

 <b>UDECA</b> UNIVERSIDAD DE CUNDINAMARCA	<b>MACROPROCESO DE APOYO</b>	<b>CÓDIGO: AAAR113</b>
	<b>PROCESO GESTIÓN APOYO ACADÉMICO</b>	<b>VERSIÓN: 6</b>
	<b>DESCRIPCIÓN, AUTORIZACIÓN Y LICENCIA DEL REPOSITORIO INSTITUCIONAL</b>	<b>VIGENCIA: 2021-09-14</b>
		<b>PAGINA: 1 de 9</b>

16.

<b>FECHA</b>	Lunes, 02 de enero del 2024
--------------	-----------------------------

Señores  
**UNIVERSIDAD DE CUNDINAMARCA**  
 BIBLIOTECA  
 Facatativa

<b>UNIDAD REGIONAL</b>	Extensión Facatativá
------------------------	----------------------

<b>TIPO DE DOCUMENTO</b>	Trabajo De Grado
--------------------------	------------------

<b>FACULTAD</b>	Ingeniería
-----------------	------------

<b>NIVEL ACADÉMICO DE FORMACIÓN O PROCESO</b>	Pregrado
---	----------

<b>PROGRAMA ACADÉMICO</b>	<b>Ingeniería de Sistemas</b>
---------------------------	-------------------------------

El Autor(Es):

<b>APELLIDOS COMPLETOS</b>	<b>NOMBRES COMPLETOS</b>	<b>No. DOCUMENTO DE IDENTIFICACIÓN</b>
Ruiz Quintero	Vanessa	1000713851
Salamanca Martínez	Juan Diego	1007770477

Director(Es) y/o Asesor(Es) del documento:

<b>APELLIDOS COMPLETOS</b>	<b>NOMBRES COMPLETOS</b>
Morera Zarate	Oscar Javier
Ortega Ortega	Rafael
Jiménez Pinzón	Neil Andrés

### **TÍTULO DEL DOCUMENTO**

Diagonal 18 No. 20-29 Fusagasugá – Cundinamarca  
 Teléfono: (091) 8281483 Línea Gratuita: 018000180414  
[www.ucundinamarca.edu.co](http://www.ucundinamarca.edu.co) E-mail: [info@ucundinamarca.edu.co](mailto:info@ucundinamarca.edu.co)  
 NIT: 890.680.062-2

*Documento controlado por el Sistema de Gestión de la Calidad  
 Asegúrese que corresponde a la última versión consultando el Portal Institucional*

	<b>MACROPROCESO DE APOYO</b>	<b>CÓDIGO: AAAR113</b>
	<b>PROCESO GESTIÓN APOYO ACADÉMICO</b>	<b>VERSIÓN: 6</b>
	<b>DESCRIPCIÓN, AUTORIZACIÓN Y LICENCIA DEL REPOSITORIO INSTITUCIONAL</b>	<b>VIGENCIA: 2021-09-14</b>
		<b>PAGINA: 2 de 9</b>

Diseño e implementación de un prototipo de estación meteorológica autosostenible para el análisis del comportamiento climático, modulo Machine Learning.


**SUBTÍTULO**  
(Aplica solo para Tesis, Artículos Científicos, Disertaciones, Objetos Virtuales de Aprendizaje)

<b>EXCLUSIVO PARA PUBLICACIÓN DESDE LA DIRECCIÓN INVESTIGACIÓN</b>	
<b>INDICADORES</b>	<b>NÚMERO</b>
ISBN	
ISSN	
ISMN	

<b>AÑO DE EDICIÓN DEL DOCUMENTO</b>	<b>NÚMERO DE PÁGINAS</b>
02/02/2024	148

<b>DESCRIPTORES O PALABRAS CLAVES EN ESPAÑOL E INGLÉS</b> (Usar 6 descriptores o palabras claves)	
<b>ESPAÑOL</b>	<b>INGLÉS</b>
1. Datos	Data
2. Aprendizaje automático	Machine learning
3. Sistemas de predicción	Prediction systems
4. Análisis	Analytics
5. Patrones	Patterns
6.	

<b>FUENTES (Todas las fuentes de su trabajo, en orden alfabético)</b>
<p>Bosilovich, M. G. ((2015). ). MERRA-2:. File specification (No. GSFC-E-DAA-TN27096.  Cano Alor, C. G. ( 2020). Diseño de una aplicación de monitoreo de emanación de gases en la ciudad mediante el uso de tecnologías de inteligencia artificial.  Cardozo Vásquez, A. (s.f.). Desarrollo de un modelo de red neuronal artificial para la reducción de escala (downscaling) de datos de temperatura del modelo Climático Global Canadiense 3.1 a estaciones meteorológicas colombianas. (Master's thesis, Universidad de La Sabana.  Cortes, D. (2021). predicción temperatura bogota: enfoque estocástico vs aprendizaje .  Cruz, I. B. (2007). Redes neuronales recurrentes para el análisis de secuencias. . Revista Cubana de Ciencias Informáticas.  Cundinamarca, A. M. (2023 ). Alcaldía Municipal Ubaté Cundinamarca . De la Barrera Castillo, J. C. (2023). Metodologías de ahorro energético aplicadas a los sistemas HVAC</p>

	<b>MACROPROCESO DE APOYO</b>	<b>CÓDIGO: AAAR113</b>
	<b>PROCESO GESTIÓN APOYO ACADÉMICO</b>	<b>VERSIÓN: 6</b>
	<b>DESCRIPCIÓN, AUTORIZACIÓN Y LICENCIA DEL REPOSITORIO INSTITUCIONAL</b>	<b>VIGENCIA: 2021-09-14</b>
		<b>PAGINA: 3 de 9</b>

utilizando inteligencia artificial: una revisión del estado del arte. Deulofeu Gómez, R. (2020). Introducción y optimización estocástica de redes neuronales profundas MLP.

Dillon, J. V. (2017). Tensorflow distributions. arXiv preprint arXiv:.. Egas Arizala, K. O. (2020). Diseño de un modelo predictivo basado .

Fonseca-González, W. V.-C.-V. ((2019). ). Potencial de mitigación del cambio climático de los ecosistemas forestales caducifolios en Costa Rica: modelos predictivos de biomasa y carbono. . Revista de Ciencias Ambientales.

GISTFA. (2022). Protocolo de GISTFA. , 6ta Versión. Universidad de Cundinamarca, .

Guerrero-Álvarez, J. A. (2021). Predicción niveles de precipitación para la identificación de zonas agrícolas . (Master's thesis).

Hao, J. &. ((2019). ). Machine learning made easy: a review of scikit-learn package in python programming language. Journal of Educational and Behavioral Statistics.

Mahesh, B. ((2020)). Machine learning algorithms-a review. International Journal of Science and Research (IJSR).

Montes, C. (2006). Sobreajuste-overfitting. centro de Investigaciones Energicas, Medioambientales y Tecnologicas.

Montesdeoca Santana, B. (2016). Estudios de predicción en series temporales de datos meteorológicos utilizando redes neuronales recurrentes. Bachelor's thesis.

Murcia Ramirez, J. S. (s.f.). Modelo de máxima entropía para la predicción de criminalidad con tecnicas de Machine Learning .

Murcia, D. S. (2015). .Diseño e implementación de un prototipo de estación . Paniagua Urquijo, D. &. ( 2021). Diseño, desarrollo e implementacion de una plataforma web de datos climatológicos mediante una boya construida en un arduino en el municipio de Guataqui Cundinamarca .

Python, W. (2021). Python. . Python Releases for Windows, 24

. Randles, B. M. ( 2017 ). Using the Jupyter notebook as a tool for open science: An empirical study. ACM/IEEE Joint Conference on Digital Libraries (JCDL) (pp. 1 2). IEEE.

Rodríguez., &. D. ( 2015). ¿Por qué implementar Scrum? Why to implement . Revista Ontare, 3(1), 125–144.

Ruiz Palomo, G. J. ( 2021). Klimachingaza: visor geográfico web .

Sommerville. (2005). Ingenieria de software. pearson educacion .

Vázquez Ruiz, T. (2023). Desarrollo y aplicación de metodología de modelado predictivo de series temporales.

Villavicencio, J. ( 2010). Introducción a series de tiempo. . Puerto Rico.

Yana Puma, J. W. (2020). Modelo predictivo agroclimático para detectar cambios climatologicos con machine learning e IoT para el sector agrario de la región de Pun.

Bosilovich, M. G. ((2015). ). MERRA-2:. File specification (No. GSFC-E-DAA-TN27096.

Cano Alor, C. G. ( 2020). Diseño de una aplicación de monitoreo de emanación de gases en la ciudad mediante el uso de tecnologías de inteligencia artificial.


Cardozo Vásquez, A. (s.f.). Desarrollo de un modelo de red neuronal artificial para la reducción de escala (downscaling) de datos de temperatura del modelo Climático Global Canadiense 3.1 a estaciones meteorológicas colombianas. (Master's thesis, Universidad de La Sabana.

Cortes, D. (2021). predicción temperatura bogota: enfoque estocástico vs aprendizaje .

Cruz, I. B. (2007). Redes neuronales recurrentes para el análisis de secuencias. . Revista Cubana de Ciencias Informáticas.

Cundinamarca, A. M. (2023). Alcaldía Municipal Ubaté Cundinamarca .

De la Barrera Castillo, J. C. (2023). Metodologías de ahorro energético aplicadas a los sistemas HVAC utilizando inteligencia artificial: una revisión del estado del arte.

	<b>MACROPROCESO DE APOYO</b>	<b>CÓDIGO: AAAR113</b>
	<b>PROCESO GESTIÓN APOYO ACADÉMICO</b>	<b>VERSIÓN: 6</b>
	<b>DESCRIPCIÓN, AUTORIZACIÓN Y LICENCIA DEL REPOSITORIO INSTITUCIONAL</b>	<b>VIGENCIA: 2021-09-14</b>
		<b>PAGINA: 4 de 9</b>

Deulofeu Gómez, R. (2020). Introducción y optimización estocástica de redes neuronales profundas MLP.

Fonseca-González, W. V.-C.-V. ((2019). ). Potencial de mitigación del cambio climático de los ecosistemas forestales caducifolios en Costa Rica: modelos predictivos de biomasa y carbono. . Revista de Ciencias Ambientales.

Murcia Ramirez, J. S. (s.f.). Modelo de máxima entropía para la predicción de criminalidad con técnicas de Machine Learning .

## RESUMEN DEL CONTENIDO EN ESPAÑOL E INGLÉS

(Máximo 250 palabras – 1530 caracteres, aplica para resumen en español):

El pronóstico preciso del clima a corto, mediano y largo plazo se ha vuelto un objetivo de creciente importancia mundial debido a su impacto en las actividades humanas. Los estudios sobre el calentamiento global y el cambio climático, que muestran escenarios adversos para la biodiversidad, han intensificado el interés en reforzar la capacidad de predecir con mayor exactitud las variaciones climáticas espaciales y temporales. Esta necesidad se acentúa ante el aumento en la frecuencia de eventos extremos como sequías, inundaciones e incendios forestales. Considerando la influencia del clima en diversos factores sociales, se describe en el presente documento el desarrollo de un modelo de predicción meteorológica utilizando técnicas de aprendizaje automático (machine learning). El objetivo es generar pronósticos precisos de variables climáticas mediante el entrenamiento de redes neuronales con datos históricos, bajo la metodología de investigación cuantitativa, empleando registros de datos para entrenar el modelo y demostrando resultados. Se adopta Scrum para un desarrollo colaborativo y focalizado en objetivos. Para modelar las relaciones temporales en las series de tiempo meteorológicas, se implementó una red neuronal recurrente LSTM, adecuada para capturar dependencias a largo plazo. El modelo fue entrenado con un extenso conjunto de datos de múltiples parámetros climáticos de estaciones meteorológicas y del conjunto MERRA-2 de la NASA y en el que se aplicaron técnicas de preprocesamiento y normalización de los datos, así como validación cruzada, para mejorar el aprendizaje y evitar problemas como el sobreajuste. La 8 precisión del modelo, medida con métricas como el error cuadrático medio, mejoró constantemente al incrementar las épocas de entrenamiento. Esto demuestra la capacidad del modelo para aprender relaciones complejas. Los resultados obtenidos indican que las técnicas de machine learning, particularmente las redes LSTM, tienen gran potencial para generar modelos predictivos meteorológicos precisos y confiables. Este trabajo representa una contribución relevante sobre el uso de inteligencia artificial para la comprensión y predicción de fenómenos climáticos complejos.

The accurate forecasting of climate in the short, medium and long term has become an objective of growing global importance due to its impact on human activities. Studies on global warming and climate change, which show adverse scenarios for biodiversity, have

 <b>UDECA</b> UNIVERSIDAD DE CUNDINAMARCA	<b>MACROPROCESO DE APOYO</b>	<b>CÓDIGO: AAAR113</b>
	<b>PROCESO GESTIÓN APOYO ACADÉMICO</b>	<b>VERSIÓN: 6</b>
	<b>DESCRIPCIÓN, AUTORIZACIÓN Y LICENCIA DEL REPOSITORIO INSTITUCIONAL</b>	<b>VIGENCIA: 2021-09-14</b>
		<b>PAGINA: 5 de 9</b>

intensified interest in reinforcing the ability to predict spatial and temporal climate variations more accurately. This need is accentuated by the increase in the frequency of extreme events such as droughts, floods and forest fires. Considering the influence of climate on various social factors, the present document describes the development of a weather forecasting model using machine learning techniques. The objective is to generate accurate forecasts of climate variables through the training of neural networks with historical data, under a quantitative research methodology, using data records to train the model and demonstrate results. Scrum is adopted for collaborative and objective-focused development. To model temporal relationships in meteorological time series, a recurrent LSTM neural network was implemented, suitable for capturing long-term dependencies. The model was trained with an extensive dataset of multiple climate parameters from weather stations and NASA's MERRA-2 dataset. Data preprocessing and normalization techniques were applied, as well as cross validation, to improve learning and avoid problems such as overfitting. The accuracy of the model, measured with metrics such as the mean square error, improved steadily as the training epochs increased. This demonstrates the model's ability to learn complex relationships. The results obtained indicate that machine learning techniques, particularly LSTM networks, have great potential to generate accurate and reliable weather predictive models. This work represents a relevant contribution on the use of artificial intelligence for understanding and predicting complex climate phenomena.

### AUTORIZACIÓN DE PUBLICACIÓN

Por medio del presente escrito autorizo (Autorizamos) a la Universidad de Cundinamarca para que, en desarrollo de la presente licencia de uso parcial, pueda ejercer sobre mí (nuestra) obra las atribuciones que se indican a continuación, teniendo en cuenta que, en cualquier caso, la finalidad perseguida será facilitar, difundir y promover el aprendizaje, la enseñanza y la investigación.

En consecuencia, las atribuciones de usos temporales y parciales que por virtud de la presente licencia se autoriza a la Universidad de Cundinamarca, a los usuarios de la Biblioteca de la Universidad; así como a los usuarios de las redes, bases de datos y demás sitios web con los que la Universidad tenga perfeccionado una alianza, son: Marque con una "X":

AUTORIZO (AUTORIZAMOS)	SI	NO
1. La reproducción por cualquier formato conocido o por conocer.	X	
2. La comunicación pública, masiva por cualquier procedimiento o medio físico, electrónico y digital.	X	

 <b>UDECA</b> UNIVERSIDAD DE CUNDINAMARCA	<b>MACROPROCESO DE APOYO</b>	<b>CÓDIGO: AAAR113</b>
	<b>PROCESO GESTIÓN APOYO ACADÉMICO</b>	<b>VERSIÓN: 6</b>
	<b>DESCRIPCIÓN, AUTORIZACIÓN Y LICENCIA DEL REPOSITORIO INSTITUCIONAL</b>	<b>VIGENCIA: 2021-09-14</b>
		<b>PAGINA: 6 de 9</b>


3. La inclusión en bases de datos y en sitios web sean éstos onerosos o gratuitos, existiendo con ellos previa alianza perfeccionada con la Universidad de Cundinamarca para efectos de satisfacer los fines previstos. En este evento, tales sitios y sus usuarios tendrán las mismas facultades que las aquí concedidas con las mismas limitaciones y condiciones.	X	
4. La inclusión en el Repositorio Institucional.	X	

De acuerdo con la naturaleza del uso concedido, la presente licencia parcial se otorga a título gratuito por el máximo tiempo legal colombiano, con el propósito de que en dicho lapso mi (nuestra) obra sea explotada en las condiciones aquí estipuladas y para los fines indicados, respetando siempre la titularidad de los derechos patrimoniales y morales correspondientes, de acuerdo con los usos honrados, de manera proporcional y justificada a la finalidad perseguida, sin ánimo de lucro ni de comercialización.

Para el caso de las Tesis, Trabajo de Grado o Pasantía, de manera complementaria, garantizo(garantizamos) en mi(nuestra) calidad de estudiante(s) y por ende autor(es) exclusivo(s), que la Tesis, Trabajo de Grado o Pasantía en cuestión, es producto de mi(nuestra) plena autoría, de mi(nuestro) esfuerzo personal intelectual, como consecuencia de mi(nuestra) creación original particular y, por tanto, soy(somos) el(los) único(s) titular(es) de la misma. Además, aseguro (aseguramos) que no contiene citas, ni transcripciones de otras obras protegidas, por fuera de los límites autorizados por la ley, según los usos honrados, y en proporción a los fines previstos; ni tampoco contempla declaraciones difamatorias contra terceros; respetando el derecho a la imagen, intimidad, buen nombre y demás derechos constitucionales. Adicionalmente, manifiesto (manifestamos) que no se incluyeron expresiones contrarias al orden público ni a las buenas costumbres. En consecuencia, la responsabilidad directa en la elaboración, presentación, investigación y, en general, contenidos de la Tesis o Trabajo de Grado es de mí (nuestra) competencia exclusiva, eximiendo de toda responsabilidad a la Universidad de Cundinamarca por tales aspectos.

Sin perjuicio de los usos y atribuciones otorgadas en virtud de este documento, continuaré (continuaremos) conservando los correspondientes derechos patrimoniales sin modificación o restricción alguna, puesto que, de acuerdo con la legislación colombiana aplicable, el presente es un acuerdo jurídico que en ningún caso conlleva la enajenación de los derechos patrimoniales derivados del régimen del Derecho de Autor.

De conformidad con lo establecido en el artículo 30 de la Ley 23 de 1982 y el artículo 11 de la Decisión Andina 351 de 1993, *“Los derechos morales sobre el trabajo son propiedad de los autores”*, los cuales son irrenunciables, imprescriptibles, inembargables e inalienables. En consecuencia, la Universidad de Cundinamarca

	<b>MACROPROCESO DE APOYO</b>	<b>CÓDIGO: AAAR113</b>
	<b>PROCESO GESTIÓN APOYO ACADÉMICO</b>	<b>VERSIÓN: 6</b>
	<b>DESCRIPCIÓN, AUTORIZACIÓN Y LICENCIA DEL REPOSITORIO INSTITUCIONAL</b>	<b>VIGENCIA: 2021-09-14</b>
		<b>PAGINA: 7 de 9</b>

está en la obligación de RESPETARLOS Y HACERLOS RESPETAR, para lo cual tomará las medidas correspondientes para garantizar su observancia.

**NOTA:** (Para Tesis, Trabajo de Grado o Pasantía):

**Información Confidencial:**

Esta Tesis, Trabajo de Grado o Pasantía, contiene información privilegiada, estratégica, secreta, confidencial y demás similar, o hace parte de la investigación que se adelanta y cuyos resultados finales no se han publicado.

**SI** \_\_\_ **NO** **X**.

En caso afirmativo expresamente indicaré (indicaremos) en carta adjunta, expedida por la entidad respectiva, la cual informa sobre tal situación, lo anterior con el fin de que se mantenga la restricción de acceso.

**LICENCIA DE PUBLICACIÓN**

Como titular(es) del derecho de autor, confiero(erimos) a la Universidad de Cundinamarca una licencia no exclusiva, limitada y gratuita sobre la obra que se integrará en el Repositorio Institucional, que se ajusta a las siguientes características:

a) Estará vigente a partir de la fecha de inclusión en el repositorio, por un plazo de 5 años, que serán prorrogables indefinidamente por el tiempo que dure el derecho patrimonial del autor. El autor podrá dar por terminada la licencia solicitándolo a la Universidad por escrito. (Para el caso de los Recursos Educativos Digitales, la Licencia de Publicación será permanente).

b) Autoriza a la Universidad de Cundinamarca a publicar la obra en formato y/o soporte digital, conociendo que, dado que se publica en Internet, por este hecho circula con un alcance mundial.

c) Los titulares aceptan que la autorización se hace a título gratuito, por lo tanto, renuncian a recibir beneficio alguno por la publicación, distribución, comunicación pública y cualquier otro uso que se haga en los términos de la presente licencia y de la licencia de uso con que se publica.

d) El(Los) Autor(es), garantizo(amos) que el documento en cuestión es producto de mi(nuestra) plena autoría, de mi(nuestro) esfuerzo personal intelectual, como consecuencia de mi (nuestra) creación original particular y, por tanto, soy(somos) el(los) único(s) titular(es) de la misma. Además, aseguro(aseguramos) que no contiene citas, ni transcripciones de otras obras protegidas, por fuera de los límites autorizados por la ley, según los usos honrados, y en proporción a los fines previstos; ni tampoco contempla declaraciones difamatorias contra terceros; respetando el derecho a la imagen, intimidad, buen nombre y demás derechos constitucionales.

Diagonal 18 No. 20-29 Fusagasugá – Cundinamarca

Teléfono: (091) 8281483 Línea Gratuita: 018000180414

[www.ucundinamarca.edu.co](http://www.ucundinamarca.edu.co) E-mail: [info@ucundinamarca.edu.co](mailto:info@ucundinamarca.edu.co)

NIT: 890.680.062-2

	<b>MACROPROCESO DE APOYO</b>	<b>CÓDIGO: AAAR113</b>
	<b>PROCESO GESTIÓN APOYO ACADÉMICO</b>	<b>VERSIÓN: 6</b>
	<b>DESCRIPCIÓN, AUTORIZACIÓN Y LICENCIA DEL REPOSITORIO INSTITUCIONAL</b>	<b>VIGENCIA: 2021-09-14</b>
		<b>PAGINA: 8 de 9</b>

Adicionalmente, manifiesto (manifestamos) que no se incluyeron expresiones contrarias al orden público ni a las buenas costumbres. En consecuencia, la responsabilidad directa en la elaboración, presentación, investigación y, en general, contenidos es de mí (nuestro) competencia exclusiva, eximiendo de toda responsabilidad a la Universidad de Cundinamarca por tales aspectos.

e) En todo caso la Universidad de Cundinamarca se compromete a indicar siempre la autoría incluyendo el nombre del autor y la fecha de publicación.

f) Los titulares autorizan a la Universidad para incluir la obra en los índices y buscadores que estimen necesarios para promover su difusión.

g) Los titulares aceptan que la Universidad de Cundinamarca pueda convertir el documento a cualquier medio o formato para propósitos de preservación digital.

h) Los titulares autorizan que la obra sea puesta a disposición del público en los términos autorizados en los literales anteriores bajo los límites definidos por la universidad en el “Manual del Repositorio Institucional AAAM003”

i) Para el caso de los Recursos Educativos Digitales producidos por la Oficina de Educación Virtual, sus contenidos de publicación se rigen bajo la Licencia Creative Commons: Atribución- No comercial- Compartir Igual.



j) Para el caso de los Artículos Científicos y Revistas, sus contenidos se rigen bajo la Licencia Creative Commons Atribución- No comercial- Sin derivar.




**Nota:**

Si el documento se basa en un trabajo que ha sido patrocinado o apoyado por una entidad, con excepción de Universidad de Cundinamarca, los autores garantizan que se ha cumplido con los derechos y obligaciones requeridos por el respectivo contrato o acuerdo.

La obra que se integrará en el Repositorio Institucional está en el(los) siguiente(s) archivo(s).

<b>Nombre completo del Archivo Incluida su Extensión (Ej. Nombre completo del proyecto.pdf)</b>	<b>Tipo de documento (ej. Texto, imagen, video, etc.)</b>
1. Diseño e implementación de un prototipo de estación meteorológica autosostenible	Texto

	<b>MACROPROCESO DE APOYO</b>	<b>CÓDIGO: AAAR113</b>
	<b>PROCESO GESTIÓN APOYO ACADÉMICO</b>	<b>VERSIÓN: 6</b>
	<b>DESCRIPCIÓN, AUTORIZACIÓN Y LICENCIA DEL REPOSITORIO INSTITUCIONAL</b>	<b>VIGENCIA: 2021-09-14</b>
		<b>PAGINA: 9 de 9</b>

para el análisis del comportamiento climático, modulo Machine Learning.pdf	
2.	
3.	
4.	

En constancia de lo anterior, Firmo (amos) el presente documento:

<b>APELLIDOS Y NOMBRES COMPLETOS</b>	<b>FIRMA (autógrafo)</b>
Ruiz Quintero Vanessa	<i>Vanessa Ruiz Quintero</i>
Salamanca Martinez Juan Diego	<i>Juan Salamanca.</i>

21.1-51-20.

**DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE ESTACIÓN  
METEOROLÓGICA AUTOSOSTENIBLE PARA EL ANÁLISIS DEL  
COMPORTAMIENTO CLIMÁTICO**

**- MODULO MACHINE LEARNING-**

**AUTOR(ES)**

**VANESSA RUIZ QUINTERO**

**JUAN DIEGO SALAMANCA MARTINEZ**

**UNIVERSIDAD DE CUNDINAMARCA**

**Facultad de Ingeniería**

**Programa de Ingeniería de Sistemas**

**Facatativá, 2022**

**DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE ESTACIÓN  
METEOROLÓGICA AUTOSOSTENIBLE PARA EL ANÁLISIS DEL  
COMPORTAMIENTO CLIMÁTICO**

**- MODULO MACHINE LEARNING-**

**AUTOR(ES)**

**DIRECTOR(A): ING. OSCAR JAVIER MORERA ZARATE**

**VANESSA RUIZ QUINTERO**

**JUAN DIEGO SALAMANCA MARTINEZ**

**GRUPO DE INVESTIGACIÓN DE SISTEMAS Y TECNOLOGÍA DE  
FACATATIVÁ**

**(GISTFA)**

**UNIVERSIDAD DE CUNDINAMARCA**

**Facultad de Ingeniería**

**Programa de Ingeniería de Sistemas**

**Facatativá, 2023**

Nota de Aceptación

---

---

---

---

---

Presidente jurado

---

Jurado

---

Jurado

Facatativá, 2023

### **COMPROMISO DE AUTOR**

Yo, Vanessa Ruiz Quintero con cédula de identidad No 1000713851 y con cód. 461218271 estudiante del programa de Ingeniería de Sistemas de la Universidad de Cundinamarca, declaro que:

El contenido del presente documento es un reflejo de mi trabajo personal y manifiesto que, ante cualquier notificación de plagio, copia o falta a la fuente original, soy responsable directo legal, económico y administrativo sin afectar al director del trabajo, a la Universidad y a cuantas instituciones han colaborado en dicho trabajo, asumiendo las consecuencias derivadas de tales prácticas.

Firma:

Vanessa Ruiz Quintero

Facatativá, 2023

### **COMPROMISO DE AUTOR**

Yo, Juan Diego Salamanca Martínez con cédula de identidad No 1007770477 y con cód. 461218272 estudiante del programa de Ingeniería de Sistemas de la Universidad de Cundinamarca, declaro que:

El contenido del presente documento es un reflejo de mi trabajo personal y manifiesto que, ante cualquier notificación de plagio, copia o falta a la fuente original, soy responsable directo legal, económico y administrativo sin afectar al director del trabajo, a la Universidad y a cuantas instituciones han colaborado en dicho trabajo, asumiendo las consecuencias derivadas de tales prácticas.

firma:

Juan Salamanca.

## Resumen

El pronóstico preciso del clima a corto, mediano y largo plazo se ha vuelto un objetivo de creciente importancia mundial debido a su impacto en las actividades humanas. Los estudios sobre el calentamiento global y el cambio climático, que muestran escenarios adversos para la biodiversidad, han intensificado el interés en reforzar la capacidad de predecir con mayor exactitud las variaciones climáticas espaciales y temporales. Esta necesidad se acentúa ante el aumento en la frecuencia de eventos extremos como sequías, inundaciones e incendios forestales.

Considerando la influencia del clima en diversos factores sociales, se describe en el presente documento el desarrollo de un modelo de predicción meteorológica utilizando técnicas de aprendizaje automático (machine learning). El objetivo es generar pronósticos precisos de variables climáticas mediante el entrenamiento de redes neuronales con datos históricos, bajo la metodología de investigación cuantitativa, empleando registros de datos para entrenar el modelo y demostrando resultados. Se adopta Scrum para un desarrollo colaborativo y focalizado en objetivos.

Para modelar las relaciones temporales en las series de tiempo meteorológicas, se implementó una red neuronal recurrente LSTM, adecuada para capturar dependencias a largo plazo. El modelo fue entrenado con un extenso conjunto de datos de múltiples parámetros climáticos de estaciones meteorológicas y del conjunto MERRA-2 de la NASA y en el que se aplicaron técnicas de preprocesamiento y normalización de los datos, así como validación cruzada, para mejorar el aprendizaje y evitar problemas como el sobreajuste. La

precisión del modelo, medida con métricas como el error cuadrático medio, mejoró constantemente al incrementar las épocas de entrenamiento. Esto demuestra la capacidad del modelo para aprender relaciones complejas.

Los resultados obtenidos indican que las técnicas de machine learning, particularmente las redes LSTM, tienen gran potencial para generar modelos predictivos meteorológicos precisos y confiables. Este trabajo representa una contribución relevante sobre el uso de inteligencia artificial para la comprensión y predicción de fenómenos climáticos complejos.

**Palabras Clave:** Datos, Aprendizaje automático, sistemas de predicción, análisis y patrones.

### **Abstract**

The accurate forecasting of climate in the short, medium and long term has become an objective of growing global importance due to its impact on human activities. Studies on global warming and climate change, which show adverse scenarios for biodiversity, have intensified interest in reinforcing the ability to predict spatial and temporal climate variations more accurately. This need is accentuated by the increase in the frequency of extreme events such as droughts, floods and forest fires.

Considering the influence of climate on various social factors, the present document describes the development of a weather forecasting model using machine learning

techniques. The objective is to generate accurate forecasts of climate variables through the training of neural networks with historical data, under a quantitative research methodology, using data records to train the model and demonstrate results. Scrum is adopted for collaborative and objective-focused development.

To model temporal relationships in meteorological time series, a recurrent LSTM neural network was implemented, suitable for capturing long-term dependencies. The model was trained with an extensive dataset of multiple climate parameters from weather stations and NASA's MERRA-2 dataset. Data preprocessing and normalization techniques were applied, as well as cross validation, to improve learning and avoid problems such as overfitting. The accuracy of the model, measured with metrics such as the mean square error, improved steadily as the training epochs increased. This demonstrates the model's ability to learn complex relationships.

The results obtained indicate that machine learning techniques, particularly LSTM networks, have great potential to generate accurate and reliable weather predictive models. This work represents a relevant contribution on the use of artificial intelligence for understanding and predicting complex climate phenomena.

**Keywords:** Data, machine learning, prediction systems, analytics and patterns.

## ÍNDICE GENERAL

INTRODUCCIÓN .....	16
I. Informe de investigación .....	17
1.1. Estado del arte .....	17
1.2. Línea de investigación.....	20
1.3. Planteamiento del problema y pregunta de investigación .....	20
1.4. Objetivo general y específico.....	22
1.5. Alcance e impacto del proyecto .....	22
1.6. Metodología .....	23
1.7. Marcos de referencia .....	24
1.7.1. Marco teórico .....	24
1.7.2. Marco legal .....	31
II. Documentación del software .....	33
2.1. Plan de proyecto.....	34
2.2. Determinación de requerimientos .....	34
2.2.1 Introducción. ....	34
2.2.2 Propósito. ....	35
2.2.3 Ámbito del sistema. ....	35

	11
2.2.4 Definiciones, acrónimos y abreviaturas.....	35
2.2.5 Referencias.....	39
2.2.6 Descripción general. ....	42
2.2.7 Requisitos comunes de las interfaces.....	43
2.2.8 Funciones. ....	46
2.2.9 Requisitos de Rendimiento .....	48
2.3. Especificación del diseño.....	48
2.3.1 Diagrama de flujo .....	49
2.3.2 Diagrama de bloques.....	52
2.3.3 Recopilación y preparación de los datos.....	55
2.3.4 Análisis y graficas.....	57
2.4 DISEÑO DE LOS CASOS DE PRUEBA .....	85
2.4.1 Prueba de Dickey-Fuller aumentada.....	85
2.4.2 Prueba de entrenamiento.....	88
2.5 ANALISIS ECONOMICO .....	109
2.6 RESULTADOS .....	113
2.6.1 EVALUACIÓN TECNOLÓGICA .....	113
2.7 CONCLUSIONES Y RECOMENDACIONES .....	117
2.8 Bibliografía .....	118
2.9 ANEXOS .....	121

## ÍNDICE DE TABLAS

Ilustración 1 Plan de proyecto realizado.....	34
Ilustración 2 Diagrama de flujo.....	49
Ilustración 3 Diagrama de bloques.....	53
Ilustración 4 QV2M: Humedad específica de a 2 metros por año.....	58
Ilustración 5 QV2M: Humedad específica de a 2 metros.....	59
Ilustración 6 PS: Presión de superficie por año.....	60
Ilustración 7 PS: Presión de superficie.....	61
Ilustración 8 WS10M: Velocidad del viento a 10 metros por año.....	62
Ilustración 9 WS10M: Velocidad del viento a 10 metros.....	63
Ilustración 10 WS10M_MIN: Velocidad mínima del viento a 10 metros por año ..	63
Ilustración 11 WS10M_MIN: Velocidad mínima del viento a 10 metros.....	65
Ilustración 12 PRECTOTCORR: Precipitación corregida por año.....	66
Ilustración 13 PRECTOTCORR: Precipitación corregida.....	67
Ilustración 14 RH2M: Humedad relativa de a 2 metros por año.....	68
Ilustración 15 RH2M: Humedad relativa de a 2 metros.....	69
Ilustración 16 T2M: Temperatura a 2 metros por año.....	70
Ilustración 17 T2M: Temperatura a 2 metros.....	71
Ilustración 18 T2M_MIN: Temperatura mínima a 2 metros por año.....	72
Ilustración 19 T2M_MIN: Temperatura mínima a 2 metros.....	73
Ilustración 20 T2M_MAX: Temperatura máxima a 2 metros por año.....	74
Ilustración 21 T2M_MAX: Temperatura máxima a 2 metros.....	75

Ilustración 22	T2M_RANGE: Rango de temperatura a 2 metros por año .....	76
Ilustración 23	T2M_RANGE: Rango de temperatura a 2 metros .....	77
Ilustración 24	T2MWET: Temperatura de bulbo húmedo a 2 metros por año .....	78
Ilustración 25	T2MWET: Temperatura de bulbo húmedo a 2 metros .....	80
Ilustración 26	T2MDEW: Punto de rocío/escarcha a 2 metros por año .....	81
Ilustración 27	T2MDEW: Punto de rocío/escarcha a 2 metros .....	82
Ilustración 28	TS: Temperatura de la superficie terrestre por año .....	83
Ilustración 29	TS: Temperatura de la superficie terrestre .....	84
Ilustración 30	Prueba de Dickey-Fuller aumentada por consola .....	86
Ilustración 31	QV2M: Humedad específica de a 2 metros Entrenamientos.....	90
Ilustración 32	PS: Presión de superficie de Entrenamientos .....	92
Ilustración 33	WS10M: Velocidad del viento a 10 metros Entrenamientos .....	93
Ilustración 34	WS10M_MAX: Velocidad máxima del viento a 10 metros	
Entrenamientos .....		95
Ilustración 35	WS10M_MIN: Velocidad mínima del viento a 10 metros	
Entrenamientos .....		96
RH2M: Humedad relativa de a 2 metros (%) Ilustración 36	RH2M: Humedad	
relativa de a 2 metros Entrenamientos.....		97
Ilustración 37	T2M: Temperatura a 2 metros Entrenamientos.....	99
Ilustración 38	T2M_MIN: Temperatura mínima a 2 metros Entrenamientos.....	101
Ilustración 39	T2M_MAX: Temperatura máxima a 2 metros Entrenamientos.....	102
Ilustración 40	T2M_RANGE: Rango de temperatura a 2 metros Entrenamientos	104
Ilustración 41	T2MWET: Temperatura de bulbo húmedo a 2 metros Entrenamientos	
.....		105

Ilustración 42 T2MDEW: Punto de rocío/escarcha a 2 metros Entrenamientos ....	106
Ilustración 43 TS: Temperatura de la superficie terrestre Entrenamientos .....	108
Ilustración 44 Sitio Oficial De Anaconda.....	124
Ilustración 45 Descarga Anaconda .....	124
Ilustración 47 Instalador Anaconda.....	125
Ilustración 46 Acuerdo de licencia Anaconda.....	125
Ilustración 48 Selección de la ubicación del directorio.....	126
Ilustración 49 Paso final Instalación.....	127
Ilustración 50 Anaconda.....	127
Ilustración 51 Jupyter Notebook.....	128
Ilustración 52 Abrir Terminal.....	129
Ilustración 53 Anaconda Prompt .....	129
Ilustración 54 Confirmación de biblioteca .....	130
Ilustración 55 Abrir proyecto. ....	130
Ilustración 56 Correr Proyecto .....	131
Ilustración 57 Correr bloque de código. ....	131
Ilustración 58 Librerías para el análisis de los datos .....	136
Ilustración 59 Información de los datos. ....	137
Ilustración 60 Bloque de código para la corrección del csv de entrenamiento .....	138
Ilustración 61 Cabecera de los datos de entrenamiento.....	138
Ilustración 62 Últimos registros del CSV de entrenamiento. ....	139
Ilustración 63 función “graficar por año”.....	139
Ilustración 64 Librerías para el modelo.....	140
Ilustración 65 Cargar los datos de la serie de tiempo .....	141

Ilustración 66 Dividir los datos en conjunto de entrenamiento y prueba. ....	141
Ilustración 67 Función para crear secuencias de entrenamiento. ....	142
Ilustración 68 Graficar resultados.....	142
Ilustración 69 Crear el modelo de red neuronal .....	143
Ilustración 70 Predicciones.....	145
Ilustración 71 Interpretación de resultados.....	146

## **INDICE DE TABLAS**

Tabla 1 Descripciones de los acrónimos. ....	36
Tabla 2 Descripción del diagrama de flujo fase 1. ....	49
Tabla 3 Descripción del diagrama de flujo fase 2. ....	50
Tabla 4 Descripción del diagrama de flujo fase 3. ....	52
Tabla 5 Descripción del diagrama de bloques.....	53
Tabla 6 Prueba de Dickey-Fuller aumentada .....	86
Tabla 7 Unidad funcional de los requerimientos.....	109
Tabla 8 Cálculo de puntos de función sin ajustar.....	110
Tabla 9 Puntos de función ajustados. ....	111
Tabla 10 Cuadro comparativo evaluación tecnológica.....	113
Tabla 11 Ventajas y desventajas de las principales tecnologías utilizadas .....	115

## INTRODUCCIÓN

En la actualidad, la predicción precisa de las condiciones climáticas se ha convertido en un elemento crucial para una variedad de industrias y aplicaciones, desde la agricultura hasta la planificación urbana y la gestión de recursos naturales. Para abordar este desafío, se ha vuelto esencial establecer un modelo predictivo sólido que aproveche las técnicas avanzadas de machine learning.

El objetivo principal de este proyecto es desarrollar un modelo predictivo y confiable que permita generar estimaciones precisas sobre las condiciones climáticas futuras. Para lograrlo, se emplearán técnicas de machine learning que analizarán una amplia gama de datos climáticos históricos, como la temperatura, la humedad, la velocidad del viento, la presión atmosférica y más. Estos datos se obtendrán de fuentes confiables, como estaciones meteorológicas y satélites, y se someterán a un proceso de limpieza y preparación.

La ventaja de este modelo predictivo es su capacidad para proporcionar estimaciones de las condiciones climáticas en un futuro próximo, lo que permitirá a las personas y organizaciones tomar decisiones informadas. Por ejemplo, los agricultores podrán planificar sus cultivos en función de las previsiones meteorológicas, las empresas de energía podrán optimizar la producción de energía en función de la demanda anticipada y las ciudades podrán prepararse para eventos climáticos extremos con mayor antelación.

En resumen, la creación de un modelo predictivo basado en machine learning para estimar condiciones climáticas es un paso crucial en la mejora de nuestra capacidad para comprender y anticipar los cambios en el clima. Esto no solo beneficia a la sociedad en general, sino que

también contribuye a la toma de decisiones más sostenibles y efectivas en una amplia variedad de sectores.

## **I. Informe de investigación**

### **1.1. Estado del arte**

En la actualidad, los factores de predicción son fundamentales en la toma de decisiones de la agricultura moderna, orientan los procesos de producción y permite planear con mayor precisión los procesos químicos, es por esto, que la variabilidad del clima ha llevado al desarrollo de modelos que permiten encontrar patrones de predicción.

