UNIVERSIDAD NACIONAL DE CHIMBORAZO
FACULTAD DE INGENIERÍA
CARRERA DE TELECOMUNICACIONES

Diseño de un modelo de predicción de temperatura en la provincia de Chimborazo basado en técnicas de Machine Learning.

Trabajo de Titulación para optar al título de
INGENIERO EN TELECOMUNICACIONES

Autor:
Coronel Mendoza, Joan Mauricio

Tutor:
PhD. Luis Patricio Tello Oquendo

Riobamba, Ecuador 2026

DECLARATORIA DE AUTORÍA

Yo, Joan Mauricio Coronel Mendoza, con cédula de ciudadanía 0704432814, autor del trabajo de investigación titulado: Diseño de un modelo de predicción de temperatura en la provincia de Chimborazo basado en técnicas de Machine Learning, certifico que la producción, ideas, opiniones, criterios, contenidos y conclusiones expuestas son de mí exclusiva responsabilidad.

Asimismo, cedo a la Universidad Nacional de Chimborazo, en forma no exclusiva, los derechos para su uso, comunicación pública, distribución, divulgación y/o reproducción total o parcial, por medio físico o digital; en esta cesión se entiende que el cesionario no podrá obtener beneficios económicos. La posible reclamación de terceros respecto de los derechos de autor (a) de la obra referida, será de mi entera responsabilidad; librando a la Universidad Nacional de Chimborazo de posibles obligaciones.

En Riobamba, 05 de diciembre de 2025.

Joan Mauricio Coronel Mendoza
C.I: 0704432814

DICTAMEN FAVORABLE DEL PROFESOR TUTOR

Quien suscribe, **PhD. Luis Patricio Tello Oquendo** catedrático adscrito a la **Facultad de Ingeniería**, por medio del presente documento certifico haber asesorado y revisado el desarrollo del trabajo de investigación titulado: "**Diseño de un modelo de predicción de temperatura en la provincia de Chimborazo basado en técnicas de machine Learning"**, bajo la autoría de **Joan Mauricio Coronel Mendoza**; por lo que se autoriza ejecutar los trámites legales para su sustentación.

Es todo cuanto informar en honor a la verdad; en Riobamba, a los 05 días del mes de Diciembre de 2025

Firmado electrónicamente por:
LUIS PATRICIO TELLO OQUENDO
Validar únicamente con FirmaEC

PhD. Luis Patricio Tello Oquendo
C.I: 0604235242

CERTIFICADO DE LOS MIEMBROS DEL TRIBUNAL

Quienes suscribimos, catedráticos designados Miembros del Tribunal de Grado para la evaluación del trabajo de investigación **Diseño de un modelo de predicción de temperatura en la provincia de Chimborazo basado en técnicas de Machine Learning** por **Joan Mauricio Coronel Mendoza**, con cédula de identidad número **070443281-4**, bajo la tutoría de **PhD. Luis Patricio Tello Oquendo**; certificamos que recomendamos la APROBACIÓN de este con fines de titulación. Previamente se ha evaluado el trabajo de investigación y escuchada la sustentación por parte de su autor; no teniendo más nada que observar.

De conformidad a la normativa aplicable firmamos, en Riobamba 06 de enero de 2026.

Ing. Radicelli García Ciro Diego PhD
PRESIDENTE DEL TRIBUNAL DE GRADO

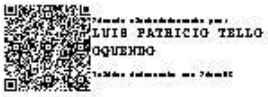
Dr. Klever Hernán Torres Rodríguez
MIEMBRO DEL TRIBUNAL DE GRADO

Mgs. José Luis Jinez Tapia
MIEMBRO DEL TRIBUNAL DE GRADO

C E R T I F I C A C I Ó N

Que, **Coronel Mendoza Joan Mauricio** con CC: **070443281-4**, estudiante de la Carrera **Telecomunicaciones**, Facultad de **Ingeniería**; ha trabajado bajo mi tutoría el trabajo de investigación titulado " **Diseño de un modelo de predicción de temperatura en la provincia de Chimborazo basado en técnicas de Machine Learning**", cumple con el 3% de similitud y 10% de Inteligencia Artificial, de acuerdo con el reporte del sistema Anti plagio **Compilatio Magister+**, porcentaje aceptado de acuerdo a la reglamentación institucional, por consiguiente autorizo continuar con el proceso.

Riobamba, 16 de diciembre de 2025



PhD. Luis Patricio Tello Oquendo
TUTOR

DEDICATORIA

A mi amado hijo Joan, quien llegó a mi vida en uno de los momentos más significativos y transformadores. Culminar este trabajo ha sido una meta importante, pero tu llegada ha marcado, sin duda, el comienzo de la etapa más hermosa de todas.

Tu nacimiento llenó mi corazón de esperanza, fortaleza y un amor que jamás imaginé. Y por esta razón esta dedicatoria es para ti, porque cada esfuerzo, cada desvelo y cada página escrita llevan el anhelo profundo de construir un futuro mejor en el que crezcas rodeado de amor, sabiduría y oportunidades. Eres y serás siempre mi mayor inspiración, mi motor y mi bendición más grande.

Con amor,
Papá

Joan Mauricio Coronel Mendoza

AGRADECIMIENTO

A Dios, por ser mi guía, mi fortaleza y mi refugio en cada etapa de este proceso. Gracias por darme la vida, por llenarme de sabiduría y por acompañarme incluso en los momentos más difíciles.

A mi papá, **Franklin Coronel**, por ser un ejemplo de esfuerzo, dedicación y firmeza. Gracias por enseñarme el valor del trabajo honesto y por darme siempre tu apoyo silencioso pero constante, que me ha impulsado a seguir adelante.

A mi mamá, **Patricia Mendoza**, por tu amor incondicional, por cada palabra de aliento, por tus oraciones y por estar siempre a mi lado, incluso cuando no lo decía en voz alta. Tu entrega y tu fe en mí han sido una fuerza invaluable.

A mi esposa, **Hermelinda Taisha**, por ser mi compañera en cada paso, por tu paciencia, tu comprensión y tu amor. Gracias por apoyarme cuando el cansancio me vencía y por creer en mí en todo momento. Este logro también es tuyo.

A mis hermanas **Mishel**, **Gerly y** a mi hermano **Frank**, por su cariño, sus palabras de motivación y por estar siempre presentes con su alegría y compañía. Gracias por ser parte de este proceso con tanto amor y entusiasmo.

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

RESUMEN

En el presente proyecto se diseñaron modelos ML para la predicción de temperatura utilizando datos de estaciones meteorológicas ubicadas en la provincia de Chimborazo. Para llegar al objetivo se utilizó información histórica del período 2013-2024 que contenía variables climáticas significativas como temperatura, radiación solar, velocidad del viento y presión atmosférica.

Se realizó un procedimiento de depuración de datos, suprimiendo registros que contenían valores inválidos en las variables esenciales. Además, se emplearon métodos de interpolación temporal para calcular los datos ausentes y, cuando los valores interpolados no eran coherentes, se utilizaron medias históricas por hora para preservar la coherencia de la serie temporal.

En la ingeniería de características se generaron nuevas variables para mejorar la capacidad predictiva de los modelos. Entre ellas se incluyeron variables de rezago considerando valores previos de temperatura a 1 y 24 horas, y variables cíclicas, transformando las variables de hora y mes mediante funciones seno y coseno para representar su naturaleza repetitiva.

Finalmente, se entrenaron y evaluaron tres modelos de regresión: Random Forest Regressor, Prophet y XGBoost. El desempeño de cada modelo se evaluó utilizando métricas como el MAE (Error Absoluto Medio) y el R^2 (Coeficiente de determinación).

Palabras claves: temperatura, predicción, aprendizaje automático, métricas de análisis.

ABSTRACT

In this project, ML models were designed for temperature prediction using data from meteorological stations located in the province of Chimborazo. To achieve this objective, historical data from the 2013–2024 period were used, which included significant climatic variables such as temperature, solar radiation, wind speed, and atmospheric pressure.

A data-cleaning procedure was performed, removing records with invalid values in the essential variables. In addition, temporal interpolation methods were used to estimate missing data; when interpolated values were inconsistent, hourly historical averages were applied to preserve the time series' coherence.

During the feature engineering stage, new variables were created to improve the models' predictive capacity. Among them were lag variables, using previous temperature values at 1 and 24 hours, and cyclic variables, transforming the hour and month using sine and cosine functions to capture their periodicity.

Finally, three regression models were trained and evaluated: Random Forest Regressor, Prophet, and XGBoost. The performance of each model was assessed using metrics such as MAE (Mean Absolute Error) and R^2 (Coefficient of Determination).

Keywords: Temperature, prediction, machine learning, analysis metrics.

