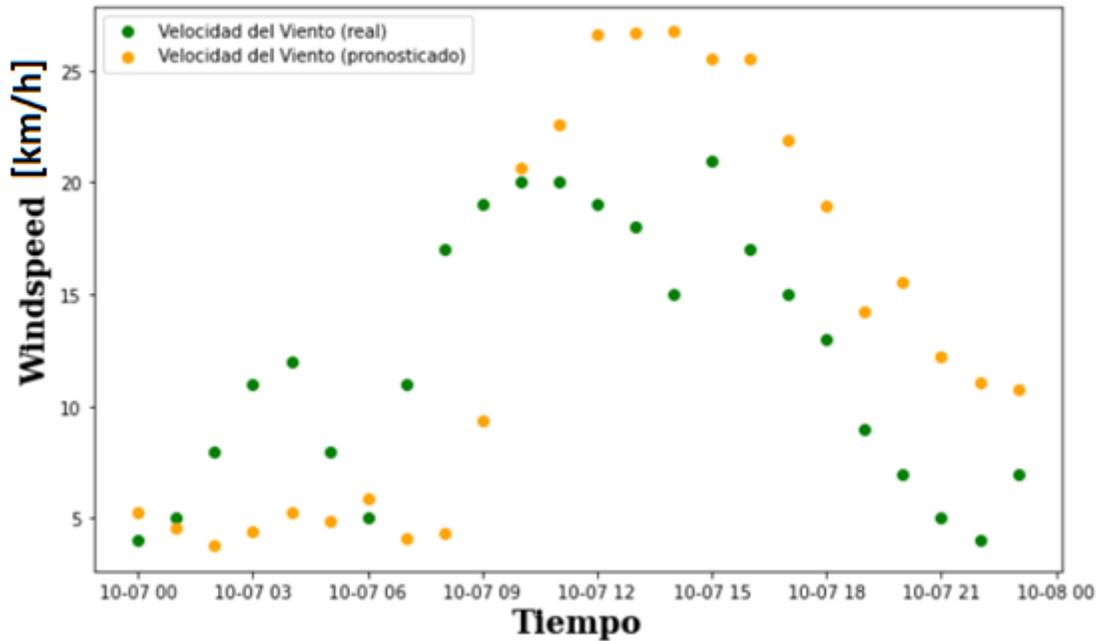
plt.figure(figsize=(10,6))
plt.scatter(x,
            y,
            color='green',
            label='Velocidad del Viento (real)')

plt.scatter(x_pred,
            y_pred,
            color='ORANGE',
            label='Velocidad del Viento (pronosticado)')

plt.suptitle('Gráfica de intervalos de la VELOCIDAD DEL VIENTO (m/s) ',
            fontsize = 18,
            fontweight = 'bold',
            fontfamily='serif')
plt.xlabel('Tiempo',
            fontsize = 18,
            fontweight = 'bold',
            fontfamily='serif')
plt.ylabel('Windspeed',
            fontsize = 18,
            fontweight = 'bold',
            fontfamily='serif')

plt.legend()
plt.show()
```

**Gráfica de intervalos de la VELOCIDAD DEL VIENTO (km/h)**



## Análisis de hiperparámetros para entrenamientos

### Algoritmo de optimización

#### ADAM

```
model_adam, history_adam = NN_LSTM(optimizador = 'adam', neuronas = 10, epocas = 15)  
model_adam.save('modelos/model_adam.h5')
```

```
Epoch 1/15  
110/110 [=====] - 20s 111ms/step - loss: 0.0242 - mean_absolute_error: 0.1225  
Epoch 2/15  
110/110 [=====] - 12s 111ms/step - loss: 0.0141 - mean_absolute_error: 0.0943  
Epoch 3/15  
110/110 [=====] - 12s 110ms/step - loss: 0.0095 - mean_absolute_error: 0.0757  
Epoch 4/15  
110/110 [=====] - 12s 110ms/step - loss: 0.0084 - mean_absolute_error: 0.0714  
Epoch 5/15  
110/110 [=====] - 11s 100ms/step - loss: 0.0076 - mean_absolute_error: 0.0679  
Epoch 6/15  
110/110 [=====] - 10s 91ms/step - loss: 0.0073 - mean_absolute_error: 0.0664  
Epoch 7/15
```

```
110/110 [=====] - 10s 91ms/step - loss: 0.0073 - mean_absolute_error: 0.0664  
Epoch 7/15  
110/110 [=====] - 11s 103ms/step - loss: 0.0070 - mean_absolute_error: 0.0647  
Epoch 8/15  
110/110 [=====] - 12s 107ms/step - loss: 0.0065 - mean_absolute_error: 0.0628  
Epoch 9/15  
110/110 [=====] - 12s 105ms/step - loss: 0.0064 - mean_absolute_error: 0.0622  
Epoch 10/15  
110/110 [=====] - 12s 105ms/step - loss: 0.0061 - mean_absolute_error: 0.0606  
Epoch 11/15  
110/110 [=====] - 12s 111ms/step - loss: 0.0059 - mean_absolute_error: 0.0600  
Epoch 12/15  
110/110 [=====] - 12s 111ms/step - loss: 0.0058 - mean_absolute_error: 0.0594  
Epoch 13/15  
110/110 [=====] - 12s 110ms/step - loss: 0.0056 - mean_absolute_error: 0.0580  
Epoch 14/15  
110/110 [=====] - 12s 111ms/step - loss: 0.0054 - mean_absolute_error: 0.0570  
Epoch 15/15  
110/110 [=====] - 12s 111ms/step - loss: 0.0053 - mean_absolute_error: 0.0562
```

```
print('El valor de la pérdida obtenido es: {}'.format(history_adam.history['loss'][-1]))  
print('El valor del error medio absoluto es: {}'.format(history_adam.history['mean_absolute_error'][-1]))
```

```
El valor de la pérdida obtenido es: 0.005283871199935675  
El valor del error medio absoluto es: 0.05622221902012825
```

Activar V

#### ADADELTA

```
model_adadelta, history_adadelta = NN_LSTM(optimizador = 'adadelta', neuronas = 10, epocas = 15)  
model_adadelta.save('modelos/model_adadelta.h5')
```

```
Epoch 1/15  
110/110 [=====] - 21s 111ms/step - loss: 0.0739 - mean_absolute_error: 0.2243  
Epoch 2/15  
110/110 [=====] - 12s 112ms/step - loss: 0.0734 - mean_absolute_error: 0.2230  
Epoch 3/15  
110/110 [=====] - 12s 111ms/step - loss: 0.0721 - mean_absolute_error: 0.2208  
Epoch 4/15  
110/110 [=====] - 12s 110ms/step - loss: 0.0712 - mean_absolute_error: 0.2187  
Epoch 5/15
```

Act  
Ve a

```
110/110 [=====] - 12s 111ms/step - loss: 0.0672 - mean_absolute_error: 0.2098
Epoch 9/15
110/110 [=====] - 12s 109ms/step - loss: 0.0658 - mean_absolute_error: 0.2070
Epoch 10/15
110/110 [=====] - 11s 101ms/step - loss: 0.0648 - mean_absolute_error: 0.2047
Epoch 11/15
110/110 [=====] - 10s 93ms/step - loss: 0.0633 - mean_absolute_error: 0.2014
Epoch 12/15
110/110 [=====] - 8s 75ms/step - loss: 0.0625 - mean_absolute_error: 0.1997
Epoch 13/15
110/110 [=====] - 10s 94ms/step - loss: 0.0613 - mean_absolute_error: 0.1968
Epoch 14/15
110/110 [=====] - 11s 103ms/step - loss: 0.0605 - mean_absolute_error: 0.1946
Epoch 15/15
110/110 [=====] - 12s 109ms/step - loss: 0.0592 - mean_absolute_error: 0.1918
```

```
print('El valor de la pérdida obtenido es: {}'.format(history_adadelta.history['loss'][-1]))
print('El valor del error medio absoluto es: {}'.format(history_adadelta.history['mean_absolute_error'][-1]))
```

El valor de la pérdida obtenido es: 0.05919744074344635  
El valor del error medio absoluto es: 0.1918332874774933

## RMSPROP

```
model_rmsprop, history_rmsprop = NN_LSTM(optimizador = 'rmsprop', neuronas = 10, epocas = 15)
model_rmsprop.save('modelos/model_rmsprop.h5')
```

```
Epoch 1/15
110/110 [=====] - 21s 109ms/step - loss: 0.0229 - mean_absolute_error: 0.1158
Epoch 2/15
110/110 [=====] - 12s 112ms/step - loss: 0.0106 - mean_absolute_error: 0.0804
Epoch 3/15
110/110 [=====] - 12s 109ms/step - loss: 0.0079 - mean_absolute_error: 0.0687
Epoch 4/15
110/110 [=====] - 11s 99ms/step - loss: 0.0070 - mean_absolute_error: 0.0649
Epoch 5/15
110/110 [=====] - 12s 105ms/step - loss: 0.0065 - mean_absolute_error: 0.0627
Epoch 6/15
110/110 [=====] - 11s 105ms/step - loss: 0.0063 - mean_absolute_error: 0.0615
Epoch 7/15
110/110 [=====] - 11s 103ms/step - loss: 0.0059 - mean_absolute_error: 0.0593
Epoch 8/15
110/110 [=====] - 12s 109ms/step - loss: 0.0056 - mean_absolute_error: 0.0578
Epoch 9/15
110/110 [=====] - 12s 112ms/step - loss: 0.0054 - mean_absolute_error: 0.0574
Epoch 10/15
110/110 [=====] - 12s 107ms/step - loss: 0.0053 - mean_absolute_error: 0.0565
```

```
Epoch 8/15
110/110 [=====] - 12s 109ms/step - loss: 0.0056 - mean_absolute_error: 0.0578
Epoch 9/15
110/110 [=====] - 12s 112ms/step - loss: 0.0054 - mean_absolute_error: 0.0574
Epoch 10/15
110/110 [=====] - 12s 107ms/step - loss: 0.0053 - mean_absolute_error: 0.0565
Epoch 11/15
110/110 [=====] - 12s 110ms/step - loss: 0.0052 - mean_absolute_error: 0.0558
Epoch 12/15
110/110 [=====] - 12s 106ms/step - loss: 0.0050 - mean_absolute_error: 0.0547
Epoch 13/15
110/110 [=====] - 12s 109ms/step - loss: 0.0049 - mean_absolute_error: 0.0543
Epoch 14/15
110/110 [=====] - 12s 113ms/step - loss: 0.0048 - mean_absolute_error: 0.0536
Epoch 15/15
110/110 [=====] - 12s 112ms/step - loss: 0.0047 - mean_absolute_error: 0.0528
```