Se han hallado diferentes técnicas, tales como, modelos físicos matemáticos, herramientas estadísticas empleadas en el campo de la inteligencia artificial (redes bayesianas), Deep Learning, Machine Learning, entre otros. Sin embargo, la predicción varía de la cantidad de datos disponibles y el nivel de complejidad en los métodos aplicados.

Según (OMM), en los últimos tiempos ha aumentado considerablemente el interés por los servicios climáticos; pese al aumento de los riesgos, se ha dejado de limitar las predicciones meteorológicas básicas tradicionales, abarcando también las tendencias a largo plazo y los sistemas de alerta temprana, permitiendo mejorar la calidad y la utilización de los servicios climáticos en los planos nacional, regional y mundial. (OMM, 2020).

Cabe destacar que para poder estudiar y predecir los eventos se debe conocer los parámetros meteorológicos y el manejo de herramientas de código en variables ambientales, esto conlleva a estrategias de preparación que permiten limitar las pérdidas provocadas por los desastres hidrometeorológicos.

Son algunos los proyectos de predicción climática basado en servicios de monitoreo,

gestión de alarmas y predicciones del clima. Un proyecto que compacta estos componentes es el diseño de modelo predictivo realizado en la Universidad Guayaquil (Egas Arizala, 2020), llevado a cabo en técnicas Machine Learning que está fundamentado en determinar la temperatura, usando datos proporcionados durante un año por el Instituto Nacional de Meteorología e Hidrología, con una obtención predictiva de un día en el futuro; este proyecto también integro un sitio web donde se visualiza los datos de la mini estación y las variaciones diarias de la temperatura analizadas mediante gráficos estadísticos, y las predicciones obtenidas por el modelo.

En los proyectos afines se encuentra el “Diseño e implementación de un prototipo de estación meteorológica remota de bajo costo empleando el enfoque de internet de las cosas”; como principal problemática este proyecto dedujo que las estaciones meteorológicas analógicas a menudo son voluminosas y de altos costos, por lo que su enfoque se proyecta en la importancia de apoyar áreas de prueba con dispositivos que le permiten obtener mediciones de variables físicas, como temperatura, humedad, presión. Atmosférica, velocidad del viento y detecciones basadas en la investigación, haciendo uso de las nuevas tecnologías de software y hardware gratuitos, combinándolas con Internet y los costos reducidos y así conducir al modelo de estación meteorológica con la capacidad de recopilar y procesar datos que permiten acceder a información remota. Este prototipo de esta estación puede ser muy útil en el desarrollo futuro no solo en el campo de la adquisición y transmisión de variables atmosféricas, sino también útil en el campo de pronóstico, ablandamiento y adaptación a fenómenos de recolección, desde el cambio climático. (Paternina, 2018)

Como parte investigativa se encontró un desarrollo geovisor web que permite conocer las condiciones venideras bajo escenarios de cambio climático, partiendo de datos

de precipitación y temperatura. Este visor proporciona herramientas clave para la toma de decisiones orientadas a la conservación y mitigación de los efectos del cambio climático, buscando evitar la escasez de agua en el futuro por medio de las fases de especificación, diseño y desarrollo con el uso de metodologías ágiles y la reutilización de software, este proyecto surgió a través de la necesidad de conocer la disponibilidad hídrica a futuro de la zonas principales que abastece el Páramo de Chingaza, considerando que es el principal abastecedor de agua de Bogotá y varios municipios de Meta y Cundinamarca. (Yépez, 2019).

Por otra parte, las prioridades investigativas expuestas y descritas en “Proyecto de predicción subestacional a estacional: tendiendo un puente entre el tiempo y el clima” nos orienta a la búsqueda de sistemas de predicción del tiempo y del clima que presenten una mayor continuidad e integración con base en el entendimiento de los mecanismos de la predecibilidad subestacional a estacional, la pericia de los pronósticos subestacionales, incluyendo la identificación de ventanas de oportunidad que permitan incrementar la destreza predictiva, el Comprender los errores sistemáticos y los sesgos en el rango predictivo subestacional a estacional, y por último, la influencia que tiene el Comparar, verificar y poner a prueba combinaciones multimodelo a partir de las predicciones y cuantificar su incertidumbre. (Vitart, 2019)

Unas de las características a tener en cuenta para la realización de este proyecto, son los distintos procesos y enfoques de cada modelo para predecir la variabilidad en los comportamientos atmosféricos. Como parte comparativa se resalta el proyecto de grado “predicción temperatura Bogotá: enfoque estocástico vs aprendizaje profundo (deep

learning)”, realizado en la universidad de Bogotá Jorge Tadeo Lozano; donde la información que ofrece se basa, en el uso de algoritmos de aprendizaje automático (Machine Learning), que incluye aquellos basados en el aprendizaje supervisado y en particular, en las diversas aplicaciones de las redes neuronales artificiales (ANN) con diferentes tipos de arquitecturas; También conceptualiza la terminología empleada para el uso de redes neuronales, nombradas como aprendizaje profundo (Deep Learning). (RAMÍREZ, 2021)

Entrando a revisar un poco más el detalle sobre la información con la que se trabajó en cada proyecto, es importante resaltar la frecuencia del uso de arquitecturas de aprendizaje profundo, debido a que la medición de las variables climáticas es constante y se debe fijar cómo se tomarán dentro del tratamiento del modelo.

## **1.2. Línea de investigación**

Al tratarse de un análisis predicativo se utiliza la línea de investigación MEDIT del grupo GISTFA (aprendizaje, conocimiento, tecnologías, comunicación y digitalización) fundamentado en dar paso a la creación de propuestas serias hacia la solución de problemas, con la tarea de convertir los resultados de sus semilleros en proyectos eficientes, óptimos y de impacto en la comunidad académica y en su entorno, así pues, se lleva a cabo el proyecto de diseño e implementación de un prototipo de estación meteorológica autosostenible para el análisis del comportamiento climático, enfocado en el módulo machine learning, el cual está basado en el análisis de información recolectada para el entrenamiento y ejecución del modelo, buscando la obtención de resultados adecuados para una correcta predicción y además de ello, aplicando métodos de investigación y de desarrollo usados en el campo de la ingeniería.

## **1.3. Planteamiento del problema y pregunta de investigación**

Mejorar la predicción de las variables climáticas en el corto, mediano y largo plazo es un objetivo que toma cada día mayor relevancia alrededor del mundo, esto debido al impacto que tiene en actividades humanas, tales como la agricultura, la generación de energía, la construcción, el aprendizaje, entre otras. Adicionalmente, los estudios sobre calentamiento global y cambio climático que muestran escenarios adversos para la biodiversidad del planeta han reflejado un incremento sobre el interés por fortalecer la capacidad de pronosticar con mayor exactitud el estado climático, desde el punto de vista de las variaciones espaciales y temporales. El interés se convierte en necesidad cuando los datos muestran además un incremento de los eventos extremos como sequías, inundaciones, deslizamientos, incendios forestales, entre otros, es decir, fenómenos que, si bien son reconocidos, su frecuencia no es tan alta. Para mejorar los pronósticos tanto climáticos como meteorológicos se han planteado diferentes alternativas que estimen las posibles condiciones futuras, sin embargo, los resultados que han sido obtenidos evidencian que aún existen diferencias y observaciones en el corto plazo. Por lo anterior la continua exploración de nuevos modelos o metodologías que aporten información cada vez más confiable se ha vuelto un factor importante.

Por otra parte, el poder brindar toda la información mediante el uso de herramientas de desarrollo es uno de las revoluciones sociales y económicas que permiten la optimización en niveles de producción, como en este caso la agricultura de la región del municipio de Ubaté es por ello que conocer de forma anticipada el comportamiento climático permitirá mejorar la preparación ante posibles eventos adversos y a su vez reducir el margen de error en la predicción de temperaturas; dicho esto se contempla en la problemática: ¿De qué manera un modelo de predicción climático proporcionará una solución adecuada al manejo

de la sostenibilidad en la producción agrícola en la Villa de san Diego de Ubaté?.

#### **1.4. Objetivo general y específico**

Objetivo general.

Establecer un modelo predictivo mediante el uso de técnicas de Machine Learning que permita generar una estimación sobre condiciones climáticas.

Objetivos específicos.

- Establecer un estado del arte que permita identificar los modelos de Machine Learning empleados en sistemas de monitorio y predicción climática.
- Construir un modelo Machine Learning para el proceso de preparación e integración de los datos meteorológicos con base a la información recolectada del “prototipo de estación meteorológica autosostenible”.
- Aplicar pruebas funcionales y técnicas que permitan determinar la calidad de modelo.
- Documentar el desarrollo del modelo y las pruebas realizadas.

#### **1.5. Alcance e impacto del proyecto**

La propuesta que se presenta considera principalmente brindar un aporte de información a la comunidad del sector agricultor, en la que se genere una proyección del posible comportamiento de diferentes aspectos climáticos en la región de la provincia de Ubaté.

Al recolectar los datos climáticos de forma periódica el proceso analítico se dirige principalmente a tomar mejores decisiones en el sector agricultor y así evitar pérdidas materiales, económicas y de tiempo. Además de ello también puede aportar a diferentes investigadores guiándolos para poder desarrollar proyectos similares o que contengan componentes que este posee.

Al no determinar una solución en la región se evidenciaría aún más las pérdidas de los agricultores y que los productos se vuelvan cada vez menos asequibles; en consecuencia, toda la comunidad se vería afectada tanto a nivel municipal, como en la parte externa, representada por los consumidores de los productos que el municipio provee.

Con la necesidad de brindar solución a lo evidenciado, el desarrollo genera un análisis del cambio climático a partir de los datos suministrados por la estación meteorológica en la provincia de Ubaté con el fin de optimizar sus procesos productivos.

### **1.6. Metodología**

La metodología de investigación propuesta para este proyecto será con un enfoque cuantitativo. Puesto que se utilizará como base principal para la recolección del registro extendido de los datos, la predicción y demostración de los resultados.

Al tener este enfoque, el proyecto frente a la recolección de los datos se encamina a la comprensión, observación e interacción informativa de las variables climáticas que se usarán para el desarrollo de un modelo predictivo.

Para la metodología de desarrollo se hará uso del marco de trabajo Scrum, puesto que de acuerdo con Cesar Rodríguez y Rubén Dorado en su artículo: ¿por qué implementar Scrum?, aclaran que la metodología Scrum es una de las metodologías de desarrollo ágiles de software más usadas por permitir resalta el trabajo en equipo, para el desarrollo de productos y la autonomía que estos deben tener (Rodríguez & Dorado, 2015).

Algunas de las ventajas del uso de Scrum como metodología de desarrollo son las siguientes:

La inspección en Scrum debe cumplir con varios objetivos, donde tres son

esenciales: “La reunión diaria, la revisión del sprint y la retrospectiva de este último”(Rodríguez & Dorado, 2015), de acuerdo con Cesar Rodríguez y Rubén Dorado, el cumplir estos objetivos permite una mejor organización para el equipo y su proceso, esto hace que las organizaciones prefieran este tipo de marco de trabajo , con una adaptación fácil a los cambios y como generalidad una visualización en los avances planteados (Rodríguez & Dorado, 2015).

En Scrum el trabajo en equipo es importante debido a que, en cada ciclo de desarrollo, “el mismo equipo se adapta para mejorar, y también implica que cada individuo sea reconocido como parte esencial del equipo” (Rodríguez & Dorado, 2015), ya que como cada uno tiene la asignación de tareas diferentes esto hace que cada integrante pueda aportar de alguna manera sobre el producto a desarrollar.

La usabilidad de los elementos de la metodología se llevará a cabo a partir de reuniones y actividades con las que se procederá a una retroalimentación semanal, procurando siempre el cumplimiento de cada asignación en los respectivos Sprint y llevando a cabo el trabajo en equipo de forma constante.

## **1.7. Marcos de referencia**

### **1.7.1. Marco teórico**

El software se define como el conjunto de código, datos y algoritmos que permiten al modelo aprender y realizar tareas. En este el código define la estructura y el funcionamiento del modelo, los datos proporcionan la estructura sobre el que el modelo aprende, y los algoritmos son los que permiten al modelo identificar patrones en los datos y generar las predicciones.

El software del modelo de machine learning se puede dividir en dos categorías principales, por una parte, está el Software de entrenamiento, el cual aplica los algoritmos de aprendizaje automático al conjunto de datos para que el modelo aprenda a identificar patrones, y el Software de ejecución que utiliza el modelo para generar predicciones o tomar decisiones.

El desarrollo del modelo puede ser implementado en una variedad de lenguajes de programación y plataformas. Los lenguajes de programación más comunes para el desarrollo de modelos de machine learning incluyen Python, R y Java junto con las plataformas más comunes para ejecución de modelos de machine learning como lo son TensorFlow, PyTorch y scikit-learn.( Raschka, S. 2020)

Por otro lado, el concepto de aprendizaje automático se comprende como una subdisciplina de la inteligencia artificial, se enfoca en el desarrollo de algoritmos y modelos estadísticos que permiten a los sistemas mejorar su desempeño en una tarea específica a través de la experiencia adquirida al procesar y analizar datos. En lugar de depender de instrucciones de programación explícitas, los algoritmos de aprendizaje automático utilizan patrones e información extraída de los datos para hacer predicciones, tomar decisiones y realizar acciones.( J. C. 2023)

El aprendizaje automático involucra varios conceptos que incluyen el aprendizaje supervisado, en este los modelos se entrenan con datos etiquetados; el aprendizaje no supervisado, donde los modelos identifican patrones en datos no etiquetados y por último el aprendizaje por refuerzo, en el cual los modelos aprenden a partir de interacciones con un entorno. La idea principal del aprendizaje automático es crear modelos o sistemas predictivos

que puedan generalizar su aprendizaje a partir de datos pasados y tomar decisiones o hacer predicciones precisas en nuevos datos no vistos.

Con relación a la anterior, para entender la parte informativa de los datos se define el concepto de Modern-Era Retrospective Analysis for Research and Applications, como “conjunto de datos producido por la NASA que proporciona una visión detallada y coherente de la evolución del clima global desde 1980 hasta el presente”. Al proporcionar una amplia gama de datos climáticos históricos, como temperaturas, presiones atmosféricas, vientos y humedad, MERRA-2 permite al modelo tener un enfoque más informado y preciso al predecir patrones climáticos futuros. La integración de MERRA-2 en el modelo permite la incorporación de información climática detallada y de alta calidad. (Zhao, B. 2017).

Python al ser un lenguaje de programación de alto nivel ampliamente utilizado en una variedad de aplicaciones de desarrollo de software. Con su sintaxis clara y legible, así como su amplia gama de bibliotecas y frameworks, Python se ha convertido en una herramienta clave en el campo del machine learning.

La aplicación de Python en el modelo se debe a diversas ventajas. En primer lugar, Python ofrece una amplia gama de bibliotecas y herramientas específicamente diseñadas para el machine learning, como TensorFlow, Keras, PyTorch y Scikit-learn, que facilitan el desarrollo y la implementación de algoritmos de aprendizaje automático de manera eficiente. (Drake, F. L. 2009).

Con relación a lo anterior, la razón fundamental por la que se aplica TensorFlow en el modelo radica en su capacidad para gestionar el flujo de datos y cálculos complejos de manera eficiente. Además, optimiza automáticamente el uso de recursos computacionales, lo

que resulta fundamental para el procesamiento rápido de grandes conjuntos de datos y la creación de modelos. Al ser una biblioteca de código abierto para realizar cálculos numéricos de alto rendimiento que se utiliza principalmente en el campo del aprendizaje automático y la inteligencia artificial, facilita la creación y entrenamiento de modelos de machine learning y redes neuronales a través de su arquitectura flexible y eficiente. (Torres, J. (2020)

Otra biblioteca de aprendizaje empleada es Scikit-learn, comúnmente abreviado como sklearn, puesto que ofrece una amplia gama de herramientas y algoritmos para tareas de minería de datos y análisis predictivo. Scikit-learn se basa en las bibliotecas NumPy, SciPy y matplotlib, lo que lo hace muy eficiente y fácil de integrar en entornos de desarrollo existentes.

Esta biblioteca es ampliamente utilizada en el campo del aprendizaje automático debido a su facilidad de uso y su capacidad para implementar rápidamente una variedad de algoritmos de aprendizaje supervisado y no supervisado. Desde algoritmos simples, como regresión lineal y clasificación, hasta técnicas más avanzadas, como máquinas de vectores de soporte y agrupamiento, scikit-learn proporciona una interfaz coherente y fácil de usar para todos estos métodos (Duchesnay, É. 2011)

Continuando con los conceptos de desarrollo, un entorno interactivo que permite la creación y compartición de documentos en tiempo real, integrando código en distintos lenguajes de programación, visualizaciones y texto explicativo es Jupyter Notebook; su flexibilidad y capacidad para ejecutar código de manera interactiva lo convierten en una herramienta fundamental en el desarrollo y la implementación del modelo de machine learning.

Su aplicación en el modelo radica en varios aspectos claves, En primer lugar, proporciona un entorno integrado para la experimentación y el desarrollo de modelo, lo que nos permite probar diferentes algoritmos y enfoques de modelado de forma iterativa.

Además, su capacidad para integrar código con visualizaciones y explicaciones detalladas facilita la comprensión y comunicación de los resultados del modelo a otros usuarios o partes interesadas. Esto resulta crucial en el contexto de machine learning, donde la interpretación y comunicación de resultados complejos son fundamentales para la toma de decisiones informada. (Willing, C. 2016)

Teniendo en cuenta las series de tiempo, como un tipo de datos secuenciales que representan observaciones climáticas registradas a lo largo del tiempo. Donde las observaciones incluyen parámetros como temperatura, humedad específica y relativa, velocidad del viento, presión atmosférica, punto de rocío, y otros factores climáticos. Se define para analizar y predecir patrones, tendencias y variaciones climáticas a lo largo de días, semanas, meses o años. Las técnicas de análisis de series de tiempo, como el modelo de aprendizaje automático en las Redes Neuronales Recurrentes (RNN) y las Redes Neuronales con Memoria a Largo Plazo (LSTM), son fundamentales para realizar predicciones precisas en el ámbito de la climatología (Santana, B.2016).

Como concepto del primer elemento clave tenemos que una Red Neuronal Recurrente (RNN) para un modelo machine learning se entiende como una arquitectura de redes neuronales diseñada para procesar secuencias de datos relacionados con el clima, como lo son las temperaturas, presión atmosférica, humedad, y otros factores meteorológicos, con la finalidad de hacer predicciones sobre las condiciones climáticas futuras.

Con relación a lo anterior, una RNN toma en consideración la naturaleza secuencial de los datos, lo que significa que puede capturar dependencias temporales y patrones de cambio a lo largo del tiempo. Al utilizar datos históricos, como registros climáticos previos encontrados en MERRA2, la RNN aprende a identificar tendencias, estacionalidades y relaciones entre diferentes variables climáticas, lo que le permite generar pronósticos climáticos para un período futuro específico, en este caso un de uno a cinco días en el futuro.

El modelo RNN considera datos de la ubicación geográficas del municipio de Ubaté con relación a la estación, además incorpora información adicional, como datos de satélites para mejorar la precisión de las predicciones climáticas.

El siguiente elemento clave se define como Las Redes Neuronales Recurrentes con Memoria a Largo Plazo (LSTM), la cual es una arquitectura de redes que procesan y modelan datos de cada variable climática a lo largo del tiempo. Estas redes ayudan a capturar relaciones temporales complejas en los datos, como patrones climáticos estacionales al mantener y gestionar la memoria a largo plazo de las observaciones pasadas. Esto es esencial para generar predicciones lo más precisas posibles y comprender mejor las tendencias climáticas junto con los eventos meteorológicos extremos. Las LSTM son particularmente útiles en aplicaciones de predicción del clima, puesto que consideran la información histórica y las dependencias temporales en los datos para mejorar la calidad de las predicciones climáticas. (Ruiz, T. 2023)

Por otro lado para entender los resultados generados por el modelo se definen situaciones como, el subajuste, o underfitting en inglés, que ocurre cuando un modelo de machine learning es demasiado simple para capturar la estructura subyacente de los datos.

En consecuencia, el modelo no logra ajustarse suficientemente a los datos de entrenamiento, lo que resulta en un rendimiento deficiente tanto en los datos de entrenamiento como en los datos no vistos. El subajuste generalmente se manifiesta cuando un modelo es incapaz de capturar la complejidad de las relaciones entre las variables de entrada y salida, lo que resulta en predicciones inexactas y poco fiables.

Según lo observado la importancia de comprender el subajuste, radica en su impacto negativo en la capacidad de generalización del modelo. La falta de capacidad para capturar la complejidad de los datos puede limitar la capacidad del modelo para adaptarse a patrones más complejos, lo que resulta en un rendimiento deficiente en tareas de predicción y análisis. (Koehrsen, W. - 2018).

Otra de las situaciones que se pueden presentar es el sobreajuste, también conocido como *overfitting* en inglés, lo cual se comprende como “un fenómeno común en el aprendizaje automático donde un modelo se ajusta excesivamente a los datos de entrenamiento” (Cárdenas-Montes, M. - 2006), lo que resulta en un rendimiento deficiente cuando se enfrenta a nuevos datos no vistos previamente. Esto implica que el modelo aprende y memoriza el ruido y las fluctuaciones aleatorias presentes en los datos de entrenamiento, en lugar de capturar las tendencias y patrones subyacentes que son generalizables a otros conjuntos de datos.

La principal razón por la que se presta atención al sobreajuste en el modelo es su facultad para comprometer la capacidad de generalización del modelo. Si un modelo está sobreajustado, puede tener un excelente rendimiento en los datos de entrenamiento, pero se comportará mal al predecir nuevos datos. En contraste, un modelo que no esté sobreajustado

tendrá una mejor capacidad para generalizar y predecir con precisión en conjuntos de datos no vistos previamente.

### **1.7.2. Marco legal**

En un primer plano, es crucial mencionar que el artículo 15 de la Constitución Política de Colombia de 1991 establece los principios fundamentales que rigen la recopilación, tratamiento y difusión de datos personales, junto con los procedimientos destinados a preservar la libertad y las demás garantías consagradas en dicha Constitución (Corte Constitucional, 1991).

Por otro lado, la Ley 23 de 1982 establece disposiciones fundamentales relacionadas con la protección de los derechos de autor para obras literarias, científicas y artísticas, independientemente de la forma en que se expresen (LEY NÚMERO 23 DE 1982, 1982).

En un contexto más actual, la Ley Estatutaria 1266 de 2008 se encarga de definir "las normas generales del habeas data y regular el manejo de la información contenida en bases de datos personales, en particular aquellas de naturaleza financiera, crediticia, comercial, de servicios y la proveniente de terceros países, además de otras disposiciones relacionadas con el tema" (Ley 1266 de 2008, 2008).

En cuanto a las herramientas tecnológicas se puede empezar por Python la cual en su licencia de código abierto conocida como la "Licencia Pública General de Python" (Python Software Foundation License) indica:

“Este CONTRATO DE LICENCIA se establece entre BeOpen.com ("BeOpen"), con oficinas en 160 Saratoga Avenue, Santa Clara, CA 95051, y el Individuo u Organización

("Licenciario") que accede y utiliza este software en su forma de código fuente o binario y su documentación asociada ("el Software").

Sujeto a los términos y condiciones de este Acuerdo de Licencia de Python de BeOpen, BeOpen otorga al Licenciario una licencia no exclusiva, sin regalías y mundial para reproducir, analizar, probar, realizar y/o mostrar públicamente, preparar obras derivadas, distribuir y utilizar de otra manera el Software solo o en cualquier versión derivada, siempre y cuando se mantenga la Licencia de Python de BeOpen en el Software, solo o en cualquier versión derivada preparada por el Licenciario.

BeOpen pone el Software a disposición del Licenciario "TAL CUAL". BEOPEN NO HACE REPRESENTACIONES O GARANTÍAS, EXPRESAS O IMPLÍCITAS. A MODO DE EJEMPLO, PERO SIN LIMITACIÓN, BEOPEN NO HACE Y RENUNCIA A CUALQUIER REPRESENTACIÓN O GARANTÍA DE COMERCIALIZACIÓN O IDONEIDAD PARA UN PROPÓSITO PARTICULAR O DE QUE EL USO DEL SOFTWARE NO INFRINGIRÁ LOS DERECHOS DE TERCEROS.

BEOPEN NO SERÁ RESPONSABLE ANTE EL LICENCIARIO U OTROS USUARIOS DEL SOFTWARE POR CUALQUIER DAÑO INCIDENTAL, ESPECIAL O CONSECUENTE O PÉRDIDA COMO RESULTADO DEL USO, MODIFICACIÓN O DISTRIBUCIÓN DEL SOFTWARE O CUALQUIER DERIVADO DEL MISMO, INCLUSO SI SE ADVIERTE DE LA POSIBILIDAD DE ELLO.

Este Contrato de Licencia terminará automáticamente en caso de incumplimiento sustancial de sus términos y condiciones.

Este Contrato de Licencia se registrará e interpretará en todos los aspectos por la ley del Estado de California, excluyendo las disposiciones de conflicto de leyes. Nada en este Contrato de Licencia se considerará como que crea ninguna relación de agencia, asociación o empresa

conjunta entre BeOpen y el Licenciatarlo. Este Contrato de Licencia no otorga permiso para utilizar las marcas comerciales o nombres comerciales de BeOpen en un sentido de marca para respaldar o promocionar productos o servicios del Licenciatarlo o de cualquier tercero. Como excepción, los logotipos "BeOpen Python" disponibles en <http://www.pythonlabs.com/logos.html> pueden usarse de acuerdo con los permisos otorgados en esa página web.

Al copiar, instalar o utilizar de otra manera el software, el Licenciatarlo acepta quedar sujeto a los términos y condiciones de este Contrato de Licencia.”

Aunque la licencia en sí no está directamente relacionada con un modelo de machine learning, es crucial comprenderla al utilizar bibliotecas y herramientas de Python para el desarrollo y la distribuir de modelo.

Python es un lenguaje popular para implementar algoritmos de machine learning debido a su sintaxis simple y su amplia gama de bibliotecas especializadas, como TensorFlow, scikit-learn y Keras, que facilitan el proceso de desarrollo y entrenamiento de modelos. La licencia Python de BeOpen establece los términos y condiciones bajo los cuales se puede utilizar y distribuir el software, incluidas las bibliotecas y herramientas utilizadas que se basan en este lenguaje.

Por lo tanto, al utilizar Python para desarrollar el modelo de machine learning, es importante tener en cuenta las restricciones y los requisitos de la licencia que garantiza un uso adecuado del software y sus componentes. Esto es crucial para cumplir con las regulaciones legales y éticas asociadas con el desarrollo general de los modelos de y para evitar posibles problemas legales en el uso y distribución de software basado en Python.

## **II. Documentación del software**



de machine learning. Esta especificación se ha estructurado basándose en las directrices dadas por el estándar IEEE Práctica Recomendada para Especificaciones de Requisitos Software ANSI/IEEE 830,1998.

**2.2.2 Propósito.** El propósito de este documento es definir las especificaciones funcionales y no funcionales para el desarrollo de un modelo predictivo basado en técnicas de machine learning que permitirá obtener resultados de una predicción de un día en el futuro de una variable climática. Éste será utilizado por diferentes actores, entre los cuales se encuentran la comunidad académica que integra a docentes y estudiantes de pregrado en las seccionales Facatativá y Ubaté

**2.2.3 Ámbito del sistema.** En el municipio de la provincia de Ubaté como en varios lugares del país, las condiciones climatológicas han tenido una variación constante que hacen que cada vez sea menos predecible su comportamiento; teniendo en cuenta que el clima influye en tantos factores sociales y para muchos de estos es considerablemente beneficioso, se plantea un desarrollo de modelo predictivo basándonos en los datos climáticos, los cuales se recolectan de forma periódica, según los instrumentos de medición y la estación climática que lo realiza de manera autónoma, de tal forma que la información registrada son series de tiempo , para poder analizarlos o ingresarlos dentro de un proceso analítico con variables que representen el total de los valores dentro de un periodo de tiempo determinado que pueden ser días, semanas, meses o más, por esta razón su proceso de desarrollo educativo se dirige a brindar información relevante para la población agricultora dando un mejor aprovechamiento con relación a las condiciones climáticas.

#### **2.2.4 Definiciones, acrónimos y abreviaturas**

*Tabla 1 Descripciones de los acrónimos.*

Nombre	Descripción
<b>UBATÉ</b>	Municipio colombiano ubicado en el departamento de Cundinamarca, en la región central del país.
<b>MACHINE LEARNING</b>	El aprendizaje automático es un enfoque de la inteligencia artificial que implica el desarrollo de algoritmos y modelos que permiten a las computadoras aprender de datos y mejorar su capacidad para tomar decisiones o hacer predicciones sin necesidad de programación directa.
<b>RNN</b>	Es una variante de redes neuronales artificiales conocida como Red Neuronal Recurrente (RNN) se emplea en la manipulación de información secuencial o series temporales. Estos algoritmos avanzados de aprendizaje profundo tienen aplicaciones habituales en contextos que implican secuencia y tiempo, tales como la traducción de idiomas, el procesamiento de voz y la generación de subtítulos para imágenes.
<b>SOFTWARE</b>	Son los programas de aplicación y los sistemas operativos que permiten que la computadora pueda desempeñar tareas inteligentes, dirigiendo a los componentes físicos o hardware con instrucciones y datos a través de diferentes tipos de programas.

<b>LSTM</b>	Una LSTM (Long Short-Term Memory) es una arquitectura avanzada de red neuronal diseñada para recordar y aprender patrones en datos secuenciales, como texto, voz o series temporales, a largo plazo. Ayuda a superar el problema del desvanecimiento del gradiente y es ampliamente utilizada en tareas de procesamiento del lenguaje natural y predicción.
<b>MERRA-2</b>	Significa "Modern-Era Retrospective Analysis for Research and Applications, Version 2", es un conjunto de datos desarrollado y mantenido por la NASA (Administración Nacional de Aeronáutica y del Espacio) que proporciona una extensa recopilación de información sobre el clima y la atmósfera de la Tierra. la parte del desarrollo web que se encarga de que toda la lógica de una página web funcione. Se trata del conjunto de acciones que pasan en una web pero que no vemos cómo.
<b>JUPYTER</b>	Se define como una herramienta versátil que combina texto, código y resultados en un solo documento, lo que lo hace ideal para la documentación, la investigación, el análisis de datos y la creación de scripts interactivos.
<b>SERIES DE TIEMPO</b>	Proporciona la base de datos estructurada necesaria para comprender y predecir variables que cambiarán en el futuro a través del análisis de patrones temporales y la construcción de modelos predictivos; el aprovechamiento de esta información histórica se toma

	para lograr decisiones más informadas y anticipar eventos futuros en una variedad de campos.
<b>MODELO PREDICTIVO</b>	Se define como herramienta analítica que se utiliza para hacer estimaciones o predicciones sobre eventos futuros en función de datos históricos y patrones identificados en esos datos.
<b>PYTHON</b>	La implementación de Python se basa en Corregir y eliminar datos incorrectos, lo que se conoce como limpieza de datos. Además de ello, extrae y selecciona varias características de la información buscando diferentes estadísticas y Visualizando mediante el uso de tablas y gráficos, como son los gráficos de líneas, los de barras, los circulares y los histogramas.
<b>TENSOR FLOW</b>	TensorFlow es una biblioteca de código abierto desarrollada por Google que se utiliza principalmente para la creación y entrenamiento de modelos de aprendizaje automático y redes neuronales.
<b>SKLEARN</b>	Scikit-learn (también conocido como sklearn) es una biblioteca de código abierto en Python que se utiliza para el aprendizaje automático (machine learning) y la minería de datos. proporciona una amplia variedad de herramientas y algoritmos para tareas comunes en el aprendizaje automático, como clasificación,

	regresión, agrupación, reducción de dimensionalidad y selección de características.
<b>SUBAJUSTE (UNDERFITTING)</b>	El subajuste se refiere a una situación en la que un modelo es demasiado simple para capturar las relaciones dentro de los datos, lo que resulta en un rendimiento deficiente en la predicción.
<b>SOBREAJUSTE (OVERFITTING)</b>	El sobreajuste ocurre cuando un modelo es excesivamente complejo y se adapta demasiado a los datos de entrenamiento. Como resultado, el modelo captura el ruido o detalles irrelevantes en los datos, lo que lleva a un rendimiento deficiente en la generalización a nuevos datos.

### 2.2.5 Referencias

Bosilovich, M. G. ((2015). ). *MERRA-2*: File specification (No. GSFC-E-DAA-TN27096.

Cano Alor, C. G. ( 2020). *Diseño de una aplicación de monitoreo de emanación de gases en la ciudad mediante el uso de tecnologías de inteligencia artificial.*

Cardozo Vásquez, A. (s.f.). *Desarrollo de un modelo de red neuronal artificial para la reducción de escala (downscaling) de datos de temperatura del modelo Climático Global Canadiense 3.1 a estaciones meteorológicas colombianas.* (Master's thesis, Universidad de La Sabana.

Cortes, D. (2021). *predicción temperatura bogota: enfoque estocástico vs aprendizaje .*

- Cruz, I. B. (2007). *Redes neuronales recurrentes para el análisis de secuencias*. . Revista Cubana de Ciencias Informáticas.
- Cundinamarca, A. M. (2023 ). *Alcaldía Municipal Ubaté Cundinamarca* .
- De la Barrera Castillo, J. C. (2023). *Metodologías de ahorro energético aplicadas a los sistemas HVAC utilizando inteligencia artificial: una revisión del estado del arte*.
- Deulofeu Gómez, R. (2020). *Introducción y optimización estocástica de redes neuronales profundas MLP*.
- Dillon, J. V. (2017). *Tensorflow distributions*. arXiv preprint arXiv:.
- Egas Arizala, K. O. (2020). *Diseño de un modelo predictivo basado* .
- Fonseca-González, W. V.-C.-V. ((2019). ). *Potencial de mitigación del cambio climático de los ecosistemas forestales caducifolios en Costa Rica: modelos predictivos de biomasa y carbono*. . Revista de Ciencias Ambientales.
- GISTFA. (2022). *Protocolo de GISTFA*. , 6ta Versión. Universidad de Cundinamarca, .
- Guerrero-Álvarez, J. A. (2021). *Predicción niveles de precipitación para la identificación de zonas agrícolas* . (Master's thesis).
- Hao, J. &. ((2019). ). *Machine learning made easy: a review of scikit-learn package in python programming language*. Journal of Educational and Behavioral Statistics.
- Mahesh, B. ((2020)). *Machine learning algorithms-a review*. International Journal of Science and Research (IJSR).
- Montes, C. (2006). *Sobreajuste-overfitting*. centro de Investigaciones Energicas, Medioambientales y Tecnologicas.
- Montesdeoca Santana, B. (2016). *Estudios de predicción en series temporales de datos meteorológicos utilizando redes neuronales recurrentes*. Bachelor's thesis.
- Murcia Ramirez, J. S. (s.f.). *Modelo de máxima entropía para la predicción de*

*criminalidad con tecnicas de Machine Learning .*

Murcia, D. S. (2015). *.Diseño e implementación de un prototipo de estación .*

Paniagua Urquijo, D. &. ( 2021). *Diseño, desarrollo e implementacion de una plataforma web de datos climatológicos mediante una boya construida en un arduino en el municipio de Guataqui Cundinamarca .*

Python, W. (2021). *Python. . Python Releases for Windows, 24.*

Randles, B. M. ( 2017 ). *Using the Jupyter notebook as a tool for open science: An empirical study.* ACM/IEEE Joint Conference on Digital Libraries (JCDL) (pp. 1-2). IEEE.

Rodríguez., &. D. ( 2015). *¿Por qué implementar Scrum? Why to implement .* Revista Ontare, 3(1), 125–144. .

Ruiz Palomo, G. J. ( 2021). *Klimachingaza: visor geográfico web .*

Sommerville. (2005). *Ingenieria de software.* pearson educacion .

Vázquez Ruiz, T. (2023). *Desarrollo y aplicación de metodología de modelado predictivo de series temporales.*

Villavicencio, J. ( 2010). *Introducción a series de tiempo. .* Puerto Rico.

Yana Puma, J. W. (2020). *Modelo predictivo agroclimático para detectar cambios climatologicos con machine learning e IoT para el sector agrario de la región de Puno .*

## **2.2.6 Descripción general.**

**2.2.6.1 Perspectiva del producto.** El diseño de un modelo predictivo basado en técnicas de Machine Learning, tomando en cuenta una cantidad de datos en un rango de tiempo sobre variables climáticas como (temperatura, humedad y presión atmosférica (mínima y máxima)), datos que se utilizaron para el entrenamiento del modelo y así obtener una predicción

**2.2.6.2 Funcionalidad del producto.** El desarrollo del diseño de un modelo predictivo basado en técnicas de Machine Learning, donde se utilizará la arquitectura de red neuronal recurrente (LSTM), usando datos de prueba para el entrenamiento del modelo, y

así poder obtener en la ejecución una predicción de un día en el futuro con los datos de una miniestación meteorológica en la provincia de Ubaté.

### **2.2.6.3 Suposiciones y dependencias.**

- Actualización de librerías y últimas versiones de Python.
- Los archivos de entrenamiento deberán estar formato CSV.

### **2.2.6.4 Requisitos futuros.**

- Reducción de error cuadrático medio para porcentaje de variación menor, haciendo que sea más acertada la predicción.

## **2.2.7 Requisitos comunes de las interfaces**

### **2.2.7.1 Interfaces de usuario**

La interfaz de usuario se compone de una ventana sencilla para el ingreso de las variables necesarias, vista de graficas necesarias y resultados de predicción.

### **2.2.7.2 Interfaces de hardware**

Será necesario disponer de equipos de cómputos en buen estado, con los periféricos necesarios y que cuenten con conectividad a la red para poder acceder a

la página web.

### **2.2.7.3 Interfaces de software**

- Sistema Operativo: Windows XP o superior.

- Explorador: Mozilla o Chrome.
- Procesador de 2.10GHz o superior.
- Memoria mínima de 4G.
- Mouse.
- Teclado.

#### **2.2.7.5 Interfaces de comunicación**

El modelo predictivo tendrá comunicación con una base de datos no relacional en caso de que exista, si no trabajara con datos ingresados manualmente.

#### **2.2.7.6 Requisitos funcionales**

- **Requisito funcional 1**

Selección en las variables climáticas: Compresión y clasificación variables climáticas necesarias.

✓ Comprensión de las variables climáticas necesarias para la recolección de los datos

- **Requisito funcional 2**

Uso del aprendizaje profundo para el modelo: Identificación de la función de aprendizaje profundo adecuada.

✓ Investigación previa al desarrollo del modelo predictivo

- **Requisito funcional 3**

Datos de entrenamiento en rango de cinco a ocho años: Preparación de los datos de entrenamiento para la red neuronal con la captación de información organizada en un CSV sobre las variables climáticas necesarias para los resultados del modelo predictivo.

- **Requisito funcional 4**

Entrenamiento del modelo predictivo: Entrenamiento y validación de la red neuronal para asegurar la precisión haciendo uso de la captación de datos almacenada y organizada en un CSV.

- **Requisito funcional 5**

Reportes de anomalías: Detección y análisis de anomalías.

✓ Hacer las correcciones necesarias alrededor de los resultados, mejorando la precisión de estos.

- **Requisito funcional 6**

Resultados: Análisis de los indicadores y resultados.

✓ Realizar cambios que mejoren la precisión de los resultados del modelo predictivo.

### 2.2.8 Funciones.

**Tabla 2** Requerimientos funcionales

Cantidad de actividades:	7
Actividad #1	
Nombre de la actividad	Selección en las variables climáticas
Compresión y clasificación variables climáticas	Funcionalidad del modelo con distintas variables climáticas dentro del modelo de datos del entrenamiento.
Prioridad de la actividad	Alta

Actividad #2	
Nombre de la actividad	Uso del aprendizaje profundo para el modelo.
Identificación de la función de aprendizaje profundo adecuada	Investigación previa al desarrollo del modelo predictivo.
Prioridad de la actividad	Alta

Actividad #3	
Nombre de la actividad	Implementación de librerías para arreglos de datos y graficas.

Instalación de las librerías necesarias para el modelo predictivo	Investigación e instalación de las librerías necesarias en sus correctas versiones.
Prioridad de la actividad	Alta

Actividad #4	
Nombre de la actividad	Datos de entrenamiento en rango de cinco a ocho años.
Preparación de los datos de entrenamiento para la red neuronal.	Captación de información organizada en un CSV sobre las variables climáticas necesarias.
Prioridad de la actividad	Alta

Actividad #5	
Nombre de la actividad	Entrenamiento del modelo predictivo
Entrenamiento y validación de la red neuronal para asegurar la precisión.	Entrenar el modelo predictivo con los datos recolectado en el CSV.
Prioridad de la actividad	Alta

Actividad #7	
Nombre de la actividad	Resultados.

Análisis de los indicadores y resultados.	Realizar cambios que mejoren la precisión de los resultados del modelo predictivo.
Prioridad de la actividad	Alta

### 2.2.9 Requisitos de Rendimiento.

Es necesario garantizar que el modelo esté en funcionamiento los 365 días del año, programando mantenimientos o soportes en tiempos muertos de consulta.