Firmado electrónicamente por:
SOFIA FERNANDA
FREIRE CARRILLO
Validar únicamente con FirmaEC

Reviewed by:

Mgs. Sofía Freire Carrillo

ENGLISH PROFESSOR

C.C. 0604257881

CAPÍTULO I. INTRODUCCIÓN

Hoy en día, el cambio climático es una prioridad esencial para la humanidad, y tanto los sectores económicos como los sociales deben tomar medidas preventivas para evitar que afecte las diversas actividades humanas en los próximos años [1], esto ha despertado un interés creciente en la investigación y el desarrollo de modelos de predicción climática.

La temperatura constituye uno de los factores más relevantes, ya que afecta directamente a los ecosistemas, la agricultura, la disponibilidad de los recursos hídricos y, en general, sobre las condiciones de vida de la población [2]. En la provincia de Chimborazo, ubicada en la región central del Ecuador y caracterizada por su geografía montañosa, contar con predicciones de temperatura confiables resulta especialmente importante para la planificación territorial y la adaptación frente a los cambios climáticos. La combinación de su elevada altitud y su compleja ubicación geográfica genera una alta variabilidad climática, lo que dificulta la aplicación de los modelos meteorológicos tradicionales.

Ante este contexto, el desarrollo de modelos de predicción de temperatura ajustados a las particularidades de Chimborazo puede beneficiarse del uso de técnicas de Machine Learning, las cuales han demostrado resultados favorables en regiones montañosas de características similares. Un caso representativo es el empleo de algoritmos como Random Forest y Redes Neuronales Artificiales (ANN) en estudios realizados en la cuenca del río Xiying, en China, donde se lograron predicciones precisas de temperatura y procesos de deshielo mediante la integración de datos de teledetección y reanálisis atmosférico [3].

En los últimos años, las técnicas de Machine Learning han ganado relevancia en diversos campos del conocimiento, y su aplicación en el ámbito climático está transformando la forma en que se analizan y proyectan las variables meteorológicas [4]. A diferencia de los métodos convencionales, que suelen basarse en modelos físicos complejos y demandan altos recursos computacionales, estos enfoques permiten identificar patrones directamente a partir de datos históricos, capturando relaciones no lineales entre múltiples variables. Entre los métodos más utilizados se encuentran la regresión lineal, los árboles de decisión, las redes neuronales y los modelos de tipo ensemble, cada uno con diferentes niveles de precisión y capacidad de adaptación a escenarios complejos [5].

En este contexto, la presente investigación tiene como objetivo principal diseñar un modelo de predicción de temperatura para la provincia de Chimborazo, utilizando técnicas de Machine Learning como una alternativa moderna y eficiente frente a los modelos climáticos tradicionales. Los resultados obtenidos buscan aportar información útil para la gestión de los efectos de la variabilidad climática, beneficiando a la población local y sentando bases metodológicas que puedan ser replicadas en otras regiones con condiciones geográficas y climáticas similares.

1.1 ANTECEDENTES

El cambio climático ha intensificado la frecuencia e intensidad de fenómenos meteorológicos extremos, generando impactos sin precedentes en distintas regiones del planeta. De acuerdo con el Grupo Intergubernamental de Expertos sobre el Cambio Climático *(IPCC),* si el calentamiento global alcanza los 1.5 °C en las próximas décadas, se prevé un aumento significativo en las olas de calor, una prolongación de las estaciones cálidas y una reducción de las estaciones frías. Asimismo, se anticipan alteraciones en los patrones de precipitación, así como un incremento en la frecuencia e intensidad de inundaciones y sequías [6].

Los efectos del cambio climático en Ecuador se reflejan en el deshielo de los glaciares andinos, el aumento de la temperatura, las sequías, las inundaciones, el incremento del nivel del mar, los deslizamientos de tierras agrícolas, las pérdidas en la producción de cultivos y amenazas a la seguridad alimentaria y la biodiversidad [7]. A nivel provincial, uno de los impactos más alarmantes es el retroceso del glaciar del nevado Chimborazo, el más alto del país, el cual ha perdido entre un 38 % y un 42.5 % de su superficie en las últimas décadas. Este proceso ha disminuido de manera significativa el caudal de laderas y ríos que abastecen a comunidades rurales y urbanas, afectando el acceso al agua para riego, consumo humano y conservación ambiental [8].

Frente a esto, la comunidad científica ha comenzado a aplicar herramientas de análisis como el Machine Learning (ML) para predecir variables climáticas en lo cual, diversos estudios han demostrado que modelos como, **Random Forest** y **XGBoost** superan a los enfoques estadísticos tradicionales en la predicción de temperatura y precipitaciones [9].

1.2 PLANTEAMIENTO DEL PROBLEMA

La provincia de Chimborazo enfrenta una creciente vulnerabilidad climática, que se manifiesta en una combinación de fenómenos interrelacionados como la irregularidad en las precipitaciones, el incremento sostenido de la temperatura y la disminución del recurso hídrico debido al retroceso glaciar del nevado Chimborazo. Estas condiciones han generado efectos adversos en diversos sectores. En la agricultura, se han producido pérdidas significativas en los cultivos y alteraciones en los calendarios de siembra y cosecha, comprometiendo la seguridad alimentaria de la población rural [10].

La reducción del caudal de vertientes y acuíferos ha provocado problemas de abastecimiento de agua para el consumo humano y para el riego, afectando tanto a comunidades rurales como a centros urbanos. La infraestructura también se ha visto afectada por deslizamientos de tierra e inundaciones causadas por lluvias intensas, lo que ha generado daños en caminos rurales, viviendas y centros educativos, y ha incrementado el riesgo de desastres naturales. Como consecuencia de la disminución de las oportunidades agrícolas y la presión ambiental, muchas familias han optado por migrar, generando abandono de tierras, pérdida de saberes ancestrales y desestructuración comunitaria [11].

Ante esta problemática, surge la posibilidad de aplicar técnicas de Machine Learning para el diseño de un modelo de predicción de temperatura que sea capaz de identificar patrones en los datos climáticos de Chimborazo. La importancia de invertir a corto plazo en estrategias de mitigación y generación de información que prevean posibles escenarios futuros del clima en la provincia es esencial para fortalecimiento de nuestros sistemas de alerta temprana, permitiendo a autoridades y comunidades locales implementar medidas preventivas y a tomar decisiones anticipadas frente a desastres [4].

1.3 OBJETIVOS

1.3.1 General

- Diseñar un modelo de predicción de los patrones de temperatura mediante técnicas de Machine Learning para la provincia de Chimborazo.

1.3.2 Específicos

- Estudiar el estado del arte sobre modelos de predicción climática, evaluando las metodologías y algoritmos para una estimación precisa de la temperatura.
- Desarrollar un modelo de predicción de temperatura utilizando algoritmos de predicción climática.
- Evaluar el modelo mediante métricas de análisis utilizando datos de temperatura de la provincia de Chimborazo.

CAPÍTULO II. MARCO TEÓRICO

2.1 ESTADO DEL ARTE

En el artículo titulado **"South America Seasonal Precipitation Prediction by Gradient-Boosting Machine-Learning Approach",** los autores Vinicius Schmidt Monego, Juliana Aparecida Anochi y Haroldo Fraga de Campos Velho (2022) propusieron un modelo de predicción estacional de la precipitación sobre América del Sur utilizando el algoritmo XGBoost, una variante del método de aprendizaje automático Gradient Boosting. El objetivo principal del estudio fue comparar el rendimiento de este modelo con el de métodos tradicionales como el modelo numérico BAM (Brazilian Atmospheric Model) y una red neuronal profunda desarrollada en TensorFlow [12].

Los resultados obtenidos por los autores evidencian que el modelo basado en XGBoost, optimizado mediante la herramienta Optuna a través de técnicas de optimización bayesiana, alcanzó un mejor desempeño en términos de precisión predictiva a lo largo de distintas estaciones del año. En particular, durante los periodos de verano de 2018 y 2019, este modelo presentó valores de error (RMSE) inferiores en comparación con BAM y TensorFlow, lo que pone de manifiesto su capacidad para identificar de manera más precisa los núcleos de precipitación intensa en zonas estratégicas como el norte de Brasil, Colombia y Perú [12].

El estudio también resalta la relevancia de una adecuada selección de las variables meteorológicas, entre las que se incluyen la temperatura, la humedad, la presión atmosférica y las componentes del viento, así como del ajuste detallado de los hiperparámetros del modelo. Estos aspectos resultan determinantes para mejorar la capacidad de generalización de los algoritmos. En este sentido, se concluye que los métodos de aprendizaje automático, y en particular XGBoost, pueden superar a los modelos numéricos tradicionales en la predicción estacional de la precipitación, constituyéndose como una alternativa eficiente y robusta para el fortalecimiento de los sistemas de pronóstico climático en Sudamérica [12].