```
print('El valor de la pérdida obtenido es: {}'.format(history_rmsprop.history['loss'][-1]))
print('El valor del error medio absoluto es: {}'.format(history_rmsprop.history['mean_absolute_error'][-1]))
```

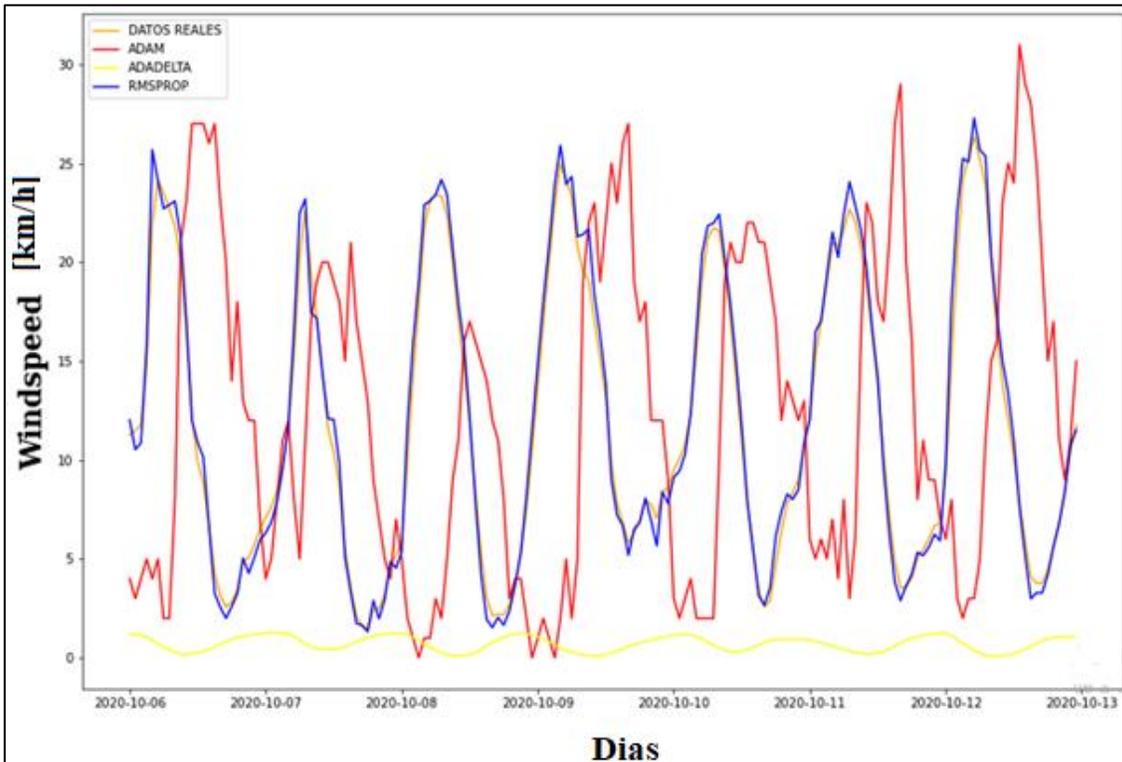
El valor de la pérdida obtenido es: 0.004725858569145203  
El valor del error medio absoluto es: 0.05282999947667122

## COMPARATIVA DE OPTIMIZADORES

*# Resultados variación algoritmo de entrenamiento de La red neuronal*

```
plt.figure(figsize=(15,10))

limite = len(semana_estudio_test)
plt.plot(semana_estudio_test.index,
prediccion_adam[:limite],
color='orange',
label='DATOS REALES')
plt.plot(semana_estudio_test,
color='red',
label='ADAM')
plt.plot(semana_estudio_test.index,
prediccion_adadelta[:limite],
color='yellow',
label='ADADELTA')
plt.plot(semana_estudio_test.index,
prediccion_rmsprop[:limite],
color='blue',
label='RMSPROP')
```





## Número de neuronas

### 10 Neuronas

```
model_10N, history_10N = NN_LSTM(optimizador = 'rmsprop', neuronas = 10, epocas = 15)
```

```
Epoch 1/15  
110/110 [=====] - 21s 110ms/step - loss: 0.0237 - mean_absolute_error: 0.1203  
Epoch 2/15  
110/110 [=====] - 12s 110ms/step - loss: 0.0136 - mean_absolute_error: 0.0926  
Epoch 3/15  
110/110 [=====] - 12s 111ms/step - loss: 0.0089 - mean_absolute_error: 0.0734  
Epoch 4/15  
110/110 [=====] - 13s 117ms/step - loss: 0.0078 - mean_absolute_error: 0.0687  
Epoch 5/15  
110/110 [=====] - 13s 119ms/step - loss: 0.0071 - mean_absolute_error: 0.0658  
Epoch 6/15  
110/110 [=====] - 13s 119ms/step - loss: 0.0067 - mean_absolute_error: 0.0641  
Epoch 7/15  
110/110 [=====] - 13s 117ms/step - loss: 0.0064 - mean_absolute_error: 0.0626  
Epoch 8/15  
110/110 [=====] - 12s 107ms/step - loss: 0.0062 - mean_absolute_error: 0.0613  
Epoch 9/15  
110/110 [=====] - 12s 110ms/step - loss: 0.0060 - mean_absolute_error: 0.0603  
Epoch 10/15  
110/110 [=====] - 12s 113ms/step - loss: 0.0057 - mean_absolute_error: 0.0592  
Epoch 11/15  
110/110 [=====] - 11s 98ms/step - loss: 0.0056 - mean_absolute_error: 0.0586  
Epoch 12/15  
110/110 [=====] - 12s 105ms/step - loss: 0.0054 - mean_absolute_error: 0.0572  
Epoch 13/15
```

```
Epoch 9/15  
110/110 [=====] - 12s 110ms/step - loss: 0.0060 - mean_absolute_error: 0.0603  
Epoch 10/15  
110/110 [=====] - 12s 113ms/step - loss: 0.0057 - mean_absolute_error: 0.0592  
Epoch 11/15  
110/110 [=====] - 11s 98ms/step - loss: 0.0056 - mean_absolute_error: 0.0586  
Epoch 12/15  
110/110 [=====] - 12s 105ms/step - loss: 0.0054 - mean_absolute_error: 0.0572  
Epoch 13/15  
110/110 [=====] - 12s 107ms/step - loss: 0.0051 - mean_absolute_error: 0.0558  
Epoch 14/15  
110/110 [=====] - 12s 110ms/step - loss: 0.0051 - mean_absolute_error: 0.0553  
Epoch 15/15  
110/110 [=====] - 12s 112ms/step - loss: 0.0049 - mean_absolute_error: 0.0545
```

```
model_10N.save('modelos/model_10N.h5')  
print('El valor de la pérdida obtenido es: {}'.format(history_10N.history['loss'][-1]))  
print('El valor del error medio absoluto es: {}'.format(history_10N.history['mean_absolute_error'][-1]))
```

```
El valor de la pérdida obtenido es: 0.004924391396343708  
El valor del error medio absoluto es: 0.05450465530157089
```

## 15 Neuronas

```
model_15N, history_15N = NN_LSTM(optimizador = 'rmsprop', neuronas = 15, epocas = 15)
```

```
Epoch 1/15  
110/110 [=====] - 22s 110ms/step - loss: 0.0175 - mean_absolute_error: 0.1060  
Epoch 2/15  
110/110 [=====] - 11s 102ms/step - loss: 0.0092 - mean_absolute_error: 0.0753  
Epoch 3/15  
110/110 [=====] - 12s 112ms/step - loss: 0.0076 - mean_absolute_error: 0.0680  
Epoch 4/15  
110/110 [=====] - 12s 112ms/step - loss: 0.0069 - mean_absolute_error: 0.0644  
Epoch 5/15  
110/110 [=====] - 13s 114ms/step - loss: 0.0065 - mean_absolute_error: 0.0624  
Epoch 6/15  
110/110 [=====] - 12s 110ms/step - loss: 0.0060 - mean_absolute_error: 0.0604  
Epoch 7/15  
110/110 [=====] - 12s 111ms/step - loss: 0.0057 - mean_absolute_error: 0.0586  
Epoch 8/15  
110/110 [=====] - 12s 110ms/step - loss: 0.0054 - mean_absolute_error: 0.0571  
Epoch 9/15  
110/110 [=====] - 11s 103ms/step - loss: 0.0052 - mean_absolute_error: 0.0557  
Epoch 10/15  
110/110 [=====] - 10s 88ms/step - loss: 0.0048 - mean_absolute_error: 0.0535  
Epoch 11/15  
110/110 [=====] - 11s 97ms/step - loss: 0.0047 - mean_absolute_error: 0.0528  
Epoch 12/15  
110/110 [=====] - 12s 107ms/step - loss: 0.0045 - mean_absolute_error: 0.0515  
Epoch 13/15  
110/110 [=====] - 12s 113ms/step - loss: 0.0044 - mean_absolute_error: 0.0511  
Epoch 14/15  
110/110 [=====] - 12s 110ms/step - loss: 0.0043 - mean_absolute_error: 0.0504  
Epoch 15/15
```

```
110/110 [=====] - 12s 110ms/step - loss: 0.0060 - mean_absolute_error: 0.0604  
Epoch 7/15  
110/110 [=====] - 12s 111ms/step - loss: 0.0057 - mean_absolute_error: 0.0586  
Epoch 8/15  
110/110 [=====] - 12s 110ms/step - loss: 0.0054 - mean_absolute_error: 0.0571  
Epoch 9/15  
110/110 [=====] - 11s 103ms/step - loss: 0.0052 - mean_absolute_error: 0.0557  
Epoch 10/15  
110/110 [=====] - 10s 88ms/step - loss: 0.0048 - mean_absolute_error: 0.0535  
Epoch 11/15  
110/110 [=====] - 11s 97ms/step - loss: 0.0047 - mean_absolute_error: 0.0528  
Epoch 12/15  
110/110 [=====] - 12s 107ms/step - loss: 0.0045 - mean_absolute_error: 0.0515  
Epoch 13/15  
110/110 [=====] - 12s 113ms/step - loss: 0.0044 - mean_absolute_error: 0.0511  
Epoch 14/15  
110/110 [=====] - 12s 110ms/step - loss: 0.0043 - mean_absolute_error: 0.0504  
Epoch 15/15  
110/110 [=====] - 12s 113ms/step - loss: 0.0042 - mean_absolute_error: 0.0501
```

```
model_15N.save('modelos/model_15N.h5')  
print('El valor de la pérdida obtenido es: {}'.format(history_15N.history['loss'][-1]))  
print('El valor del error medio absoluto es: {}'.format(history_15N.history['mean_absolute_error'][-1]))
```

```
El valor de la pérdida obtenido es: 0.004241943825036287  
El valor del error medio absoluto es: 0.050088416785001755
```

## 25 neuronas

```
model_25N, history_25N = NN_LSTM(optimizador = 'rmsprop', neuronas = 25, epocas = 15)
```

```
Epoch 1/15  
110/110 [=====] - 23s 115ms/step - loss: 0.0149 - mean_absolute_error: 0.0955  
Epoch 2/15  
110/110 [=====] - 13s 117ms/step - loss: 0.0075 - mean_absolute_error: 0.0673  
Epoch 3/15  
110/110 [=====] - 13s 117ms/step - loss: 0.0066 - mean_absolute_error: 0.0630  
Epoch 4/15  
110/110 [=====] - 13s 118ms/step - loss: 0.0060 - mean_absolute_error: 0.0603  
Epoch 5/15  
110/110 [=====] - 13s 117ms/step - loss: 0.0056 - mean_absolute_error: 0.0580  
Epoch 6/15  
110/110 [=====] - 13s 117ms/step - loss: 0.0053 - mean_absolute_error: 0.0566  
Epoch 7/15  
110/110 [=====] - 13s 117ms/step - loss: 0.0050 - mean_absolute_error: 0.0544  
Epoch 8/15  
110/110 [=====] - 13s 116ms/step - loss: 0.0047 - mean_absolute_error: 0.0529  
Epoch 9/15  
110/110 [=====] - 12s 113ms/step - loss: 0.0045 - mean_absolute_error: 0.0517  
Epoch 10/15  
110/110 [=====] - 13s 119ms/step - loss: 0.0043 - mean_absolute_error: 0.0504  
Epoch 11/15  
110/110 [=====] - 13s 119ms/step - loss: 0.0042 - mean_absolute_error: 0.0497  
Epoch 12/15  
110/110 [=====] - 13s 119ms/step - loss: 0.0042 - mean_absolute_error: 0.0495  
Epoch 13/15  
110/110 [=====] - 13s 118ms/step - loss: 0.0041 - mean_absolute_error: 0.0492  
Epoch 14/15  
110/110 [=====] - 13s 115ms/step - loss: 0.0041 - mean_absolute_error: 0.0487  
Epoch 15/15
```