Se debe realizar con un lenguaje limpio, ordenado y sin redundancia con el fin de evitar bucles o demoras en el procesamiento de datos. Además, debe Garantizar que el ingreso de datos u otro proceso no afecte el desempeño de los resultados.

**Restricciones de Diseño.** El modelado del sistema se debe realizar en UML, donde se realizarán los siguientes diagramas para el diseño completo del sistema.

**Fiabilidad.** El sistema debe tener una interfaz de uso intuitiva y sencilla.

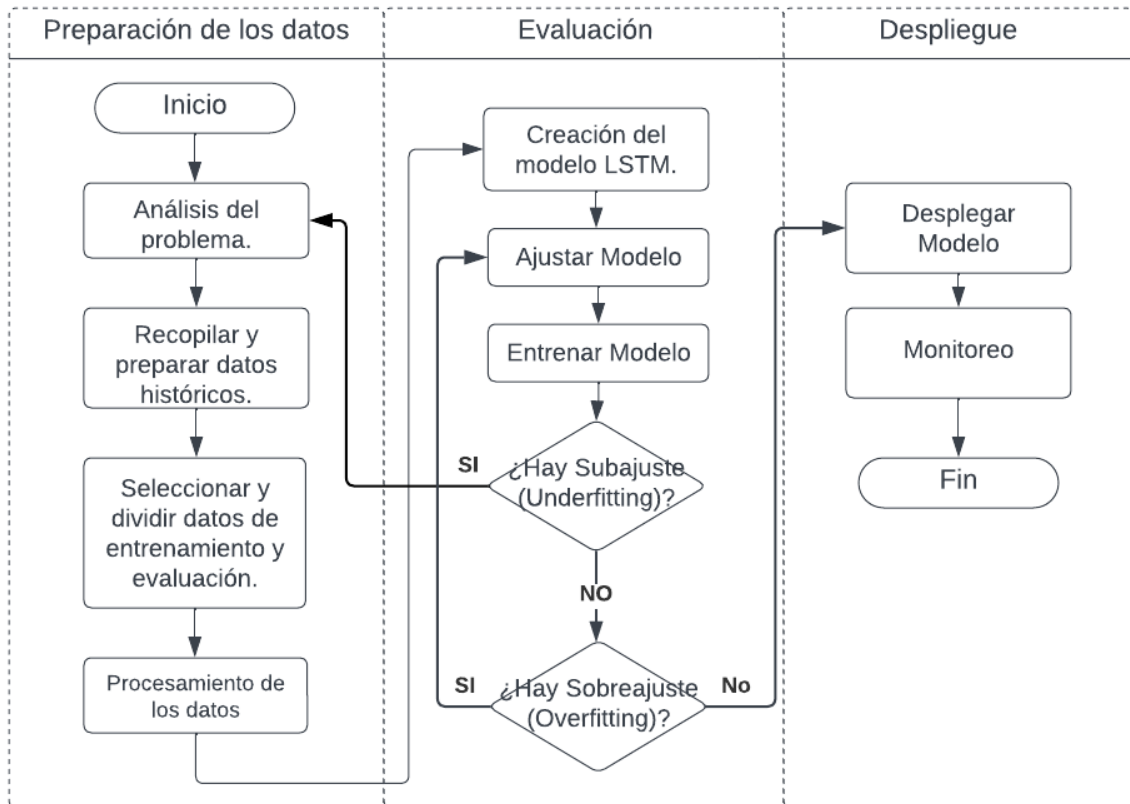
**Disponibilidad.** La disponibilidad del sistema debe ser continua con un nivel de servicio para los docentes y estudiantes de pregrado en las seccionales Facatativá y Ubaté.

**Mantenibilidad.** El sistema debe disponer de una documentación fácilmente actualizable que permita realizar operaciones de mantenimiento con el menor esfuerzo posible.

## 2.3. Especificación del diseño

### 2.3.1 Diagrama de flujo

Ilustración 2 Diagrama de flujo.



#### FASE N° 1: Preparación de datos

Tabla 2 Descripción del diagrama de flujo fase 1.

<b>PROCESO</b>	<b>DESCRIPCIÓN</b>
<b>Análisis del problema</b>	Esta etapa implica comprender el problema que se está tratando de resolver y los datos que se necesitan para entrenar el modelo.
<b>Recopilar y preparar datos históricos</b>	Esta etapa implica la recolección y organización de los datos tomados por la

	estación meteorológica y/o bases de datos tomas de otras fuentes (MERRA).
<b>Seleccionar y dividir datos de entrenamiento y evaluación</b>	<p>Implica dividir los datos en dos conjuntos:</p> <ul style="list-style-type: none"> <li>• El conjunto de entrenamiento, el cual es utilizado para entrenar el modelo.</li> <li>• El conjunto de evaluación que es utilizado para evaluar el rendimiento del modelo.</li> </ul>
<b>Procesamiento de los datos</b>	Esta etapa realiza cualquier transformación necesaria en los datos para que sean adecuados en el modelo. Las transformaciones incluyen limpieza de datos y escalado de datos

## **FASE N° 2: Evaluación**

*Tabla 3 Descripción del diagrama de flujo fase 2.*

<b><i>PROCESO</i></b>	<b><i>DESCRIPCIÓN</i></b>
<b>Creación del modelo LSTM</b>	Creación del modelo de aprendizaje automático. En este caso, el modelo Long Short-Term Memory.

<b>Ajustar Modelo</b>	Esta etapa implica ajustar los parámetros del modelo para que se ajuste mejor a los datos de entrenamiento.
<b>Entrenar Modelo</b>	Consiste en suministrar conjuntos de datos de entrenamiento al algoritmo de aprendizaje automático, permitiéndole adquirir conocimientos a partir de esos datos.
<b>Procesamiento de los datos</b>	Esta etapa realiza cualquier transformación necesaria en los datos para que sean adecuados en el modelo. Las transformaciones incluyen limpieza de datos y escalado de datos
<b>¿Hay Subajuste (Underfitting)?</b>	Evalúa si el modelo está sub ajustado, es decir, no es capaz de capturar las relaciones importantes en los datos.
<b>¿Hay Sobreajuste (Overfitting)?</b>	Evalúa si el modelo está sobre ajustado, quiere decir que se acomoda demasiado a los datos de entrenamiento y no es capaz de generalizar a nuevos datos.

**FASE N°3: Despliegue**

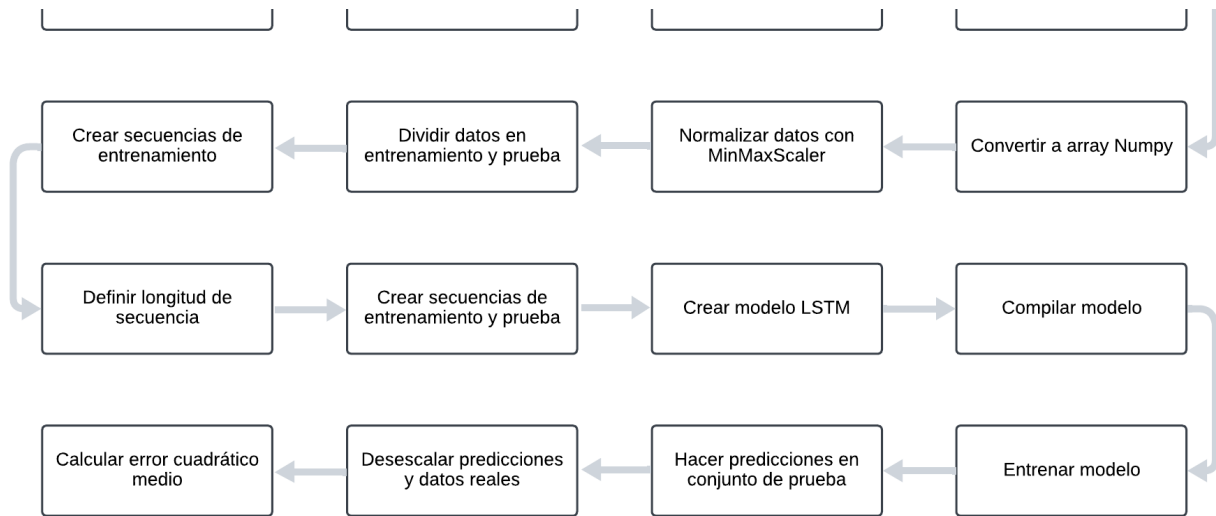
*Tabla 4 Descripción del diagrama de flujo fase 3.*

<b>PROCESO</b>	<b>DESCRIPCIÓN</b>
<b>Desplegar modelo</b>	Implica poner el modelo en producción, hacerlo disponible para su uso en un entorno real.
<b>Monitoreo</b>	Supervisión del rendimiento del modelo Machine Learning en producción.

### **2.3.2 Diagrama de bloques**

Los diagramas de bloques desempeñan un papel fundamental en la representación y comprensión de la estructura de un modelo predictivo. En el contexto de la modelización predictiva, donde se utilizan datos históricos y algoritmos para anticipar eventos futuros, la visualización y claridad son de suma importancia puesto que los diagramas de bloques actúan como una herramienta esencial que permite desglosar y presentar de manera efectiva la complejidad inherente de estos modelos; Además de ello, los diagramas de bloques son una opción sólida debido a su simplicidad, capacidad de representar flujos de datos y flexibilidad en la adaptación a las necesidades específicas.

*Ilustración 3 Diagrama de bloques.*



### DESCRIPCIÓN DIAGRAMA DE BLOQUES:

*Tabla 5 Descripción del diagrama de bloques.*

<b>BLOQUE</b>	<b>DESCRIPCIÓN</b>
<b>Importar librerías</b>	<p>Este bloque importa las librerías necesarias para el análisis de datos, incluyendo:</p> <ul style="list-style-type: none"> <li>• numpy: Para trabajar con datos numéricos.</li> <li>• pandas: Para trabajar con datos de series de tiempo.</li> <li>• tensorflow: Para crear modelos de aprendizaje automático.</li> </ul>

<b>Cargar datos de serie de tiempo</b>	Este bloque carga los datos ambientales en un formato de serie de tiempo. Los datos pueden ser cargados desde un archivo CSV, un archivo JSON o una base de datos.
<b>Establecer índice y frecuencia de datos</b>	la columna que se utilizará para identificar los datos. La frecuencia es el número de puntos de datos que se incluyen en cada intervalo de tiempo.
<b>Normalizar datos con MinMaxScaler</b>	Este bloque normaliza los datos para que tengan un rango de [0, 1]. La normalización es un proceso que ayuda a que el modelo sea más eficiente y preciso.
<b>Crear modelo LSTM</b>	Este bloque crea un modelo LSTM para predecir la variable de interés. El modelo LSTM es un tipo de red neuronal recurrente que es eficaz para predecir datos secuenciales.
<b>Seleccionar variable a predecir</b>	Selecciona la variable de interés que se desea predecir a partir de los datos.
<b>Convertir a array Numpy</b>	Este bloque convierte los datos a un array Numpy para que puedan ser utilizados por el modelo. Los arrays Numpy son un tipo de estructura de datos que es eficiente para el procesamiento de datos numéricos.

<b>Compilar modelo</b>	Este bloque compila el modelo LSTM. La compilación del modelo establece los parámetros de entrenamiento y los algoritmos de optimización.
<b>Entrenar modelo</b>	Este bloque entrena el modelo LSTM en los datos de entrenamiento. El entrenamiento del modelo implica ajustar los parámetros del modelo para que se ajusten a los datos de entrenamiento.
<b>Hacer predicciones en conjunto de prueba</b>	Este bloque hace predicciones en el conjunto de datos de prueba. El conjunto de datos de prueba se utiliza para evaluar el rendimiento del modelo en datos nuevos.
<b>Calcular error cuadrático medio</b>	Este bloque calcula el error cuadrático medio (RMSE) de las predicciones. El RMSE es una medida de la precisión de las predicciones.
<b>Des escalar predicciones y datos reales</b>	Este bloque des escala las predicciones y los datos reales para que tengan la misma escala que los datos originales. La desescalada es un proceso que ayuda a que las predicciones sean más interpretables.

### 2.3.3 Recopilación y preparación de los datos.

**Información de las bases de datos.**

Datos diarios de resolución nativa CERES/MERRA2 de la NASA/POWERMERRA-2 (Modern-Era Retrospective analysis for Research and Applications, version 2) es una reanálisis atmosférica desarrollado por la NASA. Proporciona una estimación detallada y coherente del estado de la atmósfera global, incluyendo parámetros climáticos, en una resolución espacial y temporal consistente.

Esta reanálisis proporciona información en múltiples niveles de la atmósfera, incluyendo variables como temperatura, humedad, presión, viento, precipitación, entre otros. Estos datos son útiles para una amplia gama de aplicaciones científicas y de investigación, como el estudio del clima, la predicción meteorológica, la modelización del cambio climático, la evaluación de recursos energéticos renovables, y más.

Fechas (mes/día/año): 01/01/2015 al 31/12/2022 (El 1 de enero de 2015 y el 31 de diciembre de 2022, ambos incluidos, es de 2,922 días.) y 2023

Ubicación: Latitud 5.3077, Longitud -73.8156

Elevación de MERRA-2: Promedio para la región de lat/lon de 0.5 x 0.625 grados = 2262.44 metros

El valor para los datos faltantes que no se pueden calcular o están fuera del rango de disponibilidad de las fuentes: -999

Parámetro(s):

T2M: Temperatura a 2 metros (C)

T2MDEW: Punto de rocío/escarcha a 2 metros (C)

T2MWET: Temperatura de bulbo húmedo a 2 metros (C)

TS: Temperatura de la superficie terrestre (C)

T2M\_RANGE: Rango de temperatura a 2 metros (C)

T2M\_MAX: Temperatura máxima a 2 metros (C)

T2M\_MIN: Temperatura mínima a 2 metros (C)

QV2M: Humedad específica de a 2 metros (g/kg)

RH2M: Humedad relativa de a 2 metros (%)

PRECTOTCORR: Precipitación corregida (mm/día)

PS: Presión de superficie de (kPa)

WS10M: Velocidad del viento a 10 metros (m/s)

WS10M\_MAX: Velocidad máxima del viento a 10 metros (m/s)

WS10M\_MIN: Velocidad mínima del viento a 10 metros (m/s)

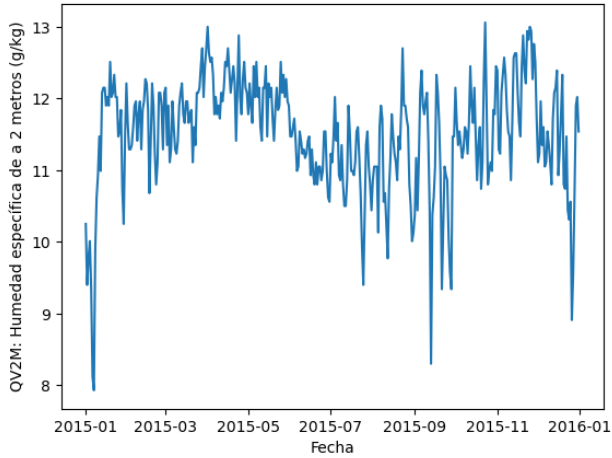
#### **2.3.4 Análisis y graficas.**

##### **QV2M: Humedad específica de a 2 metros (g/kg)**

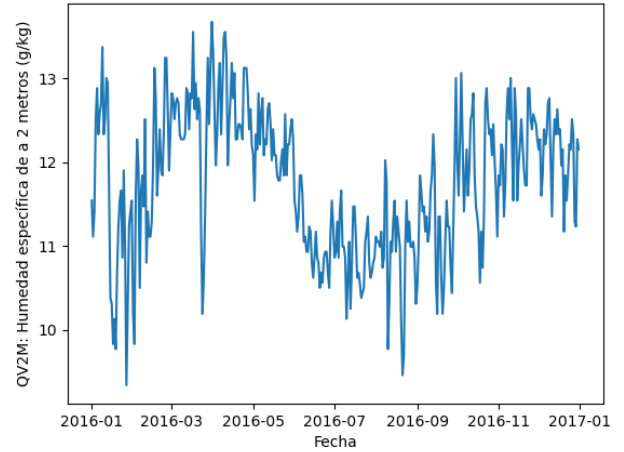
La sigla "QV2M" se refiere a la "Humedad Específica a 2 Metros", y representa la cantidad de vapor de agua presente en el aire en forma de masa por unidad de masa de aire seco, expresada en gramos por kilogramo (g/kg). Esta medida es utilizada en meteorología para cuantificar la cantidad de humedad en la atmósfera a una altura de 2 metros sobre la superficie terrestre. La humedad específica a 2 metros es importante para comprender las condiciones de humedad en un entorno determinado y para predecir la formación de nubes, lluvia y otros fenómenos meteorológicos relacionados con la humedad del aire.

*Ilustración 4 QV2M: Humedad específica de a 2 metros por año.*

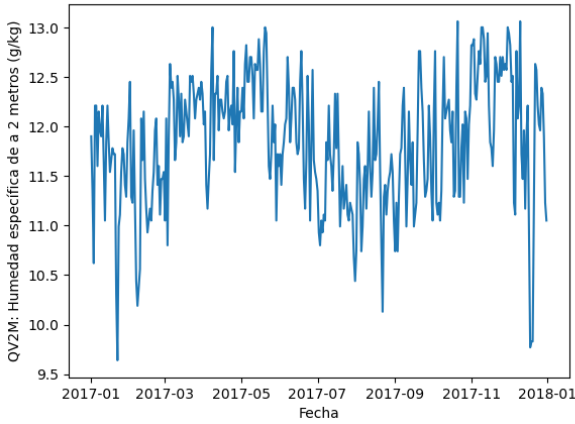
Gráfica de QV2M: Humedad específica de a 2 metros (g/kg) - Año 2015



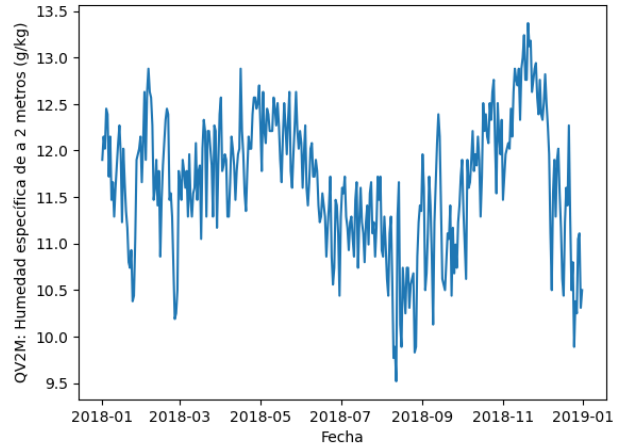
Gráfica de QV2M: Humedad específica de a 2 metros (g/kg) - Año 2016



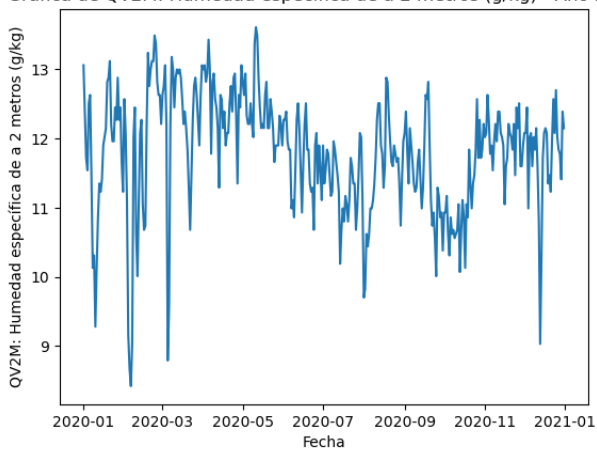
Gráfica de QV2M: Humedad específica de a 2 metros (g/kg) - Año 2017



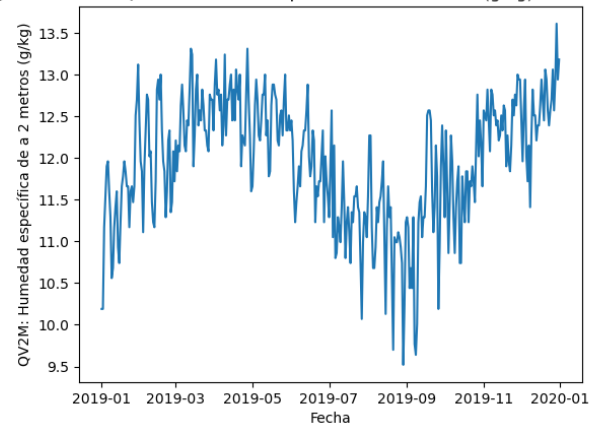
Gráfica de QV2M: Humedad específica de a 2 metros (g/kg) - Año 2018



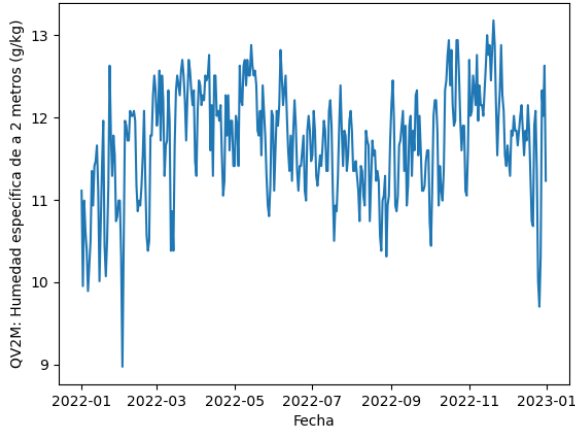
Gráfica de QV2M: Humedad específica de a 2 metros (g/kg) - Año 2020



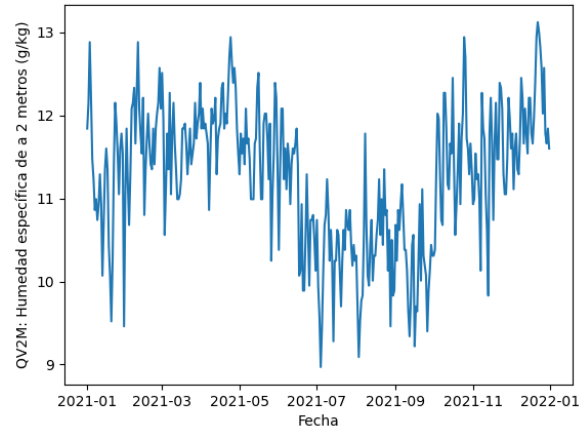
Gráfica de QV2M: Humedad específica de a 2 metros (g/kg) - Año 2019



Gráfica de QV2M: Humedad específica de a 2 metros (g/kg) - Año 2022

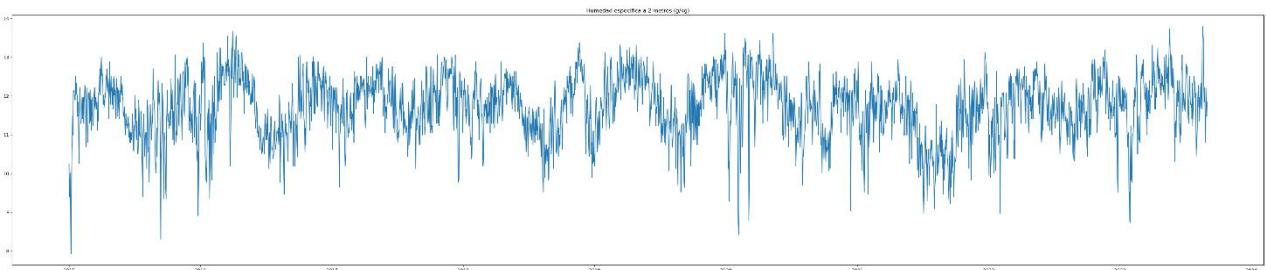


Gráfica de QV2M: Humedad específica de a 2 metros (g/kg) - Año 2021



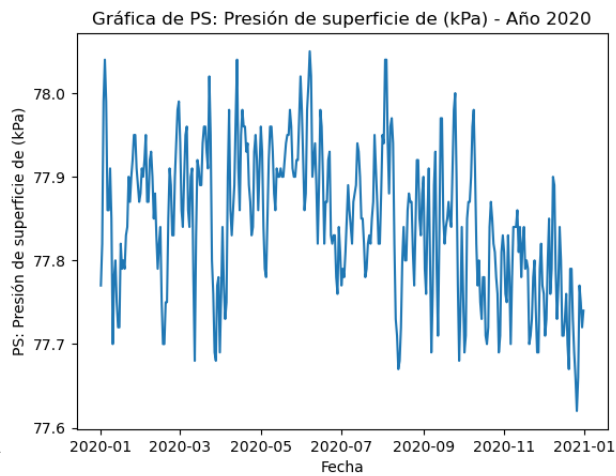
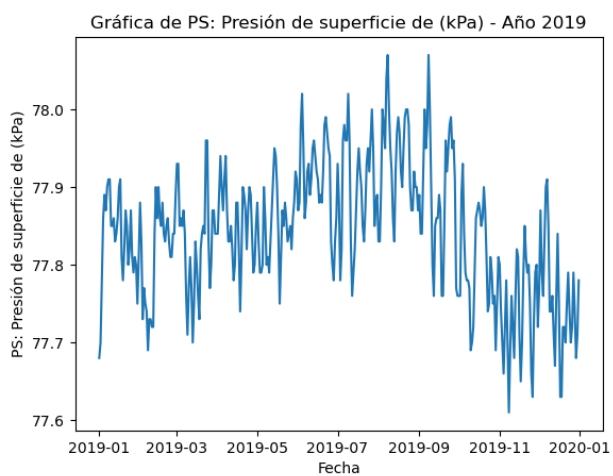
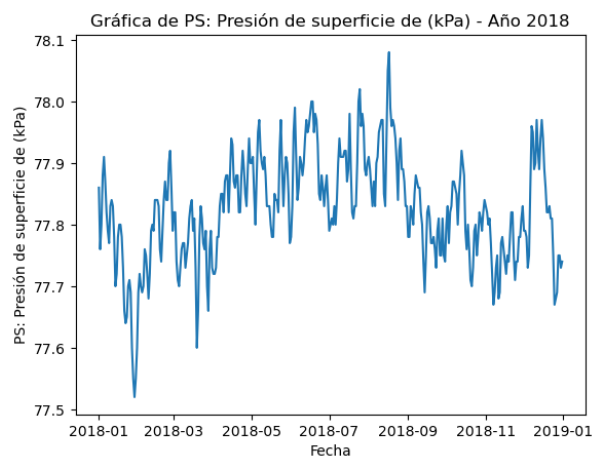
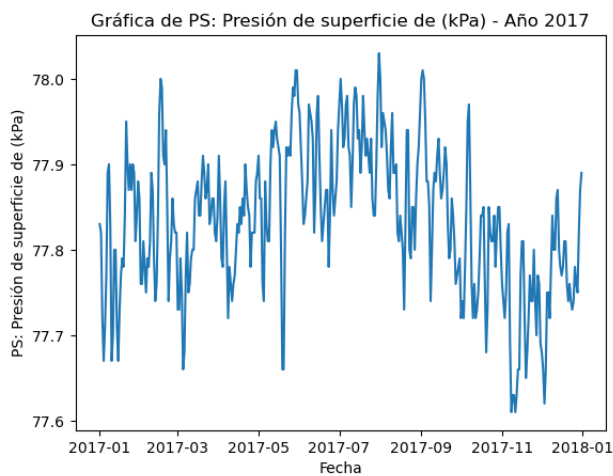
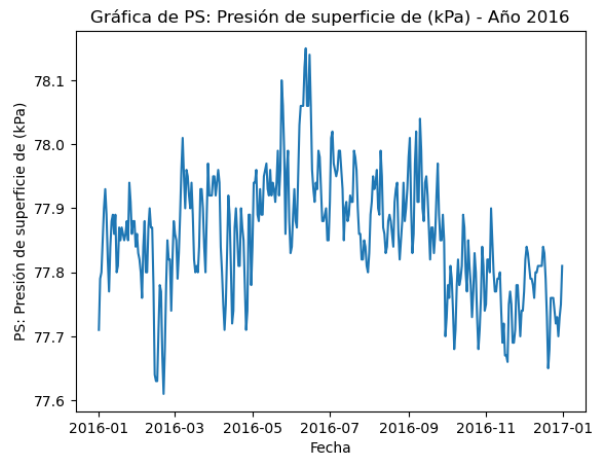
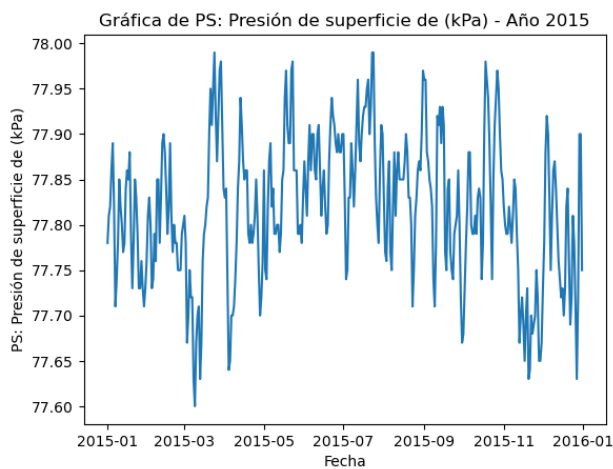
### QV2M Gráfica 2015 – 2022

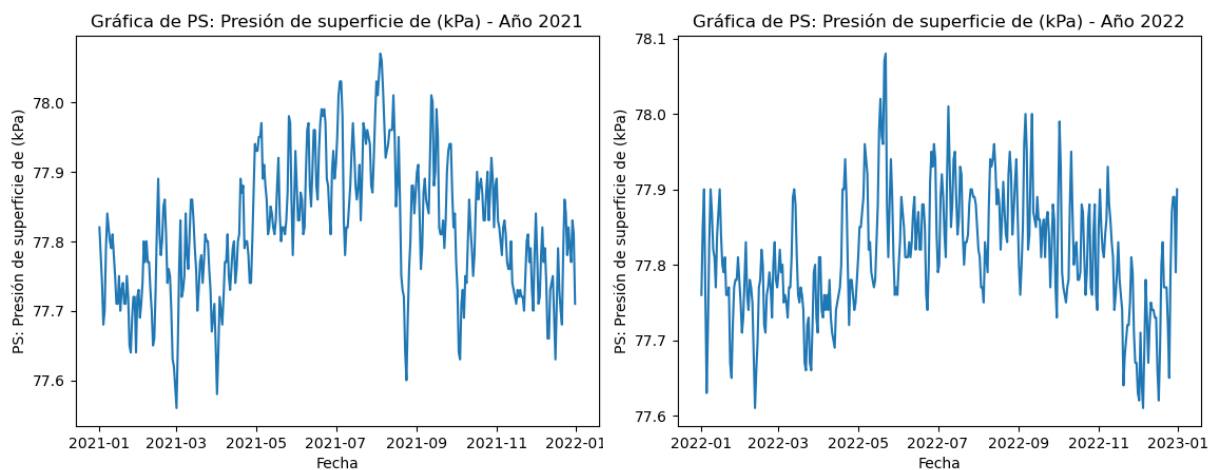
*Ilustración 5 QV2M: Humedad específica de a 2 metros*



### PS: Presión de superficie de (kPa)

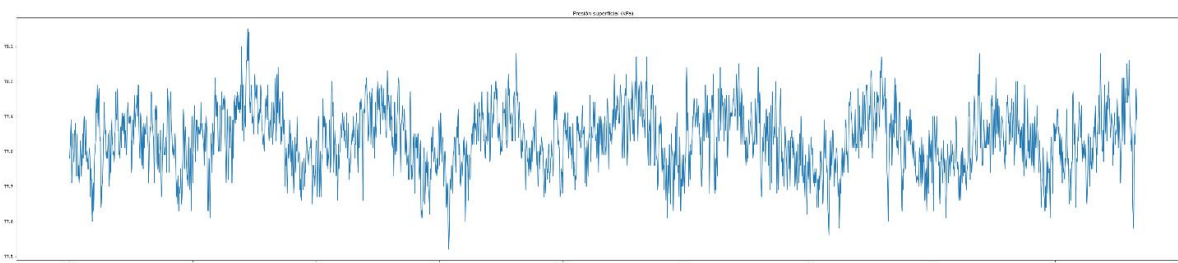
La abreviatura "PS" se refiere a la "Presión de Superficie", que representa la fuerza ejercida por la columna de aire atmosférico en un punto específico de la superficie terrestre. Esta medida se expresa en kilopascales (kPa) u otras unidades de presión. La presión de superficie es esencial para comprender la circulación atmosférica y los patrones climáticos, ya que influye en la dirección y velocidad del viento, así como en la formación y desarrollo de sistemas meteorológicos como áreas de alta y baja presión. La presión de superficie también es fundamental para la predicción del clima y el tiempo, ya que los cambios en la presión pueden indicar la llegada de sistemas frontales, cambios en la estabilidad atmosférica y otros fenómenos meteorológicos.

*Ilustración 6 PS: Presión de superficie por año*



*PS Gráfica 2015 - 2022*

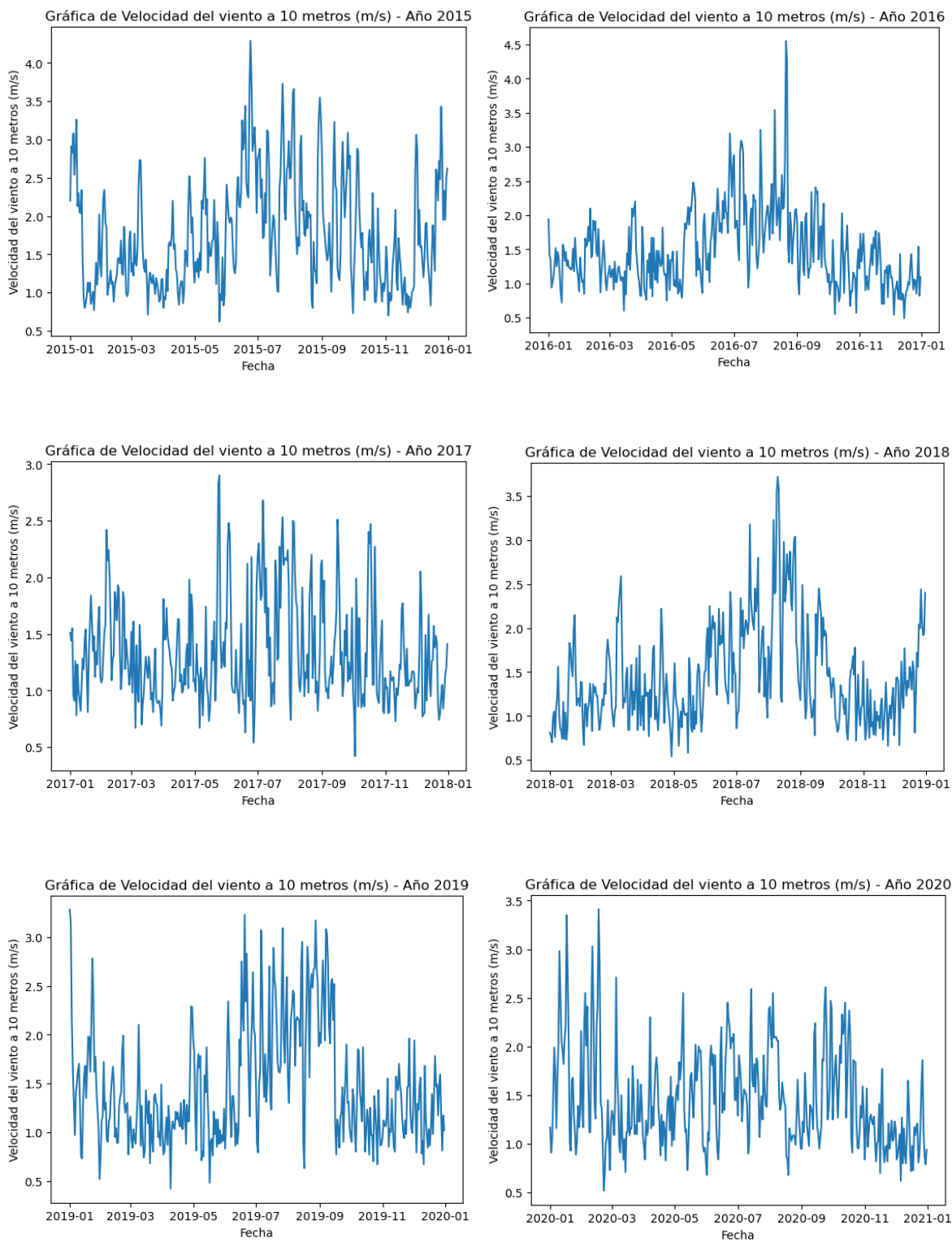
### *Ilustración 7 PS: Presión de superficie*

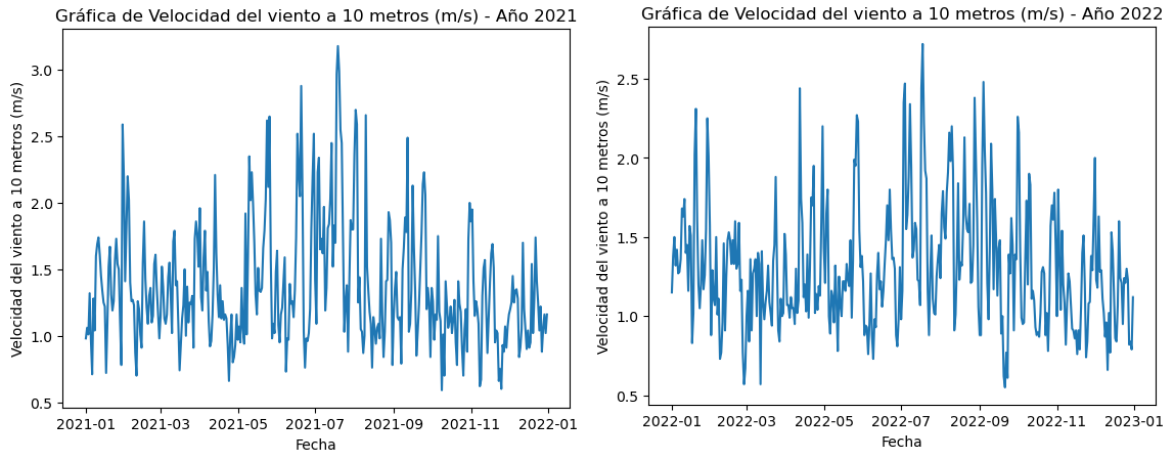


### **WS10M: Velocidad del viento a 10 metros (m/s)**

La abreviatura "WS10M" se refiere a la "Velocidad del Viento a 10 Metros", que representa la rapidez del movimiento del aire medida a una altura de 10 metros sobre la superficie terrestre. Esta medida se expresa en metros por segundo (m/s) y es utilizada en meteorología para describir la velocidad del viento en una ubicación específica. La velocidad del viento a 10 metros es importante para comprender los patrones de circulación atmosférica, la dispersión de contaminantes, la generación de energía eólica y la influencia del viento en la formación de condiciones climáticas y meteorológicas.