Por otro lado, el estudio titulado *"Temperature Prediction using Machine Learning Approaches"* presenta un análisis comparativo entre diversas técnicas de Machine Learning aplicadas a la predicción de la temperatura. Los resultados indican que modelos como la Regresión Lineal Múltiple (MLR), las Redes Neuronales Artificiales (ANN) y las Máquinas de Soporte Vectorial (SVM) ofrecen desempeños satisfactorios según el contexto de aplicación. De manera general, las ANN y las SVM sobresalen por su capacidad para modelar la naturaleza no lineal de las series temporales de temperatura. No obstante, las SVM muestran una ligera ventaja en términos de precisión cuando se analizan escenarios a escala global, mientras que las ANN, especialmente aquellas que incorporan memoria a corto y largo plazo (LSTM), resultan más adecuadas para predicciones de corto plazo, como las realizadas a nivel horario o diario [13].

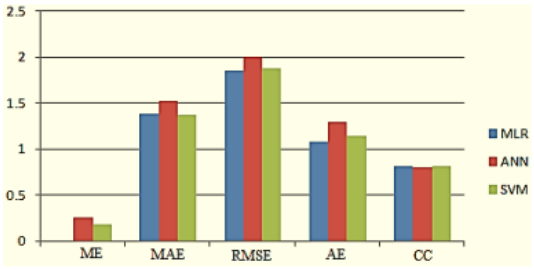


Figura 1. Comparación de la precisión de predicción de MLR, ANN y SVM [13].

Asimismo, el estudio señala que la variabilidad espacial, temporal y estacional presente en las series de temperatura representa un reto importante para los procesos de modelado. No obstante, los enfoques basados en Deep Learning, como las Redes Neuronales Convolucionales (CNN), han mostrado niveles de precisión elevados en determinados escenarios. Estos resultados ponen en evidencia la necesidad de seleccionar cuidadosamente tanto el tipo de modelo como sus parámetros, considerando el horizonte de predicción y el contexto en el que se aplican, ya sea a escala regional o global[13].

Por otro lado, en el artículo [14] titulado ***"Machine Learning in Weather Prediction and Climate Analyses: Applications and Perspectives"*,** se examina el papel de las técnicas de aprendizaje automático en la predicción del tiempo y el análisis climático. Los autores destacan que el uso de Machine Learning, y en especial de las redes neuronales artificiales y los enfoques de aprendizaje profundo, ha aumentado de manera significativa en los últimos años, debido a su capacidad para mejorar la precisión de los pronósticos meteorológicos. En este trabajo se analiza la aplicación de algoritmos como Random Forest (RF), XGBoost (XGB) y Support Vector Machines (SVM), los cuales se emplean cada vez con mayor frecuencia para complementar y optimizar los modelos numéricos de predicción del clima (NWP).

Entre los principales hallazgos del estudio se resalta la incorporación progresiva de técnicas de posprocesamiento en la predicción del viento, particularmente en investigaciones relacionadas con el sector de las energías renovables. Asimismo, se evidencia un creciente interés en la predicción probabilística mediante enfoques como el *ensemble forecasting*, así como en la aplicación de métodos orientados a la corrección de sesgos en variables climáticas como la temperatura y la presión atmosférica. Estas estrategias contribuyen a mejorar la precisión de los pronósticos tanto a corto como a largo plazo. En conjunto, los resultados del artículo reflejan el alto potencial de los métodos de Machine Learning para fortalecer las predicciones meteorológicas y climáticas, convirtiéndolas en herramientas más confiables y útiles para distintos sectores, entre ellos la gestión climática y la planificación energética [14].

2.2 FUNDAMENTO TEÓRICO

2.2.1 Clima

El clima de una región se entiende como el conjunto de condiciones atmosféricas que suelen presentarse habitualmente en ese lugar a lo largo de meses y años. Según lo establecido por la Organización Meteorológica Mundial *(OMM)* durante la Conferencia de Varsovia en 1935, el clima se define como el promedio de las variables meteorológicas mensuales y anuales, calculadas en un periodo de 30 años. Por ejemplo, cuando afirmamos que una zona tiene inviernos fríos y secos, nos referimos a lo que comúnmente ocurre en esa estación, sin descartar que pueda haber días con temperaturas agradables o niveles elevados de humedad.

Por otra parte, el tiempo meteorológico, al manifestarse a través de eventos puntuales o poco frecuentes, suele tener una incidencia limitada sobre el suelo y el relieve. En contraste, el clima, debido a su acción constante y prolongada en el tiempo, cumple un rol determinante en la configuración del paisaje, en los procesos de formación del suelo y en el desarrollo de la cobertura vegetal.

2.2.2 Temperatura

La temperatura del aire representa el nivel de energía térmica existente en la atmósfera en un instante determinado. Se trata de una de las variables meteorológicas más utilizadas para describir el estado del tiempo y, por lo general, se mide mediante termómetros. En la mayoría de los países se expresa en grados Celsius (°C), mientras que, en algunos, como Estados Unidos, se emplea la escala Fahrenheit (°F). Esta variable influye de manera directa en la sensación de frío o calor que experimentan las personas y constituye un indicador fundamental de las condiciones atmosféricas. El comportamiento de la temperatura no es uniforme, ya que depende de diversos factores ambientales, entre los que destacan la altitud, la radiación solar, la presencia de nubosidad y la dinámica del viento. Debido a esta interacción de elementos, la temperatura del aire adquiere un papel central en el análisis climático y tiene implicaciones directas en actividades como la agricultura, la salud humana y la planificación ambiental.[15].

2.2.3 Preprocesamiento de Datos

El preprocesamiento de datos es una etapa crítica en cualquier proceso de análisis o minería de datos. Consiste en preparar datos "en bruto" (no estructurados, incompletos o ruidosos) para convertirlos en datos útiles y limpios que permitan generar resultados confiables mediante técnicas de análisis [16].

2.2.4 Series de Tiempo

Una serie de tiempo es una secuencia de datos u observaciones medidos en determinados momentos, en intervalos iguales o desiguales, y ordenados cronológicamente. El análisis de series de tiempo se refiere al proceso de analizar los datos disponibles para descubrir el patrón o la tendencia en los datos. Permite extraer y modelar las relaciones entre datos a lo

largo del tiempo, sea extrapolando *(hacia futuro)* o interpolando *(hacia el pasado)* el comportamiento de datos no observados [17].

En el estudio de series temporales es frecuente trabajar con conjuntos de datos que no están completos y que presentan valores faltantes. Estas ausencias pueden originarse por diversas razones, entre ellas fallos en los sensores de medición, errores durante el registro de la información o interrupciones en los procesos de recolección de datos. Frente a esta situación, una de las estrategias más utilizadas es la interpolación, técnica que permite estimar los valores perdidos a partir de la información disponible en los registros anteriores y posteriores dentro de la serie temporal.

2.2.5 Interpolación

La interpolación en series de tiempo es un método estadístico que busca completar los vacíos en datos cronológicos continuos. Se basa en el supuesto de que los valores cercanos en el tiempo tienen cierta coherencia, por lo que es posible inferir los datos faltantes a partir de los ya conocidos.

Esta técnica cumple un rol esencial en el tratamiento de series temporales por varias razones:

- **Coherencia temporal:** Contribuye a mantener la coherencia temporal de los datos, ya que al estimar los valores ausentes se conserva la estructura original de la serie, incluyendo patrones de tendencia, estacionalidad o comportamiento cíclico.

- **Precisión en análisis**: Al contar con una serie más completa, se mejora la precisión del análisis, lo que se traduce en resultados más consistentes y confiables.

- **Aplicación de modelos:** Otro aspecto relevante es que muchos modelos estadísticos y algoritmos de aprendizaje automático requieren datos continuos y sin interrupciones; en este sentido, la interpolación facilita la aplicación de estas técnicas.

- **Disponer de información continua y confiable:** Respalda la toma de decisiones fundamentadas en áreas como la meteorología, la economía, la salud y la ingeniería.

2.1.1.1.Principales métodos de interpolación en series temporales:

Entre los métodos de interpolación más utilizados se encuentra la interpolación lineal, que asume una variación constante entre dos puntos conocidos y estima los valores faltantes mediante una línea recta. Si bien es sencilla de aplicar, puede resultar limitada cuando la serie presenta comportamientos no lineales. Otro enfoque es la interpolación del vecino más cercano, que asigna al dato ausente el valor del registro más próximo en el tiempo, siendo especialmente útil en series donde los cambios entre mediciones son poco frecuentes.