```
Epoch 6/15  
110/110 [=====] - 13s 117ms/step - loss: 0.0053 - mean_absolute_error: 0.0566  
Epoch 7/15  
110/110 [=====] - 13s 117ms/step - loss: 0.0050 - mean_absolute_error: 0.0544  
Epoch 8/15  
110/110 [=====] - 13s 116ms/step - loss: 0.0047 - mean_absolute_error: 0.0529  
Epoch 9/15  
110/110 [=====] - 12s 113ms/step - loss: 0.0045 - mean_absolute_error: 0.0517  
Epoch 10/15  
110/110 [=====] - 13s 119ms/step - loss: 0.0043 - mean_absolute_error: 0.0504  
Epoch 11/15  
110/110 [=====] - 13s 119ms/step - loss: 0.0042 - mean_absolute_error: 0.0497  
Epoch 12/15  
110/110 [=====] - 13s 119ms/step - loss: 0.0042 - mean_absolute_error: 0.0495  
Epoch 13/15  
110/110 [=====] - 13s 118ms/step - loss: 0.0041 - mean_absolute_error: 0.0492  
Epoch 14/15  
110/110 [=====] - 13s 115ms/step - loss: 0.0041 - mean_absolute_error: 0.0487  
Epoch 15/15  
110/110 [=====] - 13s 119ms/step - loss: 0.0040 - mean_absolute_error: 0.0484
```

```
model_25N.save('modelos/model_25N.h5')  
print('El valor de la pérdida obtenido es: {}'.format(history_25N.history['loss'][-1]))  
print('El valor del error medio absoluto es: {}'.format(history_25N.history['mean_absolute_error'][-1]))
```

```
El valor de la pérdida obtenido es: 0.003987045958638191  
El valor del error medio absoluto es: 0.04839063808321953
```

### 30 Neuronas

```
model_30N, history_30N = NN_LSTM(optimizador = 'rmsprop', neuronas = 30, epocas = 15)
```

```
Epoch 1/15  
110/110 [=====] - 23s 124ms/step - loss: 0.0147 - mean_absolute_error: 0.0962  
Epoch 2/15  
110/110 [=====] - 13s 121ms/step - loss: 0.0073 - mean_absolute_error: 0.0662  
Epoch 3/15  
110/110 [=====] - 13s 118ms/step - loss: 0.0065 - mean_absolute_error: 0.0623  
Epoch 4/15  
110/110 [=====] - 12s 111ms/step - loss: 0.0060 - mean_absolute_error: 0.0598  
Epoch 5/15  
110/110 [=====] - 13s 121ms/step - loss: 0.0055 - mean_absolute_error: 0.0576  
Epoch 6/15  
110/110 [=====] - 14s 124ms/step - loss: 0.0053 - mean_absolute_error: 0.0561  
Epoch 7/15  
110/110 [=====] - 13s 122ms/step - loss: 0.0049 - mean_absolute_error: 0.0540  
Epoch 8/15  
110/110 [=====] - 14s 124ms/step - loss: 0.0046 - mean_absolute_error: 0.0521  
Epoch 9/15  
110/110 [=====] - 14s 125ms/step - loss: 0.0044 - mean_absolute_error: 0.0509  
Epoch 10/15  
110/110 [=====] - 14s 123ms/step - loss: 0.0043 - mean_absolute_error: 0.0501  
Epoch 11/15  
110/110 [=====] - 14s 124ms/step - loss: 0.0042 - mean_absolute_error: 0.0496  
Epoch 12/15  
110/110 [=====] - 13s 121ms/step - loss: 0.0041 - mean_absolute_error: 0.0487  
Epoch 13/15  
110/110 [=====] - 13s 119ms/step - loss: 0.0040 - mean_absolute_error: 0.0483  
Epoch 14/15  
110/110 [=====] - 13s 122ms/step - loss: 0.0039 - mean_absolute_error: 0.0479  
Epoch 15/15  
110/110 [=====] - 13s 122ms/step - loss: 0.0040 - mean absolute error: 0.0478
```

```
model_30N.save('modelos/model_30N.h5')  
print('El valor de la pérdida obtenido es: {}'.format(history_30N.history['loss'][-1]))  
print('El valor del error medio absoluto es: {}'.format(history_30N.history['mean_absolute_error'][-1]))
```

```
El valor de la pérdida obtenido es: 0.003952177241444588  
El valor del error medio absoluto es: 0.04784652218222618
```

```
# Llamado de los modelos predcidos para graficar  
prediccion_10N = predDataTest(model_10N)  
prediccion_15N = predDataTest(model_15N)  
prediccion_25N = predDataTest(model_25N)  
prediccion_30N = predDataTest(model_30N)
```

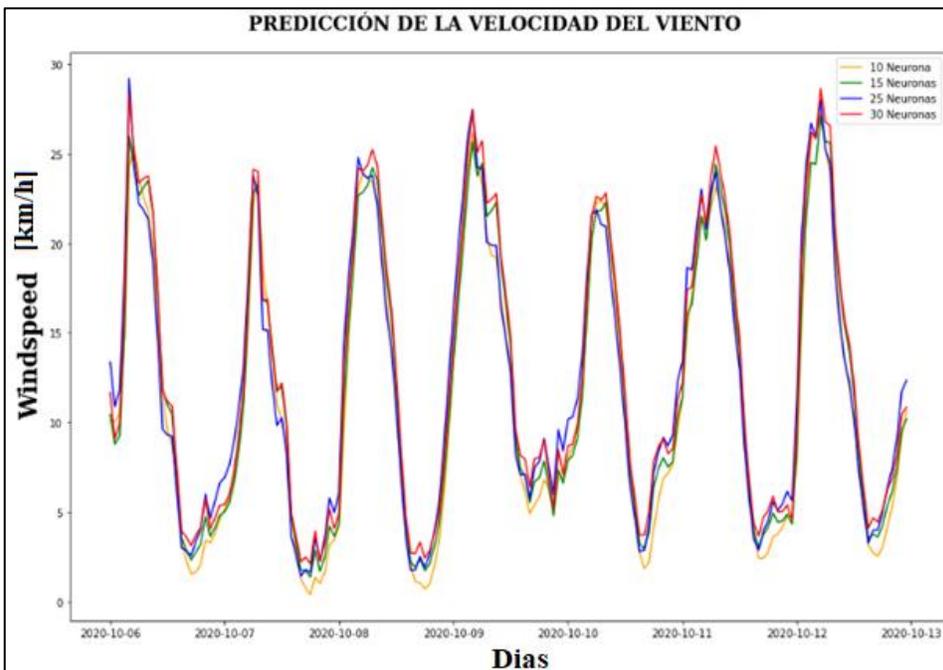
# Figura 5.43: Resultados variación número de neuronas

```
plt.figure(figsize=(15,10))

limite = len(semana_estudio_test)

plt.plot(semana_estudio_test.index,
         prediccion_10N[:limite],
         color='orange',
         label='10 Neurona')
plt.plot(semana_estudio_test.index,
         prediccion_15N[:limite],
         color='green',
         label='15 Neuronas')
plt.plot(semana_estudio_test.index,
         prediccion_25N[:limite],
         color='blue',
         label='25 Neuronas')
plt.plot(semana_estudio_test.index,
         prediccion_30N[:limite],
         color='red',
         label='30 Neuronas')
```

```
plt.suptitle('PREDICCIÓN DE LA VELOCIDAD DEL VIENTO',
            fontsize = 18,
            fontweight = 'bold',
            fontfamily='serif')
plt.xlabel('Tiempo',
          fontsize = 16,
          fontweight = 'bold',
          fontfamily='serif')
plt.ylabel('WindSpeed',
          fontsize = 16,
          fontweight = 'bold',
          fontfamily='serif')
plt.legend()
plt.show()
```



## Número de épocas

### 10 Epocas

```
model_10E,history_10E = NN_LSTM(optimizador = 'rmsprop', neuronas = 25, epocas = 10)  
model_10E.save('modelos/model_10E.h5')
```

```
Epoch 1/10  
110/110 [=====] - 23s 118ms/step - loss: 0.0133 - mean_absolute_error: 0.0904  
Epoch 2/10  
110/110 [=====] - 13s 117ms/step - loss: 0.0074 - mean_absolute_error: 0.0671  
Epoch 3/10  
110/110 [=====] - 13s 117ms/step - loss: 0.0066 - mean_absolute_error: 0.0630  
Epoch 4/10  
110/110 [=====] - 13s 118ms/step - loss: 0.0062 - mean_absolute_error: 0.0608  
Epoch 5/10  
110/110 [=====] - 13s 117ms/step - loss: 0.0058 - mean_absolute_error: 0.0588  
Epoch 6/10  
110/110 [=====] - 13s 118ms/step - loss: 0.0054 - mean_absolute_error: 0.0572  
Epoch 7/10  
110/110 [=====] - 13s 116ms/step - loss: 0.0050 - mean_absolute_error: 0.0546  
Epoch 8/10  
110/110 [=====] - 13s 117ms/step - loss: 0.0048 - mean_absolute_error: 0.0534  
Epoch 9/10  
110/110 [=====] - 13s 117ms/step - loss: 0.0045 - mean_absolute_error: 0.0517  
Epoch 10/10  
110/110 [=====] - 13s 116ms/step - loss: 0.0044 - mean_absolute_error: 0.0510
```

```
print('El valor de la pérdida obtenido es: {}'.format(history_10E.history['loss'][-1]))  
print('El valor del error medio absoluto es: {}'.format(history_10E.history['mean_absolute_error'][-1]))
```

```
El valor de la pérdida obtenido es: 0.004413412883877754  
El valor del error medio absoluto es: 0.050984352827072144
```

### 15 Epocas

```
model_15E,history_15E = NN_LSTM(optimizador = 'rmsprop', neuronas = 25, epocas = 15)  
model_15E.save('modelos/model_15E.h5')
```

```
Epoch 1/15  
110/110 [=====] - 82s 662ms/step - loss: 0.0144 - mean_absolute_error: 0.0940  
Epoch 2/15  
110/110 [=====] - 7s 63ms/step - loss: 0.0074 - mean_absolute_error: 0.0667  
Epoch 3/15  
110/110 [=====] - 13s 116ms/step - loss: 0.0064 - mean_absolute_error: 0.0620  
Epoch 4/15  
110/110 [=====] - 13s 117ms/step - loss: 0.0060 - mean_absolute_error: 0.0599  
Epoch 5/15  
110/110 [=====] - 13s 119ms/step - loss: 0.0054 - mean_absolute_error: 0.0570  
Epoch 6/15  
110/110 [=====] - 13s 116ms/step - loss: 0.0051 - mean_absolute_error: 0.0553  
Epoch 7/15  
110/110 [=====] - 13s 117ms/step - loss: 0.0048 - mean_absolute_error: 0.0535  
Epoch 8/15  
110/110 [=====] - 13s 115ms/step - loss: 0.0046 - mean_absolute_error: 0.0520
```



### 15 Epocas