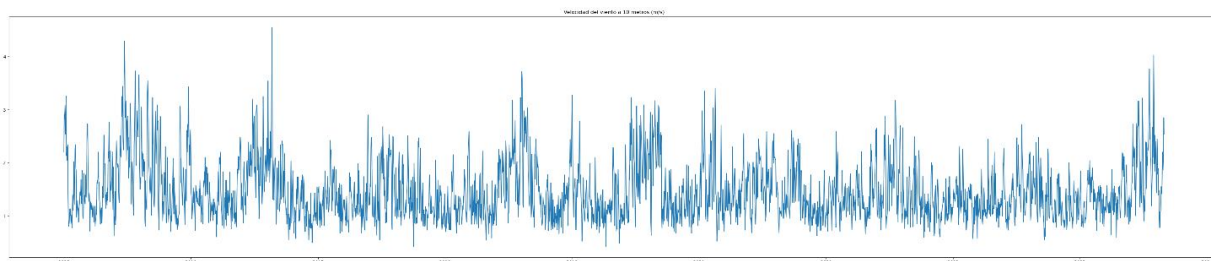
*Ilustración 8WS10M: Velocidad del viento a 10 metros por año*





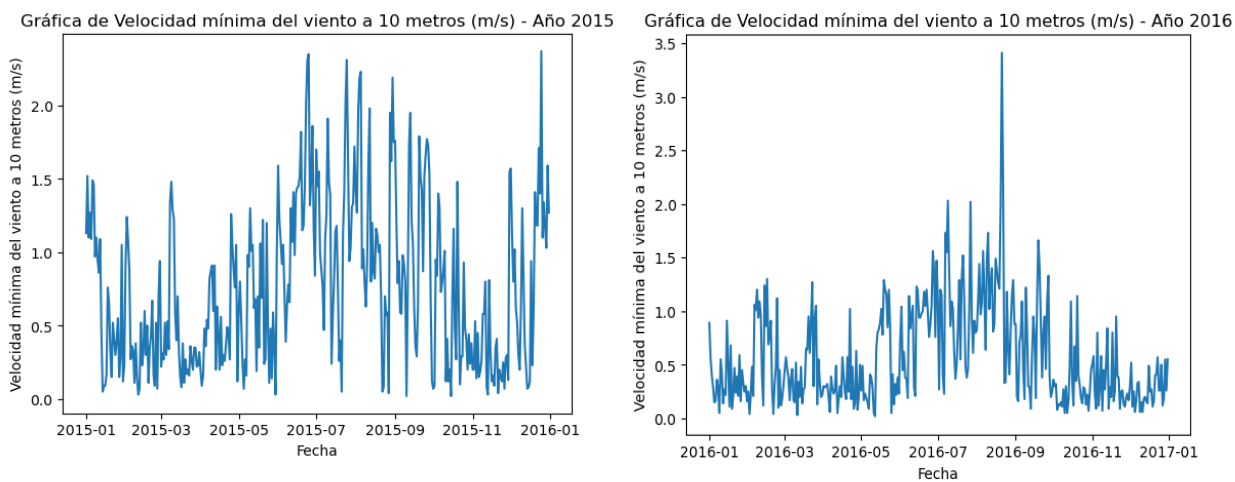
**WS10M Gráfica 2015 – 2022**

*Ilustración 9 WS10M: Velocidad del viento a 10 metros*

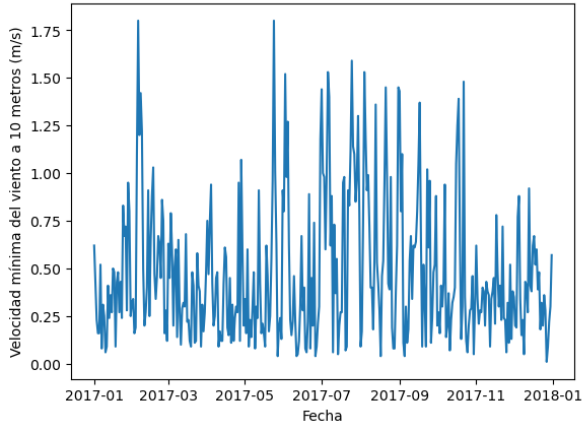


**WS10M\_MIN: Velocidad mínima del viento a 10 metros (m/s)**

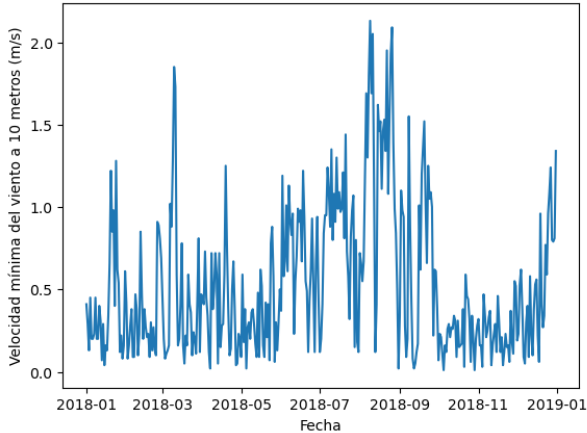
*Ilustración 10 WS10M\_MIN: Velocidad mínima del viento a 10 metros por año*



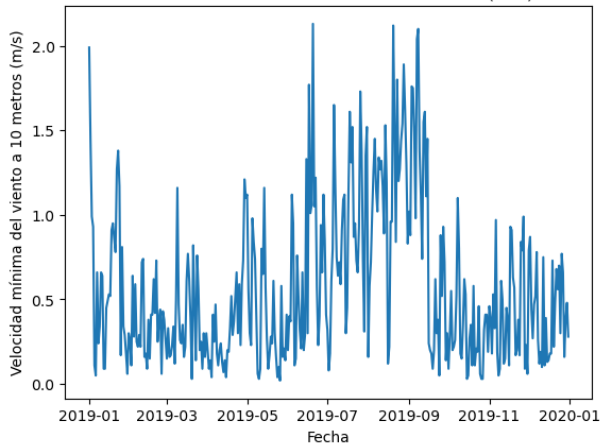
Gráfica de Velocidad mínima del viento a 10 metros (m/s) - Año 2017



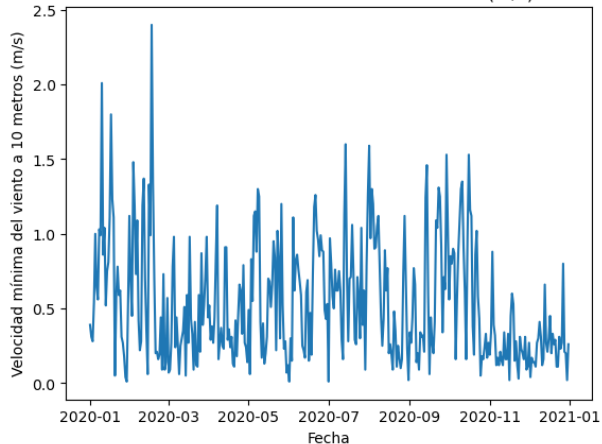
Gráfica de Velocidad mínima del viento a 10 metros (m/s) - Año 2018



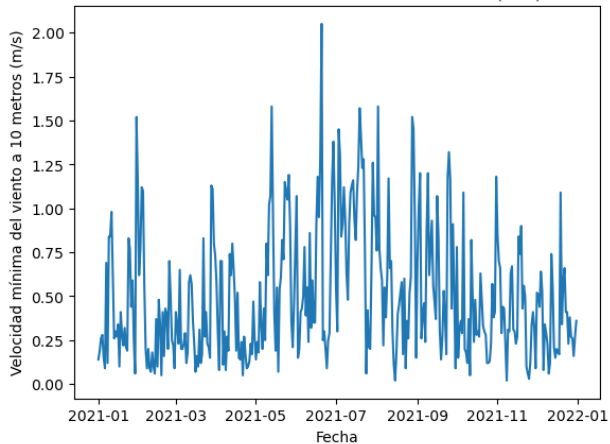
Gráfica de Velocidad mínima del viento a 10 metros (m/s) - Año 2019



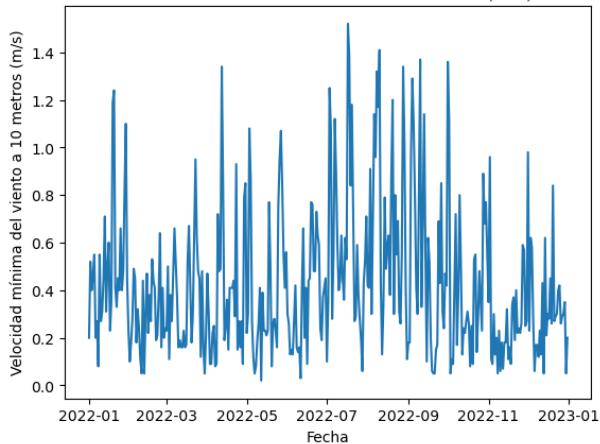
Gráfica de Velocidad mínima del viento a 10 metros (m/s) - Año 2020



Gráfica de Velocidad mínima del viento a 10 metros (m/s) - Año 2021

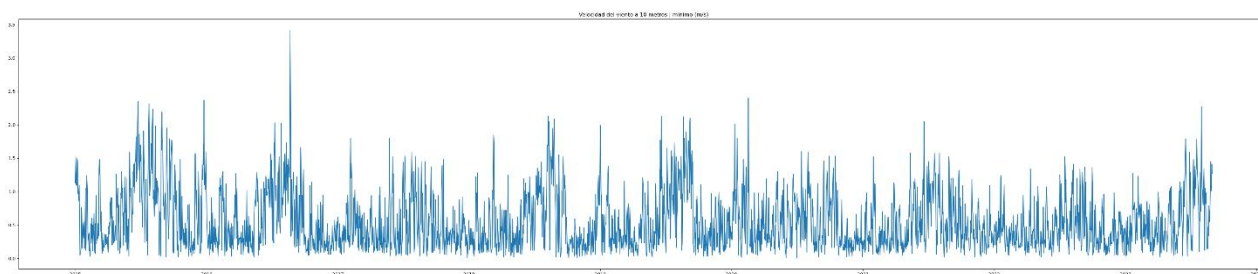


Gráfica de Velocidad mínima del viento a 10 metros (m/s) - Año 2022



## *WS10M\_MIN Gráfica 2015 - 2022*

*Ilustración 11 WS10M\_MIN: Velocidad mínima del viento a 10 metros*

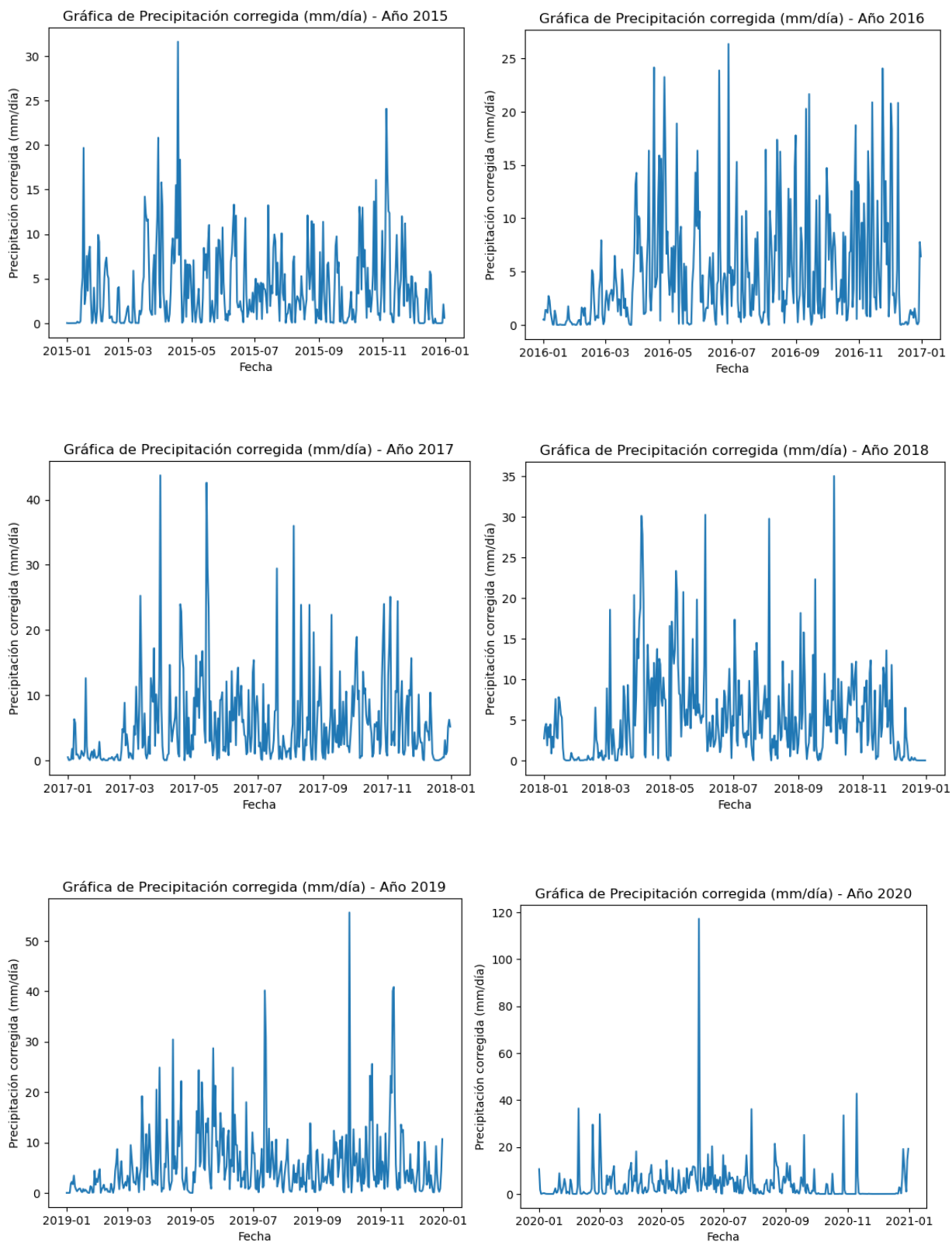


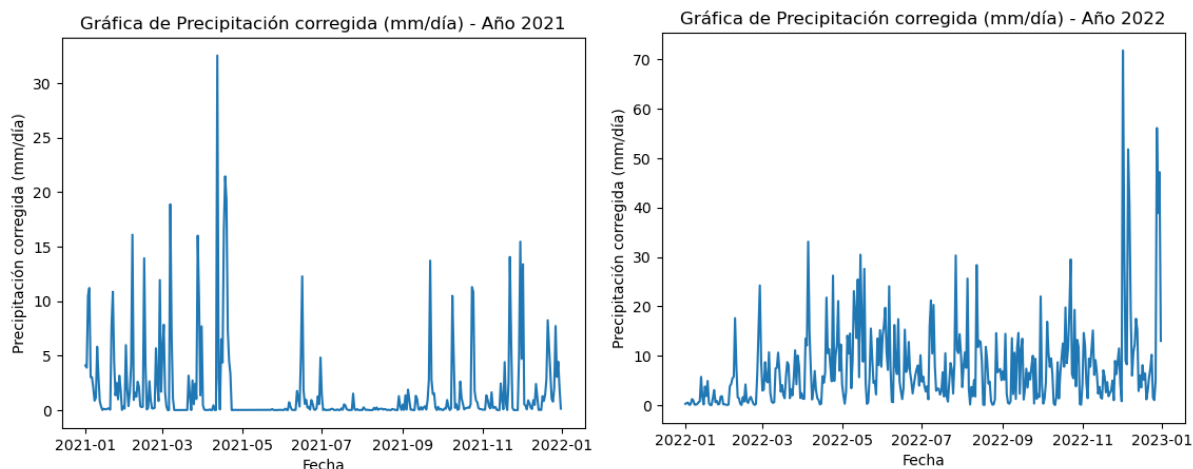
### **PRECTOTCORR: Precipitación corregida (mm/día)**

La abreviatura "PRECTOTCORR" se refiere a la "Precipitación Corregida", que representa la cantidad total de agua líquida o sólida que cae de la atmósfera y llega a la superficie terrestre en forma de precipitación en un período de un día. Esta medida se expresa en milímetros por día (mm/día). La corrección en la denominación sugiere que los datos pueden haber sido ajustados o corregidos para tener en cuenta posibles errores o variaciones en las mediciones.

La precipitación corregida es una variable meteorológica crucial para entender los patrones climáticos y las condiciones hidrológicas de una región. Ayuda a determinar la cantidad de agua disponible para fuentes de agua dulce, como ríos y embalses, y es fundamental para predecir y gestionar eventos climáticos extremos como lluvias intensas, inundaciones y sequías.

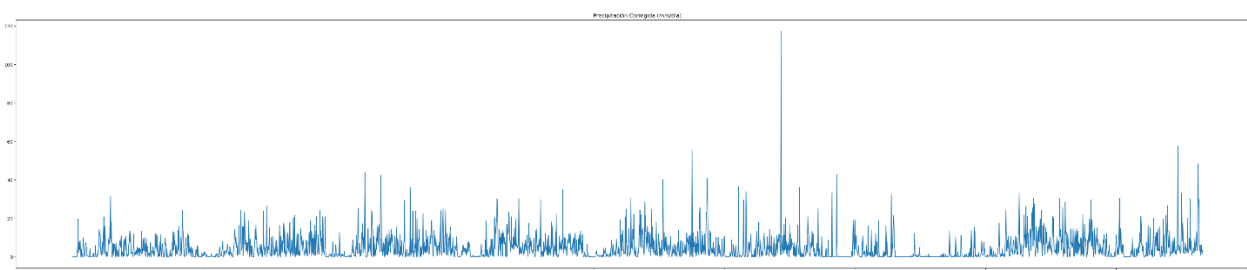
*Ilustración 12 PRECTOTCORR: Precipitación corregida por año*





### PRECTOTCORR Gráfica 2015 – 2022

*Ilustración 13 PRECTOTCORR: Precipitación corregida*



### RH2M: Humedad relativa de a 2 metros (%)

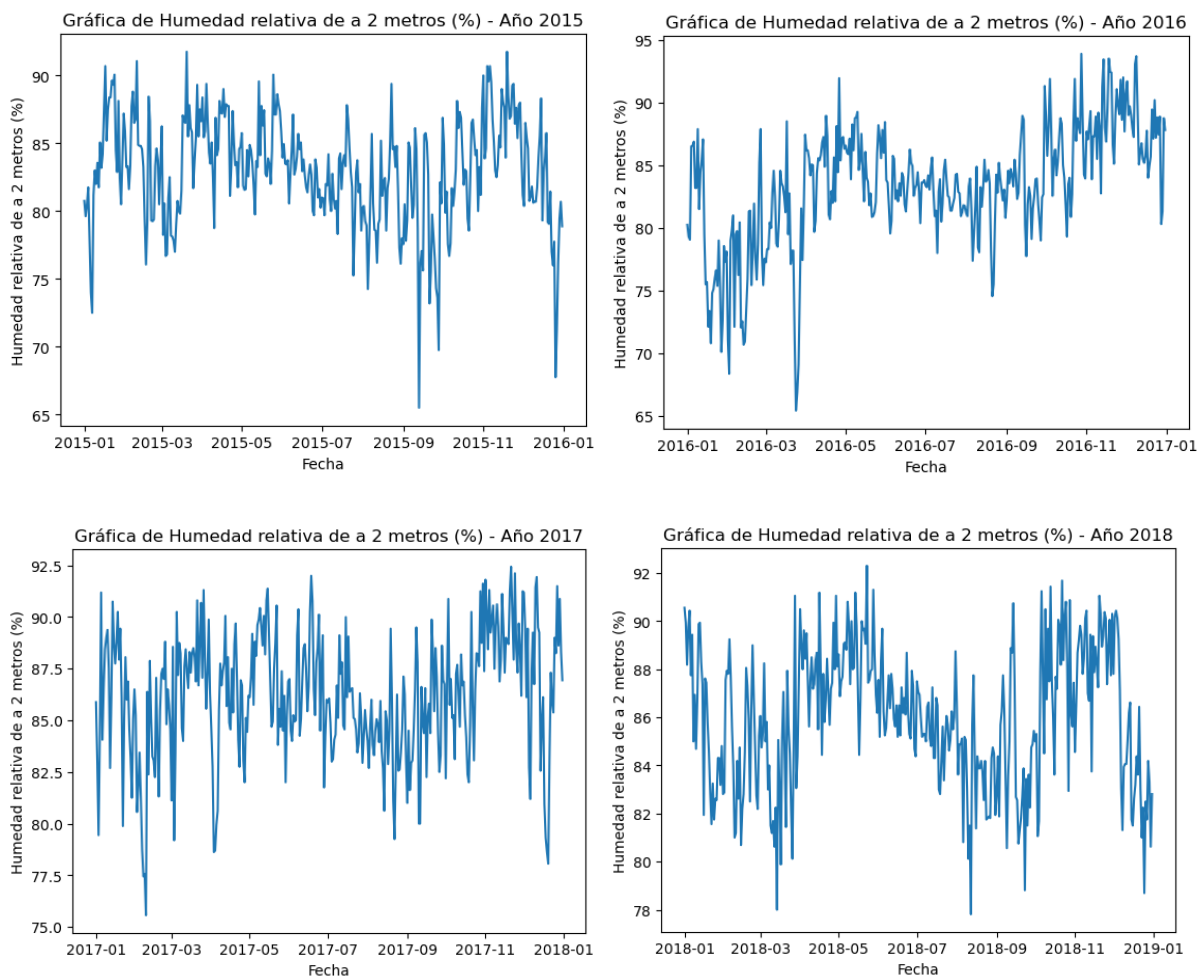
La abreviatura "RH2M" se refiere a la "Humedad Relativa a 2 Metros". La humedad relativa es una medida que expresa la cantidad de vapor de agua presente en el aire en relación con la cantidad máxima que el aire podría contener a una temperatura dada, y se expresa en porcentaje (%).

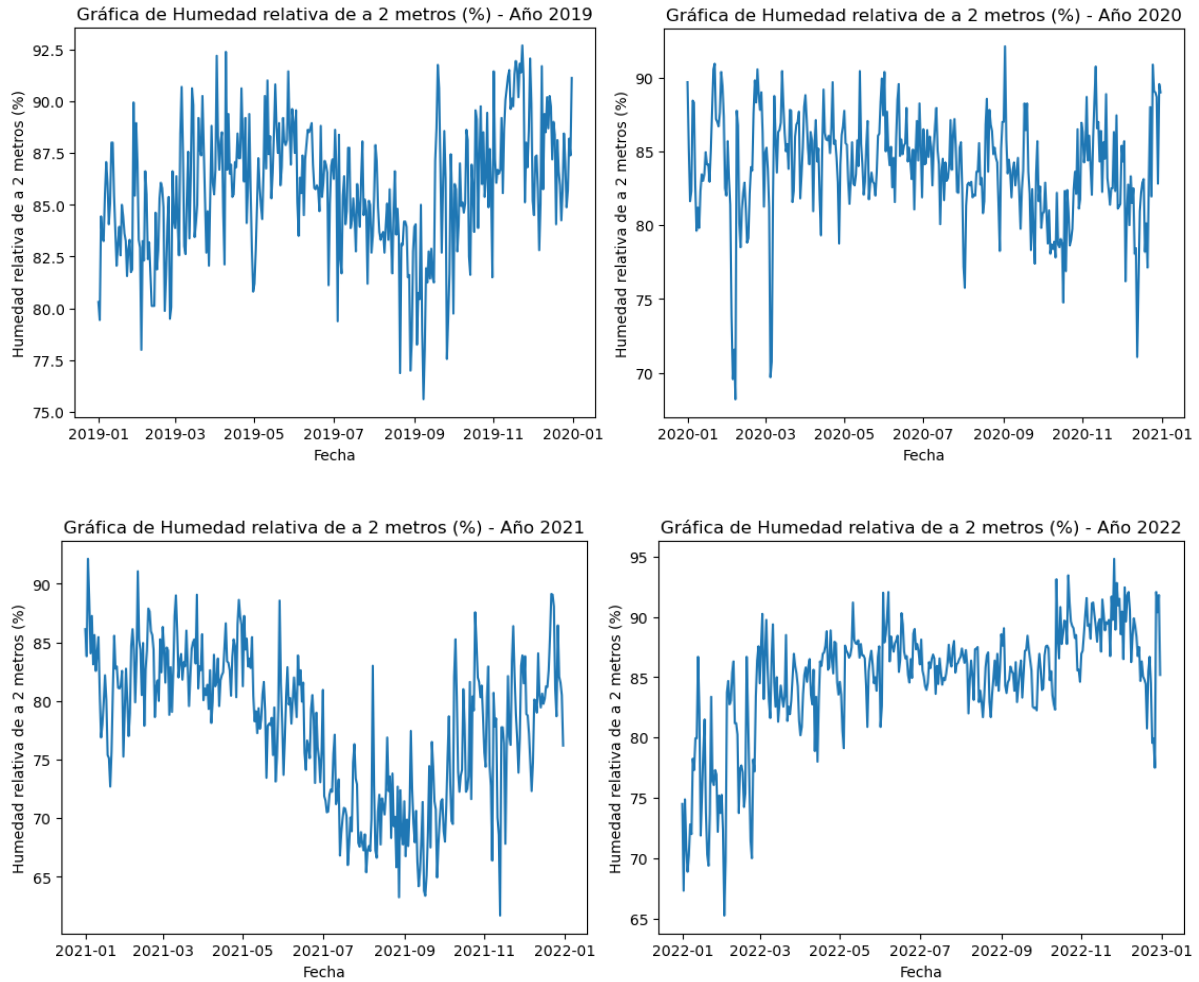
En el caso específico de "RH2M", la medición se realiza a una altura de 2 metros sobre la superficie terrestre. Esto es relevante porque la humedad relativa puede variar según la altura y las condiciones locales, y la medición a 2 metros es una referencia común en meteorología para evaluar la humedad que está más cerca del nivel donde las personas y las

actividades humanas ocurren con mayor frecuencia.

La humedad relativa es una variable importante en meteorología, ya que influye en la sensación térmica, la formación de nubes, la probabilidad de precipitación y otros fenómenos climáticos. Un valor de humedad relativa del 100% indica que el aire está completamente saturado de vapor de agua y no puede contener más, mientras que un valor más bajo indica que el aire tiene una capacidad de retener más vapor de agua.

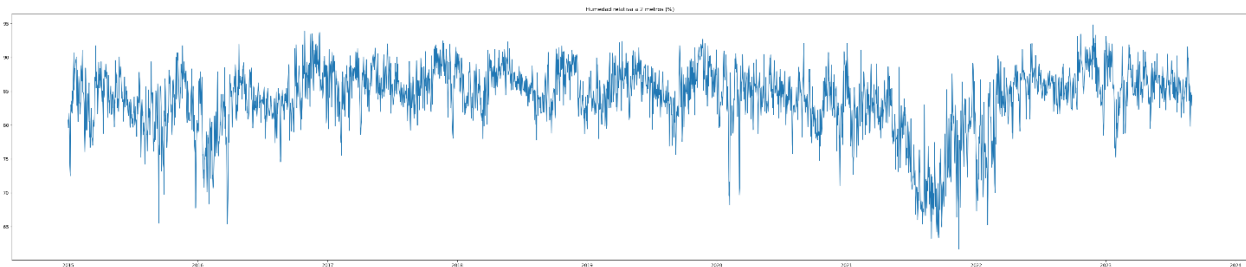
*Ilustración 14 RH2M: Humedad relativa de a 2 metros por año*





### RH2M Gráfica 2015 – 2022

*Ilustración 15 RH2M: Humedad relativa de a 2 metros*

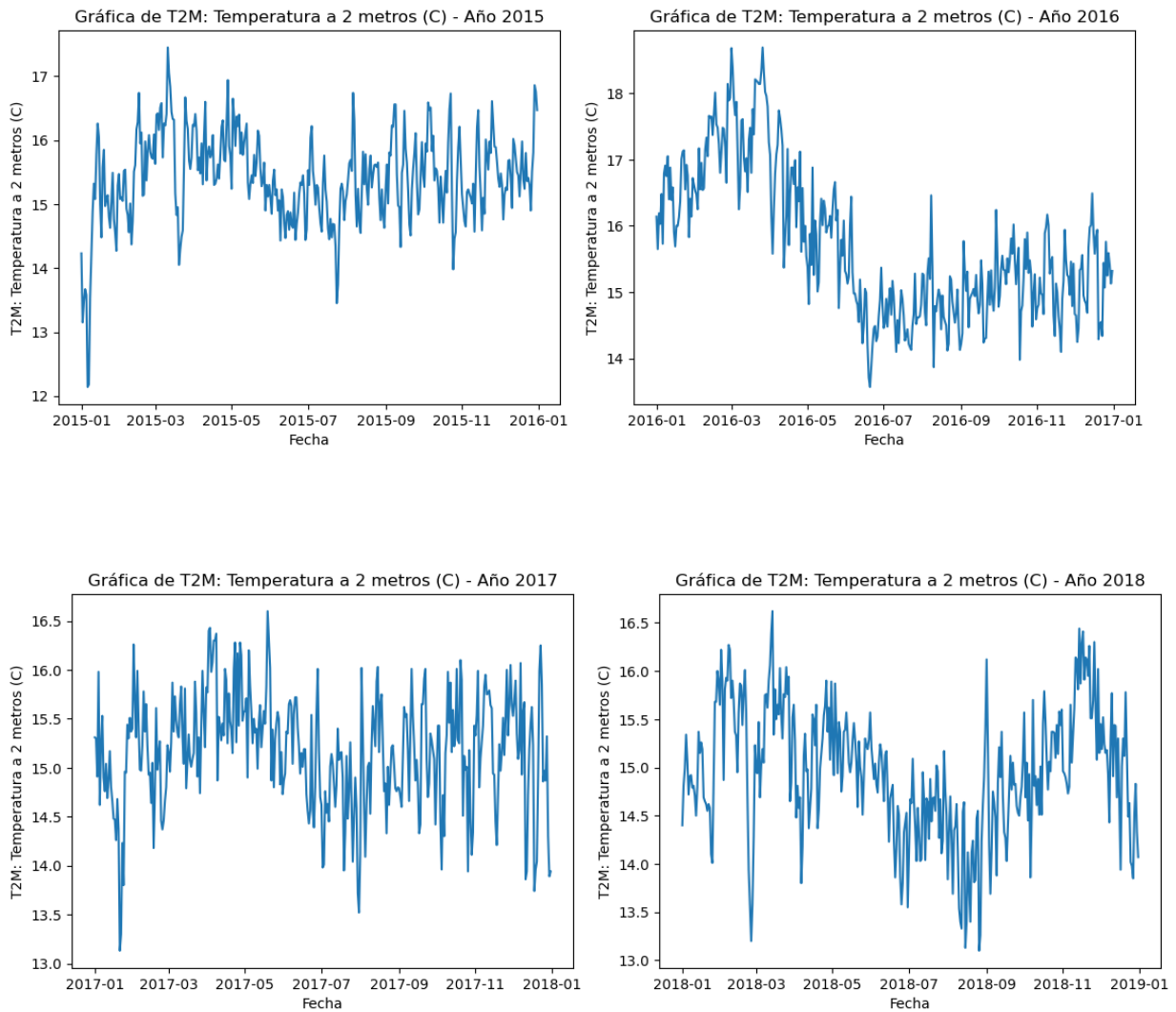


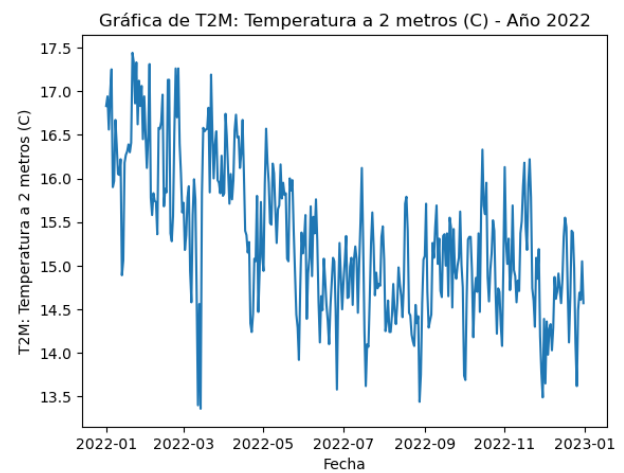
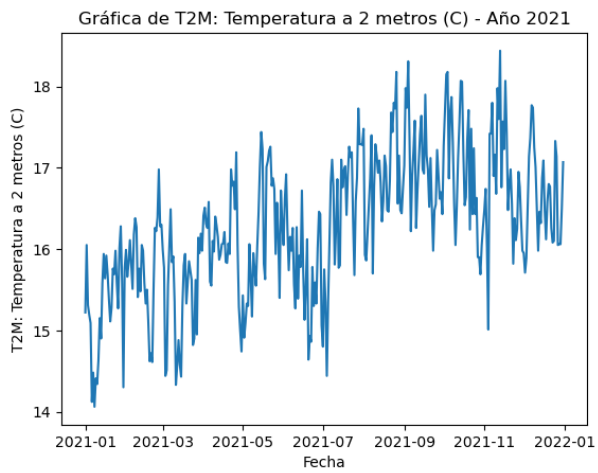
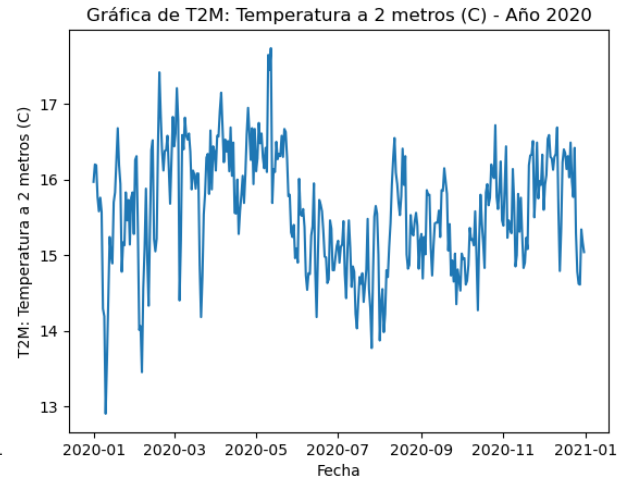
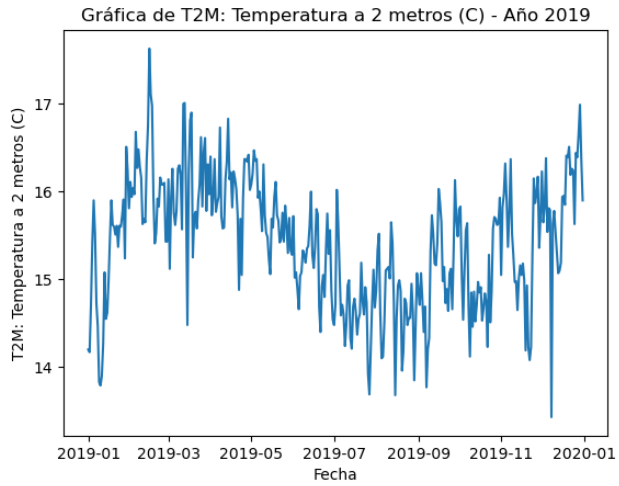
### T2M: Temperatura a 2 metros (C)

La abreviatura "T2M" se refiere a la "Temperatura a 2 Metros", que representa la temperatura del aire medida a una altura de 2 metros sobre la superficie terrestre. Esta medida se expresa en grados Celsius ( $^{\circ}\text{C}$ ) y es utilizada en meteorología y climatología para describir

las condiciones térmicas cerca de la superficie de la Tierra. La temperatura a 2 metros es una variable esencial para comprender y pronosticar el clima y el tiempo, ya que tiene un impacto significativo en los patrones climáticos, la formación de nubes, la precipitación y otros fenómenos atmosféricos.

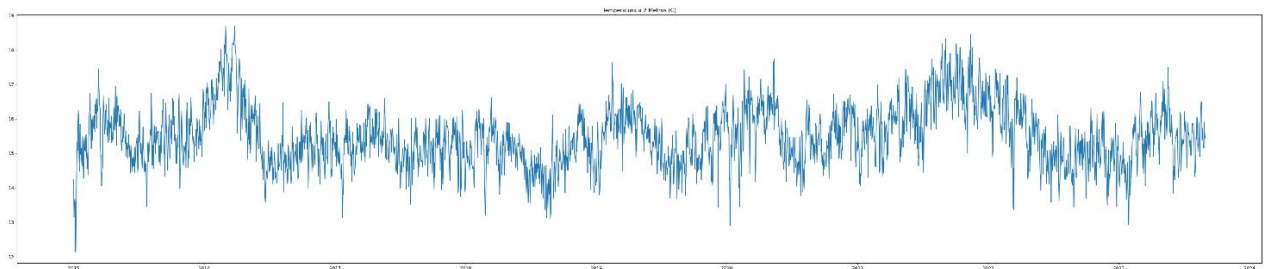
Ilustración 16 T2M: Temperatura a 2 metros por año

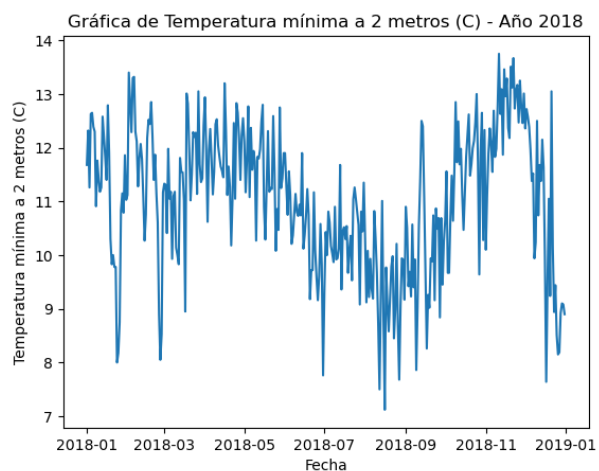
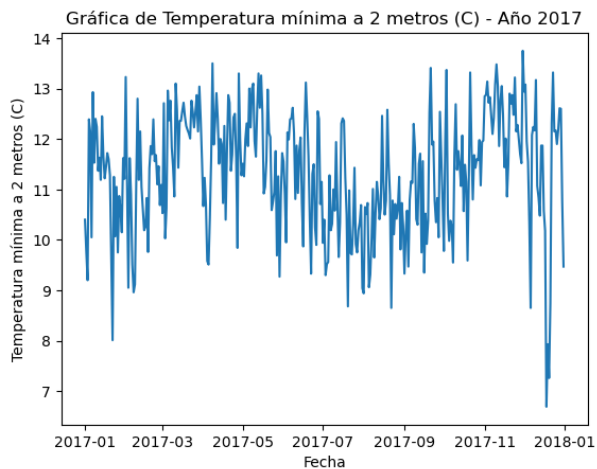
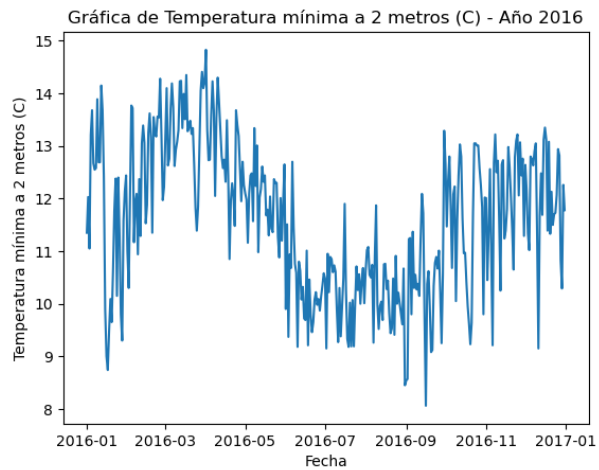
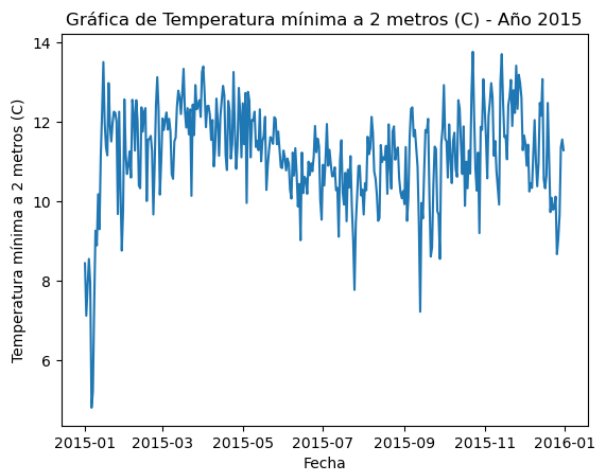


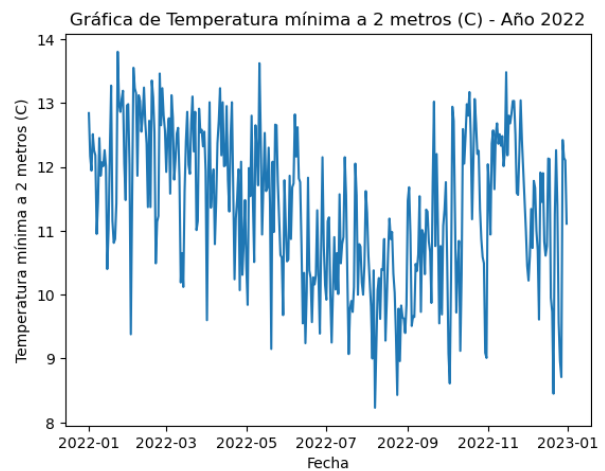
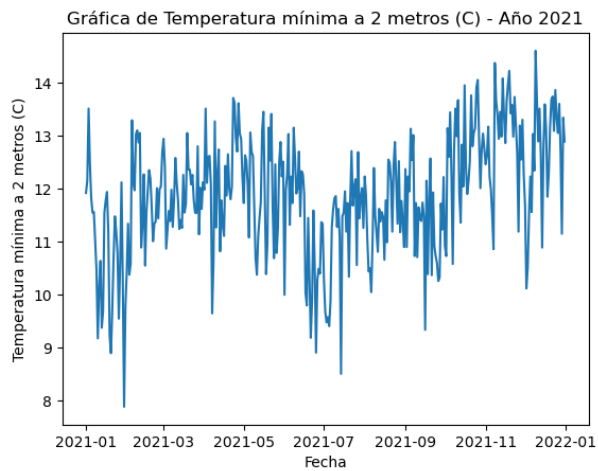
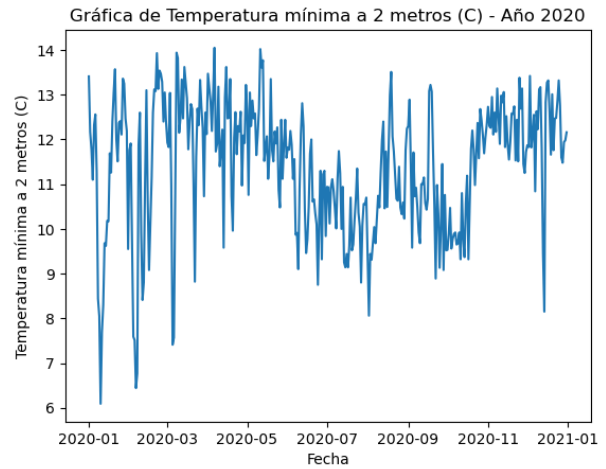
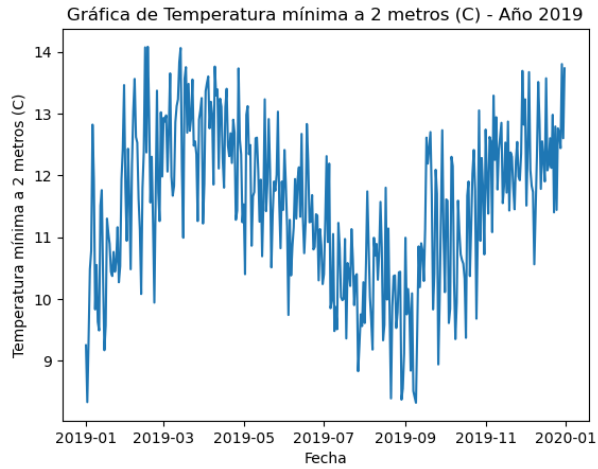