Por su parte, la interpolación polinómica emplea funciones matemáticas de mayor complejidad para ajustar los datos observados, lo que permite obtener estimaciones más precisas en series con variaciones más complejas. Finalmente, la interpolación mediante promedios móviles utiliza técnicas de suavizado para aproximar los valores faltantes, reduciendo la variabilidad local y facilitando el análisis de la tendencia general de la serie [18].

2.2.6 Machine Learning

El aprendizaje automático describe la capacidad de los sistemas para aprender a partir de datos de entrenamiento específicos del problema, con el fin de automatizar el proceso de construcción de modelos analíticos y resolver tareas asociadas. Es un campo interdisciplinario en la intersección de la estadística, la informática y la inteligencia artificial que utiliza datos históricos para descubrir patrones que pueden ser aplicados a datos nuevos, con el propósito de realizar predicciones o clasificaciones [19].

- **Un modelo**, en el contexto del aprendizaje automático, es una fórmula matemática que describe la relación entre varias variables.

Además, Machine Learning es una técnica de análisis de datos que automatiza la creación y el uso de modelos estadísticos, permitiendo a una computadora aprender de los datos. Esta disciplina se basa en la idea de que los sistemas pueden aprender, encontrar patrones y tomar decisiones con poca ayuda humana [20].

Una característica clave del ML es su naturaleza iterativa: a medida que los modelos reciben nuevos datos, pueden ajustarse por sí mismos. A diferencia de métodos anteriores, el ML se distingue por su habilidad para adaptarse a los cambios en los datos en tiempo real y aprender de sus propias acciones. Aunque el ML no es un concepto nuevo, ha ganado popularidad recientemente gracias a la gran cantidad y variedad de datos disponibles, el aumento del poder de cómputo a un costo más accesible y las mejoras en el almacenamiento de datos [20].
Los sistemas de Machine Learning pueden ser clasificados de acuerdo con la cantidad y tipo de supervisión que tienen durante su entrenamiento:

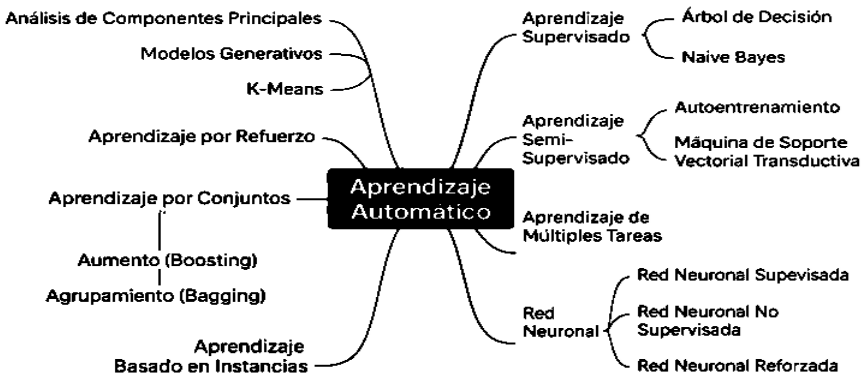


Figura 2. Tipos de aprendizaje que usan los algoritmos de Machine Learning.

2.2.7 Algoritmos ML

Un algoritmo de machine learning es un conjunto de reglas y pasos que un sistema de IA sigue para ejecutar tareas, habitualmente se usa para descubrir conocimiento y patrones en los datos o predecir valores de salida a partir de ciertas variables de entrada. En esencia, son los algoritmos los que permiten que el modelo "aprenda" a partir de los datos [21].

- **Aprendizaje Supervisado**

En el aprendizaje supervisado, el algoritmo construye un modelo a partir de un conjunto de datos etiquetados que contiene entradas y salidas asociadas; durante el entrenamiento recibe ejemplos con la salida esperada, compara sus predicciones con los valores correctos, calcula el error y ajusta el modelo para reducirlo, por lo que se utiliza cuando los datos históricos permiten anticipar resultados futuros.
Dentro de este marco, las tareas principales son la clasificación y la regresión; también se emplean procedimientos de predicción y técnicas de potenciación del gradiente *(gradient boosting)*; entre los algoritmos supervisados más habituales figuran k-vecinos más cercanos, Naïve Bayes y los árboles de decisión; la Figura 1 muestra de forma esquemática diversos métodos de aprendizaje supervisado [22].

- **Algoritmo de Clasificación**

  En un algoritmo de clasificación se pretende identificar a qué clase pertenece cada elemento; el modelo aprende patrones a partir de los datos proporcionados y, cuando recibe ejemplos nuevos, los asigna al grupo correspondiente, con lo cual puede predecir su categoría [23].

  La variante objetivo es categórica puede ser:
  - Binaria (Sí/No, Azul/Rojo, Fuga/No fuga),
  - Multiclase (Producto1, Producto2, …)
  - Ordinal (Riesgo: Bajo, Medio, Alto).

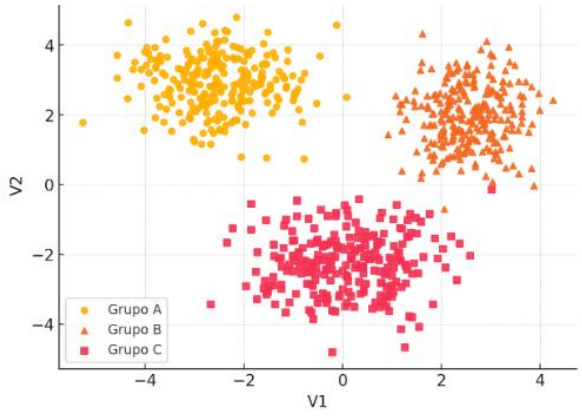


Figura 3. Ejemplo de algoritmo de clasificación [23].

- **Algoritmo de Regresión**

En este método la salida esperada es un valor numérico; no asigna el caso a una clase, sino que estima una magnitud continua a partir de las variables de entrada. Por ejemplo, usando fecha, hora, altitud, humedad y viento, el modelo puede predecir la temperatura del aire para una hora determinada y devolver un valor concreto [23].

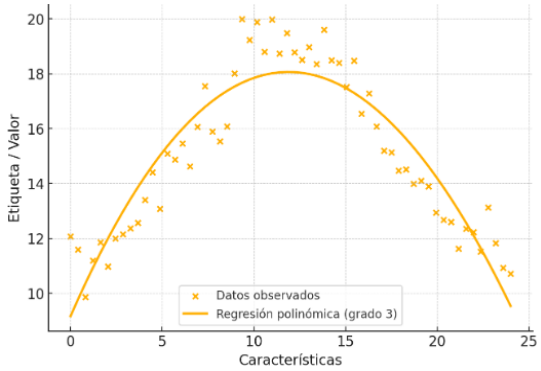


Figura 4. Ejemplo de algoritmo de clasificación [23].

➢ **Aprendizaje no Supervisado**

El aprendizaje no supervisado se emplea cuando se dispone únicamente de datos de entrada y no se cuenta con salidas previamente etiquetadas. En este enfoque, los algoritmos se encargan de identificar patrones ocultos y relaciones internas, con el propósito de describir y comprender la estructura subyacente de los datos.

Un método clave es el clustering, que identifica grupos naturales y permite asignar nuevos registros al grupo más probable, a partir del cual se pueden inferir comportamientos, un ejemplo típico es segmentar clientes y, con esos segmentos, anticipar su conducta de compra [24].

➢ **Aprendizaje en Conjunto**

El aprendizaje en conjunto o ensemble learning es una técnica de aprendizaje automático que entrena a múltiples modelos para resolver un mismo problema. La idea principal es que al combinar las predicciones de varios "aprendices" básicos, se logra un rendimiento superior al que podría obtener cualquiera de ellos por sí solo [24].

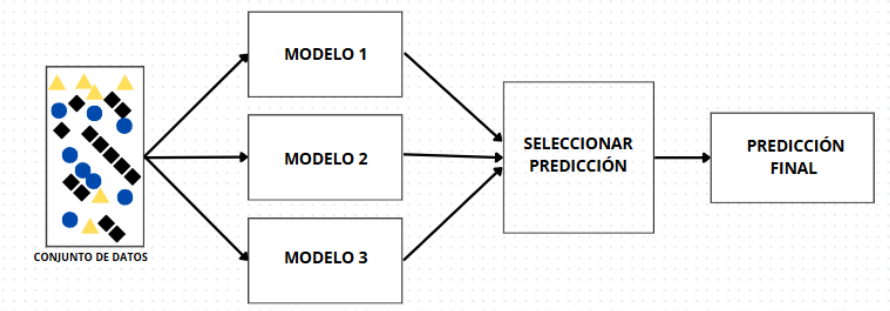


Figura 5. Resumen del aprendizaje en conjunto [24].