```
model_15E,history_15E = NN_LSTM(optimizador = 'rmsprop', neuronas = 25, epocas = 15)  
model_15E.save('modelos/model_15E.h5')
```

```
Epoch 1/15  
110/110 [=====] - 82s 662ms/step - loss: 0.0144 - mean_absolute_error: 0.0940  
Epoch 2/15  
110/110 [=====] - 7s 63ms/step - loss: 0.0074 - mean_absolute_error: 0.0667  
Epoch 3/15  
110/110 [=====] - 13s 116ms/step - loss: 0.0064 - mean_absolute_error: 0.0620  
Epoch 4/15  
110/110 [=====] - 13s 117ms/step - loss: 0.0060 - mean_absolute_error: 0.0599  
Epoch 5/15  
110/110 [=====] - 13s 119ms/step - loss: 0.0054 - mean_absolute_error: 0.0570  
Epoch 6/15  
110/110 [=====] - 13s 116ms/step - loss: 0.0051 - mean_absolute_error: 0.0553  
Epoch 7/15  
110/110 [=====] - 13s 117ms/step - loss: 0.0048 - mean_absolute_error: 0.0535  
Epoch 8/15  
110/110 [=====] - 13s 115ms/step - loss: 0.0046 - mean_absolute_error: 0.0520  
Epoch 9/15  
110/110 [=====] - 13s 114ms/step - loss: 0.0043 - mean_absolute_error: 0.0506  
Epoch 10/15  
110/110 [=====] - 13s 118ms/step - loss: 0.0043 - mean_absolute_error: 0.0498  
Epoch 11/15  
110/110 [=====] - 13s 119ms/step - loss: 0.0041 - mean_absolute_error: 0.0490  
Epoch 12/15  
110/110 [=====] - 13s 118ms/step - loss: 0.0040 - mean_absolute_error: 0.0483
```

```
print('El valor de la pérdida obtenido es: {}'.format(history_15E.history['loss'][-1]))  
print('El valor del error medio absoluto es: {}'.format(history_15E.history['mean_absolute_error'][-1]))
```

El valor de la pérdida obtenido es: 0.003954560495913029  
El valor del error medio absoluto es: 0.04801340773701668

### 25 Epocas

```
model_25E,history_25E = NN_LSTM(optimizador = 'rmsprop', neuronas = 25, epocas = 25)  
model_25E.save('modelos/model_25E.h5')
```

```
Epoch 1/25  
110/110 [=====] - 21s 117ms/step - loss: 0.0139 - mean_absolute_error: 0.0926  
Epoch 2/25  
110/110 [=====] - 13s 116ms/step - loss: 0.0075 - mean_absolute_error: 0.0670  
Epoch 3/25  
110/110 [=====] - 13s 117ms/step - loss: 0.0066 - mean_absolute_error: 0.0632  
Epoch 4/25  
110/110 [=====] - 13s 119ms/step - loss: 0.0061 - mean_absolute_error: 0.0604  
Epoch 5/25  
110/110 [=====] - 13s 118ms/step - loss: 0.0057 - mean_absolute_error: 0.0585  
Epoch 6/25  
110/110 [=====] - 13s 115ms/step - loss: 0.0053 - mean_absolute_error: 0.0564  
Epoch 7/25  
110/110 [=====] - 13s 115ms/step - loss: 0.0050 - mean_absolute_error: 0.0544  
Epoch 8/25  
110/110 [=====] - 13s 117ms/step - loss: 0.0047 - mean_absolute_error: 0.0531  
Epoch 9/25  
110/110 [=====] - 13s 117ms/step - loss: 0.0045 - mean_absolute_error: 0.0513  
Epoch 10/25  
110/110 [=====] - 13s 117ms/step - loss: 0.0044 - mean_absolute_error: 0.0506  
Epoch 11/25  
110/110 [=====] - 13s 117ms/step - loss: 0.0042 - mean_absolute_error: 0.0496
```



```
Epoch 18/25
110/110 [=====] - 13s 117ms/step - loss: 0.0039 - mean_absolute_error: 0.0476
Epoch 19/25
110/110 [=====] - 13s 118ms/step - loss: 0.0039 - mean_absolute_error: 0.0476
Epoch 20/25
110/110 [=====] - 13s 117ms/step - loss: 0.0039 - mean_absolute_error: 0.0476
Epoch 21/25
110/110 [=====] - 13s 117ms/step - loss: 0.0038 - mean_absolute_error: 0.0471
Epoch 22/25
110/110 [=====] - 13s 118ms/step - loss: 0.0038 - mean_absolute_error: 0.0471
Epoch 23/25
110/110 [=====] - 13s 120ms/step - loss: 0.0038 - mean_absolute_error: 0.0472
Epoch 24/25
110/110 [=====] - 13s 117ms/step - loss: 0.0038 - mean_absolute_error: 0.0471
Epoch 25/25
110/110 [=====] - 13s 117ms/step - loss: 0.0038 - mean_absolute_error: 0.0470
```

```
print('El valor de la pérdida obtenido es: {}'.format(history_25E.history['loss'][-1]))
print('El valor del error medio absoluto es: {}'.format(history_25E.history['mean_absolute_error'][-1]))
```

El valor de la pérdida obtenido es: 0.0037645867560058832  
El valor del error medio absoluto es: 0.04696726053953171

### 30 Epocas

```
model_30E, history_30E = NN_LSTM(optimizador = 'rmsprop', neuronas = 25, epocas = 30)
model_30E.save('modelos/model_30E.h5')
```

```
Epoch 1/30
110/110 [=====] - 22s 118ms/step - loss: 0.0158 - mean_absolute_error: 0.0994
Epoch 2/30
110/110 [=====] - 13s 118ms/step - loss: 0.0078 - mean_absolute_error: 0.0689
Epoch 3/30
110/110 [=====] - 13s 118ms/step - loss: 0.0067 - mean_absolute_error: 0.0637
Epoch 4/30
110/110 [=====] - 13s 117ms/step - loss: 0.0061 - mean_absolute_error: 0.0608
Epoch 5/30
110/110 [=====] - 13s 120ms/step - loss: 0.0056 - mean_absolute_error: 0.0580
Epoch 6/30
110/110 [=====] - 13s 117ms/step - loss: 0.0052 - mean_absolute_error: 0.0561
Epoch 7/30
110/110 [=====] - 13s 118ms/step - loss: 0.0048 - mean_absolute_error: 0.0535
```

```
Epoch 22/30
110/110 [=====] - 13s 117ms/step - loss: 0.0038 - mean_absolute_error: 0.0469
Epoch 23/30
110/110 [=====] - 13s 116ms/step - loss: 0.0038 - mean_absolute_error: 0.0471
Epoch 24/30
110/110 [=====] - 13s 119ms/step - loss: 0.0037 - mean_absolute_error: 0.0467
Epoch 25/30
110/110 [=====] - 13s 116ms/step - loss: 0.0037 - mean_absolute_error: 0.0469
Epoch 26/30
110/110 [=====] - 13s 119ms/step - loss: 0.0038 - mean_absolute_error: 0.0470
Epoch 27/30
110/110 [=====] - 13s 118ms/step - loss: 0.0037 - mean_absolute_error: 0.0464
Epoch 28/30
110/110 [=====] - 13s 117ms/step - loss: 0.0037 - mean_absolute_error: 0.0468
Epoch 29/30
110/110 [=====] - 13s 117ms/step - loss: 0.0037 - mean_absolute_error: 0.0466
Epoch 30/30
110/110 [=====] - 13s 117ms/step - loss: 0.0037 - mean_absolute_error: 0.0464
```

```
print('El valor de la pérdida obtenido es: {}'.format(history_30E.history['loss'][-1]))
print('El valor del error medio absoluto es: {}'.format(history_30E.history['mean_absolute_error'][-1]))
```

El valor de la pérdida obtenido es: 0.003675850573927164  
El valor del error medio absoluto es: 0.04641631618142128

```
# Resultados variación número de épocas
```

```
plt.figure(figsize=(15,10))
```

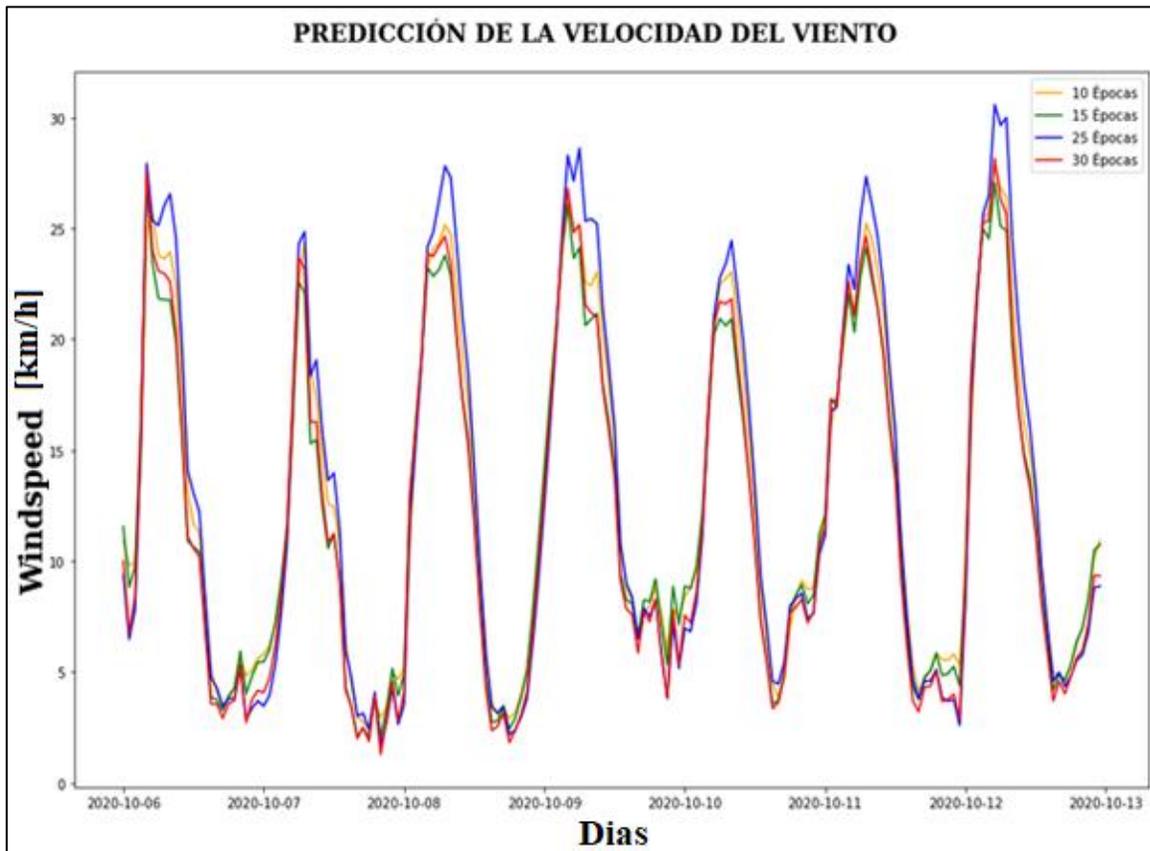
```
limite = len(semana_estudio_test)
```