### T2M Gráfica 2015 – 2022

*Ilustración 17 T2M: Temperatura a 2 metros*

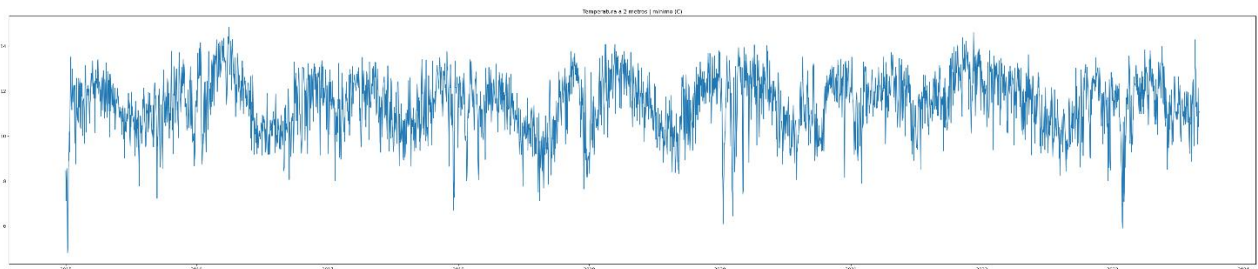


**T2M\_MIN: Temperatura mínima a 2 metros (C)***Ilustración 18 T2M\_MIN: Temperatura mínima a 2 metros por año*

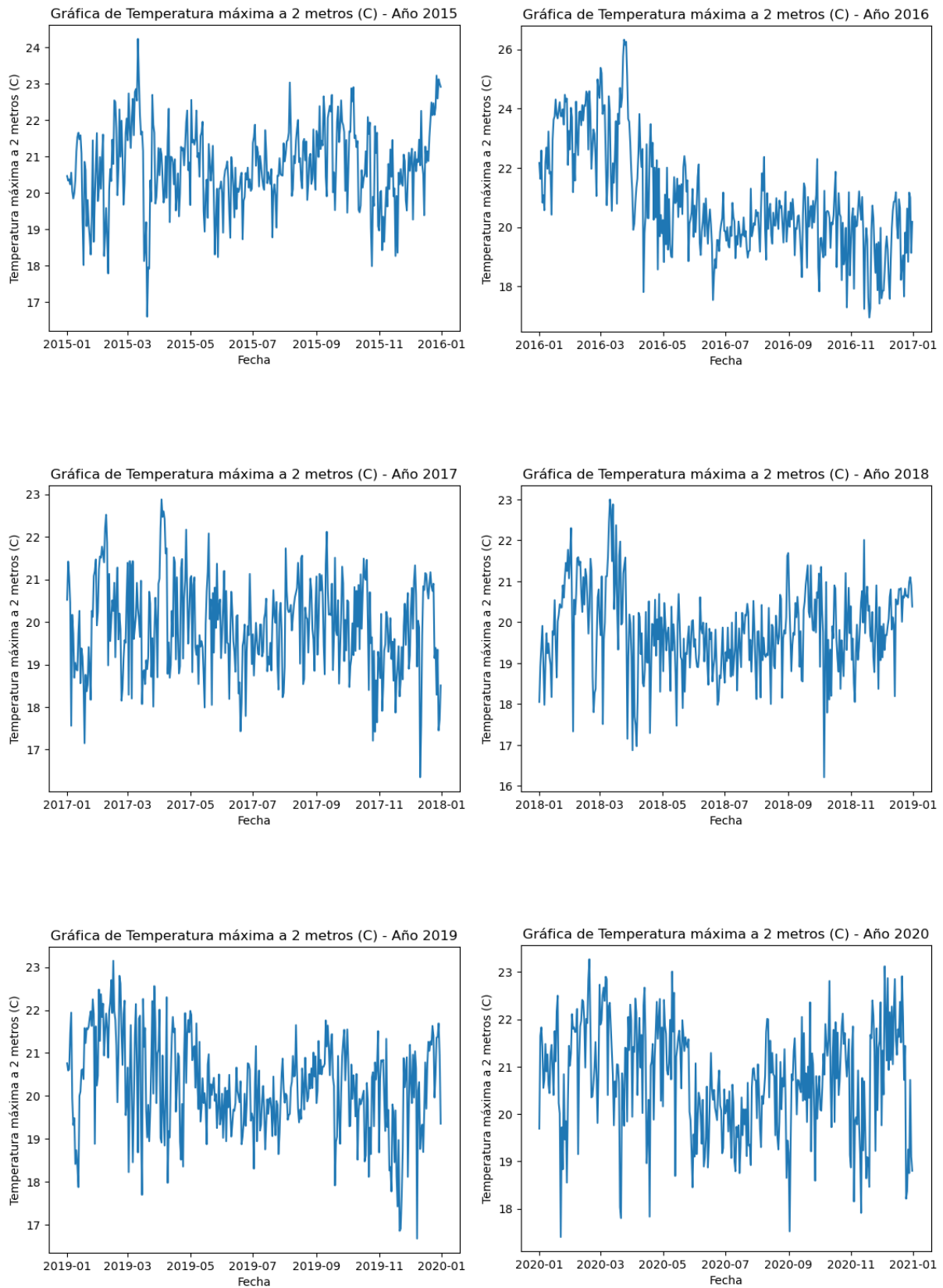


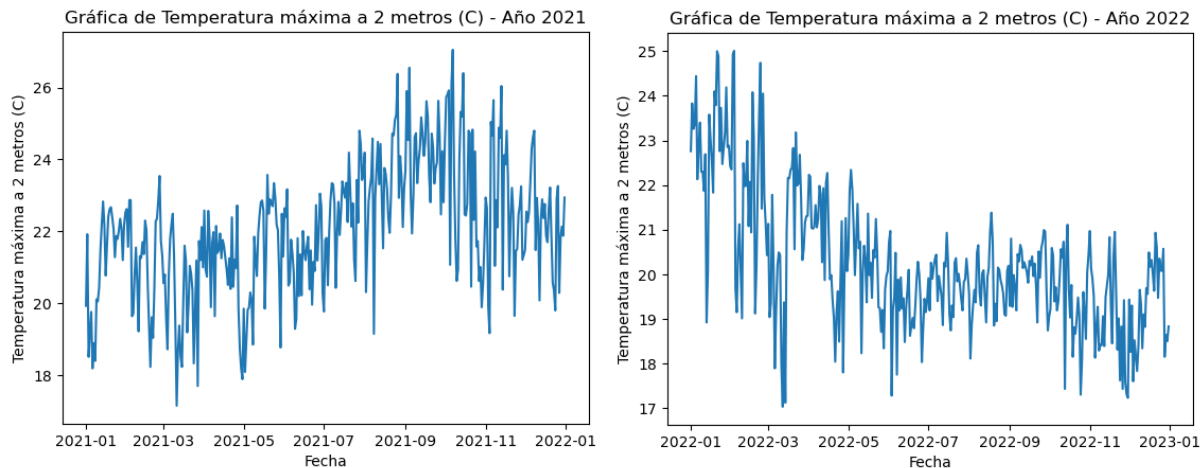
### T2M\_MIN Gráfica 2015 – 2022

*Ilustración 19 T2M\_MIN: Temperatura mínima a 2 metros*



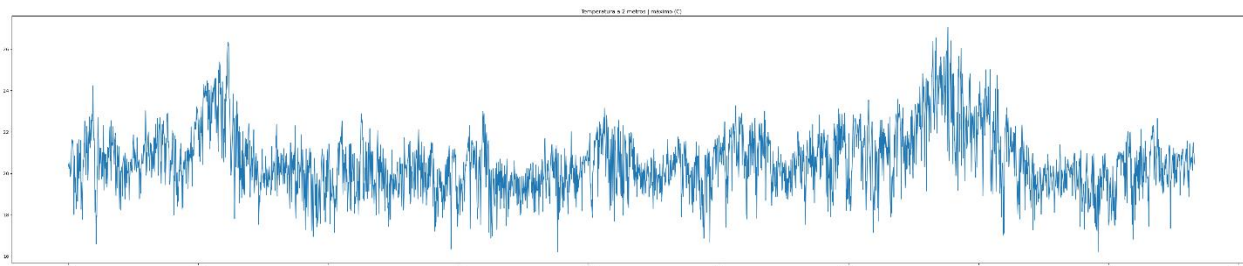
### T2M\_MAX: Temperatura máxima a 2 metros (C)

*Ilustración 20 T2M\_MAX: Temperatura máxima a 2 metros por año*



### T2M\_MAX Gráfica 2015 – 2022

*Ilustración 21 T2M\_MAX: Temperatura máxima a 2 metros*



### T2M\_RANGE: Rango de temperatura a 2 metros (C)

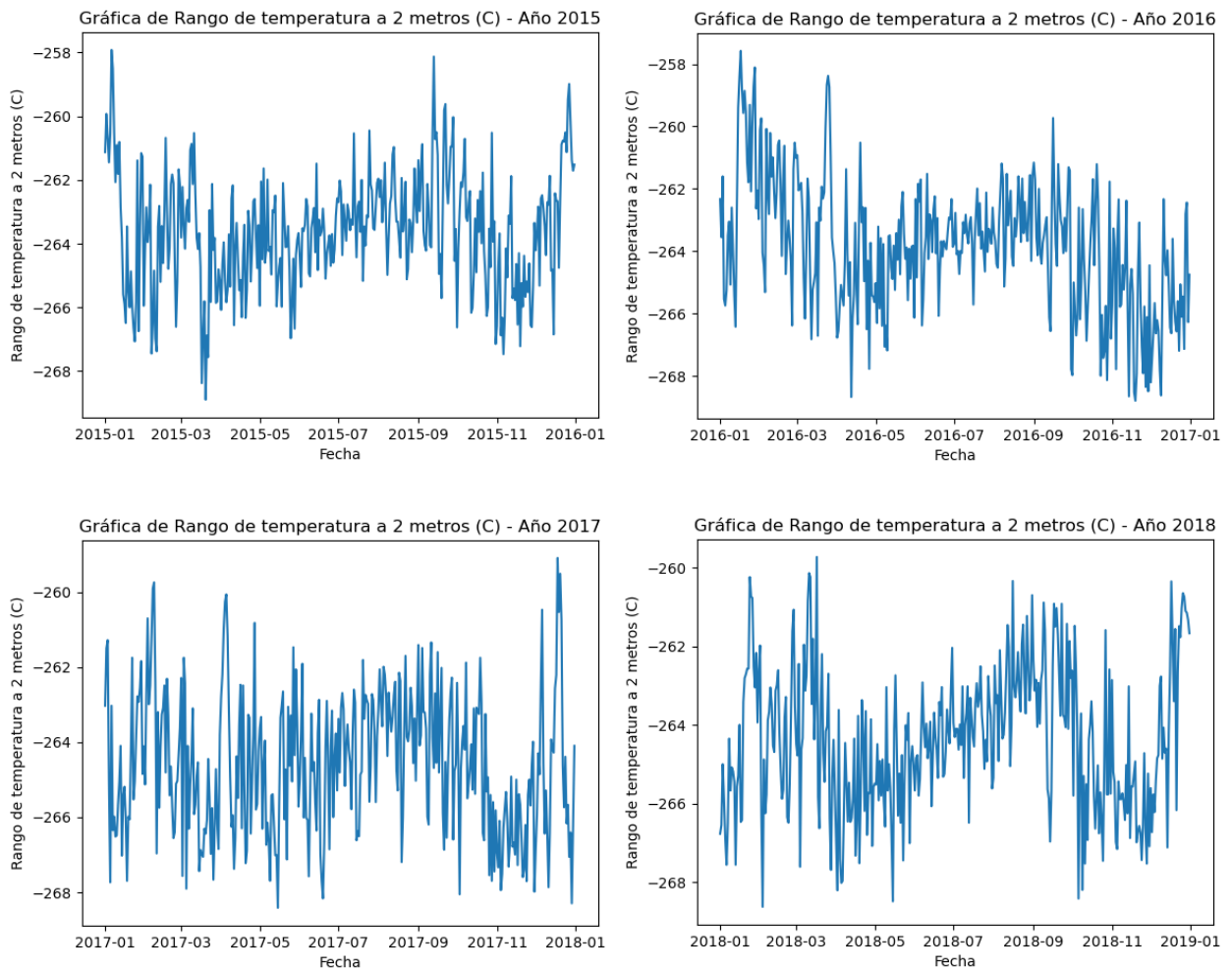
La abreviatura "T2M\_RANGE" se refiere al "Rango de Temperatura a 2 Metros", que representa la diferencia entre las temperaturas máxima y mínima registradas a una altura de 2 metros sobre la superficie terrestre en un período específico, generalmente en un día. Esta diferencia se expresa en grados Celsius (°C).

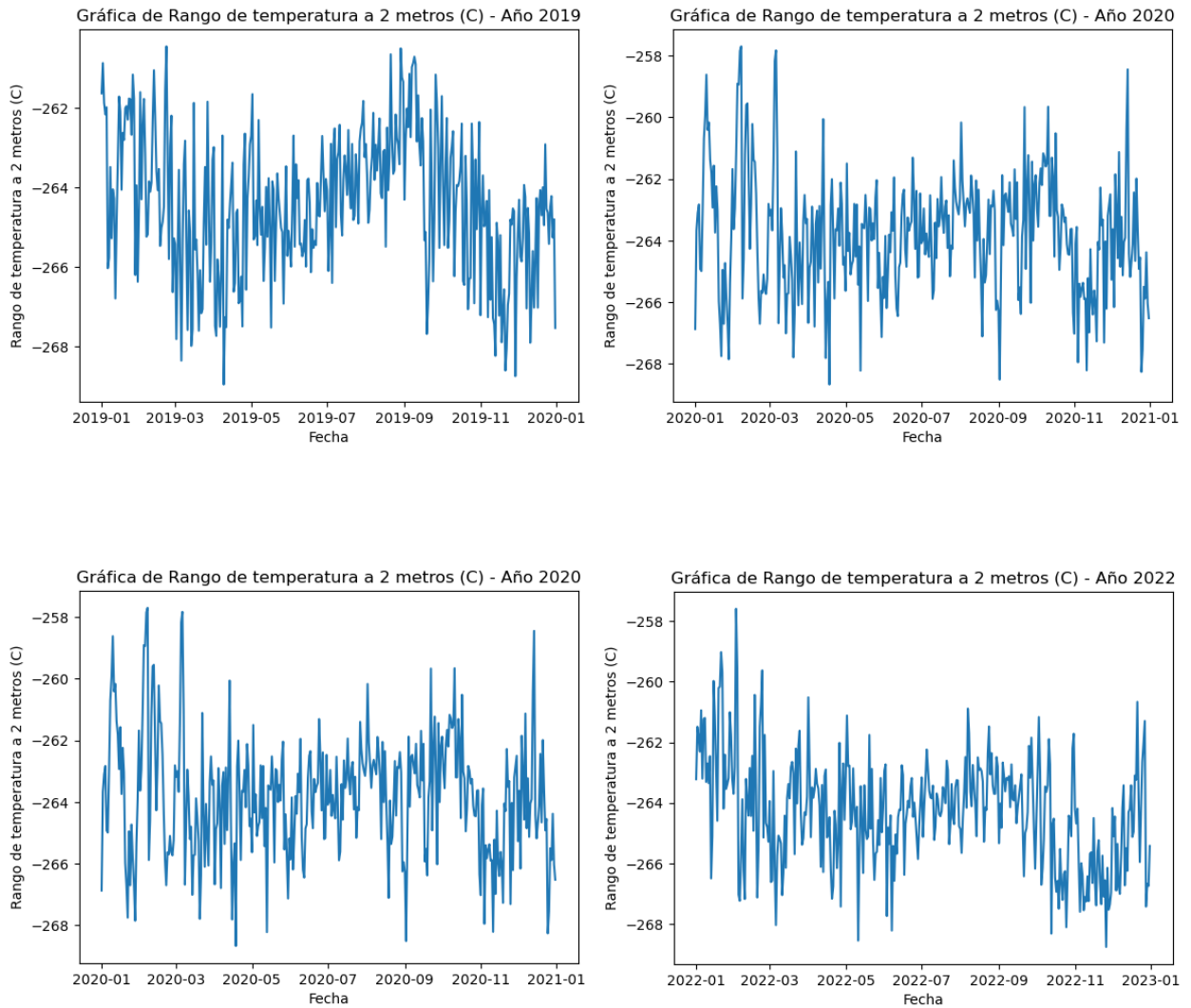
El rango de temperatura a 2 metros proporciona información sobre la variabilidad térmica diaria en una ubicación dada. Puede ser útil para comprender las fluctuaciones de temperatura durante el día y la noche, así como para evaluar la amplitud térmica diaria. El

rango de temperatura puede influir en una serie de procesos climáticos y biológicos, incluyendo la evaporación, la formación de rocío y la adaptación de organismos a cambios de temperatura.

En resumen, "T2M\_RANGE" es una medida que cuantifica la diferencia entre las temperaturas máxima y mínima a 2 metros sobre la superficie terrestre en un período de tiempo específico.

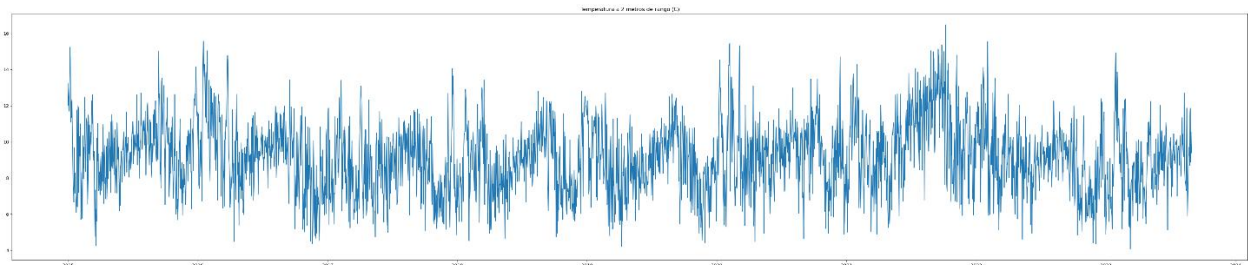
*Ilustración 22T2M\_RANGE: Rango de temperatura a 2 metros por año*





### T2M\_RANGE Gráfica 2015 – 2022

*Ilustración 23 T2M\_RANGE: Rango de temperatura a 2 metros*



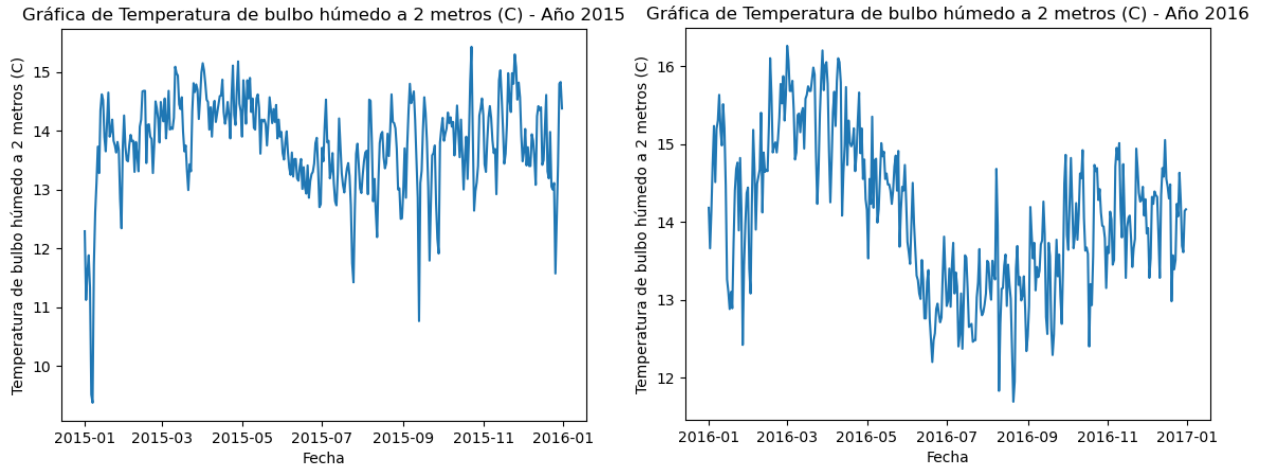
### T2MWET: Temperatura de bulbo húmedo a 2 metros (C)

La abreviatura "T2MWET" se refiere a la "Temperatura de Bulbo Húmedo a 2 Metros", que representa la temperatura a la que un termómetro de bulbo húmedo alcanzaría

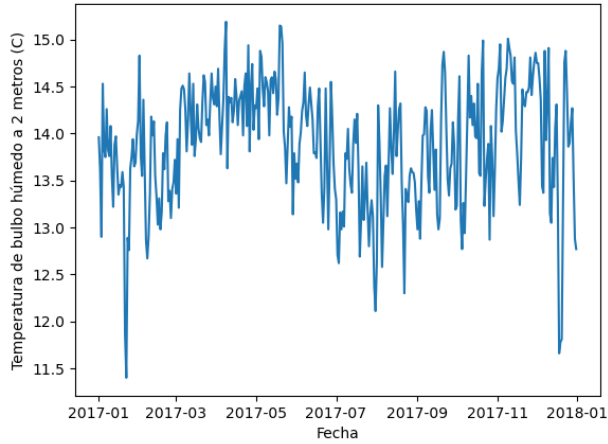
equilibrio térmico con el aire circundante a una altura de 2 metros sobre la superficie terrestre. Esta medida se expresa en grados Celsius (°C).

La temperatura de bulbo húmedo es un indicador de la humedad del aire. Cuando el bulbo húmedo está envuelto en un paño humedecido y expuesto al aire, el agua en el paño se evapora, enfriando el bulbo. La cantidad de enfriamiento depende de la humedad en el aire: cuanto más húmedo esté el aire, menos evaporación ocurrirá y menos se enfriará el bulbo. Por lo tanto, la temperatura de bulbo húmedo es una medida relacionada con la humedad y se utiliza en combinación con la temperatura del aire seco (medida comúnmente) para evaluar las condiciones de confort térmico y para calcular otras variables climáticas, como la temperatura de punto de rocío y la humedad relativa.

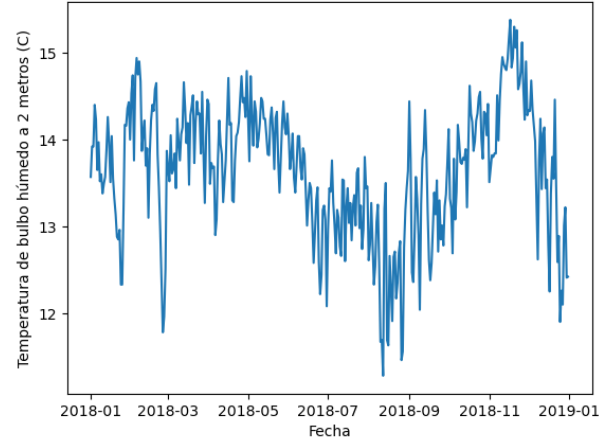
*Ilustración 24 T2MWET: Temperatura de bulbo húmedo a 2 metros por año*



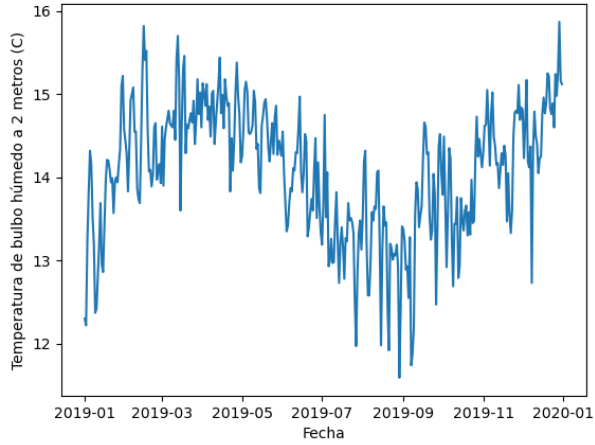
Gráfica de Temperatura de bulbo húmedo a 2 metros (C) - Año 2017



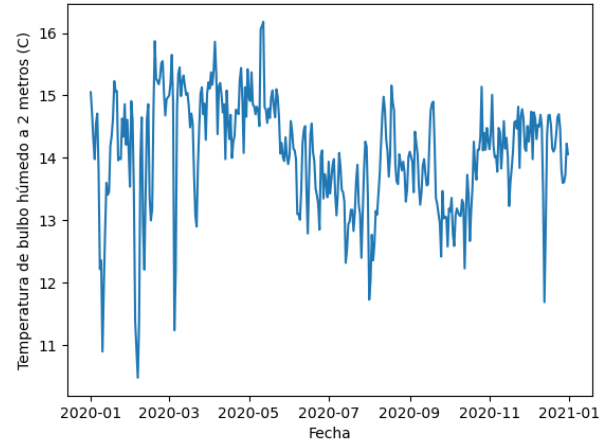
Gráfica de Temperatura de bulbo húmedo a 2 metros (C) - Año 2018



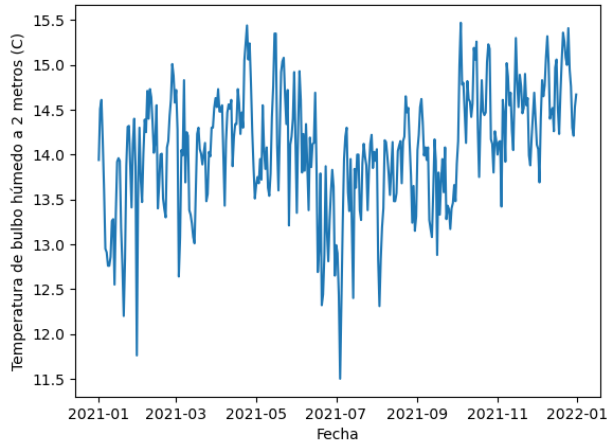
Gráfica de Temperatura de bulbo húmedo a 2 metros (C) - Año 2019



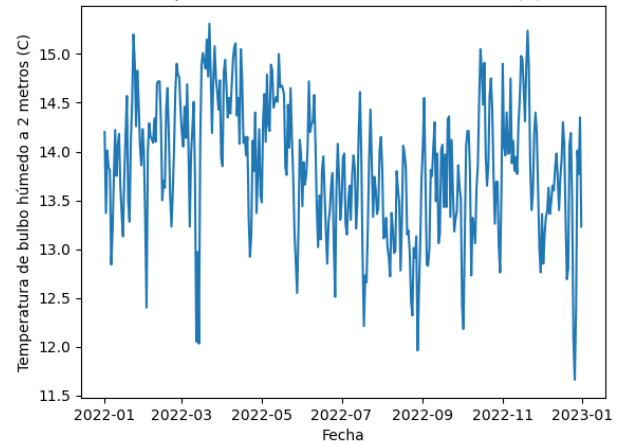
Gráfica de Temperatura de bulbo húmedo a 2 metros (C) - Año 2020



Gráfica de Temperatura de bulbo húmedo a 2 metros (C) - Año 2021

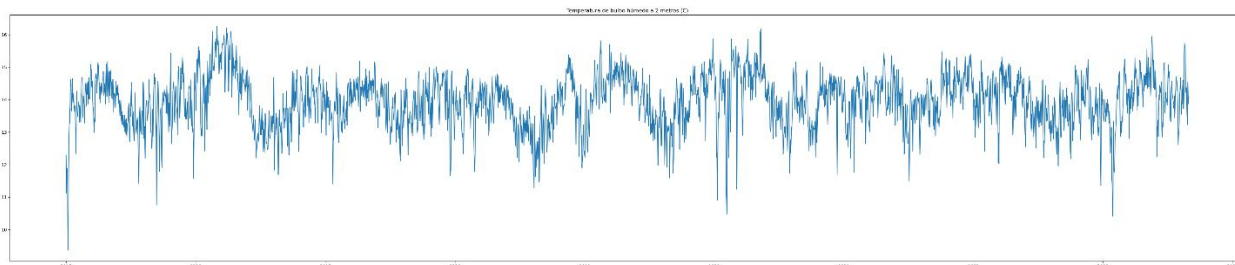


Gráfica de Temperatura de bulbo húmedo a 2 metros (C) - Año 2022



## T2MWET Gráfica 2015 – 2022

*Ilustración 25 T2MWET: Temperatura de bulbo húmedo a 2 metros*



### T2MDEW: Punto de rocío/escarcha a 2 metros (C)

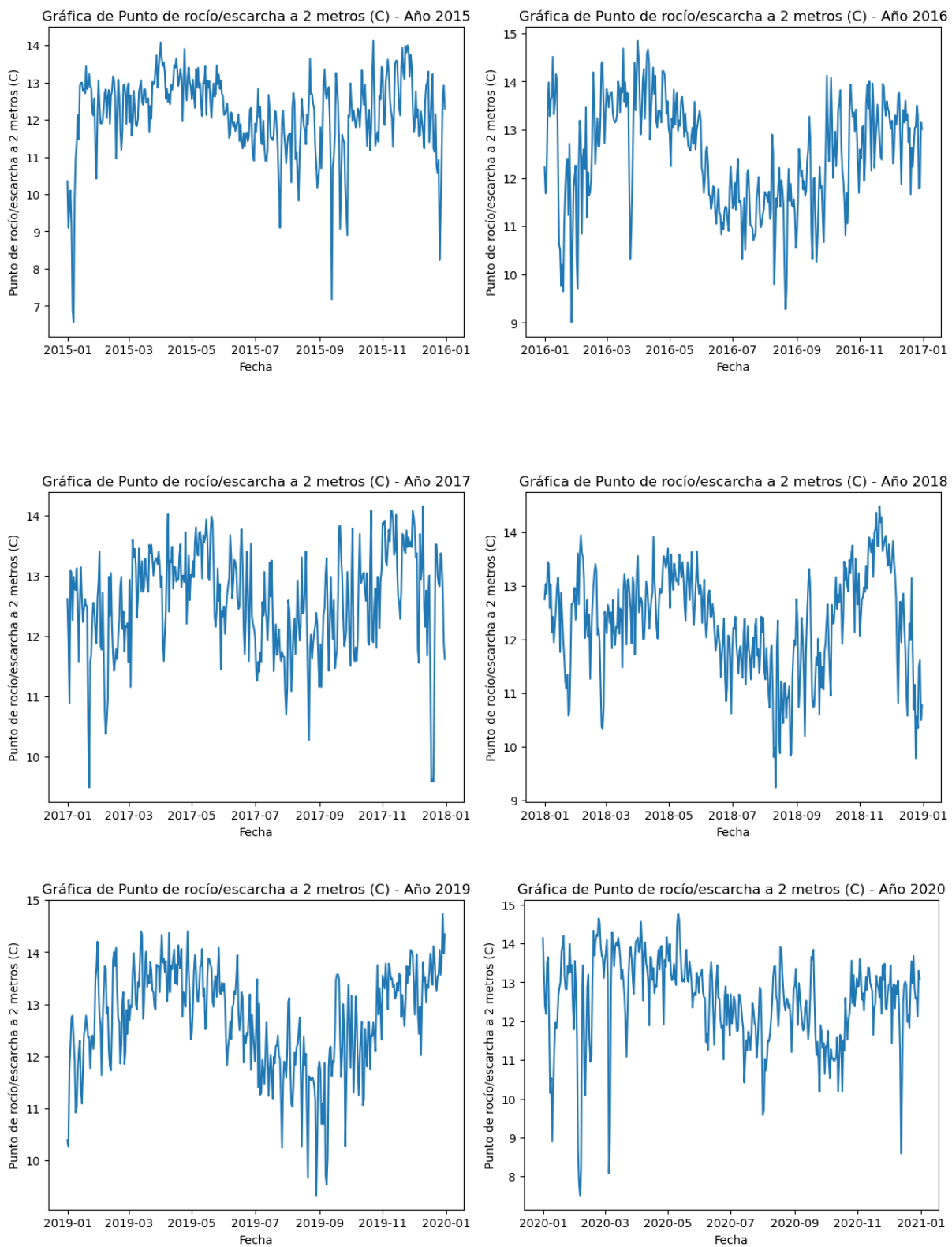
La combinación "T2MDEW" se refiere a dos mediciones meteorológicas distintas:

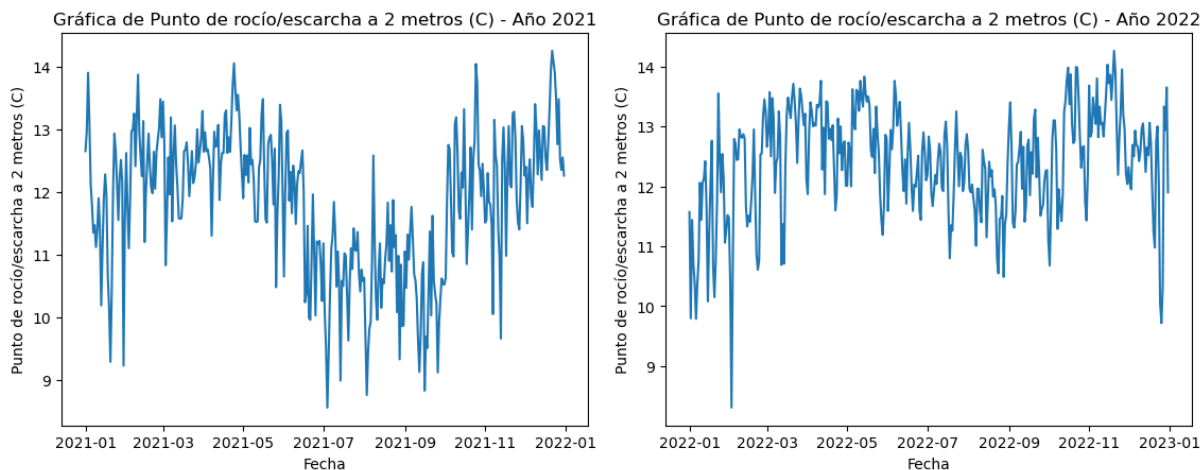
1. "T2M": Esto se refiere a la "Temperatura a 2 Metros", como se explicó en una respuesta anterior. Representa la temperatura del aire medida a una altura de 2 metros sobre la superficie terrestre, y se expresa en grados Celsius (°C).

2. "DEW": Esto se refiere al "Punto de Rocío", que es la temperatura a la cual el aire se satura de vapor de agua, causando que la humedad relativa alcance el 100%. Es la temperatura a la cual el aire debe enfriarse para que se produzca condensación y se forme rocío o escarcha. El punto de rocío se expresa en grados Celsius (°C).

En resumen, "T2MDEW" representa información sobre la temperatura del aire a 2 metros sobre la superficie terrestre y el punto de rocío o escarcha en el mismo nivel. El punto de rocío es útil para comprender cuándo la humedad en el aire se condensará y se formarán fenómenos como el rocío en las superficies o la escarcha en condiciones frías.

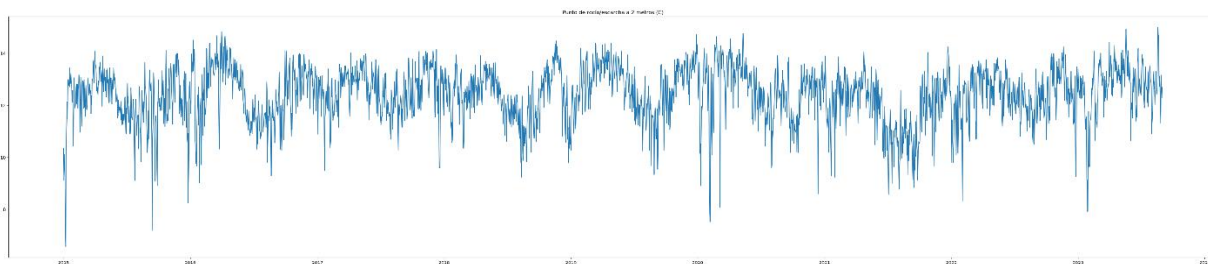
*Ilustración 26 T2MDEW: Punto de rocío/escarcha a 2 metros por año*





### T2MDEW Gráfica 2015 – 2022

*Ilustración 27 T2MDEW: Punto de rocío/escarcha a 2 metros*



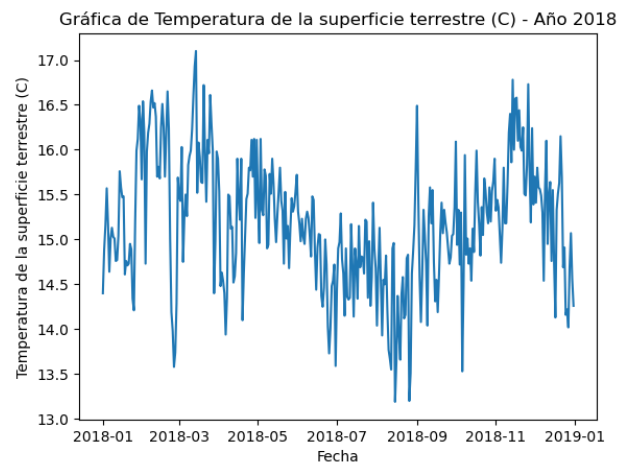
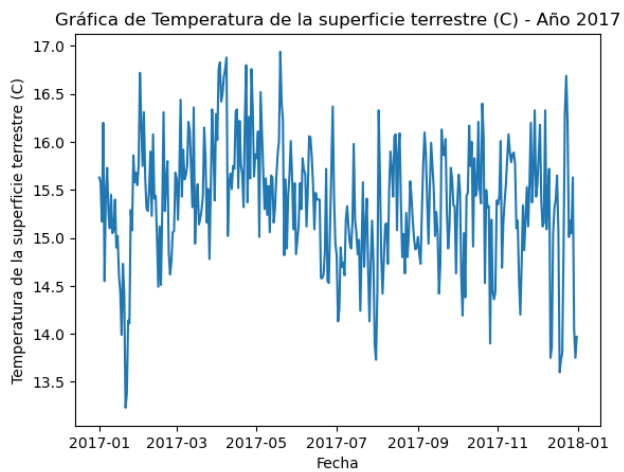
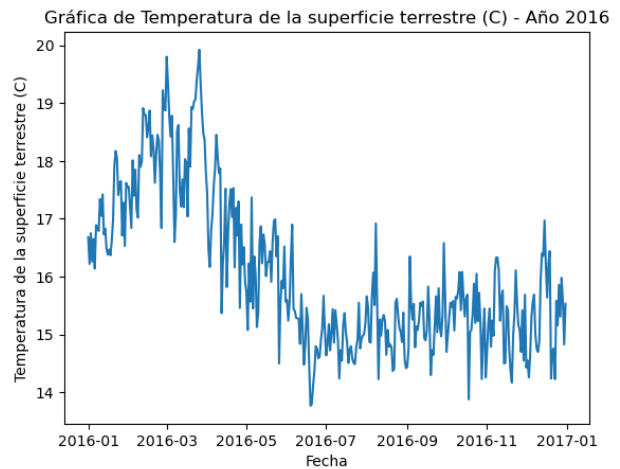
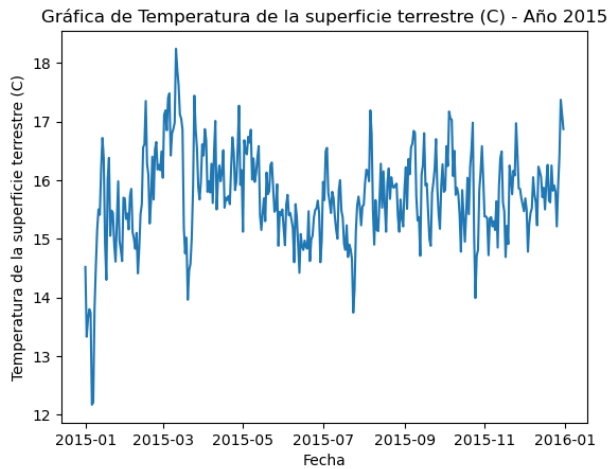
### TS: Temperatura de la superficie terrestre (C)

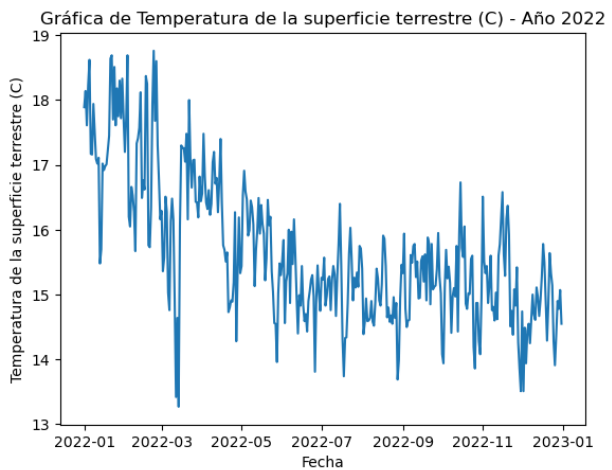
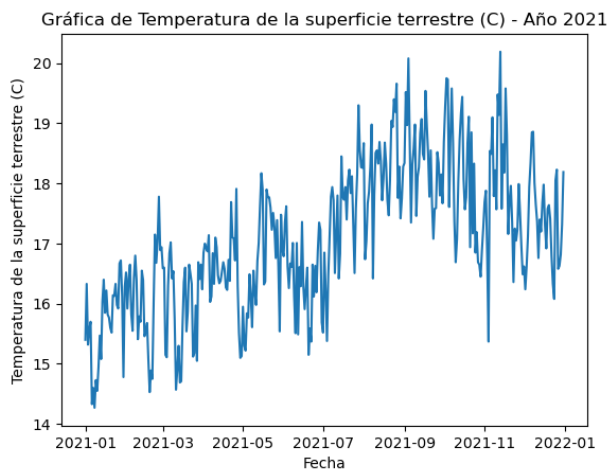
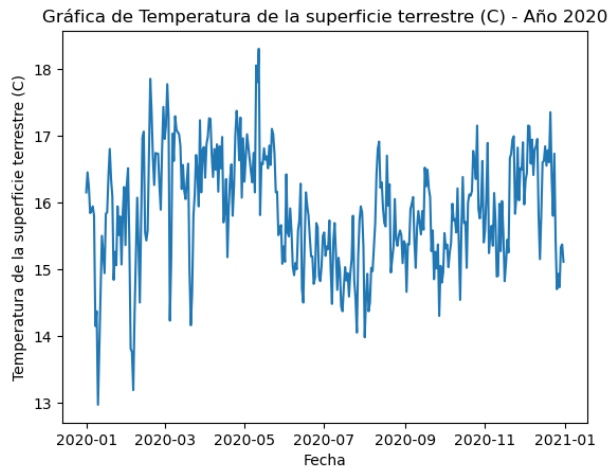
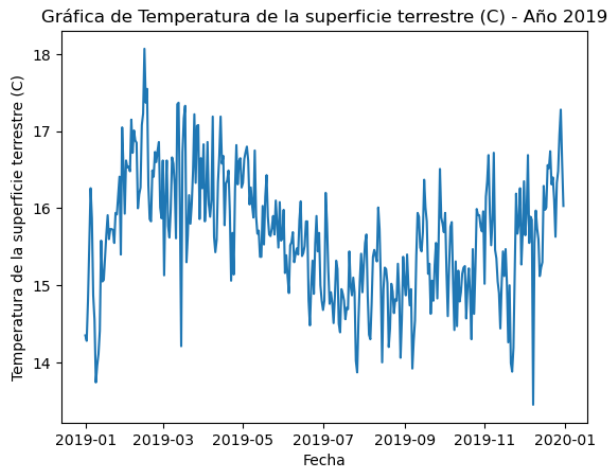
La abreviatura "TS" se refiere a la "Temperatura de la Superficie Terrestre", que representa la temperatura medida en la parte superior de la capa de suelo o tierra, en contacto directo con la atmósfera. Esta medida se expresa en grados Celsius ( $^{\circ}\text{C}$ ) u otras unidades de temperatura.