Dos de los tipos de ensemble más utilizados son:

- **Bagging**

Es una técnica de aprendizaje de conjunto que crea varios modelos. Lo hace generando múltiples subconjuntos de datos de entrenamiento a partir del conjunto original mediante muestreo aleatorio con reemplazo. Cada modelo se entrena de forma independiente en su propio subconjunto de datos. Las predicciones finales se obtienen combinando los resultados de todos los modelos individuales [25].

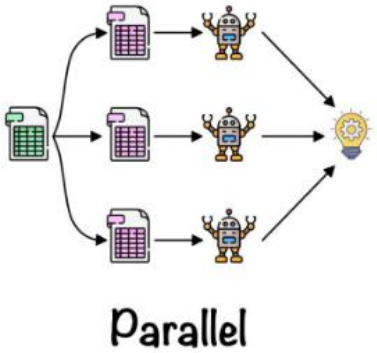


Figura 6. Bagging.- Procesamiento Paralelo de Múltiples Modelos [26].

- **Boosting**

Es una técnica de conjunto que busca construir un clasificador robusto a partir de una serie de clasificadores más débiles. Los modelos se entrenan de forma secuencial, donde cada modelo posterior corrige los errores del modelo anterior. Esto se logra ajustando los pesos de las muestras de datos: a las que fueron mal clasificadas por el modelo anterior se les asigna un peso mayor, para que el siguiente modelo se enfoque en ellas [25].

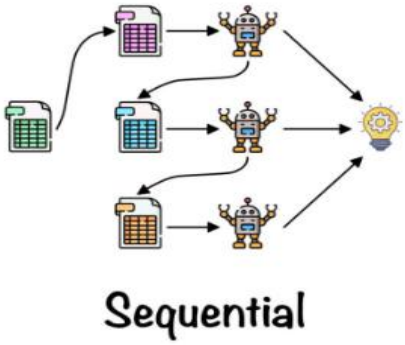


Figura 7. Boosting: Aprendizaje Secuencial y Mejoras Iterativas [26].

2.2.8 Modelos ML

Los algoritmos de Machine Learning, se pueden agrupar en tres modelos:

- **Modelos lineales**

Estos tratan de encontrar una línea que se "ajuste" bien a la nube de puntos que se disponen, aquí destacan desde los modelos muy conocidos y usados como la regresión lineal *(también*

conocida como la regresión de mínimos cuadrados), la logística *(adaptación de la lineal a problemas de clasificación cuando son variables, discretas o categóricas).* Estos dos modelos tienen el problema del "overfit", esto significa que se ajustan "demasiado" a los datos disponibles, con el riesgo que esto tiene para nuevos datos que pudieran llegar. Al ser modelos relativamente simples, no ofrecen resultados muy buenos para comportamientos más complicados [23].

- **Modelos de árbol**

Son modelos precisos, estables y más sencillos de interpretar básicamente porque construyen unas reglas de decisión que se pueden representar como un árbol. A diferencia de los modelos lineales, pueden representar relaciones no lineales para resolver problemas. En estos modelos, destacan los árboles de decisión y los random forest *(una media de árboles de decisión)*. Al ser más precisos y elaborados, obviamente ganamos en capacidad predictiva, pero perdemos en rendimiento [23].

- **Redes neuronales**

Las redes artificiales de neuronas tratan, en cierto modo, de replicar el comportamiento del cerebro, donde tenemos millones de neuronas que se interconectan en red para enviarse mensajes unas a otras. Esta réplica del funcionamiento del cerebro humano es uno de los "modelos de moda" por las habilidades cognitivas de razonamiento que adquieren. El reconocimiento de imágenes o videos, por ejemplo, es un mecanismo complejo y una red neuronal es lo mejor para realizarlo. El problema, como ocurre con el cerebro humano, es que son lentas de entrenar y necesitan mucha capacidad de cómputo. Quizás sea uno de los modelos que más ha ganado con la "revolución de los datos" [23].

2.2.9 Random Forest

Random Forest es un método de aprendizaje automático muy versátil y robusto, que se basa en la creación de un conjunto *(ensemble)* de múltiples árboles de decisión. Además, reduce la correlación entre los árboles de decisión que lo componen mediante el uso de dos tipos de aleatoriedad. En primer lugar, selecciona al azar un subconjunto de los datos de entrenamiento para crear cada árbol individual y, en segundo lugar, al construir cada árbol, elige aleatoriamente un subconjunto de características para determinar las mejores divisiones. Al utilizar estas dos estrategias, el modelo disminuye la dependencia entre los árboles, lo que ayuda a prevenir el sobreajuste y a mejorar la precisión general [27].

- **Random Forest Regressor**

Es una implementación del algoritmo Random Forest diseñada específicamente para problemas de regresión, es decir, para predecir valores continuos. En este modelo, se construye un "bosque" de árboles de decisión. Cada árbol se entrena con una muestra aleatoria del conjunto de datos, y en cada división de nodo, solo se considera un subconjunto aleatorio de predictores [28].

- **Proceso de Operación**

El funcionamiento del Random Forest Regressor se resume en los siguientes pasos:

- **Muestreo Aleatorio**: Se seleccionan muestras aleatorias con reemplazo del conjunto de datos original.
- **Construcción del Árbol**: Para cada muestra, se crea un árbol de decisión.
- **Votación y Promedio**: Las predicciones de cada árbol individual se promedian para obtener la predicción final del modelo.

2.2.10 XGBoost

XGBoost *(eXtreme Gradient Boosting)* es una técnica de aprendizaje automático de conjuntos de árboles de decisión que destaca por su escalabilidad y eficiencia. Funciona de manera similar al método de impulso de gradiente tradicional, construyendo de forma aditiva una serie de modelos *(árboles de decisión)* para minimizar una función de pérdida [29].

$$\mathbf{L_{xgb}} = \sum_{\mathbf{i=1}}^{\mathbf{N}} \mathbf{L}\left(\mathbf{y_i}, \mathbf{F}(\mathbf{X_i})\right) + \sum_{\mathbf{m=1}}^{\mathbf{M}} \boldsymbol{\Omega}(\mathbf{h_m}) \quad (1)$$

- $\sum_{i=1}^{N} L\left(y_i, F(X_i)\right)$ es el término de pérdida del entrenamiento (cuán bien predice el modelo).
- $\sum_{m=1}^{M} \Omega(h_m)$ es el término de regularización, que penaliza la complejidad del modelo.

$$\Omega(\mathbf{h}) = \boldsymbol{\gamma}\mathbf{T} + \frac{\mathbf{1}}{\mathbf{2}}\boldsymbol{\lambda}||\mathbf{w}||^{\mathbf{2}} \quad (2)$$

- γ es un parámetro que controla la ganancia mínima necesaria para una división. Un valor más alto de γ hace los árboles más simples.
- T es el número de hojas en el árbol.
- λ (lambda) es otro parámetro de regularización para la fuerza de la regularización.
- w son las puntuaciones de salida de las hojas del árbol.

XGBoost utiliza varias técnicas para mejorar el rendimiento, tanto para evitar el sobreajuste como para acelerar el entrenamiento. Una de las estrategias clave es la aleatorización. Esto se logra tomando submuestras aleatorias de datos para entrenar cada árbol y seleccionando solo un subconjunto de columnas (variables) en los niveles de árbol y nodo [29].

Además de las mejoras en precisión, XGBoost se enfoca en hacer más eficiente el proceso de creación de árboles. La parte más lenta de este proceso es encontrar la mejor división para cada nodo. En lugar de escanear todos los posibles puntos de división para cada variable de

forma repetida, utiliza una estructura de datos especial basada en columnas donde la información ya está preordenada. Esto significa que cada variable solo necesita ordenarse una vez al inicio, lo que permite que la búsqueda de la mejor división se realice en paralelo [29].

Otro método que acelera el proceso es el uso de percentiles. En lugar de revisar todas las divisiones candidatas, XGBoost solo evalúa un subconjunto de ellas basándose en estadísticas agregadas. Este enfoque de muestreo de datos a nivel de nodo ya es una característica de los árboles CART tradicionales [29].

2.2.11 Prophet

Prophet, una herramienta desarrollada por el equipo de Ciencia de Datos de Facebook y liberada como software de código abierto, se ha consolidado como un método robusto para el pronóstico de series de tiempo. Su eficacia radica en su modelo aditivo, que descompone una serie temporal en componentes de tendencia, estacionalidad (anual, semanal y diaria) y efectos de días festivos. Este enfoque lo hace particularmente adecuado para datos con marcadas periodicidades y patrones estacionales.

Una de las ventajas clave de Prophet es su resistencia a problemas comunes como datos faltantes, cambios abruptos en la tendencia y valores atípicos, lo que lo convierte en una opción fiable para análisis de datos del mundo real, además, su arquitectura matemática, si bien inspirada en principios complejos como el teorema de representación de Kolmogorov-Arnold, se traduce en una implementación práctica y altamente interpretable. La descomposición aditiva permite aislar y analizar el impacto de cada componente, facilitando la comprensión del modelo.