```
plt.plot(semana_estudio_test.index,  
prediccion_10E[:limite],  
color='orange',  
label='10 Épocas')
```

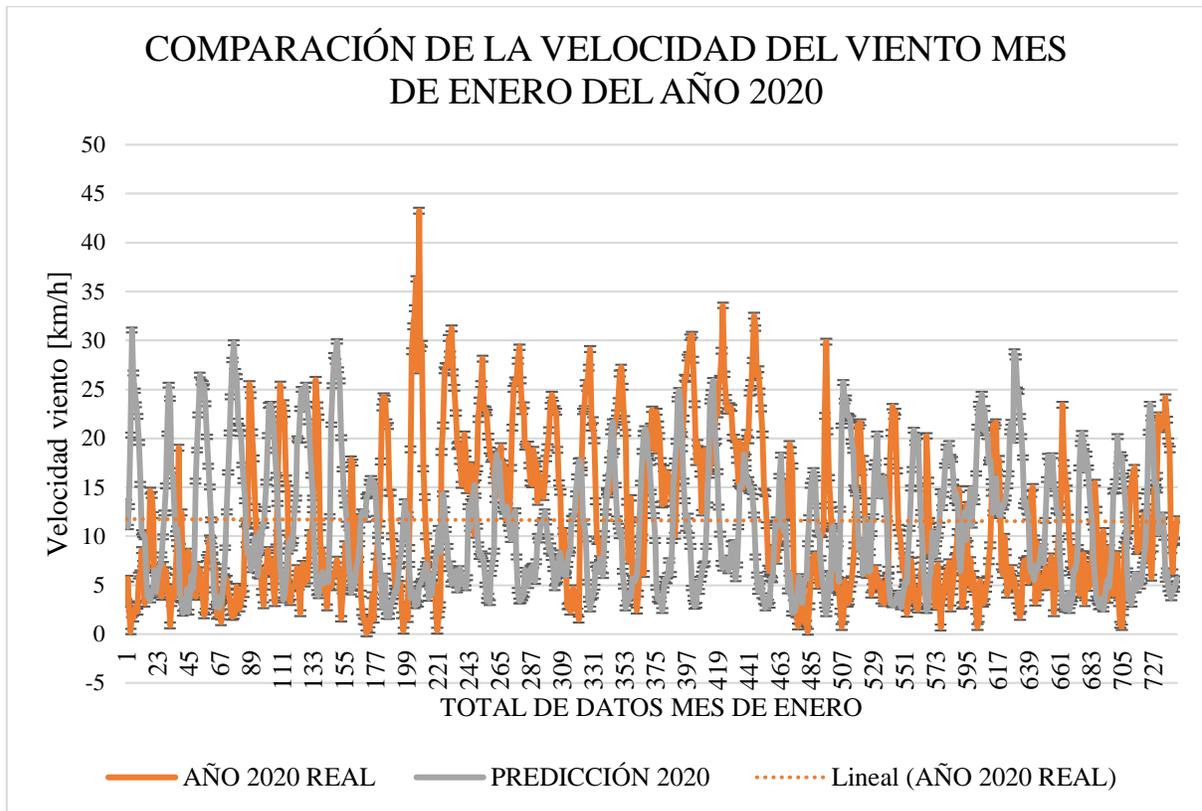
```
plt.plot(semana_estudio_test.index,  
prediccion_15E[:limite],  
color='green',  
label='15 Épocas')
```

```
plt.plot(semana_estudio_test.index,  
prediccion_25E[:limite],  
color='blue',  
label='25 Épocas')
```