La temperatura de la superficie terrestre es relevante en meteorología y climatología, ya que afecta la interacción entre la superficie de la Tierra y la atmósfera. Puede influir en la formación de nubes, la radiación térmica y la distribución de calor en la atmósfera y en la superficie. Los datos de temperatura de la superficie terrestre son fundamentales para comprender los patrones climáticos, las variaciones estacionales y los fenómenos

meteorológicos que se originan en la interacción entre la tierra y la atmósfera.

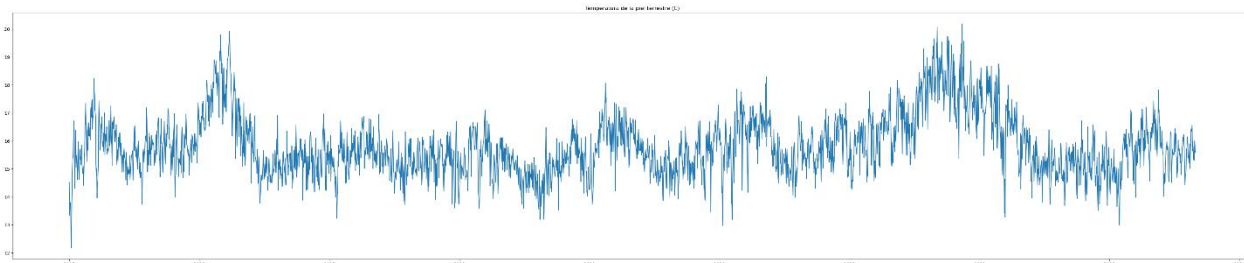
*Ilustración 28 TS: Temperatura de la superficie terrestre por año*





**TS Gráfica 2015 – 2022**

*Ilustración 29 TS: Temperatura de la superficie terrestre*



## **2.4 DISEÑO DE LOS CASOS DE PRUEBA**

### **2.4.1 Prueba de Dickey-Fuller aumentada.**

La Prueba de Dickey-Fuller aumentada o ADF es una prueba estadística utilizada para determinar si una serie de tiempo es estacionaria. Se usa comúnmente en análisis de series de tiempo y pronósticos.

Una serie de tiempo es estacionaria cuando sus propiedades estadísticas como la media, varianza y autocorrelación son constantes en el tiempo. Si una serie no es estacionaria, puede presentar tendencias o estacionalidad que cambien sus características a lo largo del tiempo.

Para hacer pronósticos precisos de una serie de tiempo, generalmente se requiere que la serie sea estacionaria. La prueba ADF se utiliza para determinar si una serie es estacionaria o no.

La prueba ADF se basa en un modelo de regresión donde la variable dependiente es la diferencia primera de la serie de tiempo y la variable independiente es el valor rezagado de la serie. Se prueba la hipótesis nula de que la serie tiene raíz unitaria (no es estacionaria) frente a la hipótesis alternativa de que no tiene raíz unitaria (es estacionaria).

Si la prueba ADF rechaza la hipótesis nula, eso significa que la serie es probablemente estacionaria y se puede usar directamente para pronósticos. Si no se rechaza la hipótesis nula, la serie es no estacionaria y se deben aplicar técnicas como diferenciación para hacerla estacionaria antes de construir modelos de pronóstico.

La prueba ADF es útil porque ayuda a determinar si se debe diferenciar una serie de

tiempo antes de hacer pronósticos con ella. Al hacer que la serie sea estacionaria, se mejora la precisión y fiabilidad de los pronósticos resultantes.

### PRUEBAS INDIVIDUALES:

#### *Ilustración 30 Prueba de Dickey-Fuller aumentada por consola*

En esta figura se observa la visualización del resultado de la prueba desde la consola

```
Prueba_Dickey_Fuller(df['QV2M'], 'QV2M')
```

```
Resultados de la prueba de Dickey-Fuller para columna: QV2M
Test Statistic          -5.584061
p-value                 0.000001
No Lags Used           23.000000
Número de observaciones utilizadas  3141.000000
Critical Value (1%)     -3.432434
Critical Value (5%)     -2.862461
Critical Value (10%)    -2.567260
dtype: float64
1.375707926629695e-06
Conclusion:====>
Rechazar la hipótesis nula
Los datos son estacionarios
```

de Jupyter Notebook, donde se identifican los valores para p-value, dado que el estadístico calculado es menor (más negativo) que los valores críticos al 1%, 5% y 10% ,se rechaza la hipótesis nula de raíz unitaria ( no estacionariedad)

Por lo tanto, los datos de la serie QV2M humedad especifica de a 2 metros, son estacionarios puesto que, se rechazó la hipótesis nula en la prueba al tener un p-valor muy pequeño

Dando explicación a las otras variables se representa la siguiente tabla:

*Tabla 6 Prueba de Dickey-Fuller aumentada*

Nombre de variable	p-value	Crítica Value 1%	Crítica Value 5%	Crítica Value 10%
<b>PS- presión de superficie</b>	1.254587e-12	-3.432427e+00	-2.862458e+00	-2.567259e+00
Conclusión: Rechazar la hipótesis nula. Los datos son estacionarios.				

<b>WS10M- velocidad del viento a 10 metros</b>	2.386624e-13	-3.432427e+00	-2.862458e+00	-2.567259e+00
Conclusión: Rechazar la hipótesis nula. Los datos son estacionarios.				
<b>WS10M_MAX- velocidad del viento a 10 metros</b>	1.017433e-17	-3.432427e+00	-2.862458e+00	-2.567259e+00
Conclusión: Rechazar la hipótesis nula. Los datos son estacionarios.				
<b>WS10M_MIN- velocidad del viento a 10 metros</b>	4.099115e-13	-3.432427e+00	-2.862458e+00	-2.567259e+00
Conclusión: Rechazar la hipótesis nula. Los datos son estacionarios.				
<b>PRECTOTCORR- precipitación corregida</b>	4.195917e-10	4.195917e-10	-2.862460e+00	-2.567260e+00
Conclusión: Rechazar la hipótesis nula. Los datos son estacionarios.				
<b>RH2M- unidad relativa de a dos metros</b>	0.000645	-3.432433	-2.862460	-2.567260
Conclusión: Rechazar la hipótesis nula. Los datos son estacionarios.				
<b>T2M- temperatura a 2 metros</b>	0.000123	-3.432432	-2.862460	-2.567260
Conclusión: Rechazar la hipótesis nula. Los datos son estacionarios.				
<b>T2M_MIN- temperatura a 2 metros</b>	9.132204e-07	-3.432436e+00	-2.862461e+00	-2.567261e+00
Conclusión: Rechazar la hipótesis nula. Los datos son estacionarios.				
<b>T2M_MAX- temperatura a 2 metros</b>	0.000139	-3.432433	-2.862460	-2.567260
Conclusión: Rechazar la hipótesis nula. Los datos son estacionarios.				

<b>T2M_RANGE- rango de temperatura a dos metros</b>	2.939031e-09	-3.432435e+00	-2.862461e+00	-2.567260e+00
Conclusión: Rechazar la hipótesis nula. Los datos son estacionarios.				
<b>T2MWET- temperatura de bulbo húmedo 2 metros</b>	8.077761e-07	-3.432434e+00	-2.862461e+00	-2.567260e+00
Conclusión: Rechazar la hipótesis nula. Los datos son estacionarios.				
<b>T2MDEW- punto de rocío/escarcha 2 metros</b>	8.281867e-07	-3.432434e+00	-2.862461e+00	-2.567260e+00
Conclusión: Rechazar la hipótesis nula. Los datos son estacionarios				
<b>TS- temperatura de la superficie terrestre</b>	0.003861	-3.432434	-2.862461	-2.567260
Conclusión: Rechazar la hipótesis nula. Los datos son estacionarios.				

En los resultados obtenidos por la prueba ADF observamos que Cálculo del p-valor, es la probabilidad de observar un valor igual o más extremo que el estadístico de prueba, asumiendo que la hipótesis nula es verdadera.

En la Interpretación Si el p-valor es muy pequeño (generalmente menor que un nivel de significancia predefinido, como 0.05), entonces tienes evidencia suficiente para rechazar la hipótesis nula. En este caso, se concluye que la serie de tiempo es estacionaria.

Si el p-valor es mayor que el nivel de significancia, no se tiene evidencia suficiente para rechazar la hipótesis nula, lo que sugiere que la serie de tiempo no es estacionaria.

#### **2.4.2 Prueba de entrenamiento**

En este proyecto se desarrolló un modelo de redes neuronales para predecir variables

relacionadas con el clima, como lo son la humedad, precipitación, temperatura y velocidad del viento. El modelo fue entrenado con conjuntos de datos históricos para aprender las complejas interrelaciones de estas variables.

Para evaluar el desempeño del modelo durante el entrenamiento, se utilizó la métrica del Error Cuadrático Medio (MSE), el cual mide la diferencia promedio entre los valores predichos por el modelo y los valores reales observados, teniendo en cuenta que un MSE más bajo indica un mejor ajuste del modelo a los datos.

En el siguiente resumen se presentan los valores de MSE obtenidos después de entrenar el modelo en diversas cantidades de entrenamientos. En general, se observa una disminución del MSE a medida que aumenta el número de rangos de tiempo para el entrenamiento.

Estos resultados proporcionan información valiosa sobre la capacidad del modelo para aprender las relaciones entre datos. El decrecimiento constante del error MSE es un indicador positivo puesto que el modelo está mejorando su capacidad de generalización y precisión con más entrenamientos.

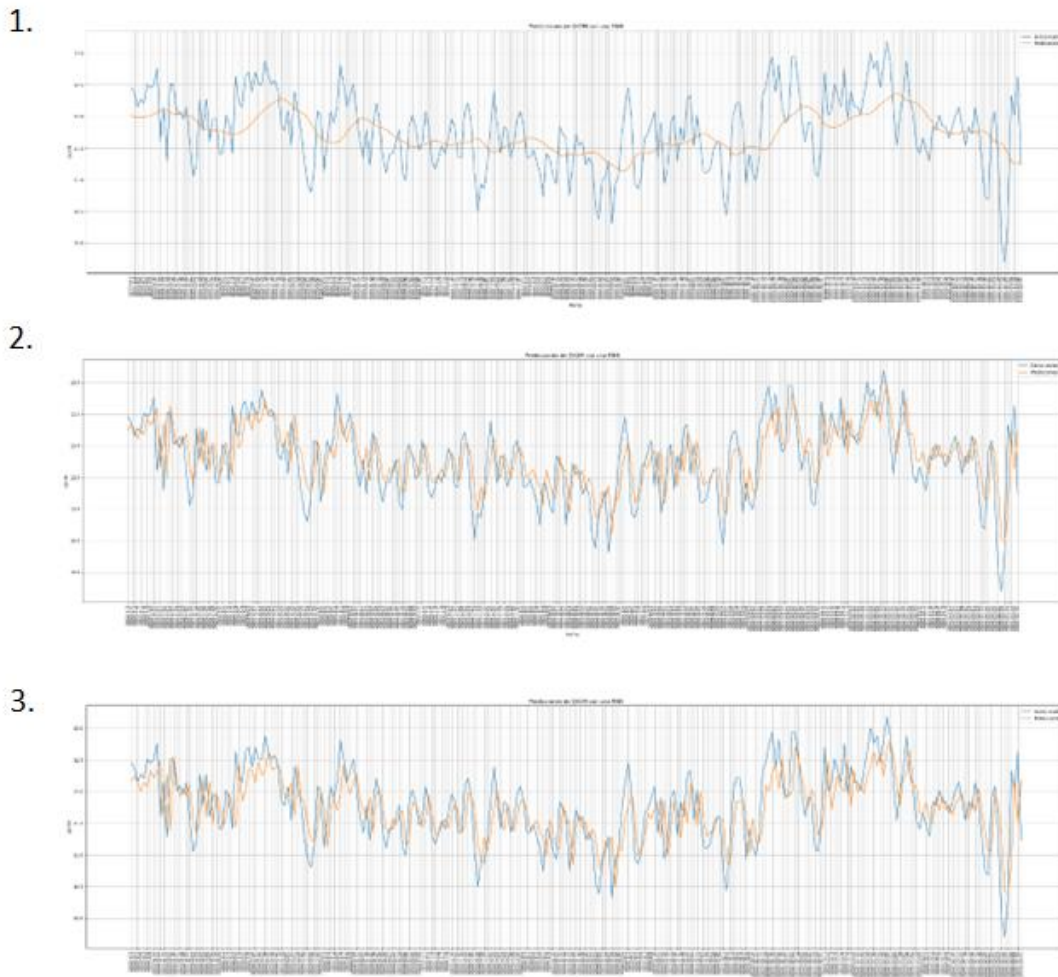
El MSE por variable es útil para evaluar qué tan puntual son las predicciones del modelo en diferentes parámetros meteorológicos. Esta información puede guiar esfuerzos futuros para mejorar el desempeño del modelo en variables específicas. En conjunto, estos resultados demuestran el progreso en el desarrollo de un modelo predictivo meteorológico basado en redes neuronales.

Con relaciona a las siguientes graficas ordenadas ascendentemente por épocas de

entrenamiento, se puede observar el comportamiento del aprendizaje del modelo, donde se representa de color **naranja** los resultados de la predicción y con un color **azul** los valores reales de acuerdo a los datos por variable.

QV2M: Humedad específica de a 2 metros (g/kg)

*Ilustración 31 QV2M: Humedad específica de a 2 metros Entrenamientos*



La humedad específica a 2 metros de altura (QV2M) refleja la cantidad de vapor de agua presente en el aire. Para esta variable, el error cuadrático medio (MSE) del modelo disminuyó constantemente con más épocas de entrenamiento:

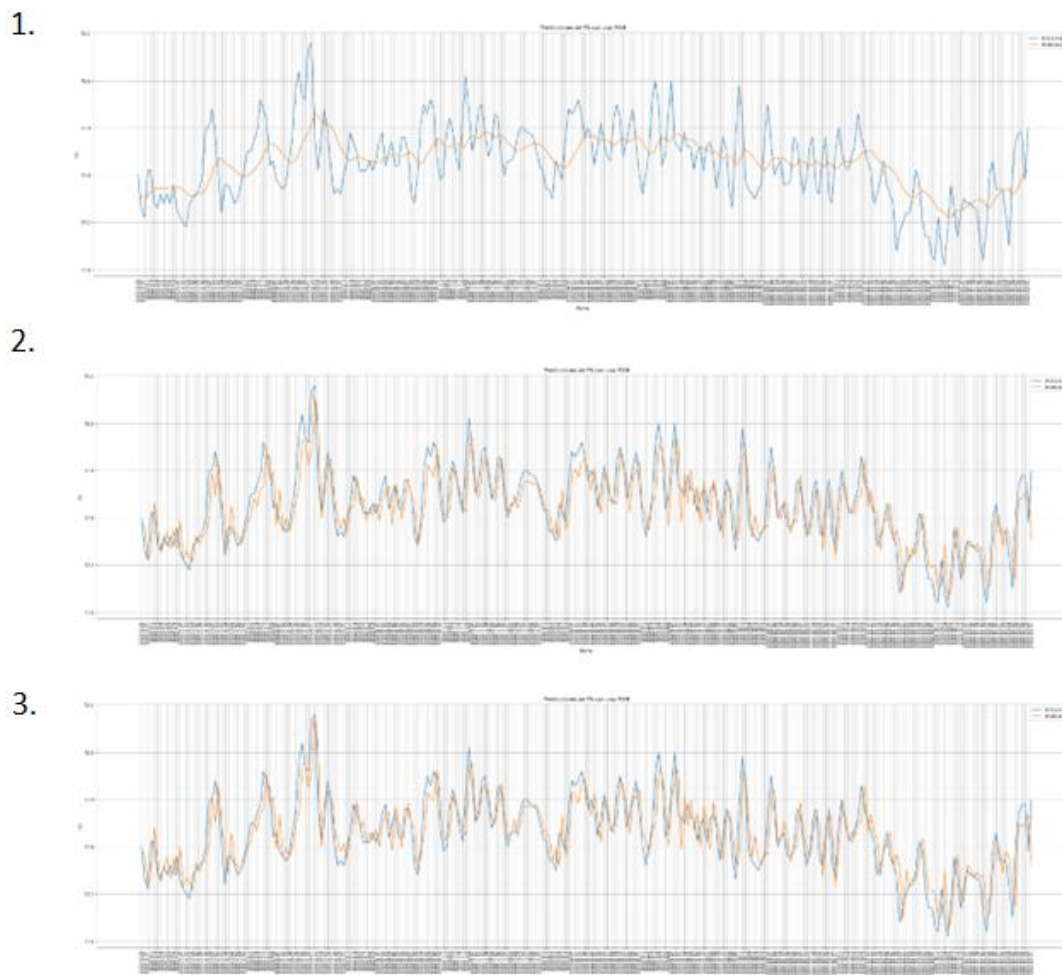
Con sólo 1 a 10 épocas, como se observa en la gráfica 1 el MSE fue relativamente alto, de 0.29866834732721675. Esto indica que el modelo no lograba capturar adecuadamente el comportamiento de la humedad específica.

De 10 a 25 épocas, el MSE de la gráfica 2, se redujo a 0.19454286163705772, demostrando una mejora en el aprendizaje del modelo.

Finalmente, con 35 épocas de entrenamiento como se observa en la parte inferior, se alcanzó un MSE de 0.1937618801144945. Este valor más bajo muestra que el modelo pudo aprender mejor los patrones de la humedad específica del aire con más entrenamiento.

PS: Presión de superficie de (kPa)

*Ilustración 32 PS: Presión de superficie de Entrenamientos*



La presión de superficie (PS) es un importante parámetro meteorológico que indica la fuerza que ejerce el aire en la superficie terrestre. Para esta variable, la precisión del modelo mejoró sustancialmente con más entrenamiento:

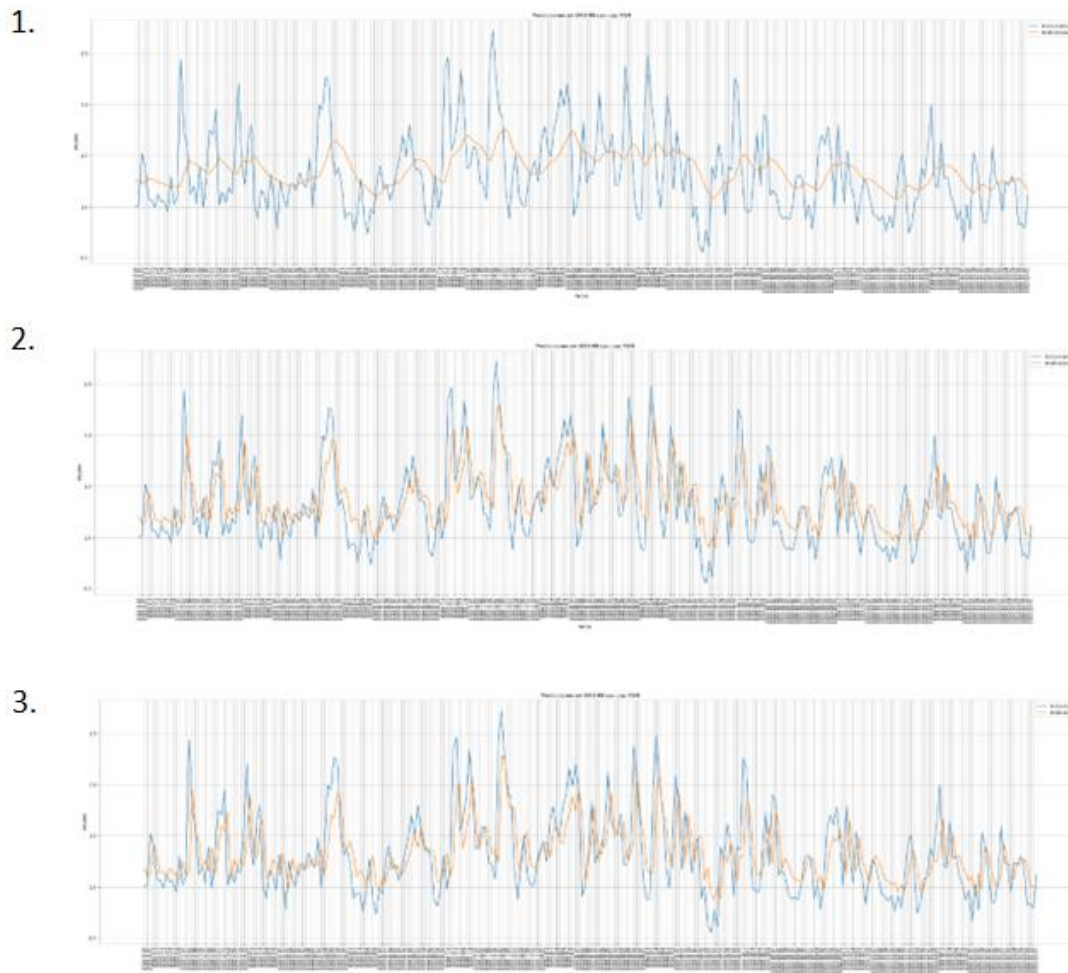
Grafica 1: Con sólo 1 a 10 épocas, el error cuadrático medio (MSE) era relativamente alto, de 0.004279807671902207. Esto revela un ajuste deficiente del modelo a los datos de presión superficial inicialmente.

Grafica 2: De 10 a 25 épocas, el MSE descendió a 0.002146847287180512, evidenciando una mejora en la capacidad predictiva del modelo para esta variable.

Finalmente, con 35 épocas de entrenamiento el MSE alcanzó 0.0020002044024518485. Este valor más bajo demuestra un mejor aprendizaje y generalización en la predicción de la presión de superficie por parte del modelo.

WS10M: Velocidad del viento a 10 metros (m/s)

*Ilustración 33WS10M: Velocidad del viento a 10 metros Entrenamientos*



La velocidad del viento a 10 metros de altura (WS10M) es un parámetro relevante para comprender las condiciones atmosféricas. Para esta variable, la precisión del modelo presentó la siguiente evolución

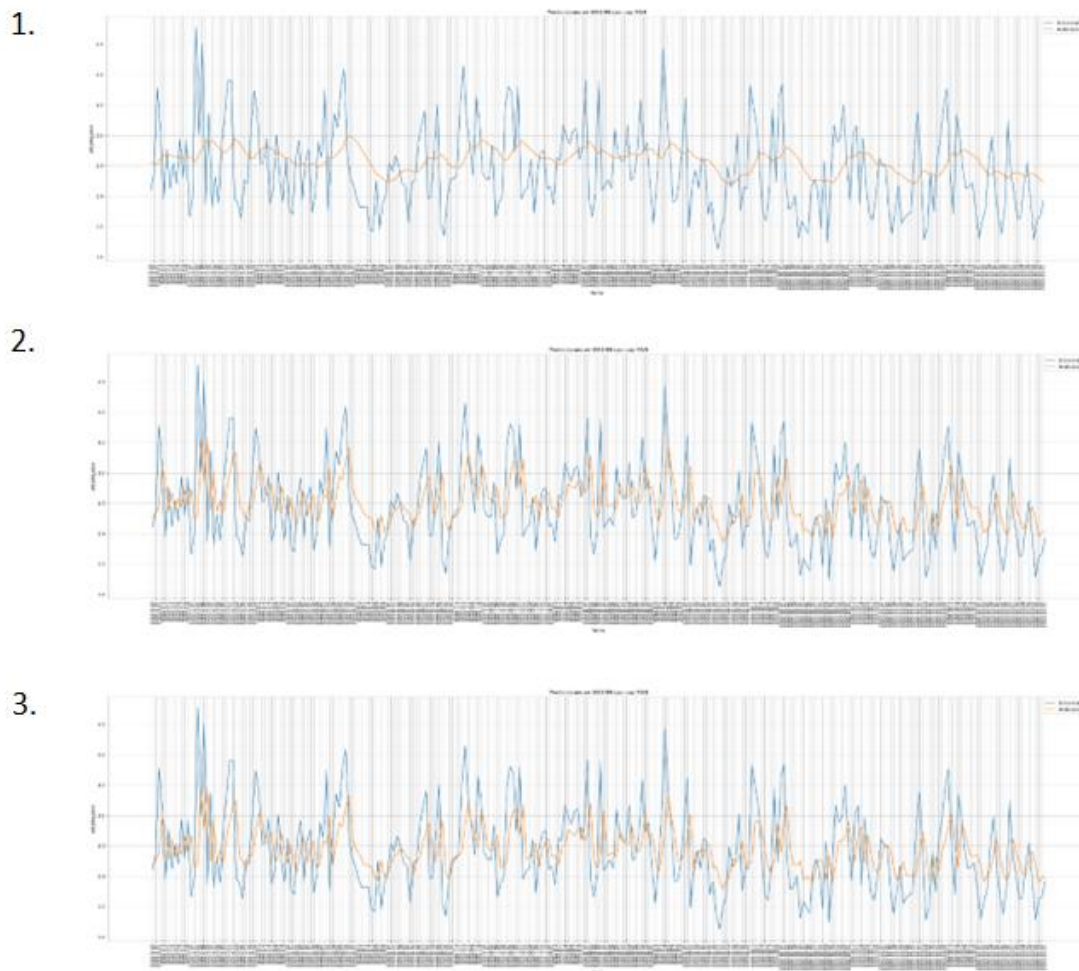
Grafica 1: Con sólo 1 a 10 épocas de entrenamiento, el error cuadrático medio (MSE) era de 0.15130821241171025, reflejando un ajuste inicialmente pobre a los datos de velocidad del viento.

Grafica 2: De 10 a 25 épocas, el MSE disminuyó a 0.109050611766311, mostrando una mejora en el aprendizaje del modelo sobre los patrones de viento.

Finalmente, con 35 épocas se obtuvo un MSE de 0.10607746287298839. Este valor más bajo indica una mayor capacidad del modelo para predecir los valores de velocidad de viento a 10 metros.

WS10M\_MAX: Velocidad máxima del viento a 10 metros (m/s)

*Ilustración 34 WS10M\_MAX: Velocidad máxima del viento a 10 metros Entrenamientos*



La velocidad máxima del viento a 10 metros de altura (WS10M\_MAX) proporciona información sobre los picos de viento en dicho nivel. Para esta variable, la precisión del modelo evolucionó de la siguiente manera:

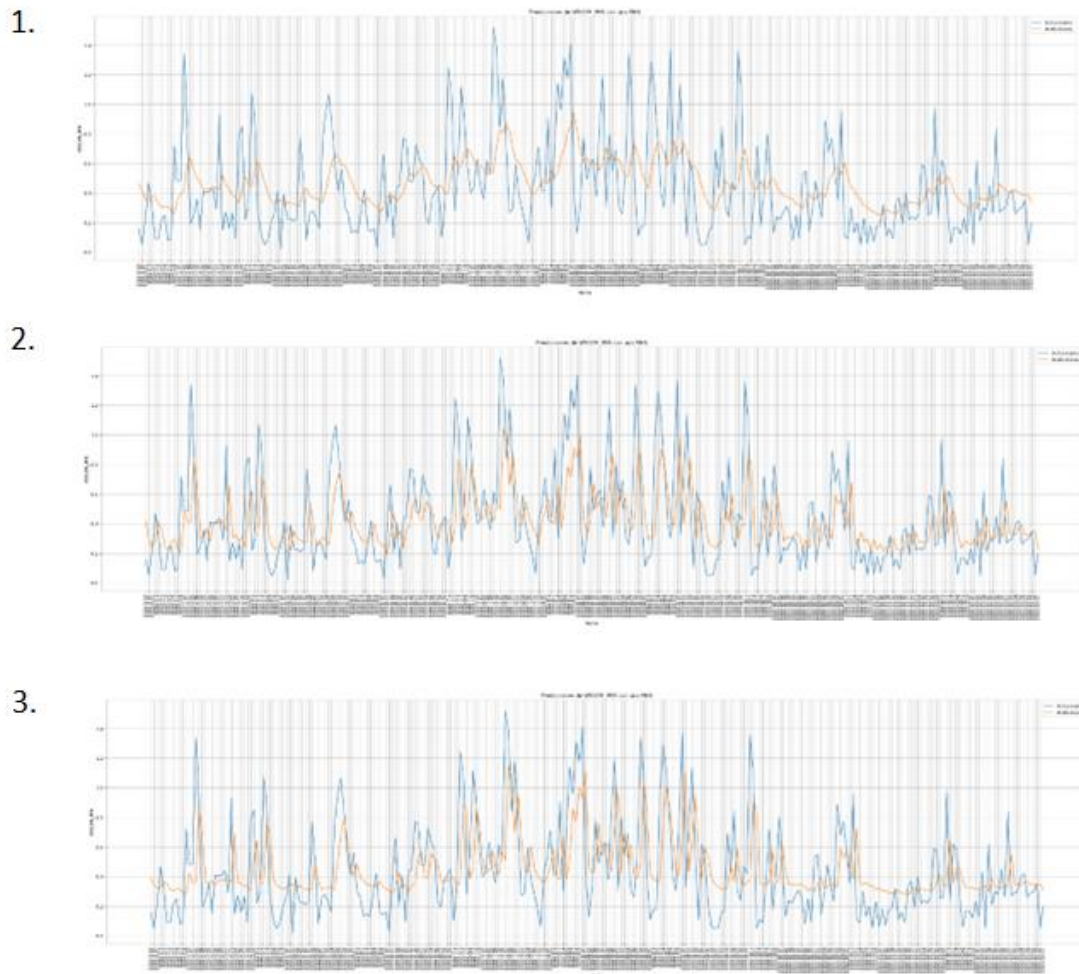
Grafica 1: Con sólo 1 a 10 épocas de entrenamiento, el error cuadrático medio (MSE) era relativamente alto, de 0.5178954775871586, denotando un ajuste pobre a los datos inicialmente.

Grafica 2: De 10 a 25 épocas, el MSE disminuyó a 0.44292187403657535, exhibiendo una mejora en la habilidad del modelo para predecir los valores pico de velocidad de viento.

Con 35 épocas, se alcanzó un MSE de 0.4321946442915721. Este valor más bajo refleja una mayor capacidad del modelo para predecir los máximos de velocidad de viento a 10 metros.

WS10M\_MIN: Velocidad mínima del viento a 10 metros (m/s)

*Ilustración 35 WS10M\_MIN: Velocidad mínima del viento a 10 metros Entrenamientos*



La velocidad mínima del viento a 10 metros (WS10M\_MIN) provee información sobre los valores piso de la velocidad del viento. Para esta variable, la precisión del modelo evolucionó así:

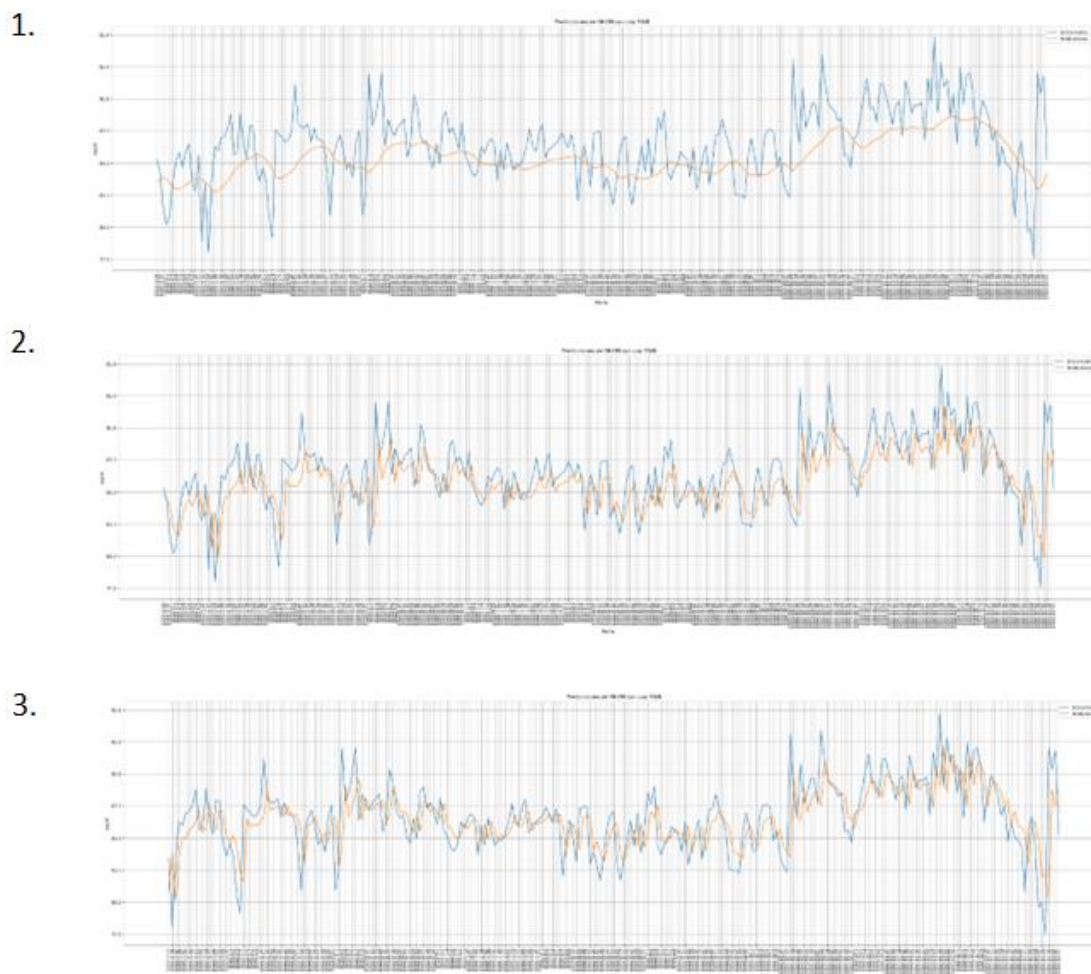
Grafica 1: Con sólo 1 a 10 épocas de entrenamiento, el error cuadrático medio (MSE) era de 0.09435720023201825, denotando un ajuste inicial pobre a los datos.

Grafica 2: De 10 a 25 épocas, el MSE descendió a 0.08529187296977825, mostrando una mejora en la habilidad del modelo para predecir los valores mínimos.

Con 35 épocas se alcanzó un MSE de 0.0822235215397374. Este valor más bajo refleja una mayor capacidad del modelo para predecir las velocidades de viento mínimas a 10 metros.

RH2M: Humedad relativa de a 2 metros (%)

Ilustración 36 RH2M: Humedad relativa de a 2 metros Entrenamientos



La humedad relativa a 2 metros de altura (RH2M) es una medida importante para evaluar la cantidad de humedad presente en el aire. Para esta variable, la precisión del modelo evolucionó de la siguiente forma:

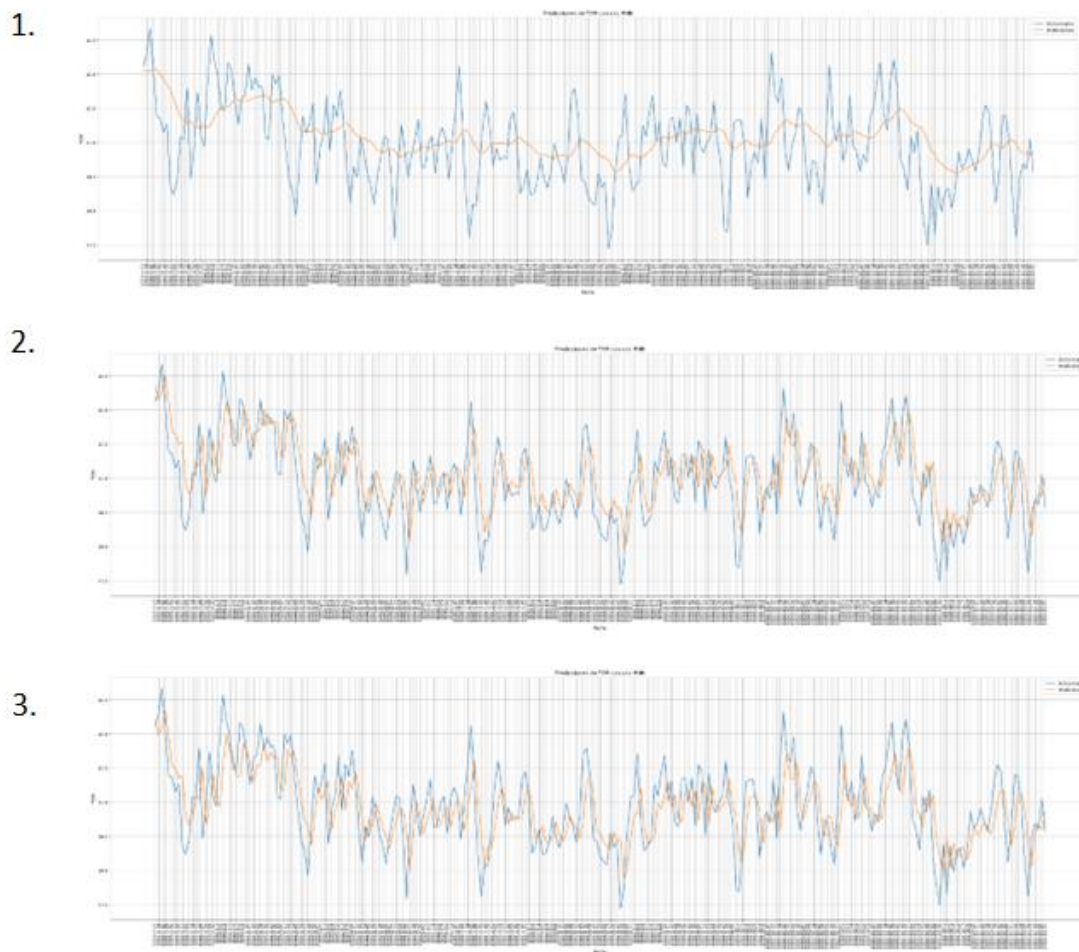
Grafica 1: Con sólo 1 a 10 épocas de entrenamiento, el error cuadrático medio (MSE) era alto, de 7.267272987580398, denotando un ajuste inicial pobre del modelo.