Para abordar la complejidad computacional en el pronóstico a gran escala, Prophet utiliza la plataforma Stan, que optimiza el proceso de ajuste del modelo. A través de este mecanismo, el modelo estima sus parámetros de manera eficiente y proporciona resultados de pronóstico de forma rápida. En esencia, Prophet adapta el marco de los Modelos Aditivos Generalizados (GAM) para el pronóstico de series de tiempo, la ecuación de Prophet:

$$y(t) = g(t) + s(t) + h(t) + \varepsilon t \quad (3)$$

Donde g(t) captura la tendencia, s(t) la estacionalidad, h(t) el impacto de los días festivos y ϵt el término de error. Este enfoque permite una estimación iterativa y convergente de cada componente, logrando un pronóstico preciso y con una alta capacidad de interpretación [30].

2.2.12 Ingeniería de Características Cíclicas en la Modelación de Series de Tiempo

Las características cíclicas son un componente crucial en la predicción de series de tiempo, ya que representan patrones repetitivos u oscilaciones inherentes a los datos. Estos patrones, que se repiten en intervalos regulares, deben ser tratados con una ingeniería de características cuidadosa para que los modelos de aprendizaje automático puedan interpretarlos correctamente [31].

Debido a su naturaleza circular, no es recomendable introducir estas variables directamente como valores numéricos en un modelo. En su lugar, es necesario transformarlas en un formato que capture su comportamiento cíclico inherente. Entre las técnicas de codificación más comunes se encuentran:

- **Codificación One-Hot:** Este método es adecuado para características cíclicas que se pueden agrupar en categorías discretas, como los meses o las estaciones. Consiste en crear variables binarias que indican la presencia o ausencia de cada categoría, permitiendo al modelo discernir entre ellas de forma efectiva [31].
- **Codificación Trigonométrica:** Para variables continuas o periódicas como la hora del día o el día de la semana, se utilizan funciones trigonométricas como el seno y el coseno. Esta técnica proyecta la variable cíclica en un círculo unitario, preservando la continuidad del ciclo. Al generar solo dos nuevas características, este método resulta muy eficiente [31].
- **Funciones Base:** Para el tratamiento de variables cíclicas se recurre a transformaciones matemáticas que facilitan su análisis dentro de los modelos; algunas de las más empleadas son las funciones de Fourier, las gaussianas y las B-splines. En particular, las B-splines destacan por su capacidad de representar comportamientos no lineales de forma flexible, ya que dividen el dominio en tramos y ajustan polinomios que reflejan mejor la dinámica real de los datos.

  Mediante la aplicación de estas técnicas de codificación, las características cíclicas pueden integrarse de manera efectiva en un modelo predictivo, lo que le permite identificar y aprovechar los patrones recurrentes y valiosos presentes en las series de tiempo [31].

2.2.13 Variables Lag

En el análisis predictivo de series de tiempo, las variables de retraso *(o lag features)* son elementos clave para capturar la dependencia temporal en los datos, estas variables consisten en observaciones pasadas de una serie, utilizadas como predictores para los valores futuros de la misma. Al integrar esta información histórica, los modelos, tanto los tradicionales como los de aprendizaje automático, pueden identificar y aprovechar patrones que influyen en el comportamiento futuro de la serie [32].

Estas variables se definen por el valor de la serie temporal en el pasado, indicado por el parámetro t, como ejemplo, un retraso de 1 toma el valor del punto temporal inmediatamente anterior, mientras que un retraso de 3 captura el valor de tres puntos temporales hacia atrás. Al ajustar el valor de t, es posible crear múltiples variables de retraso que incorporen información de diversos puntos en el pasado, lo que permite a los modelos utilizar la historia de los datos para la predicción [32].

Representada matemáticamente como:

$$y(t) = f(Y(t-1), (t-2), \dots, +Y(tn) \quad (4)$$

Donde el valor actual de una variable, Y(t), se modela como una función, f, de sus valores históricos, los valores pasados, representados por $Y(t-1), Y(t-2), \ldots, Y(t-n)$, se conocen como variables rezagadas o lags. La función f puede ser cualquier modelo estadístico o de aprendizaje automático, lo que permite capturar la dependencia temporal de la serie [32].

2.2.14 Variables Exógenas

Las variables exógenas en los modelos de predicción son factores externos que influyen en los resultados del modelo. Aunque estas variables no son parte del proceso que se está analizando, se las incluye para mejorar la precisión y la validez de las predicciones. En otras palabras, estas variables, al ser independientes, nos ayudan a considerar elementos adicionales que pueden afectar el modelo predictivo [33].

2.2.15 Métricas de Evaluación

- **Coeficiente de Determinación (R^2)**

Es una métrica que nos dice qué tan bien nuestro modelo explica la variabilidad de los datos, su valor ideal es 1, lo que significa que el modelo es perfecto, y puede ser tan bajo como menos infinito [34].

- **Error Cuadrático Medio (MSE)**

Mide el error promedio de las predicciones del modelo. Su valor ideal es 0 y se acerca a infinito cuanto peor es el modelo. El MSE es muy útil porque, al elevar los errores al cuadrado, da un peso mucho mayor a las predicciones muy malas.
Los resultados del R^2 y el MSE son inversos: un modelo con un MSE bajo tendrá un R2 alto, y viceversa. Por lo tanto, si usas cualquiera de estas métricas para ordenar el rendimiento de varios modelos, el ranking final será el mismo [34].

- **Raíz del Error Cuadrático Medio (RMSE)**

Es una métrica que se deriva del MSE (Error Cuadrático Medio). La principal ventaja del RMSE es que, al ser la raíz cuadrada del MSE, su valor se expresa en las mismas unidades que la variable que se está prediciendo. Esto lo convierte en una medida muy fácil de entender e interpretar para evaluar el error promedio de un modelo [34].

- **Error Absoluto Medio (MAE)**

Mide el error promedio, pero, a diferencia del RMSE, no penaliza tanto los errores grandes. Esto lo convierte en una mejor opción cuando los datos contienen valores atípicos o incorrectos, ya que el MAE no se verá tan afectado por ellos y proporcionará una medida de rendimiento más general y estable del modelo, sin embargo, si el conjunto de datos de prueba tiene muchos valores atípicos, el rendimiento del modelo puede parecer peor de lo que realmente es [34].

CAPÍTULO III. METODOLOGÍA

3.1 Tipo de Investigación

Este proyecto de investigación fue de tipo cuantitativo, ya que se basó en el análisis de datos numéricos provenientes de las estaciones meteorológicas en la provincia de Chimborazo, el trabajo implicó procesar y analizar grandes cantidades de datos; además, se evaluó el rendimiento de los modelos utilizando métricas numéricas. Este enfoque permitió cuantificar la precisión de las predicciones y comparar porcentualmente el desempeño de cada modelo.

Asimismo, se trató de una investigación aplicada, ya que su meta final no fue solo generar conocimiento teórico también buscó crear una solución práctica y funcional: un modelo de predicción de temperatura, con el propósito de aportar al monitoreo y análisis climático de la región.

3.2 Diseño de Investigación

El diseño de investigación fue no experimental observacional, según el grado de manipulación de variables por la razón de que no se intervinieron ni se manipularon las variables climáticas de la provincia; en lugar de eso, el trabajo consistió en observar y analizar los datos que ya existían.

Además, la construcción del modelo se basó en el comportamiento natural de estas variables a lo largo del tiempo, lo que estableció un diseño longitudinal, sin ninguna alteración o control por parte del investigador.

3.3 Técnicas de recolección de Datos

3.3.1 Revisión Bibliográfica

Se llevó a cabo una revisión profunda de literatura científica y técnica, con el objetivo de comprender los fundamentos teóricos de la predicción de temperatura mediante técnicas de Machine Learning. A partir de esta revisión se analizaron investigaciones previas, enfoques metodológicos, algoritmos y variables comúnmente utilizadas en estudios de predicción climática, priorizando aquellos modelos que pueden aplicarse de manera efectiva a las condiciones y características de los datos de la provincia de Chimborazo.

El análisis de la información permitió definir una base sólida de conocimiento que sirvió para:

- Seleccionar los algoritmos más adecuados, tomando en cuenta las características de los datos meteorológicos disponibles para la provincia.
- Evaluar mediante métricas adecuadas que permitan analizar de forma clara el desempeño y la precisión de los modelos de predicción.

3.3.2 Recolección de Datos Meteorológicos

La recolección de los datos meteorológicos para el proyecto se llevó a cabo de manera indirecta, utilizando información proveniente de fuentes secundarias instrumentales. Los registros fueron proporcionados por la Escuela Superior Politécnica de Chimborazo a través del proyecto interinstitucional IDIPI-306, el cual posee una red de estaciones meteorológicas en ubicadas diferentes lugares de la provincia de Chimborazo.