```
plt.plot(semana_estudio_test.index,  
prediccion_30E[:limite],  
color='red',  
label='30 Épocas')
```



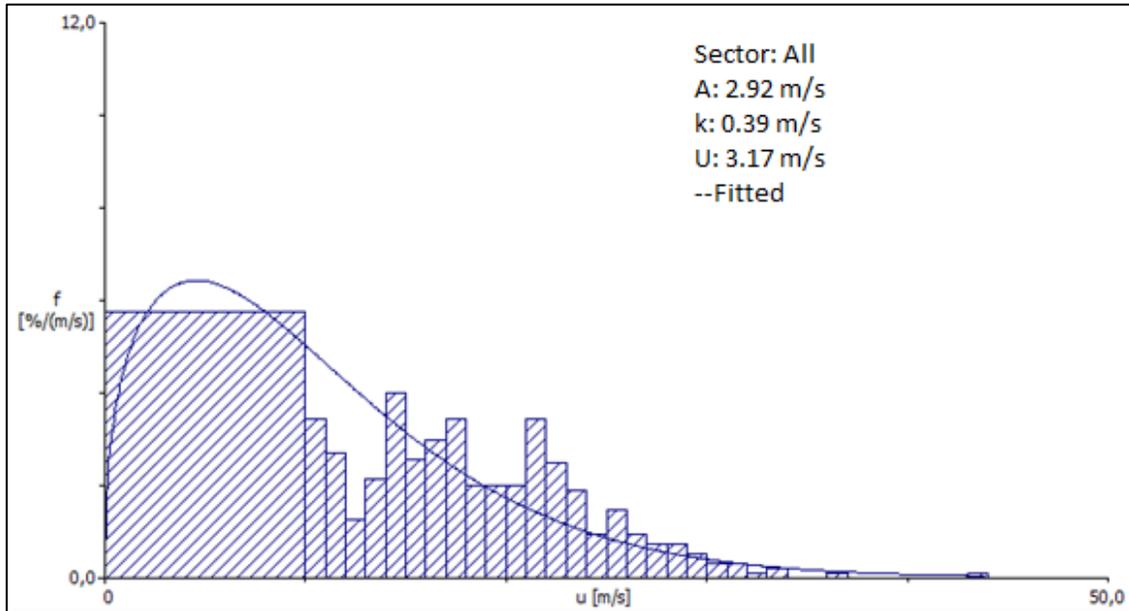
**ANEXO IV. Curva de predicción del mes de enero y datos promedios de cada mes del año 2020**



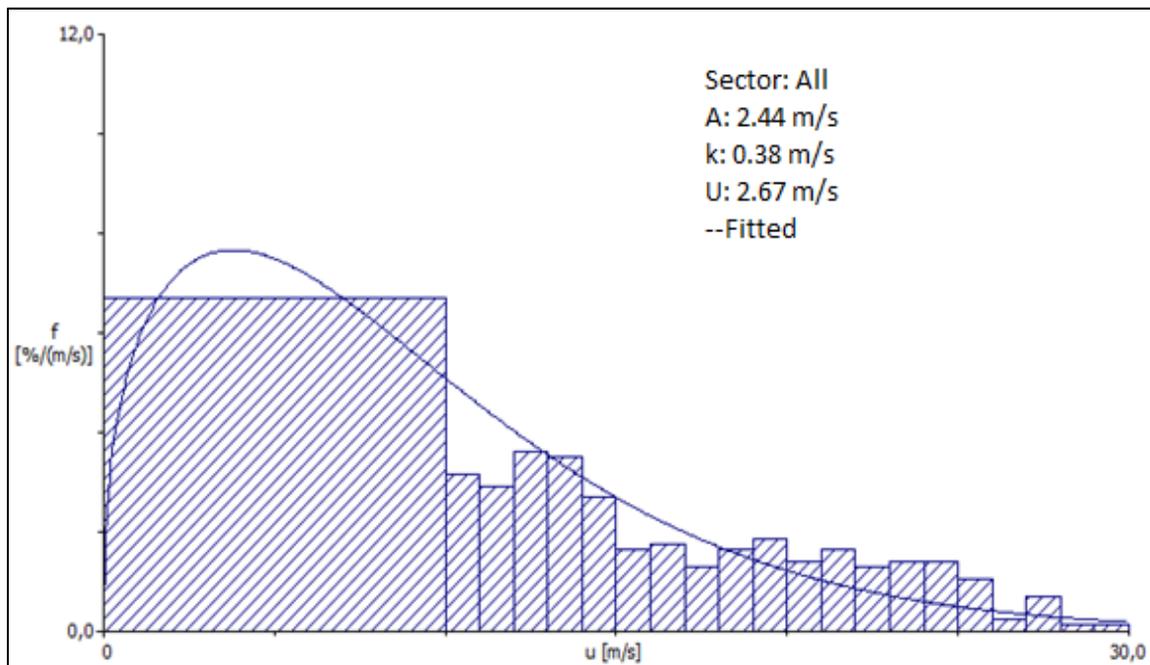
Mes	Velocidad del viento promedio [km/h]	
	Demanda Real	D. Pronosticada
Enero	11.62	11.02
Febrero	9.30	9.28
Marzo	8.54	8.60
Abril	8.47	8.57
Mayo	12.05	12.03
Junio	9.72	9.36
Julio	9.44	9.16
Agosto	11.22	11.24
Septiembre	10.30	10.90
Octubre	10.23	9.40
Noviembre	8.57	9.04
Diciembre	7.30	8.49

### ANEXO V. Velocidad promedio por cada mes del año 2020

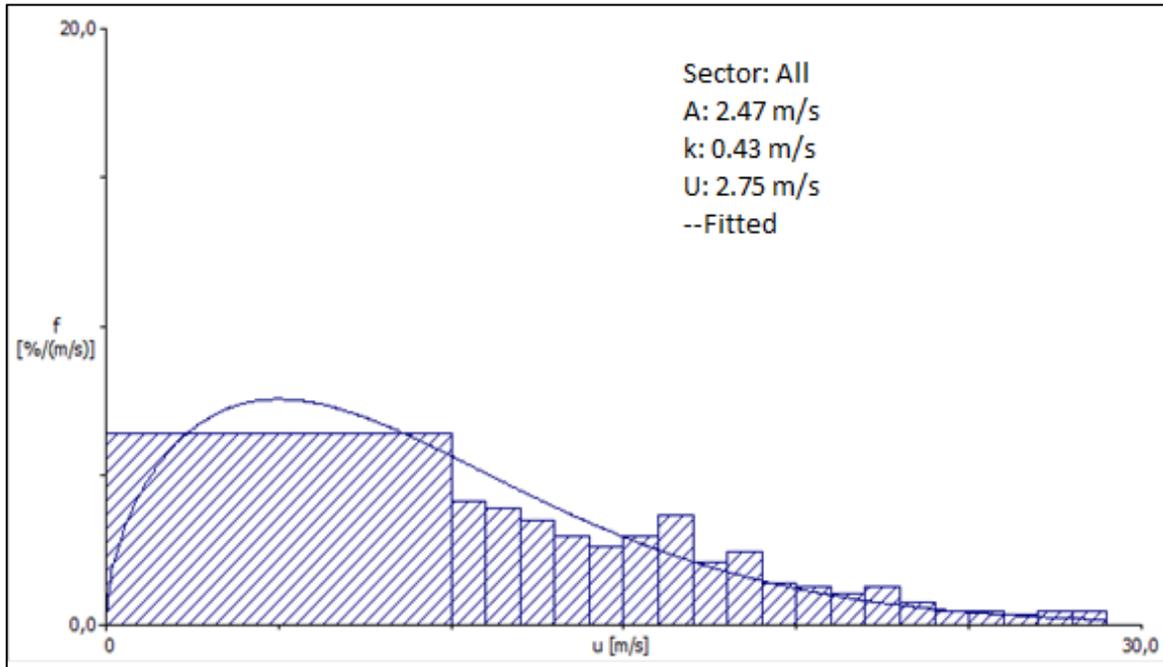
Enero 2020



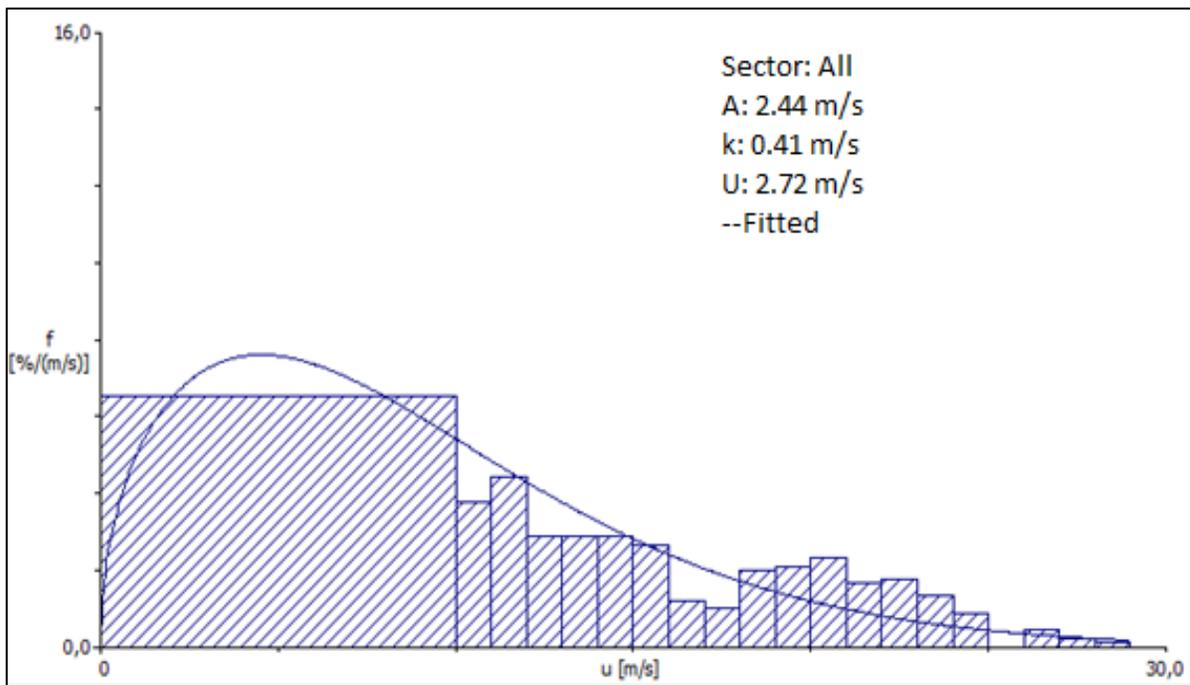
Febrero 2020



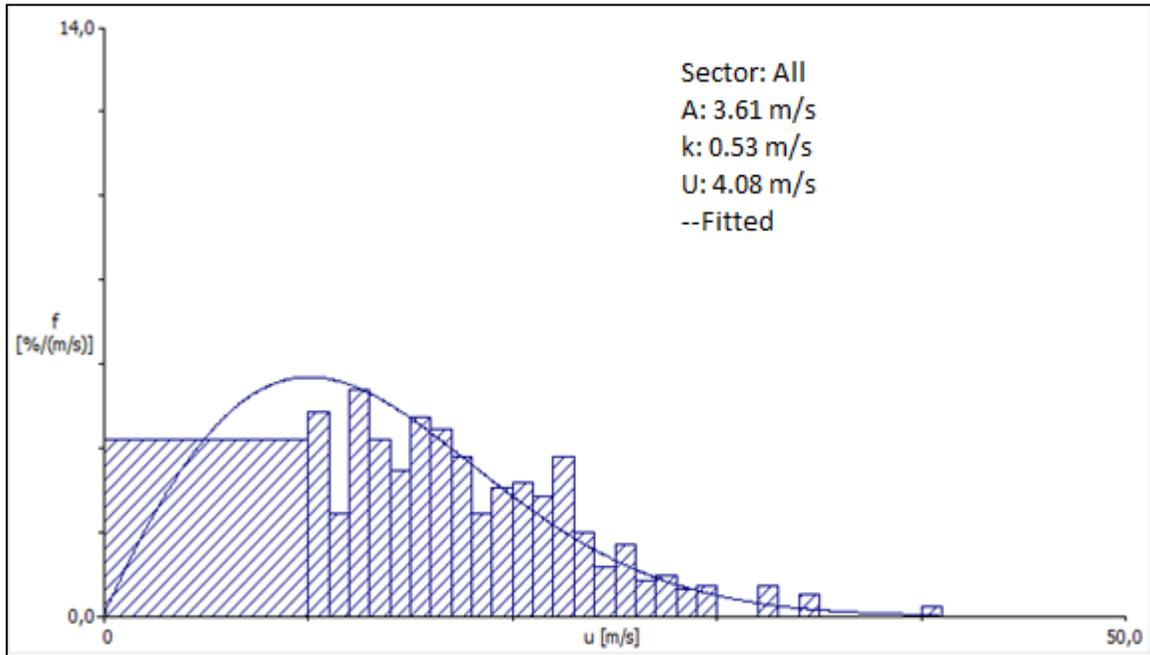
Marzo 2020



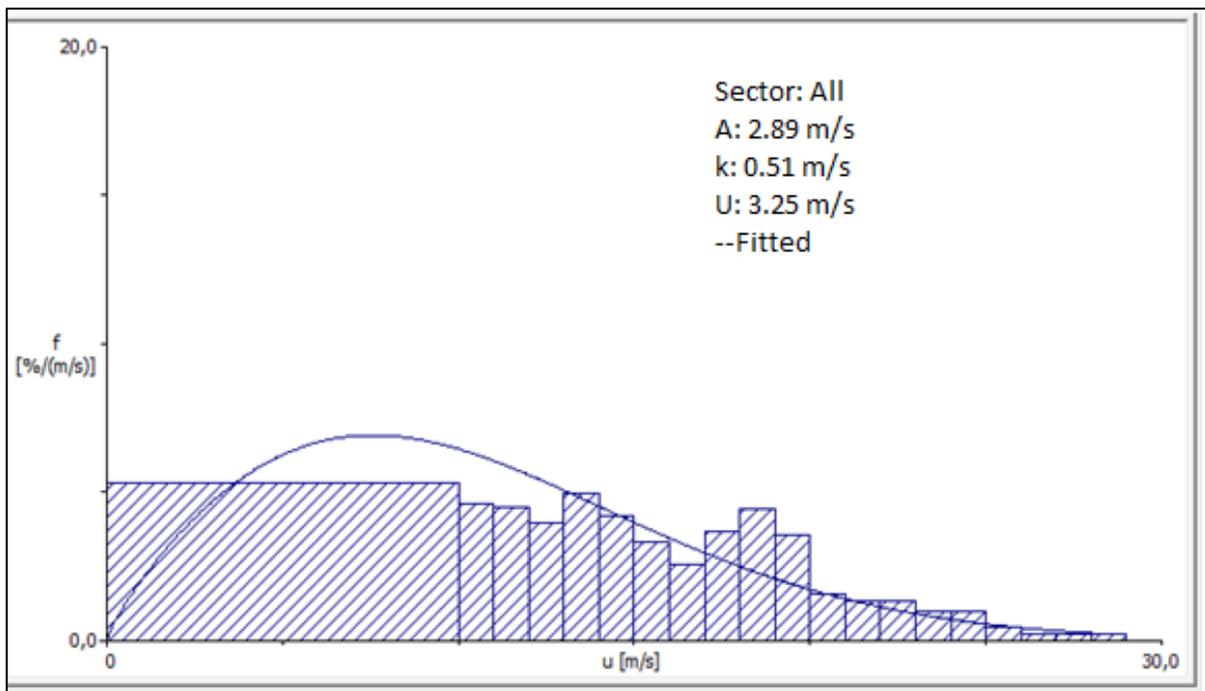
Abril 2020



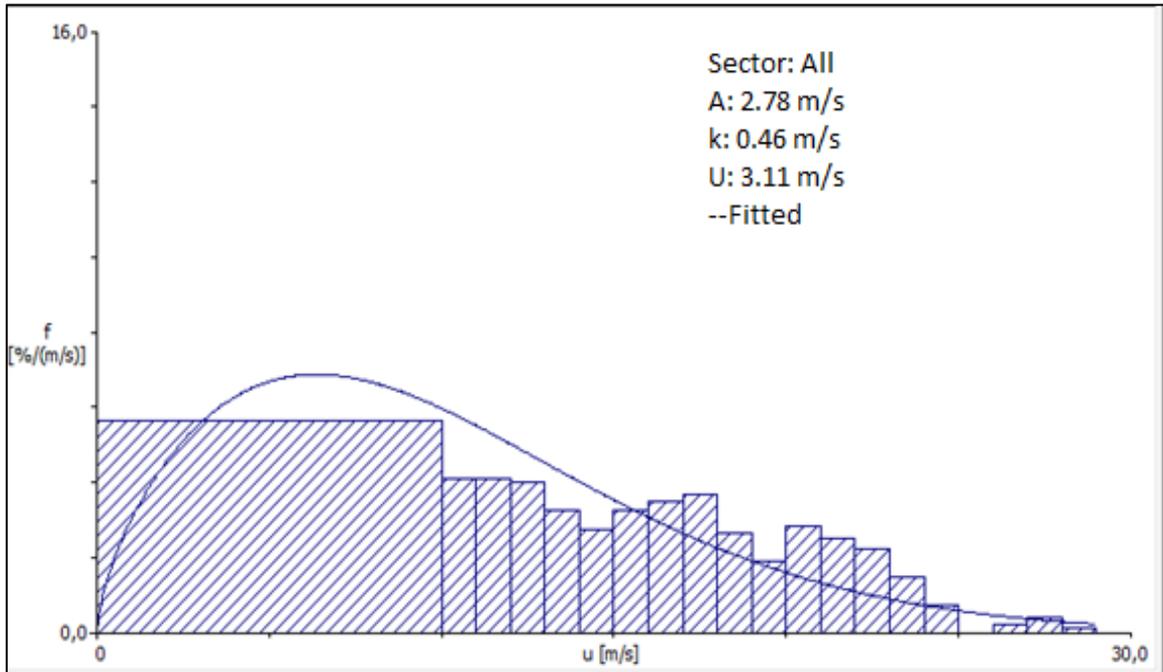
Mayo 2020



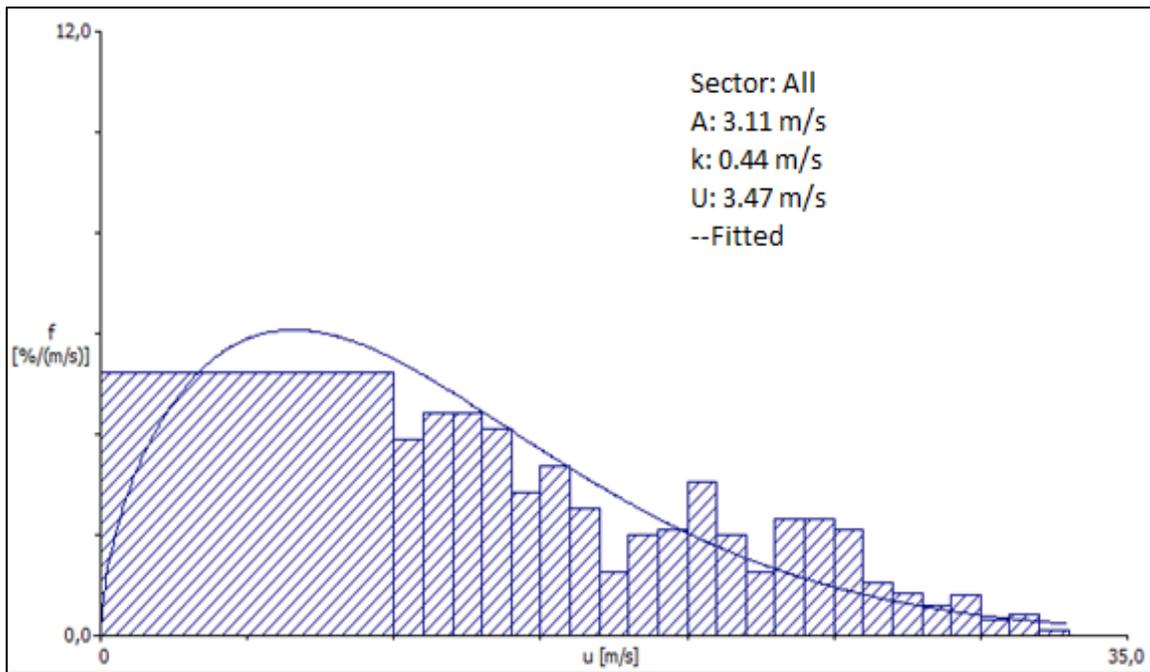
Junio 2020



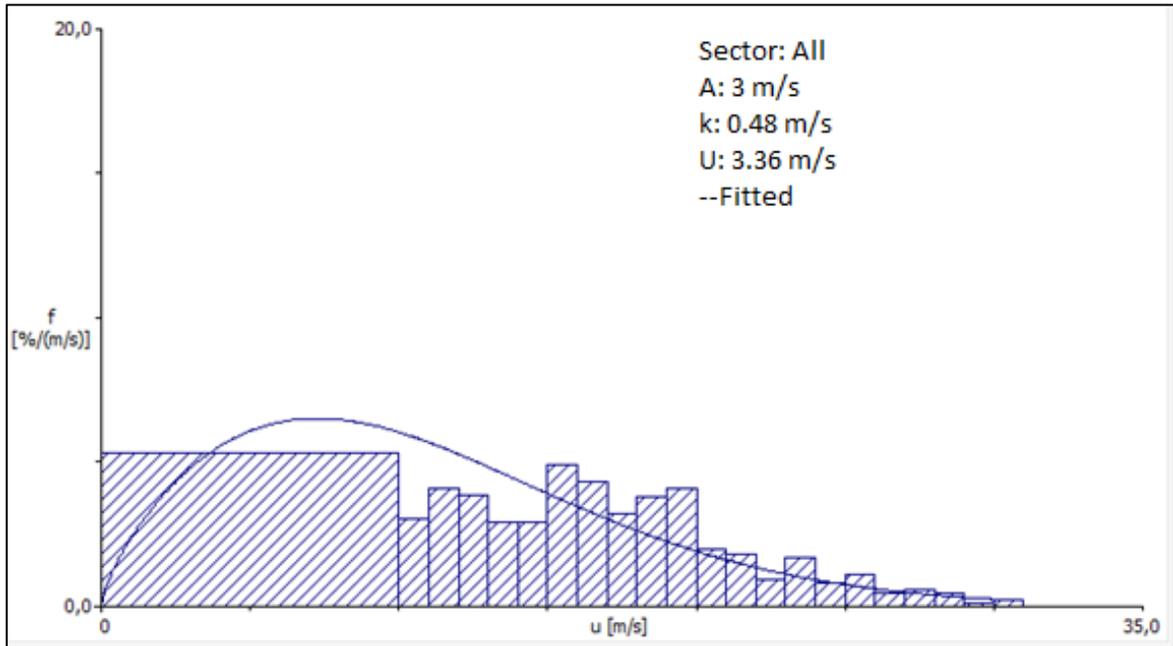
Julio 2020



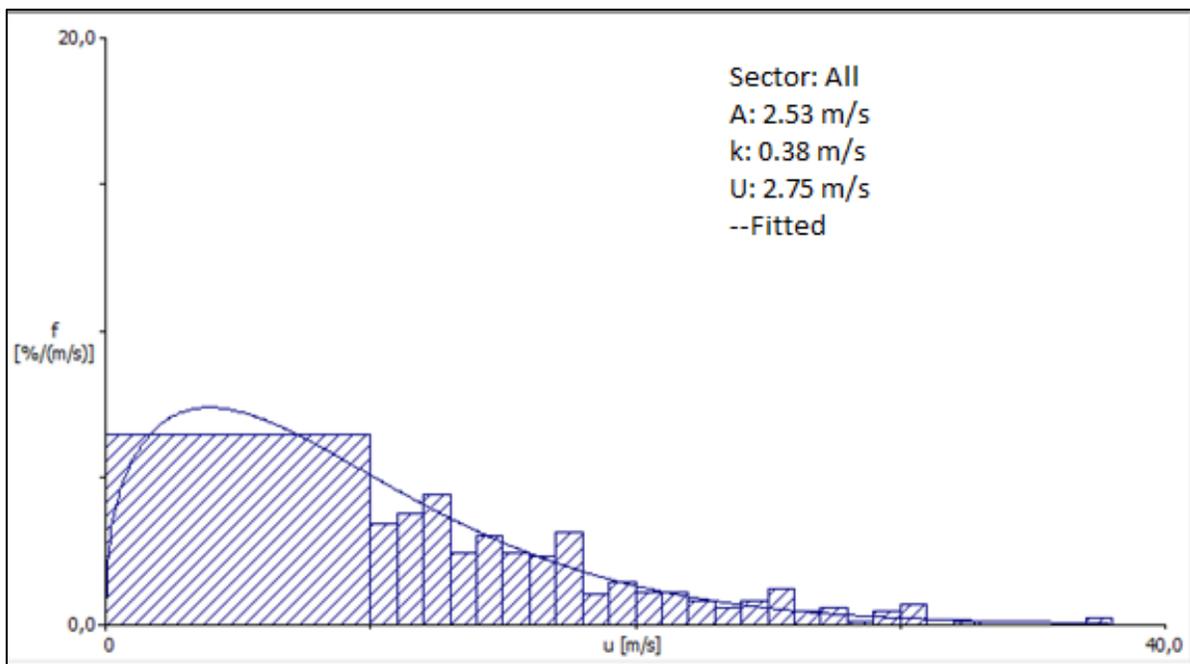
Agosto 2020



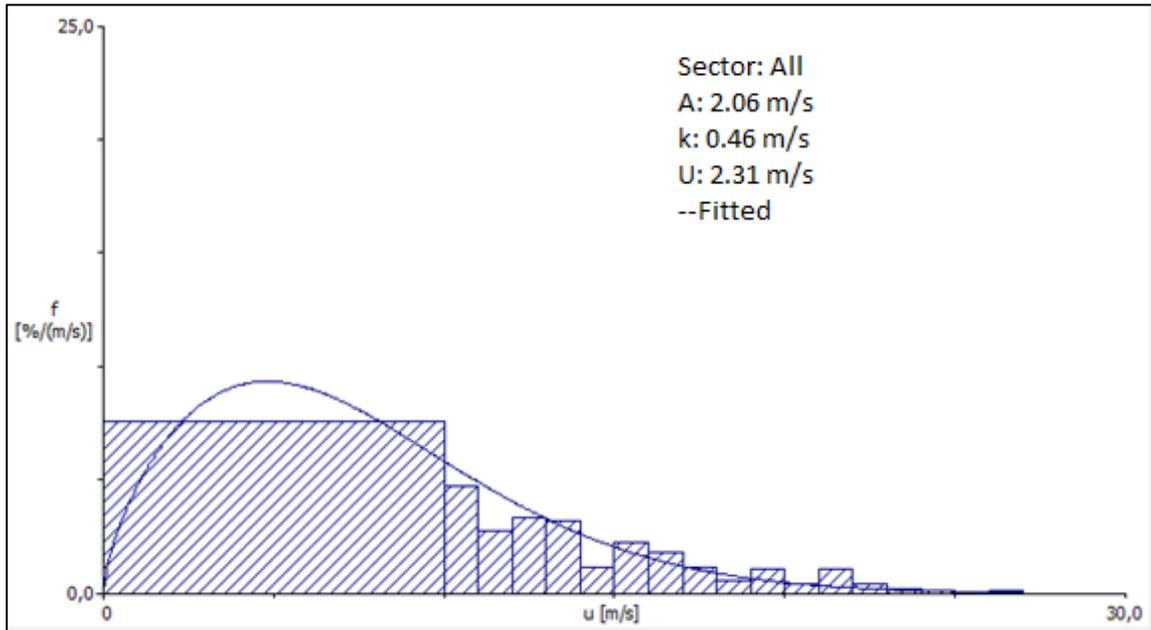
Septiembre 2020



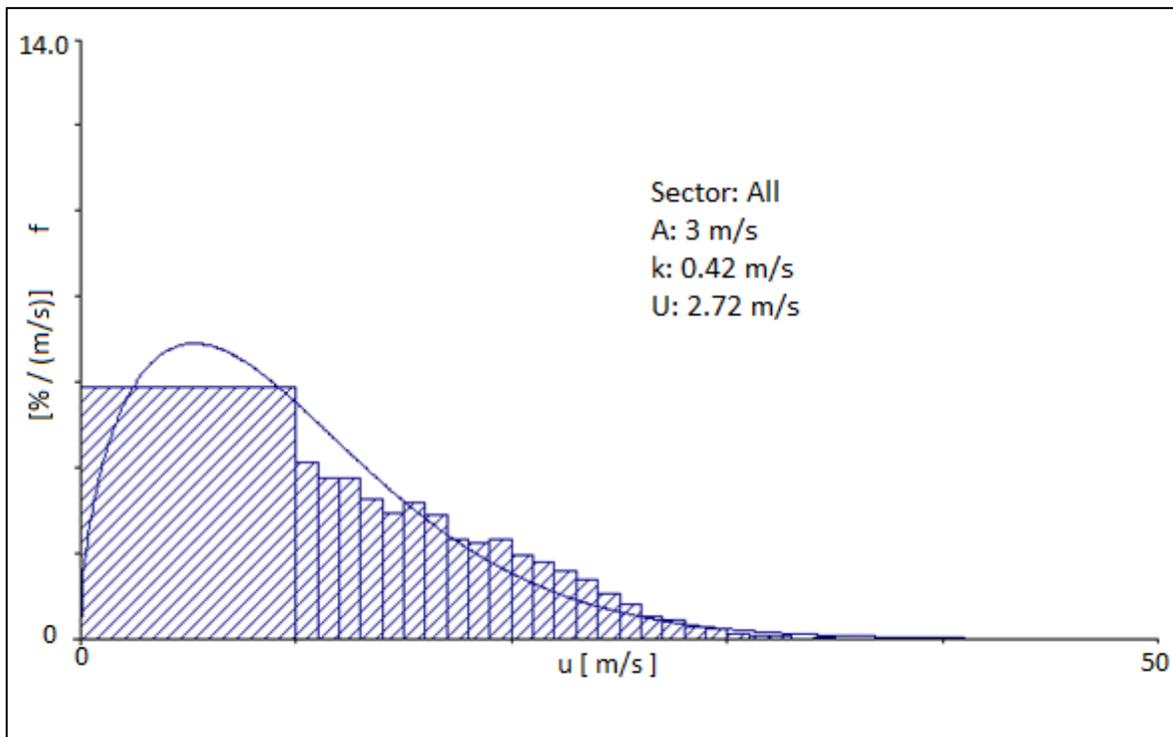
Noviembre 2020



Diciembre 2020



Total 2020



**ANEXO VI. Velocidad promedio del sector de Lasso por enair**









CONSULTORIA ▾ AEROGENERADORES ▾ APLICACIONES ▾ INSTALACIONES ▾ ACTUALIDAD ▾ EMPRESA ▾ PRECIOS

Ajustes:

---

**Modelo de aerogenerador:**

Enair 30PRO ▾

**Rugosidad del terreno:**

Tierra de cultivo campo

**Altura sobre suelo (m):**

1

**Potencia nominal:**

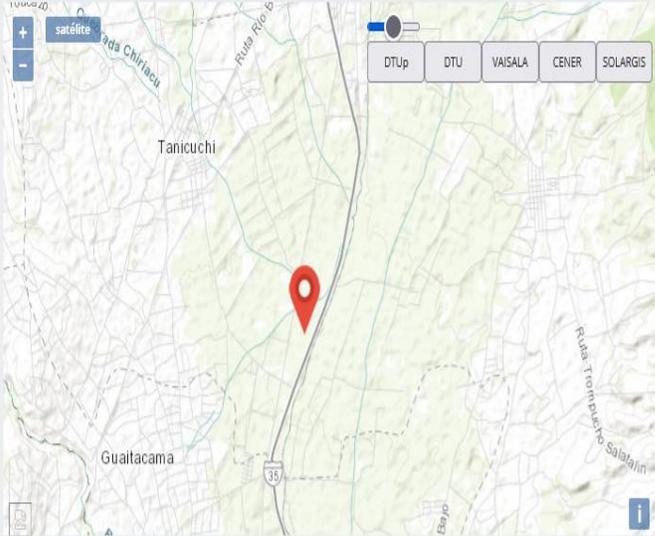
1.9kW

**Potencia máxima:**

3kW

**Área de barrido:**

11.34m<sup>2</sup>



DTUp
DTU
VAISALA
CENER
SOLARGIS

Este atlas eólico y solar combina las principales fuentes de datos internacionales, para determinar con gran exactitud los cálculos derivados del mismo. Las fuentes consultadas son: PVGIS, para energía solar y DTU, MERRA-2, VAISALA y BBDD propia la cual parte con el origen de la NASA corregido. Estas fuentes de datos son cálculos internacionales y además de ellas para cálculos en España, se usa también en combinación la fuente de datos del CENER. No obstante, y dada la complejidad de los cálculos pueden existir márgenes de error en las medias de viento que pueden variar del 1 al 15%, también es importante considerar que no se aplican pérdidas por turbulencias de objetos cercanos.

COORDENADAS MANUALES

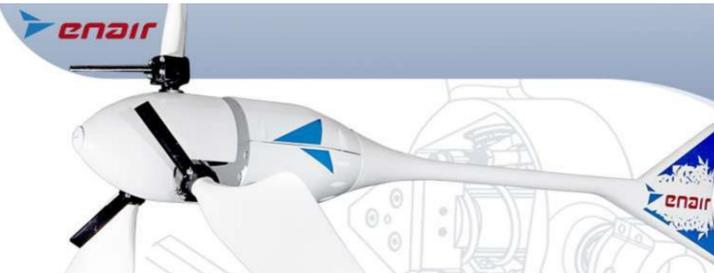
30.0 kWh/día

Vel.media 3.3 m/s

**Velocidad viento a altura de buje**

Si usted conoce la velocidad de viento exacta, puede introducirla manualmente. Esta estimación de viento puede contener un error de hasta el 15%, es importante considerar la posible merma derivada de este error.