Grafica 2: De 10 a 25 épocas, el MSE descendió a 5.558293487576882, mostrando cierta mejoría en la capacidad de predicción de la humedad relativa.

Con 35 épocas se alcanzó un MSE de 5.133188245330736. Si bien este valor aún es relativamente alto, exhibe una mejora en el aprendizaje del modelo sobre los niveles de humedad relativa del aire.

T2M: Temperatura a 2 metros (C)

*Ilustración 37 T2M: Temperatura a 2 metros Entrenamientos*



La temperatura del aire a 2 metros de altura (T2M) es un parámetro meteorológico clave. Para esta variable, la precisión del modelo presentó la siguiente evolución:

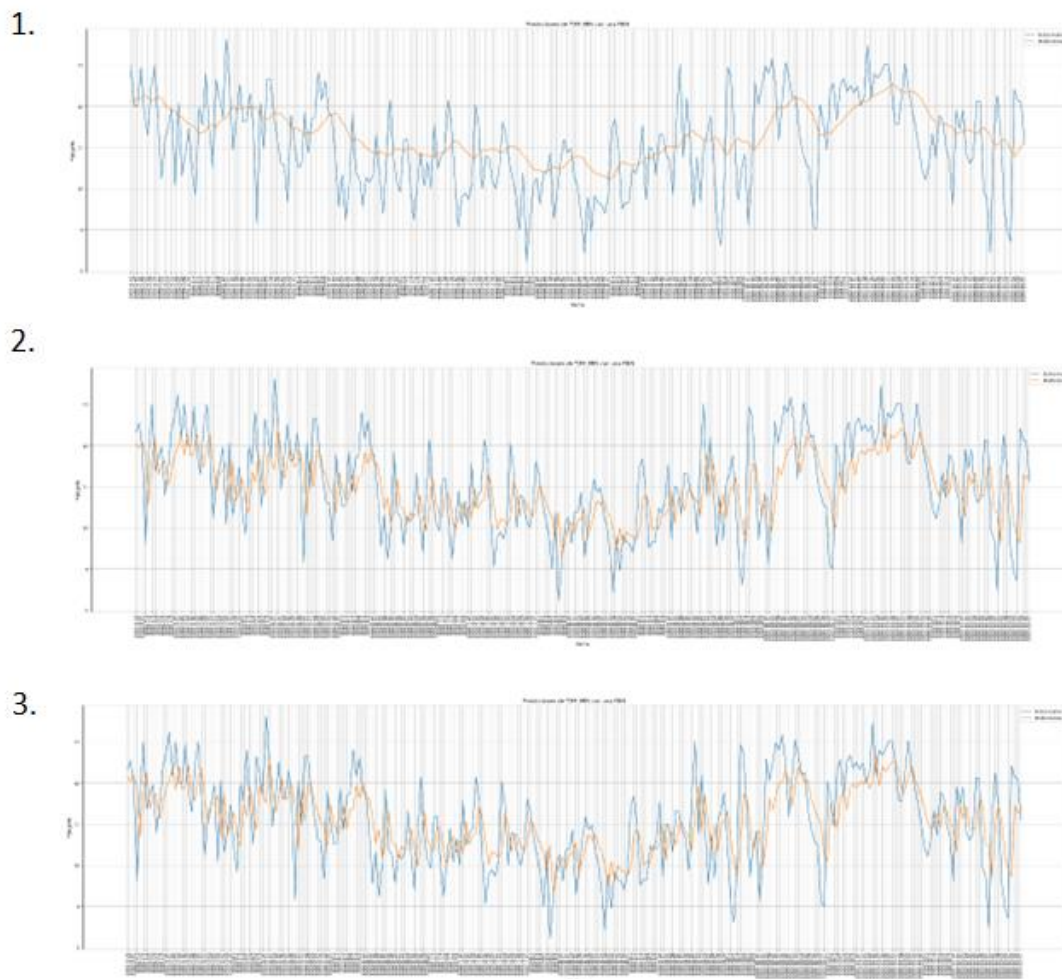
Grafica 1: Con sólo 1 a 10 épocas de entrenamiento, el error cuadrático medio (MSE) era de 0.33365955245177575, indicando un ajuste inicialmente deficiente.

Grafica 2: De 10 a 25 épocas, el MSE disminuyó a 0.22968930495760145, mostrando una mejora en la capacidad del modelo para predecir la temperatura ambiente.

Con 35 épocas se obtuvo un MSE de 0.2212387161665088. Este valor más bajo refleja un mejor aprendizaje y generalización del modelo en la predicción de la temperatura a 2 metros.

T2M\_MIN: Temperatura mínima a 2 metros (C)

*Ilustración 38 T2M\_MIN: Temperatura mínima a 2 metros Entrenamientos*



La temperatura mínima del aire a 2 metros de altura (T2M\_MIN) proporciona información sobre los valores piso de temperatura. Para esta variable, la precisión del modelo evolucionó de la siguiente manera:

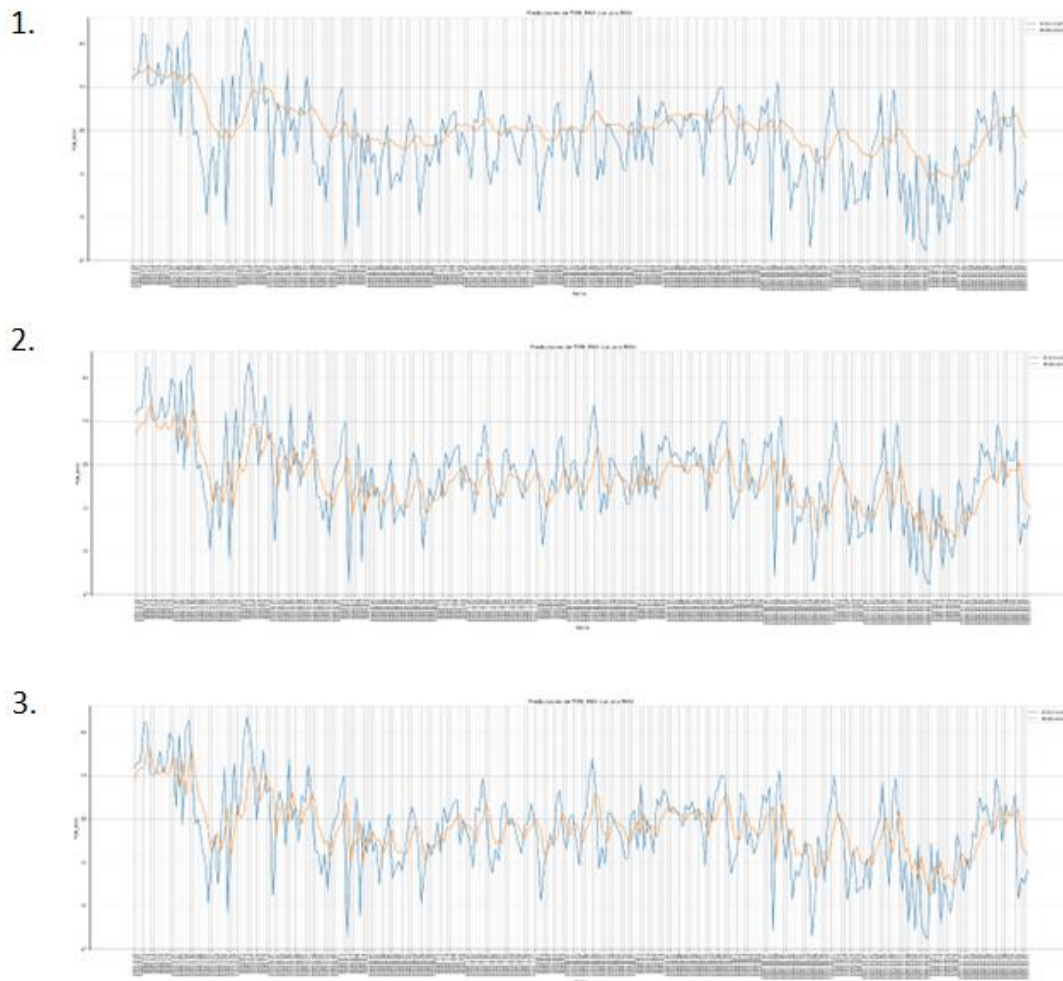
Grafica 1: Con sólo 1 a 10 épocas de entrenamiento, el error cuadrático medio (MSE) era alto, de 1.0648731890247458, denotando un ajuste inicial deficiente.

Grafica 2: De 10 a 25 épocas, el MSE disminuyó a 0.8423815920728687, mostrando una mejora en la capacidad del modelo para predecir las temperaturas mínimas.

Con 35 épocas se alcanzó un MSE de 0.825239313878881. Este valor más bajo refleja un mejor aprendizaje del modelo en la predicción de las temperaturas mínimas a 2 metros.

T2M\_MAX: Temperatura máxima a 2 metros (C)

*Ilustración 39 T2M\_MAX: Temperatura máxima a 2 metros Entrenamientos*



La temperatura máxima del aire a 2 metros de altura (T2M\_MAX) proporciona información sobre los valores pico de temperatura. Para esta variable, la precisión del modelo evolucionó así:

Grafica 1: Con sólo 1 a 10 épocas de entrenamiento, el error cuadrático medio (MSE) era de 0.8286425789853615, indicando un ajuste inicialmente pobre.

Grafica 2: De 10 a 25 épocas, el MSE descendió a 0.6885746226084809, mostrando una mejora en la habilidad del modelo para predecir las temperaturas máximas.

Con 35 épocas se identificó un MSE de 0.6768858560173326. Este valor más bajo refleja una mayor capacidad del modelo en la predicción de las temperaturas máximas a 2 metros.

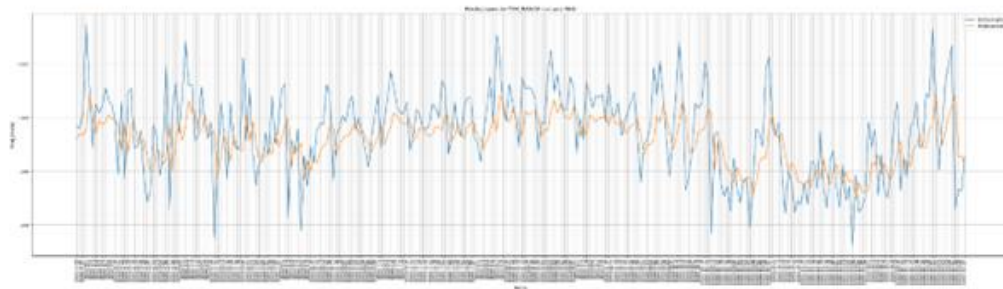
T2M\_RANGE: Rango de temperatura a 2 metros (C)

*Ilustración 40 T2M\_RANGE: Rango de temperatura a 2 metros Entrenamientos*

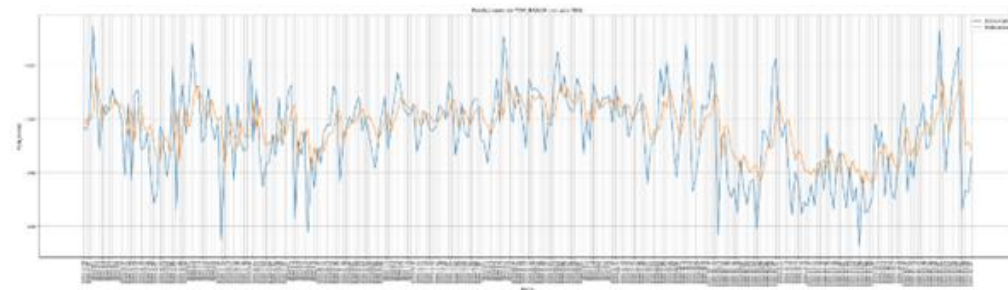
1.



2.



3.



El rango de temperatura del aire a 2 metros (T2M\_RANGE) indica la variación entre los valores máximos y mínimos. Para esta variable, la precisión del modelo evolucionó así:

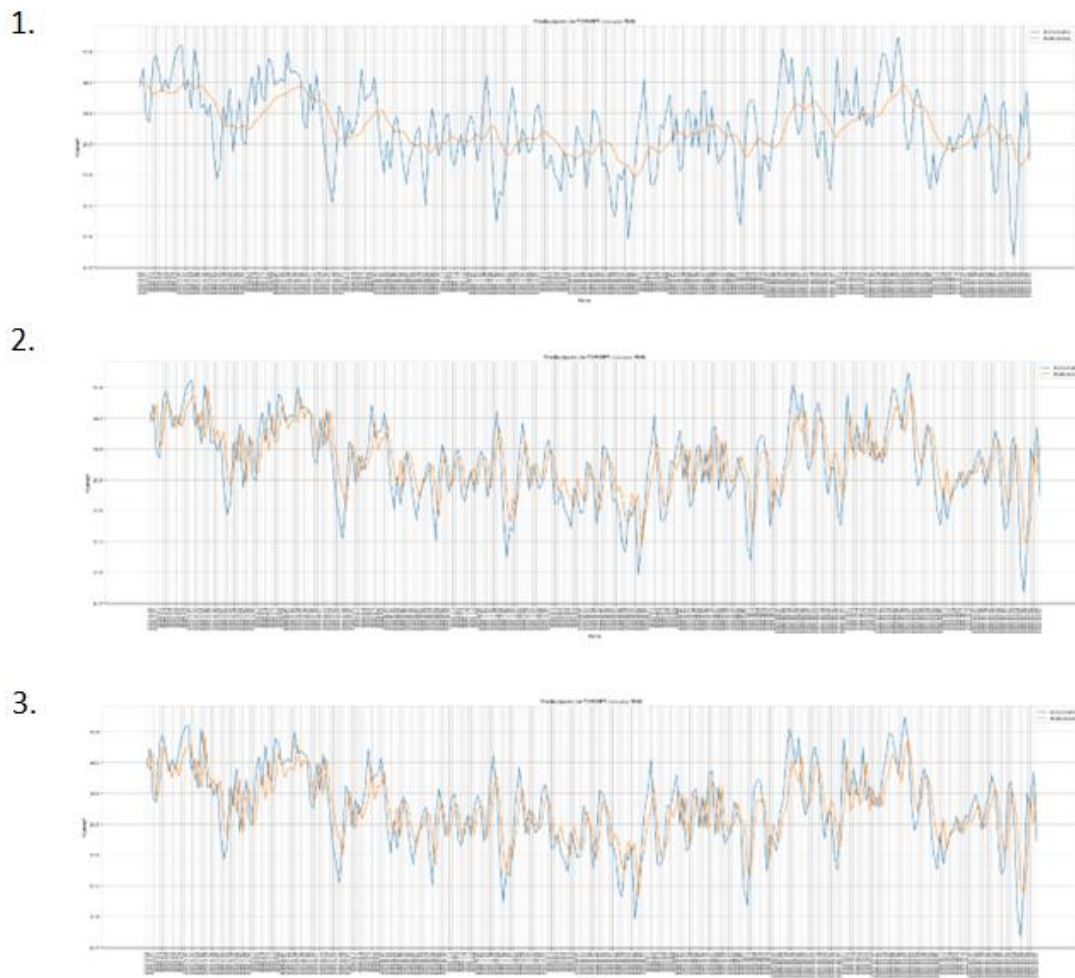
Grafica 1: Con sólo 1 a 10 épocas de entrenamiento, el error cuadrático medio (MSE) era alto, de 2.017880313383037, reflejando un ajuste inicial pobre.

Grafica 2: De 10 a 25 épocas, el MSE se redujo a 1.9493531258614458, mostrando cierta mejoría en la habilidad del modelo para predecir el rango térmico.

Con 35 épocas se alcanzó un MSE de 1.8991059485263972. Si bien aún es un valor relativamente alto, exhibe una mejora en el aprendizaje del rango de temperaturas.

T2MWET: Temperatura de bulbo húmedo a 2 metros (C)

*Ilustración 41 T2MWET: Temperatura de bulbo húmedo a 2 metros Entrenamientos*



Se entrenó un modelo para predecir la temperatura de bulbo húmedo a 2 metros (T2MWET). Se realizaron pruebas con diferentes cantidades de entrenamientos para evaluar la precisión del modelo mediante el error cuadrático medio (MSE).

Los resultados fueron:

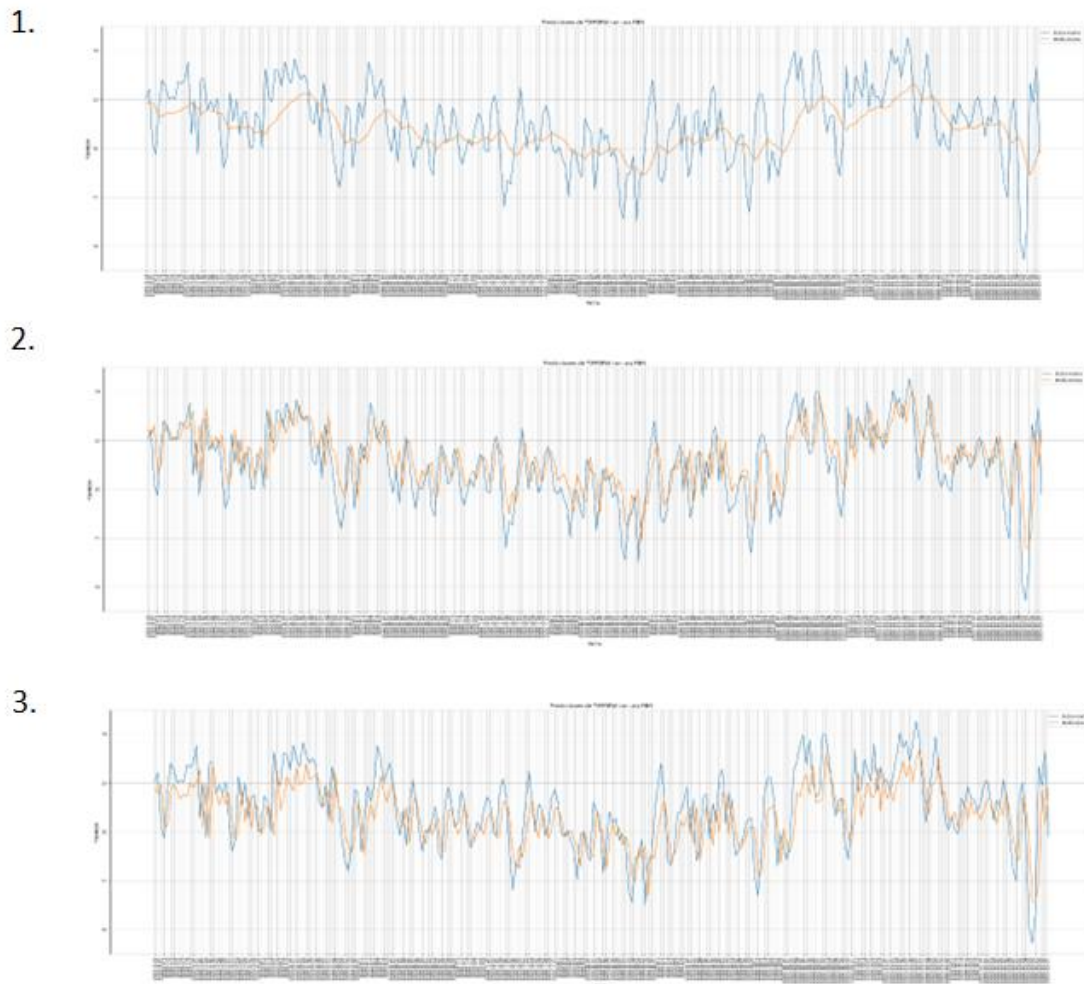
Grafica 1: Con 1 a 10 entrenamientos, el MSE fue de 0.3276282139827341.

Grafica 2: Con 10 a 25 entrenamientos, el MSE mejoró a 0.22072514001222499.

Con 35 entrenamientos, se obtuvo el mejor MSE de 0.21671446212318735.

T2MDEW: Punto de rocío/escarcha a 2 metros (C)

*Ilustración 42 T2MDEW: Punto de rocío/escarcha a 2 metros Entrenamientos*



Se entrenó un modelo para predecir el punto de rocío/escarcha a 2 metros (T2MDEW). Al igual que con T2MWET, se evaluó la precisión del modelo a través del error cuadrático medio (MSE) con diferentes cantidades de entrenamientos.

Los resultados fueron:

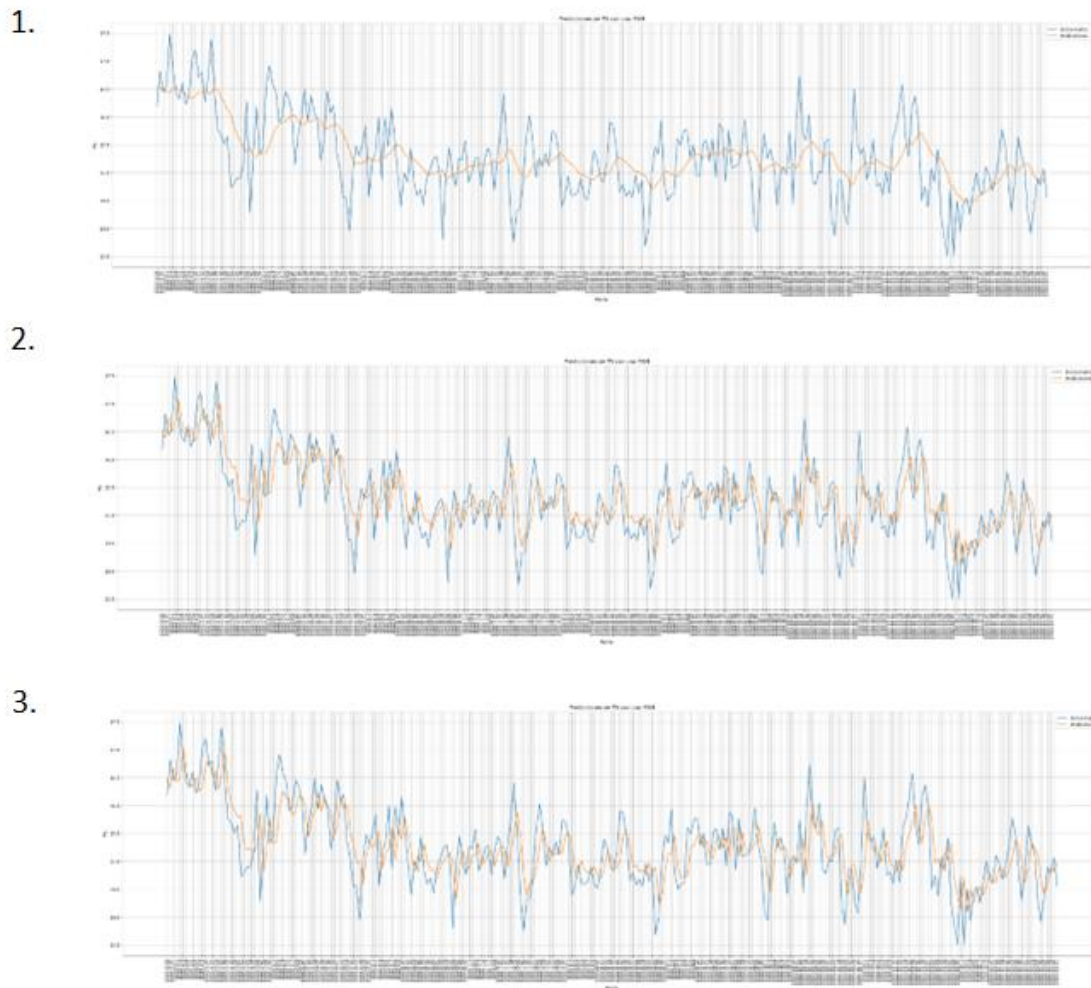
Grafica 1: Con 1 a 10 entrenamientos, el MSE fue de 0.4847191854086208.

Grafica 2: Con 10 a 25 entrenamientos, el MSE mejoró a 0.3529837244751237.

Con 35 entrenamientos, el MSE fue de 0.366595246638617, ligeramente mayor que con 25 entrenamientos.

TS: Temperatura de la superficie terrestre (C)

*Ilustración 43 TS: Temperatura de la superficie terrestre Entrenamientos*



Grafica 1: Con 1 a 10 entrenamientos, el MSE del modelo para predecir TS fue de 0.35418538240534797.

Grafica 2: Con 10 a 25 entrenamientos, la precisión mejoró, obteniendo un MSE de 0.28564226619608113.

Con 35 entrenamientos, el MSE fue de 0.28150899263393914, marginalmente mejor que con 25 entrenamientos.

Al igual que con T2MWET, se observa una mejora progresiva en la precisión del modelo para TS al incrementar la cantidad de entrenamientos, logrando el menor MSE con 35 entrenamientos. Esto sugiere que más entrenamiento contribuyó a mejorar la capacidad de predicción para esta variable.

## 2.5 ANALISIS ECONOMICO

Para la estimación de recursos se utilizó la técnica de esfuerzo de puntos de función, Esta técnica nos permite medir el tamaño funcional de un sistema de software, lo que facilita la predicción del esfuerzo requerido para su desarrollo.

Dicha estimación se fundamenta en los requerimientos funcionales delineados en el formato IEEE. Para ello, se identificarán las unidades funcionales que posibilitan la aplicación de la estimación por puntos de función, asignando el tipo y la complejidad correspondiente a cada una de ellas.

*Tabla 7 Unidad funcional de los requerimientos.*

No	Nombre del requerimiento	Tipo	Complejidad
1	Selección en las variables climáticas	EI	Bajo
2	Uso del aprendizaje profundo para el modelo	ILF	Alta
3	Implementación de librerías, para arreglos de datos y graficas.	ILF	Media
4	Datos de entrenamiento en rango de un año o 6 meses.	EI	Alta

5	Entrenamiento del modelo predictivo.	ILF	Alta
6	Reportes de anomalías.	EO	Media
7	Resultados.	EO	Media

Una vez realizada la designación de las unidades funcionales, se calcula los puntos de función no ajustados.

*Tabla 8 Cálculo de puntos de función sin ajustar*

<b>Tipo de función</b>	<b>Total, complejidad funcional</b>	<b>Total, de tipos de función</b>
<b>EI (Entrada externa)</b>	Baja ( $1 * 3PF = 3$ ) Alta ( $1 * 6PF = 6$ )	9
<b>EO (Salida externa)</b>	Media ( $1 * 5PF = 5$ )	5
<b>EQ (Consulta externa)</b>	0	0
<b>ILF (Archivo lógico interno)</b>	Media ( $1 * 7PF = 7$ ) Alta ( $2 * 10PF = 20$ )	27
<b>EIF (Archivo de interfaz externo)</b>	0	0
	<b>Total</b>	41

Obteniendo estos datos, se procede a calcular los puntos de función ajustados

*Tabla 9 Puntos de función ajustados.*

<b>Factor de ajuste</b>	<b>Puntaje</b>
<b>Comunicación de datos</b>	5
<b>Procesamiento distribuido</b>	5
<b>Objetivos de rendimiento</b>	3
<b>Configuración del equipamiento</b>	4
<b>Tasa de transacciones</b>	2
<b>Entrada de datos en línea</b>	5
<b>Interfaz con el usuario</b>	3
<b>Actualizaciones en línea</b>	1
<b>Proceso complejo</b>	3
<b>Reusabilidad del código</b>	5
<b>Facilidad de implementación</b>	3
<b>Facilidad de operación</b>	1

<b>Instalaciones múltiples</b>	2
<b>Facilidad de cambios</b>	3
<b>Factor de ajuste</b>	45

Después de obtener los datos de factor de ajuste, se realiza la operación de puntos de función ajustados.

$$PFA = PFS * [0.65 + (0.01 * \text{Factor de ajuste})]$$

$$PFA = 41 * [0.65 + (0.01 * 45)]$$

$$PFA = 41 * [0.65 + 0.45]$$

$$PFA = 41 * [1.1]$$

$$PFA = 45.1$$

Se realiza la estimación del esfuerzo, teniendo en cuenta los siguientes datos:

Estimación del esfuerzo requerido:

$$H/H \text{ (horas/hombre)} = PFA * \text{Horas de trabajo promedio}$$

$$H/H = 45.1 * 8$$

$$\text{Horas / Hombre} = 360.8$$

Se establece para este proyecto un numero de 6 horas de trabajo diario para obtener un mayor rendimiento de este. De igual manera se establece un promedio de 20 días hábiles mensuales de trabajo.

$$360.8 (H/H) / 6H = 60.13 \text{ días de trabajo}$$

$$60.13 / 20 = 3.006 \text{ meses}$$

Para desarrollar el software se requiere se 3.006 meses con la participación de un ingeniero, quien trabajara de lunes a viernes 6 horas al día. En vista de contar con un solo desarrollador, la estimación de duración del proyecto será 3.006 meses.

En cuanto la estimación de costos:

$$\text{Sueldo mensual de desarrollador} = 2.500.000$$

$$\text{Estimación de costos} = 2.500.000 * 3.006$$

$$\text{Estimación de costos} = 7.515.000$$

Con esto se concluye que para el desarrollo del software aplicando el método de puntos de función por desarrollador se determina que en promedio se requiere un valor total de **7.515.000**

## 2.6 RESULTADOS

### 2.6.1 EVALUACIÓN TECNOLÓGICA

Para la evaluación tecnológica se contemplaron algunas herramientas distintas a las utilizadas para el desarrollo y con ello se generó un cuadro comparativo.

*Tabla 10 Cuadro comparativo evaluación tecnológica*

Tecnología utilizada	Tecnología alternativa	Comparación
<b>Redes Neuronales</b>	Redes Neuronales	Las CNN son mejores para datos de imagen, video y señales. Las LSTM
<b>Recurrentes LST</b>	Convolucionales (CNN)	

		superan a las CNN en secuencias temporales largas.
<b>Redes Neuronales Recurrentes LSTM</b>	Máquinas de Vectores de Soporte (SVM)	Las SVM tienen menor poder de procesamiento y memoria. Las LSTM modelan mejores relaciones temporales complejas.
<b>Redes Neuronales Recurrentes LSTM</b>	Árboles de Decisión Aleatorios (Random Forest)	Los Random Forest no capturan bien dependencias temporales. Las LSTM se adaptan mejor a la estacionalidad.
<b>TensorFlow</b>	PyTorch	Ambos son frameworks líderes en deep learning. TensorFlow es más usable, PyTorch es más flexible.
<b>Python</b>	R	Python tiene más librerías para machine learning. R es más utilizado en estadística y análisis de datos.
<b>MERRA-2</b>	Datos de estaciones meteorológicas	MERRA-2 provee mediciones globales consistentes. Las estaciones ofrecen datos locales de alta resolución.

Seguido se contempló realizar un cuadro comparativo de las ventajas y desventajas de las principales tecnologías utilizadas en el modelo predictivo y las razones por las cuales

representan una buena opción en este desarrollo

*Tabla 11 Ventajas y desventajas de las principales tecnologías utilizadas*

Tecnología	Ventajas	Desventajas
<b>Redes Neuronales Recurrentes LSTM</b>	Manejo eficiente de dependencias temporales a largo plazo en los datos. Buen desempeño con series de tiempo y secuencias.	Requieren muchos datos de entrenamiento.
	Capacidad para hacer predicciones basadas en tendencias y estacionalidades.	Son modelos complejos que demandan alto poder de cómputo.
<b>TensorFlow</b>	Plataforma líder en deep learning. Permite crear y entrenar eficientemente modelos.	Curva de aprendizaje empinada.
	Tiene capacidades de computación distribuida.	Depende de bibliotecas externas para algunas funcionalidades.
<b>Python</b>	Lenguaje de programación versátil y fácil de aprender. Cuenta con amplio ecosistema de librerías para análisis de datos y machine learning.	Lentitud en la ejecución comparado con lenguajes compilados.
		Dificultades para desarrollar interfaces gráficas nativas.

<b>MERRA-2</b>	Conjunto de datos meteorológicos global de alta calidad.	Requiere acceso a servidores de la NASA para descargar los datos.
	Permite entrenar modelos con series de tiempo extensas y confiables.	No incluye mediciones de todas las variables en todas las ubicaciones.

## CONCLUSIÓN DE LA EVALUACIÓN TECNOLÓGICA

El proyecto hace uso de tecnologías de vanguardia en el campo del aprendizaje automático, como las redes neuronales recurrentes LSTM implementadas con TensorFlow y Keras. Esto permite un modelado predictivo avanzado. Además de ello, la arquitectura LSTM es adecuada para capturar relaciones temporales complejas en series de tiempo, como es el caso de los datos meteorológicos, y se aplican buenas prácticas de preprocesamiento y normalización de datos, cruciales para el entrenamiento efectivo del modelo.

El desarrollo aprovecha fuentes de datos abiertos y confiables como MERRA-2 de la NASA, mejorando la calidad de los datos de entrenamiento.

También, se evalúa el desempeño del modelo con métricas estándar como el error cuadrático medio, permitiendo medir su precisión.

Por otro lado, se considera que el conjunto de datos podría expandirse incorporando más variables climáticas, estaciones meteorológicas y series de tiempo más largas.

Otro aspecto que se contempló fue el evaluar otras arquitecturas y modelos de machine learning como las redes convolucionales pudiendo mejorar el rendimiento predictivo, contemplando que el monitorear continua y automáticamente el desempeño del

modelo en producción permitiría detectar la necesidad de reentrenamientos.

## **2.7 CONCLUSIONES Y RECOMENDACIONES**

- El proyecto representó un esfuerzo integral para desarrollar un modelo de pronóstico del clima basado en técnicas avanzadas de machine learning. La implementación de una red neuronal recurrente LSTM permitió capturar las complejas interrelaciones temporales en las variables meteorológicas.
- La etapa de recolección y preparación de los datos fue un paso crucial, ya que se compiló un extenso conjunto de datos históricos de múltiples parámetros climáticos. La incorporación de datos de fuentes confiables como MERRA-2 de la NASA aportó información de calidad al modelo.
- El preprocesamiento y normalización de los datos de entrenamiento y prueba fue fundamental para mejorar el aprendizaje y rendimiento de la red neuronal. Las técnicas como escalado Min-Max aplicadas optimizaron los datos de entrada al modelo.
- Los resultados de las métricas de desempeño como el error cuadrático medio mostraron consistentemente una reducción a medida que aumentaba el número de épocas de entrenamiento. Esto pone en evidencia la capacidad de aprendizaje del modelo al incrementar la exposición a los datos.
- La arquitectura LSTM implementada probó ser efectiva para capturar las dependencias temporales y estacionalidades en las series de tiempo meteorológicas. Esto se vio reflejado en la habilidad del modelo para replicar patrones en las predicciones.

- Si bien el alcance del proyecto se limitó a una región específica, demuestra el gran potencial de las técnicas de machine learning para la modelización predictiva meteorológica a gran escala.
- Los resultados obtenidos son prometedores y destacan la importancia de seguir investigando en el campo de la inteligencia artificial aplicada a la comprensión del clima y generación de pronósticos confiables.

Como conclusión, el proyecto culminó con éxito en el desarrollo de un modelo de aprendizaje automático para la predicción de variables climáticas captadas en la localidad de Ubaté, sentando las bases para continuar perfeccionando la precisión y alcance de este tipo de herramientas predictivas del clima. Los aportes en este campo pueden generar gran impacto en múltiples sectores y actividades humanas.

## **2.8 Bibliografía**

- Bosilovich, M. G. ((2015). ). *MERRA-2*: File specification (No. GSFC-E-DAA-TN27096.
- Cano Alor, C. G. ( 2020). *Diseño de una aplicación de monitoreo de emanación de gases en la ciudad mediante el uso de tecnologías de inteligencia artificial*.
- Cardozo Vásquez, A. (s.f.). *Desarrollo de un modelo de red neuronal artificial para la reducción de escala (downscaling) de datos de temperatura del modelo Climático Global Canadiense 3.1 a estaciones meteorológicas colombianas*. (Master's thesis, Universidad de La Sabana.
- Cortes, D. (2021). *predicción temperatura bogota: enfoque estocástico vs aprendizaje* .
- Cruz, I. B. (2007). *Redes neuronales recurrentes para el análisis de secuencias*. . Revista Cubana de Ciencias Informáticas.
- Cundinamarca, A. M. (2023 ). *Alcaldía Municipal Ubaté Cundinamarca* .
- De la Barrera Castillo, J. C. (2023). *Metodologías de ahorro energético aplicadas a los sistemas HVAC utilizando inteligencia artificial: una revisión del estado del arte*.
- Deulofeu Gómez, R. (2020). *Introducción y optimización estocástica de redes neuronales profundas MLP*.
- Dillon, J. V. (2017). *Tensorflow distributions*. arXiv preprint arXiv:.
- Egas Arizala, K. O. (2020). *Diseño de un modelo predictivo basado* .
- Fonseca-González, W. V.-C.-V. ((2019). ). *Potencial de mitigación del cambio climático de los ecosistemas forestales caducifolios en Costa Rica: modelos predictivos de biomasa y carbono*. . Revista de Ciencias Ambientales.
- GISTFA. (2022). *Protocolo de GISTFA*. , 6ta Versión. Universidad de Cundinamarca, .
- Guerrero-Álvarez, J. A. (2021). *Predicción niveles de precipitación para la identificación de zonas agrícolas* . (Master's thesis).
- Hao, J. &. ((2019). ). *Machine learning made easy: a review of scikit-learn package in*

- python programming language*. Journal of Educational and Behavioral Statistics.
- Mahesh, B. ((2020)). *Machine learning algorithms-a review*. International Journal of Science and Research (IJSR).
- Montes, C. (2006). *Sobreaajuste-overfitting*. centro de Investigaciones Energicas, Medioambientales y Tecnologicas.
- Montesdeoca Santana, B. (2016). *Estudios de predicción en series temporales de datos meteorológicos utilizando redes neuronales recurrentes*. Bachelor's thesis.
- Murcia Ramirez, J. S. (s.f.). *Modelo de máxima entropía para la predicción de criminalidad con tecnicas de Machine Learning* .
- Murcia, D. S. (2015). *.Diseño e implementación de un prototipo de estación* .
- Paniagua Urquijo, D. &. ( 2021). *Diseño, desarrollo e implementacion de una plataforma web de datos climatológicos mediante una boya construida en un arduino en el municipio de Guataqui Cundinamarca* .
- Python, W. (2021). *Python*. . Python Releases for Windows, 24.
- Randles, B. M. ( 2017 ). *Using the Jupyter notebook as a tool for open science: An empirical study*. ACM/IEEE Joint Conference on Digital Libraries (JCDL) (pp. 1-2). IEEE.
- Rodríguez., &. D. ( 2015). *¿Por qué implementar Scrum? Why to implement* . Revista Ontare, 3(1), 125–144. .
- Ruiz Palomo, G. J. ( 2021). *Klimachingaza: visor geográfico web* .
- Sommerville. (2005). *Ingenieria de software*. pearson educacion .
- Vázquez Ruiz, T. (2023). *Desarrollo y aplicación de metodología de modelado predictivo de series temporales*.
- Villavicencio, J. ( 2010). *Introducción a series de tiempo*. . Puerto Rico.

Yana Puma, J. W. (2020). *Modelo predictivo agroclimático para detectar cambios climatológicos con machine learning e IoT para el sector agrario de la región de Puno .*

## **2.9 ANEXOS**

**DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE ESTACIÓN  
METEOROLÓGICA AUTOSOSTENIBLE PARA EL ANÁLISIS DEL COMPORTAMIENTO  
CLIMÁTICO  
MODELO MACHINE LEARNING**

*Versión 01*



**UDECA**  
UNIVERSIDAD DE  
CUNDINAMARCA

**Manual/Guía de Instalación Desarrollo de  
Sistemas de Información Dirección de Sistemas y  
Tecnología**

## **1. Especificaciones Técnicas**

### **1.1. Especificaciones de Hardware.**

- Procesador: contar con un procesador de al menos cuatro núcleos.
- Memoria RAM: contar con una memoria RAM de por lo menos 4GB
- Almacenamiento: al menos 5 GB de almacenamiento disponible en el dispositivo.