Los registros fueron obtenidos en formato NetCDF, el cual es muy utilizado en meteorología y climatología por su capacidad para almacenar grandes cantidades de información estructurada en el tiempo y el espacio.

3.4 Población de estudio y tamaño de muestra

3.4.1 Población

La población de este estudio está conformada por la totalidad de los datos válidos registrados a lo largo de las 24 horas del día con intervalos de 1 hora por las estaciones meteorológicas de la provincia de Chimborazo desde el año 2013 a el año 2024, dando en su totalidad 167 050 datos validos con la cual se entrenó los modelos Machine Learning.

3.4.2 Muestra

No fue necesario calcular el tamaño muestral porque se trabajó con la totalidad de los datos que conforman la población, debido que para entrenar modelos de aprendizaje se necesita la mayor cantidad de información posible, esto permite que los modelos capten mayor la variabilidad y patrones de los datos, lo cual es fundamental para asegurar la precisión y fiabilidad de las predicciones de temperatura.

3.5 Operacionalización de las variables

En la tabla 1 se detallan las variables que intervinieron en la investigación, tanto dependiente como independiente, las cuales sirvieron como base para la implementación y evaluación de los modelos de Machine Learning propuestos.

Tabla 1: Variable dependiente y Variable independiente.

Variable Independiente	**Descripción**	**Indicador**
Tipo de algoritmo de predicción	Representa el modelo Machine Learning utilizada para realizar la predicción de la temperatura.	1. Random Forest 2. XGBoost 3. Prophet
Variable dependiente	**Descripción**	**Indicador**
Desempeño	Precisión o capacidad del modelo para realizar predicciones cercanas a los	Porcentaje %.

	valores reales de temperatura en la región estudiada.	

3.6 Métodos de análisis, y procesamiento de datos.

Se puede observar en la figura 8 las fases realizadas para completar la investigación, las cuales describen de manera secuencial el proceso metodológico seguido desde la recopilación y procesamiento de los datos hasta la implementación y evaluación del modelo de predicción de temperatura.

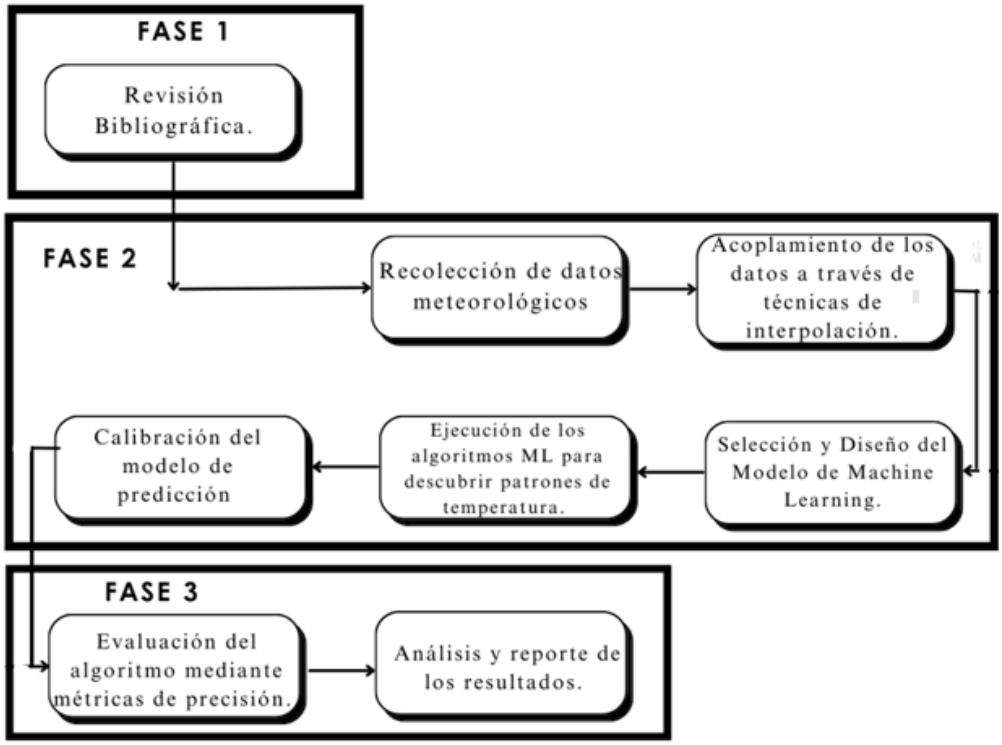


Figura 8. Fases de trabajo del proyecto de investigación.

3.7 Fase 1

3.7.1 Revisión Bibliográfica

En esta fase se realizó una revisión minuciosa de la literatura científica y técnica con el fin de conocer el estado actual de las investigaciones relacionadas con la predicción de temperatura mediante técnicas de Machine Learning. Se analizaron artículos científicos, trabajos de titulación y repositorios especializados, lo que permitió identificar tanto las metodologías más empleadas como las variables que han demostrado mayor relevancia en estudios previos sobre predicción climática.

A partir de la revisión, se seleccionaron tres modelos de aprendizaje automático:

- Random Forest

- XGBoost
- Prophet

La elección de los modelos se fundamentó en criterios técnicos relacionados con su capacidad para manejar datos meteorológicos, su precisión demostrada en la literatura y su robustez ante problemas comunes en series temporales, como valores faltantes y las relaciones no lineales.

A continuación, en la tabla 2 se presenta algunas de las ventajas detalladamente por la cuales se eligieron los tres modelos ya mencionados.

Tabla 2: Ventajas de modelos de Machine Learning.

Modelo	Ventajas principales
Random Forest	• Manejo adecuado de datos incluso con valores faltantes y ruido sin afectar significativamente la precisión. • Capaz de modelar comportamientos no lineales presentes en las variables. • Al promediar múltiples árboles de decisión reduce el sobreajuste. • facilita la comprensión del modelo a través del análisis de la importancia de las variables.
XGBoost	• Alta nivel de precisión y eficiencia computacional gracias a su optimización basada en gradiente. • Permite capturar relaciones complejas y con comportamiento no lineal. • Reduce el riesgo de sobreajuste utilizando regularización L1 y L2. • Logra un óptimo desempeño en datos tabulares con diversas características.
Prophet	• Es un modelo que se diseñó específicamente para predicciones temporales. • Identifica y modela automáticamente tendencias, estacionalidades y efectos externos. • Tolera datos faltantes y valores atípicos conservando su rendimiento. • Permite añadir variables internas como radiación solar, humedad, etc.

3.8 Fase 2

En la fase 2, el proceso de acoplamiento, diseño, calibración y ejecución se realizó para todas las estaciones meteorológicas incluidas en el estudio; sin embargo, con el fin de evitar una extensión excesiva del documento y mantener la claridad en la presentación, se tomó como estación de referencia aquella que presentó el mayor porcentaje de datos válidos, utilizando sus resultados como ejemplo representativo del procedimiento aplicado al conjunto total de estaciones.

Cabe recalcar que, a partir de esta fase el desarrollo se realizó en el entorno de programación Visual Studio Code, utilizando el lenguaje de programación Python por su versatilidad en el manejo de datos y su compatibilidad con librerías especializadas en Machine Learning.

3.8.1 Recolección de datos meteorológicos

En esta etapa se realizó la recolección de los datos meteorológicos la cual constituyó la base fundamental para el desarrollo del modelo de predicción. La información fue proporcionada por la Escuela Superior Politécnica de Chimborazo (ESPOCH), la cual cuenta con una red de estaciones meteorológicas repartidas en diferentes lugares de la provincia de Chimborazo.

Los datos se recibieron en formato NetCDF (.nc), el cual permite almacenar grandes cantidades de información climatológica estructurada en múltiples dimensiones, como tiempo, latitud, longitud y un total de 52 variables meteorológicas como se puede observar en la figura 9.