ANEXO VII. Ficha técnica del aerogenerador E30PRO



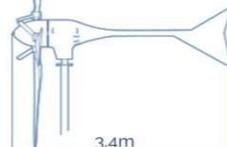
# E30PRO

## FICHA TÉCNICA

Con una velocidad de viento medio situado en la nominal de 11m/s el modelo Enair 30PRO es capaz de generar más de 30kWh/día

### CARACTERÍSTICAS TÉCNICAS, ELÉCTRICAS Y DE FUNCIONAMIENTO

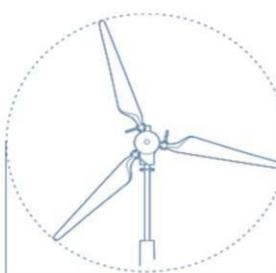
Numero de palas	3
Material de palas	Fibra de vidrio con resinas y núcleo de poliuretano
Generador	250rpm nominales   imanes de neodimio
Potencia	3000W
Potencia Nominal	1900W (según IEC 61400-2)
Tensión	24/48/220V
Clase de viento	CLASS I-IEC 61400-2/NVNI-A
Diámetro	3.80m
Sentido de Giro	Horario
Área de barrido	11.34m <sup>2</sup>
Peso	125kg
Aplicaciones	Carga de baterías 24 o 48V y conexión a red
Viento de arranque	2m/s
Velocidad nominal	11m/s
Vel. regulación paso variable	12m/s
Velocidad de supervivencia	60m/s
Rango de generación eficiente	De 2 a 60m/s
Tipo	Rotor de eje horizontal a barlovento
Orientación	Sistema pasivo con timón de orientación
Control de potencia	Sistema de paso variable pasivo centrífugo con dos vel.
Transmisión	Directa
Freno	- Electromagnético por cortocircuito - Mecánico (opcional) - Aerodinámico mediante paso variable - Manual o automático por viento o por tensión de baterías
Controlador	Carga de baterías y conexión a red
Inversor eólico	Eficiencia 97%, algoritmo MPPT
Ruido	48dB   Reducción al mínimo gracias al diseño de las palas y a las bajas revoluciones. 1% más que el ruido ambiente del viento
Protección anti-corrosión	Hermético, pintura epoxi de secado al horno de alta temperatura, generando un recubrimiento plástico
Torre	Celosía, presilla y tubular, abatibles o fijas, altura variable



3.4m



125kg

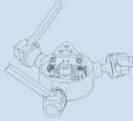


3.80m



## E30PRO Wind Turbine FICHA TÉCNICA

### PASO VARIABLE PASIVO



Tecnología patentada para maximizar la producción de energía. Es un sistema mecánico que gracias a la fuerza centrífuga, modifica el ángulo de las palas y nunca se sobrepasan sus rpm de diseño.

Obteniendo:

- Menos ruido
- Más capacidad de absorber rachas de viento
- Más constancia en la generación
- Más energía con menos viento

### CONTROL ELECTRÓNICO



Sistema de gestión de la energía inteligente

**Conexión de baterías:**

7 tipos de baterías programables (litio, plomo, gel, etc.)  
Cargador con pulsos de derivación a resistencias en caso de sobrecarga. Solo deriva el exceso que no puede cargar para proteger las baterías

**Conexión a red:**

Los inversores MPPT programados con la curva de potencia eólica maximizan la producción en todo momento. Compatibles con redes trifásicas y monofásicas en sistemas europeos y americanos