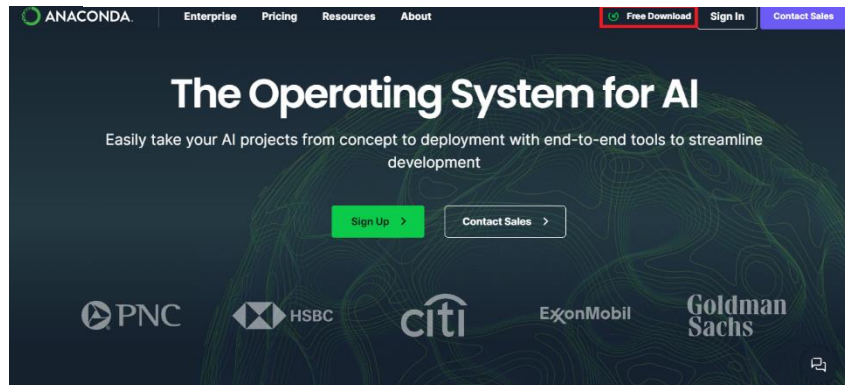
### **1.2. Especificaciones de Software.**

- El modelo de machine learning está diseñado para ejecutarse en Jupyter Notebook versión 6.5.4 o superior.
- El modelo requerirá Bibliotecas de machine learning, manipulación de datos y visualización de datos y resultados. Teniendo como mínimo las siguientes versiones: scikit-learn 1.3.0, keras 2.10.0, tensorflow 2.10.0, numpy 1.26.0, pandas 1.5.2 y matplotlib 3.7.1

## **2. Instalación Jupyter Notebook (Anaconda)**

- Paso 1: Dirigirse al sitio oficial de Anaconda y seleccionar Free

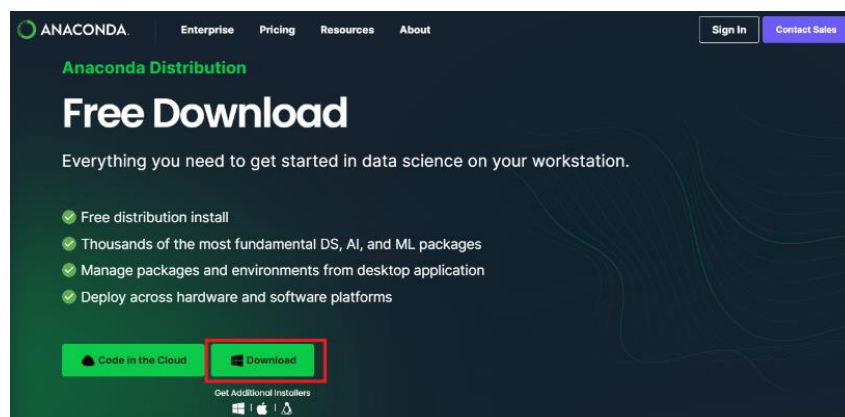
*Ilustración 44 Sitio Oficial De Anaconda*



Download en la esquina superior derecha.

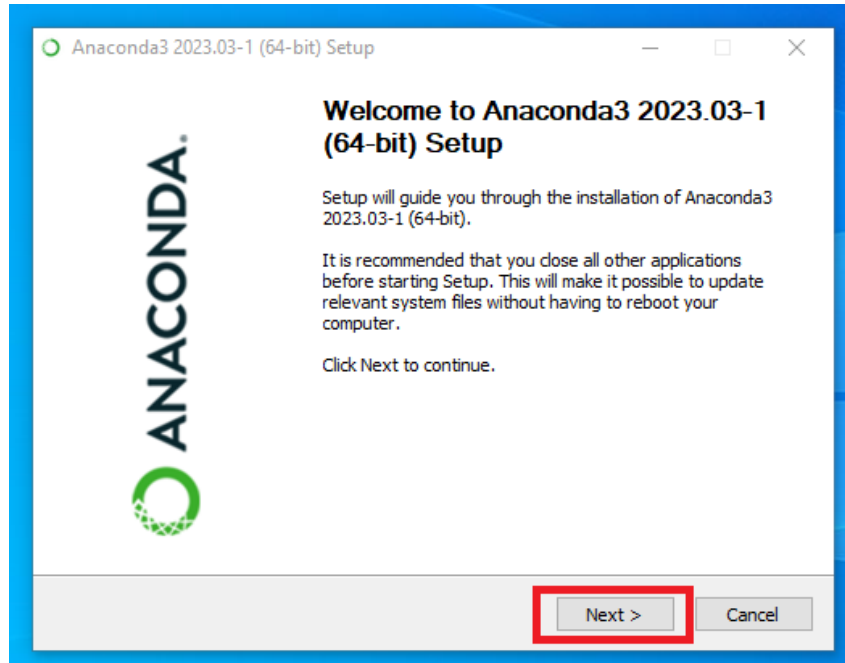
- Paso 2: seleccionamos la descarga del sistema operativo que corresponda.

*Ilustración 45 Descarga Anaconda*

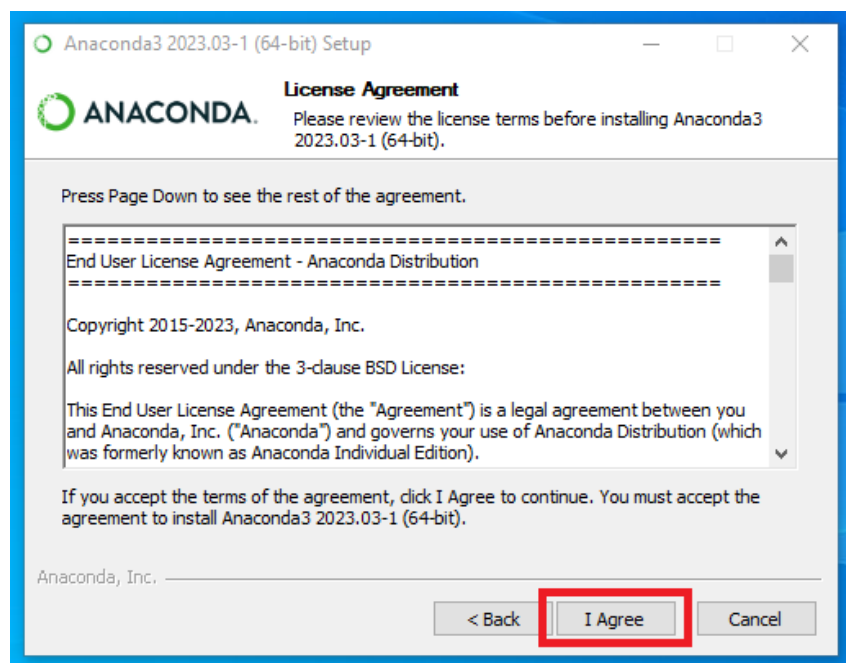


- Paso 3: Ejecutaremos el instalador y seleccionamos en Next

*Ilustración 46 Instalador Anaconda*



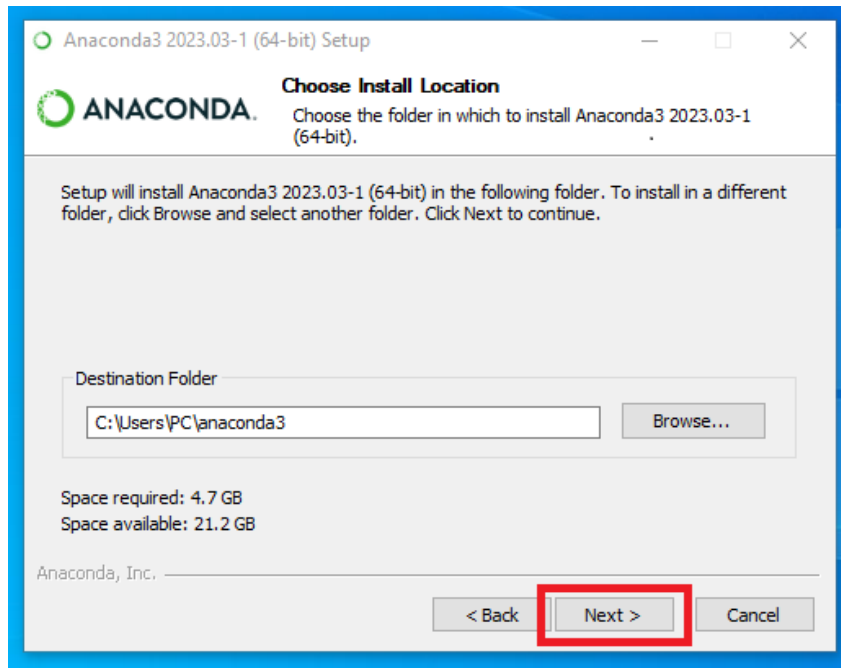
*Ilustración 47 Acuerdo de licencia Anaconda*



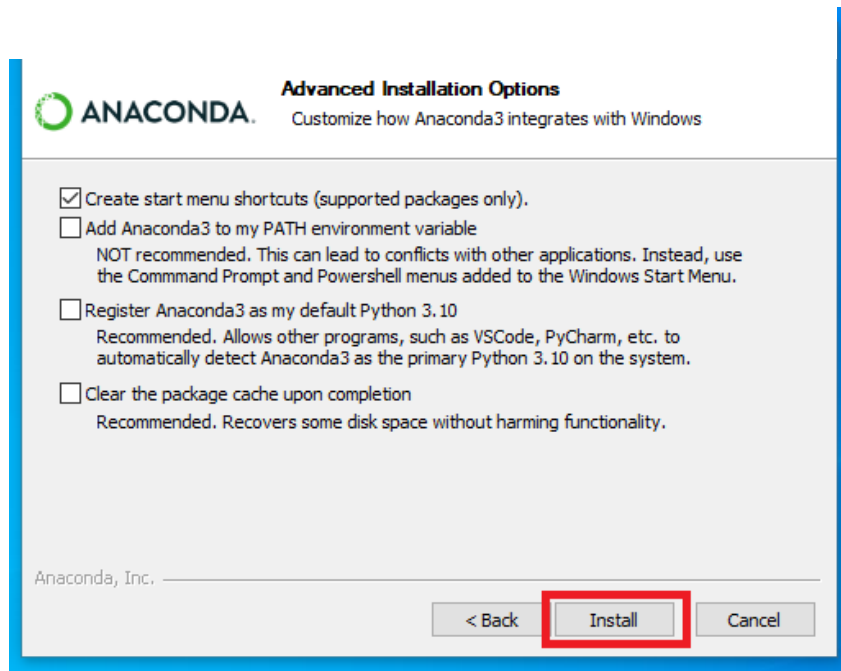
- Paso 4: Después de aceptar los términos de la licencia de Anaconda,

seleccionamos la ruta de la instalación y luego Next.

*Ilustración 48 Selección de la ubicación del directorio.*



### Ilustración 49 Paso final Instalación



- Paso 5: Como paso final seleccionamos Install y al terminar abrimos Anaconda.

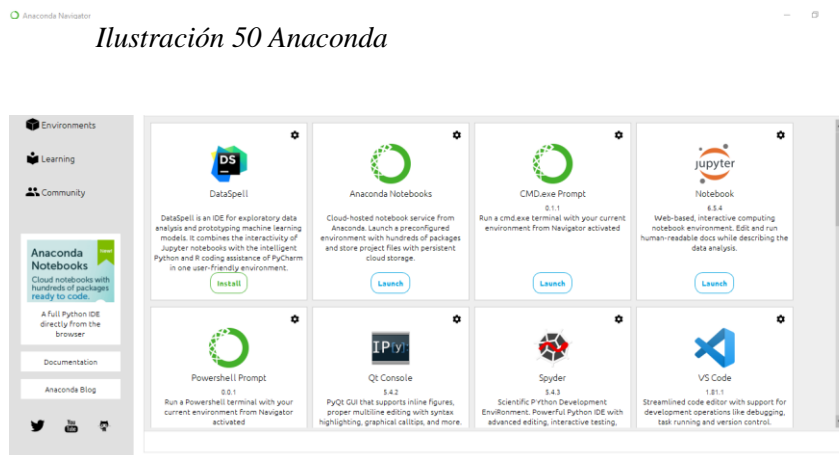
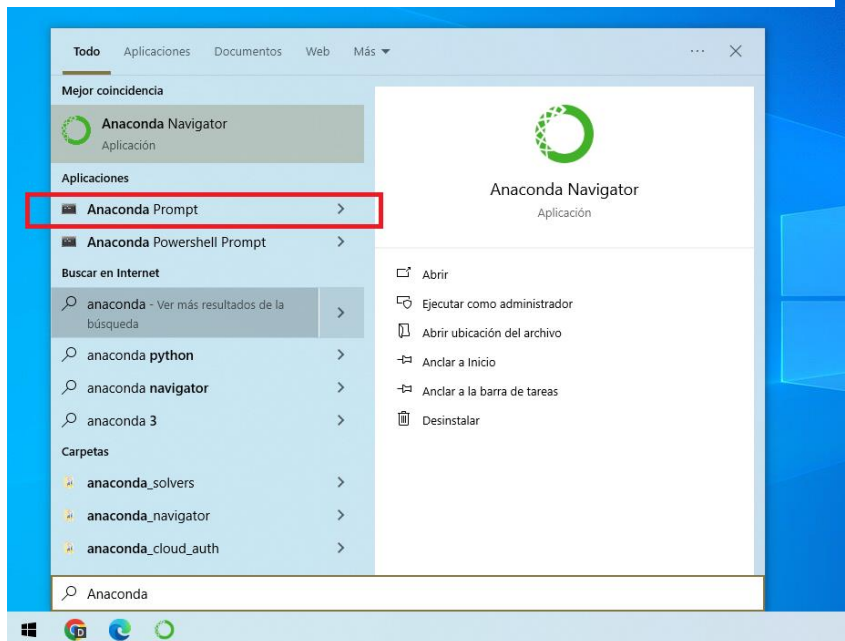


Ilustración 50 Anaconda



*Ilustración 52 Abrir Terminal*

- Paso 2: Escribimos el comando para la instalación de bibliotecas de Anaconda: `conda install Nombre De La Biblioteca`

*Ilustración 53 Anaconda Prompt*

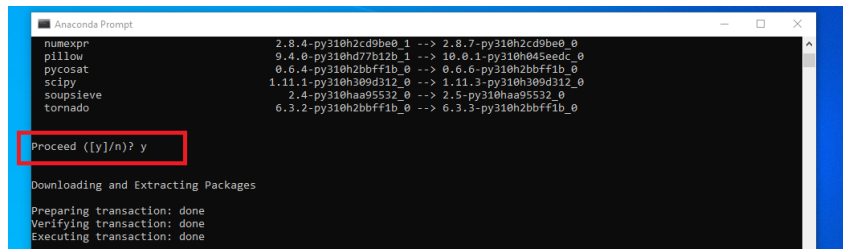
```
conda install scikit-learn
conda install tensorflow
conda install pytorch
conda install keras

conda install pandas
conda install numpy

conda install matplotlib
conda install seaborn
```

- Paso 3: Cada comando requiere una confirmación después de ejecutarse, escribimos la letra Y después presionamos la tecla Enter.

*Ilustración 54 Confirmación de biblioteca*



```

Anaconda Prompt
numexpr 2.8.4-py310h2cd0be0_1 --> 2.8.7-py310h2cd0be0_0
pillow 9.4.0-py310hd77b12b_1 --> 10.0.1-py310ha4seedfc_0
pycosat 0.6.4-py310h2bbff1b_0 --> 0.6.6-py310h2bbff1b_0
scipy 1.11.1-py310h309d312_0 --> 1.11.3-py310h309d312_0
soupsieve 2.4-py310haa95532_0 --> 2.5-py310haa95532_0
tornado 6.3.2-py310h2bbff1b_0 --> 6.3.3-py310h2bbff1b_0

Proceed ([y]/n)? y

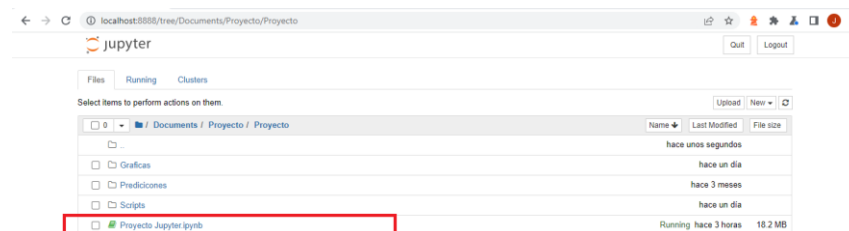
Downloading and Extracting Packages
Preparing transaction: done
Verifying transaction: done
Executing transaction: done

```

#### 4. Abrir proyecto del modelo

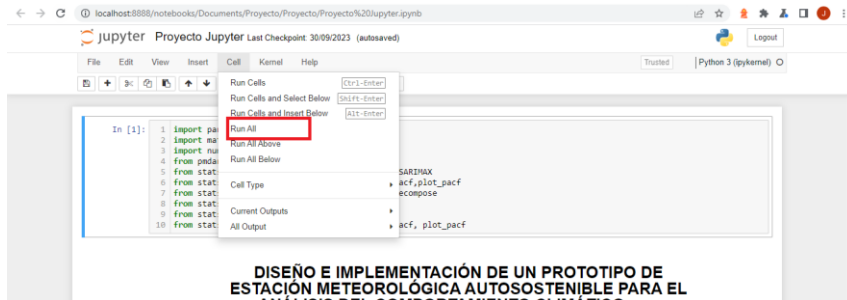
- Paso 1: Dentro de jupyter notebook dirigirse a la carpeta que contenga el proyecto y hacer doble click para abrirlo

*Ilustración 55 Abrir proyecto.*



Paso 2: Para correr el proyecto completo seleccionamos Cell/Run All

*Ilustración 56 Correr Proyecto*



Paso 3: En caso de correr un único bloque, este se selecciona y se presiona el botón de Run en la parte superior.

*Ilustración 57 Correr bloque de código.*



**DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE ESTACIÓN  
METEOROLÓGICA AUTOSOSTENIBLE PARA EL ANÁLISIS DEL  
COMPORTAMIENTO CLIMÁTICO  
- MODULO MACHINE LEARNING-**

**Versión 01**



**UDEC**  
UNIVERSIDAD DE  
CUNDINAMARCA

**Desarrollo de Sistemas de Información  
Dirección de Sistemas y Tecnología**

## **1. Conceptos básicos**

### **1.1. Definiciones**

#### **1.1.1 ¿Qué es una red neuronal LSTM?**

Las redes neuronales LSTM (Long Short-Term Memory, en inglés) son un tipo especial de red neuronal recurrente capaz de aprender dependencias a largo plazo. Las redes LSTM fueron diseñadas para lidiar con el problema del desvanecimiento del gradiente que tienen las redes recurrentes tradicionales al trabajar con secuencias largas de datos.

La estructura básica de una red LSTM consiste en una cadena de módulos repetitivos de memoria (celdas LSTM). Cada celda LSTM contiene puertas de entrada, salida y olvido que regulan el flujo de información. Estas puertas pueden aprender qué información es relevante mantener o descartar de la celda de estado a lo largo del tiempo. De esta manera, una red LSTM puede aprender relaciones temporales a largo plazo entre datos secuenciales, como series de tiempo. Son utilizadas en tareas como procesamiento de lenguaje natural, reconocimiento de voz, predicción de series temporales, entre otras. Las redes LSTM ayudan a mantener un contexto de información relevante que permita entender dependencias en los datos cuando existen largas distancias temporales.

#### **1.1.2 ¿Qué es estacionariedad en una serie de tiempo?**

El análisis de series de tiempo requiere evaluar si los datos presentan la propiedad de estacionariedad. Una serie temporal se considera estacionaria cuando sus propiedades estadísticas como media, varianza y autocorrelación se mantienen constantes a lo largo del tiempo.

En particular, en una serie estacionaria la media no varía sino que fluctúa alrededor

de un valor promedio constante. Asimismo, la variabilidad de los datos con respecto a la media tampoco cambia con el tiempo. Otra propiedad importante es que la dependencia entre observaciones con cierto desfase temporal permanece invariante.

Adicionalmente, la estacionariedad implica que la serie de tiempo no tiene tendencias deterministas ni efectos periódicos o estacionales. La ausencia de tendencias y estacionalidades hace que el proceso generador de los datos sea estable alrededor de la media y no exhiba un crecimiento sostenido en alguna dirección.

Evaluar si una serie de tiempo cumple las propiedades de estacionariedad es crucial antes de ajustar modelos como ARIMA. Esto se debe a que la mayoría de técnicas de series de tiempo requieren estacionariedad para que los parámetros estimados sean robustos y los pronósticos confiables. Las series no estacionarias deben transformarse mediante diferenciación u otros métodos.

### **1.1.3 El subajuste o underfitting en el modelo predictivo**

El subajuste o underfitting es un problema frecuente al desarrollar modelos predictivos con algoritmos de machine learning. Ocurre cuando el modelo no logra capturar de manera adecuada la relación entre las variables predictivas y la variable objetivo.

Esto sucede cuando el modelo es demasiado simple en comparación con la verdadera relación que subyace en los datos. Por ejemplo, tratar de ajustar un modelo lineal a datos que siguen una tendencia no lineal. El modelo subajustado no será capaz de adaptarse lo suficiente a los patrones en los datos.

Las consecuencias del subajuste son pronósticos poco precisos y un bajo desempeño en nuevos conjuntos de datos, dado que el modelo no puede extrapolar

adecuadamente los patrones aprendidos. Otra consecuencia es alta varianza entre los resultados en distintos conjuntos de entrenamiento.

El subajuste puede deberse a no haber incluido suficientes predictores relevantes, no haber utilizado un modelo lo suficientemente flexible o regularizado en exceso. Para prevenir el subajuste se recomienda utilizar algoritmos más flexibles, incluir más variables explicativas y evitar la regularización excesiva. Encontrar el balance adecuado entre flexibilidad y capacidad predictiva es clave para crear modelos predictivos efectivos.

#### **1.1.4 sobreajuste o overfitting en el modelo predictivo**

El sobreajuste se produce cuando un modelo predictivo se ajusta demasiado bien a los datos de entrenamiento, pero tiene un bajo desempeño en nuevos conjuntos de datos. Esto ocurre cuando el modelo es excesivamente complejo en relación a la verdadera señal en los datos.

Algunas causas típicas del sobreajuste son: tener muy pocos datos en relación a la complejidad del modelo, ruido en los datos, o dejar que el algoritmo aprenda relaciones muy específicas de la muestra de entrenamiento que no se generalizan bien.

Las consecuencias del sobreajuste son una varianza muy alta en los resultados obtenidos con distintos conjuntos de entrenamiento, así como malas predicciones en nuevos datos donde el modelo no puede extrapolar adecuadamente.

Para evitar el sobreajuste algunas buenas prácticas son: usar más datos de entrenamiento, técnicas de regularización, reducir la complejidad del modelo, validación cruzada y poda de modelos. Encontrar el justo medio entre flexibilidad y capacidad de generalización es la clave para crear buenos modelos predictivos.

## 1.2 Proceso de modelado

En este primer bloque Importamos las librerías necesarias para el análisis del archivo del Csv que contiene los datos para el entrenamiento del modelo. F1

*Ilustración 58 Librerías para el análisis de los datos*

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from statsmodels.tsa.stattools import adfuller
```

En la siguiente parte se demuestra la información de los datos recolectados, al igual que la definición de las variables como lo son, el tipo, la cantidad de registros, , memoria usada y registros nulos. F2

*Ilustración 59 Información de los datos.*

```

: 1 df = pd.read_csv('2015- Agt 2023 FC.csv',sep=',')
  2 df['DAY'] = pd.to_datetime(df['DAY'])
  3 df.info()

```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3165 entries, 0 to 3164
Data columns (total 15 columns):
#   Column                Non-Null Count  Dtype
---  ---                -
0   DAY                   3165 non-null   datetime64[ns]
1   T2M                   3165 non-null   float64
2   T2MDEW                3165 non-null   float64
3   T2MWET                3165 non-null   float64
4   TS                    3165 non-null   float64
5   T2M_RANGE             3165 non-null   float64
6   T2M_MAX               3165 non-null   float64
7   T2M_MIN               3165 non-null   float64
8   QV2M                  3165 non-null   float64
9   RH2M                  3165 non-null   float64
10  PRECTOTCORR           3165 non-null   float64
11  PS                     3165 non-null   float64
12  WS10M                  3165 non-null   float64
13  WS10M_MIN             3165 non-null   float64
14  WS10M_MAX             3165 non-null   float64
dtypes: datetime64[ns](1), float64(14)
memory usage: 371.0 KB

```

Este bloque de código hace lectura del archivo csv “2015- Agt 2023 FC.csv” como se observa en la parte superior de la imagen, es decir, la primera línea de código; en la segunda línea de código se define como index la columna que corresponde a las fechas en formato “YYYY/MM/DD” de la captación de los datos, en caso contrario que la información no se encuentren en dicho formato, se ejecutará el siguiente bloque correspondiente a la f3

*Ilustración 60 Bloque de código para la corrección del csv de entrenamiento*

```

1 #MODIFICA ACOMODA COMBINA = LA FECHA Y GUARDA LOS CAMBIOS EN UN NUEVO .csv
2 datos_clima = pd.read_csv('2020-2022.csv', sep=',')
3 datos_clima = datos_clima.astype({'YEAR': 'str'})
4 datos_clima = datos_clima.astype({'MO': 'str'})
5 datos_clima = datos_clima.astype({'DY': 'str'})
6 datos_clima['DAY'] = datos_clima['YEAR']+'-'+datos_clima['MO']+'-'+datos_clima['DY']
7 fecha = datos_clima.pop('DAY')
8 datos_clima.insert(0, 'DAY', fecha)
9 datos_clima = datos_clima.drop(['YEAR', 'DY', 'MO'], axis=1)
10 datos_clima.to_csv('2020-2022 FC.csv', index=False)

```

Este tiene por entrada el nombre del CSV con el formato de la fecha dividida para posterior mente unificar en una sola columna y condicionar dicha entrada, guardando los parámetros en un nuevo CSV. F3

*Ilustración 61 Cabecera de los datos de entrenamiento.*

```

1 df.head(10)

```

	DAY	T2M	T2MDEW	T2MWET	TS	T2M_RANGE	T2M_MAX	T2M_MIN	QV2M	RH2M	PRECTOTCORR	PS	WS10M	WS10M_MIN	WS10M_MAX
0	2015-01-01	14.23	10.35	12.29	14.52	12.02	20.46	8.44	10.25	80.75	0.02	77.78	2.20	1.13	4.41
1	2015-01-02	13.15	9.10	11.12	13.33	13.22	20.33	7.12	9.40	79.62	0.00	77.81	2.91	1.52	5.17
2	2015-01-03	13.48	9.69	11.58	13.63	12.52	20.37	7.85	9.77	80.44	0.00	77.82	2.83	1.10	4.75
3	2015-01-04	13.67	10.10	11.88	13.80	11.70	20.23	8.55	10.01	81.75	0.00	77.86	3.08	1.27	5.28
4	2015-01-05	13.58	9.09	11.33	13.74	12.60	20.55	7.95	9.34	78.00	0.00	77.89	2.54	1.09	4.67
5	2015-01-06	12.14	6.89	9.51	12.17	15.23	20.04	4.80	8.12	74.00	0.00	77.82	2.87	1.49	5.05
6	2015-01-07	12.18	6.56	9.37	12.21	14.64	19.84	5.20	7.93	72.50	0.00	77.71	3.26	1.46	4.98
7	2015-01-08	13.54	9.90	11.72	13.80	12.62	19.99	7.37	9.89	81.19	0.01	77.74	2.13	0.97	3.84
8	2015-01-09	14.24	10.98	12.61	14.53	11.08	20.34	9.26	10.62	83.00	0.01	77.78	2.30	1.10	4.53
9	2015-01-10	14.88	11.37	13.12	15.15	12.24	21.13	8.89	10.93	82.00	0.02	77.85	2.05	0.95	3.69

Esta línea de código corresponde a los 10 primero registros, los cuales nos muestran la información correspondiente a las primeras fechas de captación. F4

*Ilustración 62 Últimos registros del CSV de entrenamiento.*

```
1 df.tail(10)
```

	DAY	T2M	T2MDEW	T2MWET	TS	T2M_RANGE	T2M_MAX	T2M_MIN	QV2M	RH2M	PRECTOTCORR	PS	WS10M	WS10M_MIN	WS10M_MAX
3155	2023-08-22	15.37	13.21	14.29	15.64	9.22	20.52	11.30	12.27	87.88	6.46	77.64	1.36	0.79	1.93
3156	2023-08-23	15.73	12.72	14.22	15.98	9.27	20.78	11.52	11.84	83.62	3.00	77.76	1.55	0.65	2.73
3157	2023-08-24	15.61	13.12	14.37	15.74	9.71	20.60	10.89	12.21	86.12	4.35	77.85	1.44	0.55	2.26
3158	2023-08-25	15.23	12.69	13.97	15.33	8.68	20.15	11.48	11.84	85.69	6.03	77.84	1.41	0.79	2.14
3159	2023-08-26	15.15	11.88	13.51	15.44	11.59	21.26	9.69	11.23	82.44	0.83	77.82	1.95	1.10	3.10
3160	2023-08-27	15.16	11.30	13.23	15.30	11.86	21.49	9.64	10.80	79.81	0.89	77.87	2.20	1.45	3.18
3161	2023-08-28	15.90	13.16	14.53	16.02	8.88	21.00	12.12	12.21	84.81	6.17	77.95	1.86	1.01	2.43
3162	2023-08-29	15.59	12.75	14.17	15.80	10.54	20.95	10.41	11.90	84.44	3.50	77.98	2.08	1.06	3.11
3163	2023-08-30	15.44	12.29	13.87	15.58	9.42	20.51	11.09	11.47	82.88	1.37	77.95	2.85	1.41	4.89
3164	2023-08-31	15.48	12.69	14.09	15.62	9.38	20.45	11.08	11.84	84.44	3.42	77.89	2.54	1.27	3.77

Esta línea de código corresponde a los 10 últimos registros, los cuales nos muestran la información correspondiente a las fechas finales de la captación de los datos. F5

*Ilustración 63 función “graficar por año”*

```
1 data = pd.read_csv('DHU 2015 2022 FC.csv', sep=',')
2 data['DAY'] = pd.to_datetime(data['DAY'])
3
4 data['año'] = data['DAY'].dt.year
5 # Obtener la lista de años únicos
6 años = data['año'].unique()
7 def graficarporaño(variable, nombreV):
8     for año in años:
9         # Filtrar los datos para el año actual
10        datos_año = data[data['año'] == año]
11
12        # Crear una figura y un conjunto de subgráficos
13        fig, ax = plt.subplots()
14
15        # Graficar los datos correspondientes al año actual
16        ax.plot(datos_año['DAY'], datos_año[variable])
17
18        # Configurar los ejes y el título
19        ax.set_xlabel('Fecha')
20        ax.set_ylabel(nombreV)
21        ax.set_title(f'Gráfica de {nombreV} - Año {año}')
22        # Mostrar la gráfica
23        plt.show()
```

Para este bloque se define la función “graficar por año” que tiene como parámetros la variable y el nombre de la variable, generando un retorno de graficas por año. F6

## 2. Configuración del Modelo

### 2.1 Parámetros del modelo

#### 2.1.1 Importación de librerías

*Ilustración 64 Librerías para el modelo*

```
In [1]: 1 import pandas as pd
        2 import numpy as np
        3 import matplotlib.pyplot as plt
        4 import tensorflow as tf
        5 from sklearn.preprocessing import MinMaxScaler
        6 from sklearn.metrics import mean_squared_error
        7 from tensorflow.keras import regularizers
```

Esta parte importa las librerías para hacer la construcción del modelo de red neuronal para hacer la predicción de series de tiempo utilizando TensorFlow y Keras. F7

Las librerías necesarias son:

pandas y numpy: manejo de datos

matplotlib: graficar

tensorflow y keras: construir el modelo de red neuronal

MinMaxScaler: normalizar los datos

mean\_squared\_error: calcular error

regularizers de keras: regularización

## 2.1.2 cargue y procesamiento de los datos

*Ilustración 65 Cargar los datos de la serie de tiempo*

```

1 # Cargar Los datos de La serie de tiempo
2 data = pd.read_csv('DHU 2015 2022 FC.csv', sep=',')
3 data.set_index('DAY', inplace=True)
4 data.index.freq = 'D'
5 data = data['TS']
6 dataC = data.copy()
7 print(data.head(12))
8 data = np.array(data).reshape(-1, 1)
9 # Preprocesar Los datos
10 scaler = MinMaxScaler(feature_range=(0, 1))
11 scaled_data = scaler.fit_transform(data)

```

El código carga un conjunto de datos de serie de tiempo desde un archivo CSV que se define en la línea uno.

En la línea 5 se define la variable que se quiere predecir. Luego, se realiza un preprocesamiento de los datos, incluida la transformación de los datos a un rango específico utilizando MinMaxScaler. F8

*Ilustración 66 Dividir los datos en conjunto de entrenamiento y prueba.*

```

1 # Dividir Los datos en conjunto de entrenamiento y prueba
2 train_size = int(len(scaled_data) * 0.90)
3 train_data = scaled_data[:train_size]
4 test_data = scaled_data[train_size:]

```

Y la división de los datos en conjuntos de entrenamiento y prueba. F9

## 2.2 Creación de secuencias de entrenamiento

*Ilustración 67 Función para crear secuencias de entrenamiento.*

```

1 # Función para crear secuencias de entrenamiento
2 def create_sequences(data, seq_length):
3     X = []
4     y = []
5     for i in range(len(data) - seq_length):
6         X.append(data[i:i+seq_length])
7         y.append(data[i+seq_length])
8     return np.array(X), np.array(y)
9
10 # Definir la longitud de la secuencia de entrenamiento
11 seq_length = 15
12
13 # Crear secuencias de entrenamiento
14 X_train, y_train = create_sequences(train_data, seq_length)
15 X_test, y_test = create_sequences(test_data, seq_length)
16

```

La función `create_sequences` se utiliza para crear secuencias de entrada y salida para el modelo. Estas secuencias se utilizan para entrenar la red neuronal. F10

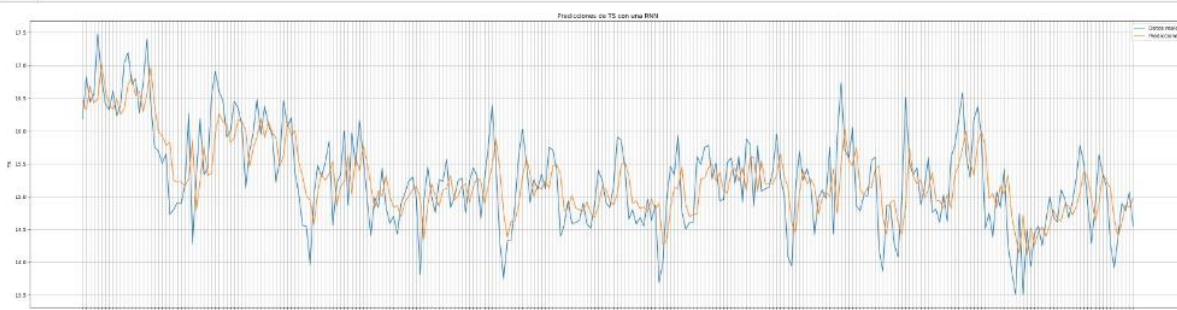
### 2.3 opciones de visualización

*Ilustración 68 Graficar resultados.*

```

1 # Graficar los resultados
2 from matplotlib.dates import DateFormatter
3
4 plt.figure(figsize=(40,10))
5 plt.plot(dataC.index[train_size+seq_length:], y_test, label='Datos reales')
6 plt.plot(dataC.index[train_size+seq_length:], predictions, label='Predicciones')
7 plt.xlabel('Fecha')
8 plt.xticks(rotation=90)
9 plt.grid(True)
10 plt.ylabel('TS')
11 plt.title('Predicciones de TS con una RNN')
12 plt.legend()
13
14 plt.show()
15

```



En esta figura se muestra una de las formas para ver los resultados de las predicciones, donde la línea color azul hace referencia a los datos reales y la línea color naranja visualiza los resultados a través de gráficos utilizando Matplotlib. F11

### 3. Realización de Predicciones.

#### 3.1 Configuración de Parámetros de Predicción

*Ilustración 69 Crear el modelo de red neuronal*

```

1 # Crear el modelo de red neuronal
2 model = tf.keras.Sequential([
3     tf.keras.layers.LSTM(64, input_shape=(seq_length, 1)),
4     tf.keras.layers.Dense(32, activation='relu'),
5     tf.keras.layers.Dense(1)
6 ])
7
8 # Compilar el modelo
9 model.compile(optimizer='adam', loss='mse')
10
11 # Entrenar el modelo
12 model.fit(X_train, y_train, epochs=30, batch_size=32)

```

Se creo un modelo de red neuronal secuencial utilizando `tf.keras.Sequential`.

El modelo consta de las siguientes capas:

- Una capa LSTM con 64 unidades y una forma de entrada definida por `input_shape=(seq_length, 1)` como se observa en la línea número 3 del bloque . Esto indica que el modelo aceptará secuencias de longitud `seq_length` con una dimensión.
- Una capa densa con 32 unidades y función de activación 'relu' lo que significa, unidad lineal rectificadora (línea número 4 del bloque).
- Una capa de salida densa con una sola unidad, que representa la salida final del modelo (línea número 5 del código).

Posterior mente a la creación de las capas, el modelo se compila utilizando el optimizador 'adam', que es un algoritmo de optimización popular basado en el descenso de gradiente estocástico.

Se especifica la función de pérdida 'mse' (mean squared error) para medir la discrepancia entre las predicciones y los valores reales. El uso de MSE es común en problemas de regresión.

El modelo se entrena utilizando el método fit de Keras como se muestra en la parte inferior de la imagen. F12

- X\_train y y\_train son los datos de entrada y las etiquetas de entrenamiento, respectivamente.
- Epochs=30 indica que el modelo se entrenará durante 30 épocas, lo que significa que el conjunto de datos completo se pasará al modelo 30 veces durante el entrenamiento.
- Batch\_size=32 indica el tamaño del lote utilizado para el entrenamiento. Aquí, se están utilizando lotes de tamaño 32.

### 3.2 Iniciar una Predicción

#### *Ilustración 70 Predicciones*

```

1 n = 10 # Cantidad de días a predecir
2
3 # Utiliza los últimos datos de prueba para inicializar la predicción
4 current_sequence = X_test[-1]
5
6 # Inicializar una lista para almacenar las predicciones
7 future_predictions = []
8
9 # Realizar predicciones para 'n' días hacia el futuro
10 for i in range(n):
11     # Predecir el siguiente valor basado en la secuencia actual
12     next_pred = model.predict(current_sequence.reshape(1, seq_length, 1))
13
14     # Agregar la predicción a la lista de predicciones futuras
15     future_predictions.append(scaler.inverse_transform(next_pred).item())
16
17     # Actualizar la secuencia actual para incluir la nueva predicción y eliminar el valor más antiguo
18     current_sequence = np.append(current_sequence[1:], next_pred)
19
20 # Imprimir las predicciones futuras
21 print("Predicciones futuras:", future_predictions)
22

```

En primera línea se Inicializa la variable `n`, la cual se refiere a la cantidad de días que se desea predecir en el futuro.

Seguido encontramos la Inicialización de `current_sequence`: Se toma la última secuencia de prueba (`X_test`) como punto de partida para la predicción de días futuros. Esto permite que el modelo haga una predicción basada en los datos de prueba más recientes.

A partir de la línea seis se muestra la Inicialización de una lista para almacenar las predicciones futuras. Se crea una lista vacía llamada `future_predictions` que se utilizará para almacenar las predicciones futuras generadas por el modelo.

El Bucle `for` de la línea número 10 para la predicción de '`n`' días en el futuro:

- Se itera '`n`' veces para predecir los siguientes valores basados en la secuencia actual.
- Se utiliza el modelo para predecir el siguiente valor, teniendo en cuenta la forma de entrada esperada del modelo.

- Se desescala la predicción utilizando el objeto scaler previamente definido y se agrega el valor desescalado a la lista de predicciones futuras.
- Se actualiza la secuencia actual para incluir la nueva predicción y eliminar el valor más antiguo.

Por último, imprime las predicciones generadas en la consola utilizando la instrucción Print. F13

#### 4. Interpretación de resultados

##### *Ilustración 71 Interpretación de resultados*

```
1
2 # Realizar predicciones en el conjunto de prueba
3 predictions = model.predict(X_test)
4
5 # Desescalar las predicciones y los datos de prueba
6 predictions = scaler.inverse_transform(predictions)
7 y_test = scaler.inverse_transform(y_test)
8
9 # Calcular el error cuadrático medio (MSE)
10 mse = mean_squared_error(y_test, predictions)
11 print("Error cuadrático medio (MSE):", mse)
```

Este fragmento de código se utiliza para realizar predicciones en el conjunto de prueba y calcular el error cuadrático medio (MSE) entre las predicciones y los datos reales.

Se realiza predicciones en el conjunto de prueba utilizando el modelo ya entrenado sobre el conjunto de prueba (`X_test`) y almacena estos resultados en la variable `predictions` como se observa en la línea tres.

Luego utilizando el objeto `scaler` que se encuentran en la línea seis y siete, desescala las predicciones y los datos de prueba.

Por último calcula el error cuadrático medio (MSE) con la función

`mean_squared_error` de scikit-learn (línea 10 del código), para calcular el error cuadrático medio entre los datos reales de prueba (desescalados) y las predicciones del modelo (desescaladas). El MSE es una métrica comúnmente utilizada para medir la calidad de las predicciones en problemas de regresión e imprime el valor del MSE en la consola utilizando la función `print` como se observa en la última línea del bloque. F14