MAYOR ENERGÍA



MAYOR EFICIENCIA



MAYOR ROBUSTEZ



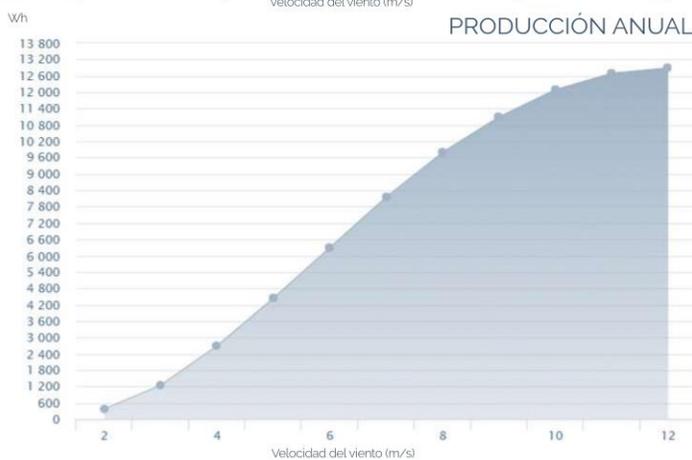
MAYOR SEGURIDAD

## CLASS I WIND TURBINE IEC 61400-2/NVI-A

CURVA DE POTENCIA



PRODUCCIÓN ANUAL



#### Minimo ruido

El ruido esta en torno a un 1% por encima del ruido ambiente siendo prácticamente inapreciable para nuestro oído.

#### Safety-Brake

Nuevo sistema de seguridad mecánica acoplada al eje que garantiza el frenado bajo las condiciones mas adversas, para vientos incluso mayores de 70m/s

#### Anticorrosivo

El uso de pintura epoxy lo convierte en un conjunto anticorrosivo y antisalino ideal para islas y costas

#### Anti-Icing and Hermetic

Uretano acrílico de resina estructural con una composición química antiadherente al hielo y de máxima resistencia para temperaturas de hasta -50°C. Sellado hermético

#### Storm-detection

Algoritmo inteligente de detección de tormentas y bloqueo de seguridad del aerogenerador totalmente automático combinado con el Safety-Brake

#### Remote-Control

Control combinado con el Venus de Victron que permite la marcha/paro del aerogenerador de forma remota

#### BBS (Battery Brake System)

Sistema inteligente que mide el nivel de carga de la batería y permite detener el aerogenerador cuando la batería llega a la tensión de consigna retomando la marcha cuando baja su carga





**ANEXO VIII. Histograma de frecuencia distribución de Weibull.**

Fechas / Horas	Windspeed 2020 (km/h)	Windspeed 2020 (m/s)	Rango mínimo	Rango Max	Promedio	Frecuencia Absoluta	Frecuencia Relativas	Frecuencia Acumulada
1/1/2020 00:00	3.29	0.91	0.00	1.00	0.50	1899.000	0.216	0.216
1/1/2020 01:00	5.65	1.57	1.00	2.00	1.50	2189.000	0.249	0.465
1/1/2020 02:00	3.04	0.85	2.00	3.00	2.50	1403.000	0.160	0.625
1/1/2020 03:00	1.00	0.28	3.00	4.00	3.50	1069.000	0.122	0.747
1/1/2020 04:00	2.02	0.56	4.00	5.00	4.50	892.000	0.102	0.848
1/1/2020 05:00	1.61	0.45	5.00	6.00	5.50	664.000	0.076	0.924
1/1/2020 06:00	2.56	0.71	6.00	7.00	6.50	428.000	0.049	0.973
1/1/2020 07:00	2.88	0.80	7.00	8.00	7.50	174.000	0.020	0.992
1/1/2020 08:00	2.30	0.64	8.00	9.00	8.50	52.000	0.006	0.998
1/1/2020 09:00	5.96	1.66	9.00	10.00	9.50	10.000	0.001	1.000
1/1/2020 10:00	6.20	1.72	10.00	11.00	10.50	2.000	0.000	1.000
1/1/2020 11:00	8.57	2.38	11.00	12.00	11.50	1.000	0.000	1.000
1/1/2020 12:00	7.20	2.00	12.00	13.00	12.50	1.000	0.000	1.000
1/1/2020 13:00	3.13	0.87				8784.00		
1/1/2020 14:00	10.19	2.83						
1/1/2020 15:00	6.59	1.83						
.....	.....	.....						
31/12/2020 18:00	6.952	1.93						
31/12/2020 19:00	6.592	1.83						
31/12/2020 20:00	2.578	0.72						
31/12/2020 21:00	2.412	0.67						
31/12/2020 22:00	1.000	0.28						
31/12/2020 23:00	1.512	0.42						