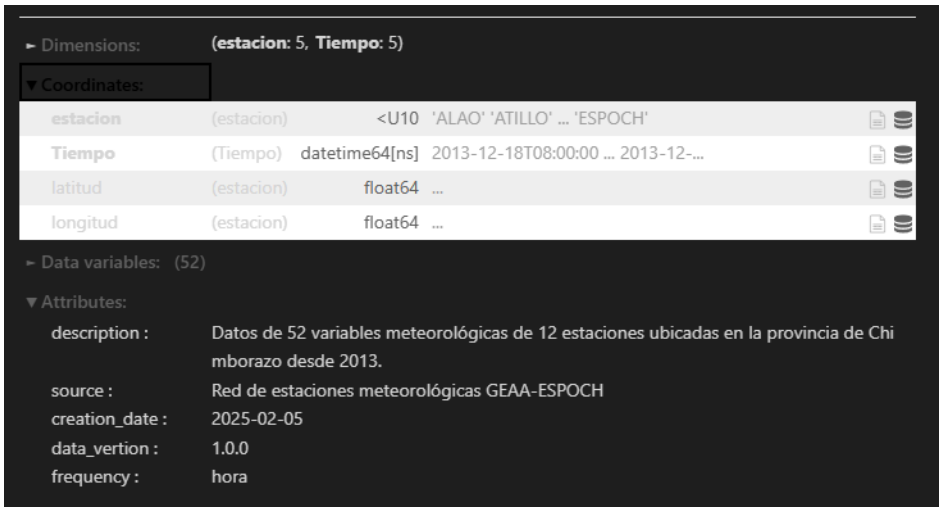


Figura 9. Archivo *NetCDF (.nc)* de los datos meteorológicos recibidos.

Posteriormente, se efectuó un control de las 52 variables meteorológicas obtenidas para la elección de las variables principales que ayudaron a lograr el objetivo de predecir la temperatura:

- temperatura promedio (TA Avg),
- presión atmosférica (PA Avg),
- humedad relativa (RH Avg),
- radiación solar global (SR_Glob Avg),
- velocidad promedio del viento (GenWind SpdAvg).

La elección de estas variables se dio por su influencia directa y comprobada en los procesos de variación térmica atmosférica; en conjunto, estos parámetros permitieron capturar los principales procesos termodinámicos y energéticos del ambiente, garantizando que los modelos de Machine Learning utilizados dispusieran de información representativa para realizar predicciones precisas y confiables de temperatura en la provincia.

Cada conjunto de datos se analizó para su posterior procesamiento, y así lograr obtener datos limpios para entrenar los modelos seleccionados previamente, garantizando la calidad y coherencia de la información.

3.8.2 Acoplamiento de los datos mediante interpolación

- **Detección de datos inválidos**

Se realizó una lectura para obtener los porcentajes de datos válidos que correspondían a todas las estaciones meteorológicas consideradas en esta investigación. Los resultados obtenidos se presentan en la Figura 10, donde se visualiza el porcentaje de datos válidos por estación, lo cual fue fundamental para determinar la necesidad de aplicar técnicas de interpolación y limpieza antes de entrenar los modelos de predicción.

	estacion	TA Avg	PA Avg	RH Avg	SR_Glob Avg	GenWind SpdAvg	Total Promedio %
0	ALAO	84.31	84.31	84.31	84.31	84.31	84.31
1	ATILLO	84.23	84.23	84.23	84.23	84.23	84.23
2	CHOCAVI	18.68	18.68	18.68	18.68	18.68	18.68
3	CUMANDA	63.84	63.84	63.84	63.84	63.84	63.84
4	ESPOCH	94.56	94.56	94.56	94.11	94.56	94.47
5	GUARGUALLA	2.83	2.83	2.83	2.83	2.83	2.83
6	MATUS	83.26	83.26	83.26	83.26	83.26	83.26
7	MULTITUD	85.34	85.34	85.34	85.34	85.34	85.34
8	QUIMIAG	80.46	80.46	80.46	80.46	80.46	80.46
9	SAN JUAN	88.65	88.65	88.65	88.65	88.65	88.65
10	TIXAN	90.46	90.46	90.46	90.46	90.46	90.46
11	TUNSHI	73.98	73.98	73.98	73.98	73.98	73.98
12	URBINA	68.91	68.91	68.91	68.91	68.91	68.91

Figura 10. Porcentajes de los datos meteorológicos por estación.

Gracias a los porcentajes de datos validos obtenidos, se tomó la decisión de descartar la estación GUARGUALLA y la estación CHOCAVI por tener muy poca información valida.

De igual manera se realizó una lectura para obtener el porcentaje de datos validos de las variables por año de cada estación como se observa en la Figura 11.

	TA Avg	PA Avg	RH Avg	SR_Glob Avg	GenWind SpdAvg
año					
2013	100.000000	100.000000	100.000000	57.317073	99.390244
2014	93.778539	93.778539	93.778539	60.171233	93.458904
2015	98.082192	98.082192	98.082192	64.783105	97.705479
2016	99.965847	99.965847	99.965847	66.427596	99.943078
2017	95.353881	95.353881	95.353881	65.684932	94.988584
2018	80.810502	80.810502	80.810502	54.417808	46.118721
2019	99.988584	99.988584	99.988584	67.123288	99.977169
2020	100.000000	100.000000	99.943078	51.571038	100.000000
2021	98.401826	98.401826	89.406393	55.057078	18.835616
2022	90.582192	90.593607	74.737443	47.363014	0.000000
2023	72.819635	86.449772	65.228311	43.949772	0.000000
2024	100.000000	100.000000	100.000000	57.302296	0.000000

Figura 11. Porcentajes de datos validos anual estación ESPOCH.

Cabe mencionar que este proceso se realizó para todas las estaciones meteorológicas y así obtener información más precisa para su procesamiento.

- **Interpolación**

En primer lugar, se desarrolló un script para eliminar todas las filas que tenían un valor NaN o inválido en la columna de temperatura TA Avg. Este proceso se realizó porque no se recomienda forzar interpolación en la variable objetivo; además, garantiza que el modelo solo use datos reales y confiables, evitando introducir valores artificiales que podrían sesgar la predicción.

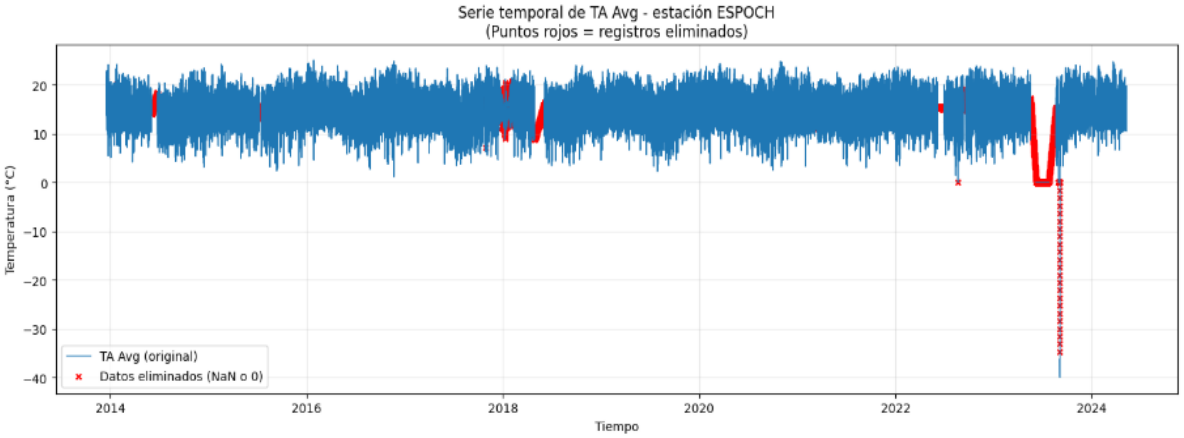


Figura 12. serie temporal de la temperatura promedio del aire (TA Avg) registrada por la estación ESPOCH durante el periodo 2013–2024.

En la Figura 12 se observa la línea azul que representa los datos válidos de temperatura, mientras que los puntos rojos corresponden a los registros eliminados por contener valores inválidos o físicamente inconsistentes de la estación ESPOCH durante el período 2013-2024.

Una vez realizado el paso anterior se procedió a interpolar los valores faltantes de las demás variables, en este caso se utilizó `interpolate(method='time'),` el cual nos sirvió para rellenar valores (NaN) en una serie temporal usando la información del tiempo como referencia.

En la Figura 13 se observa la evolución temporal de las variables de Radiación Global y Humedad Relativa, donde los valores reales e interpolados se representan mediante una línea celeste y verde respectivamente, mientras que los puntos rojos son una marcación visual para indicar exactamente en qué tiempo se aplicó la interpolación.

Figura 13. Evolución temporal de la radiación (SR_Glob Avg) y humedad relativa (RH Avg) con valores interpolados.

El método utilizado realizó la estimación de los valores ausentes en función del tiempo; es decir, utilizó la distancia temporal entre los registros válidos anteriores y posteriores al valor faltante para calcular un dato intermedio, asumiendo que el cambio entre ambos puntos ocurre de forma lineal y continua. De este modo, el método mantuvo la coherencia temporal de la serie evitando introducir saltos bruscos ni valores fuera del rango esperado.

No obstante, cuando los intervalos sin datos son extensos, la interpolación genera tramos lineales prolongados como se observa, por ejemplo, entre los años 2023 y 2024, los cuales no reflejan un comportamiento físico real de la radiación solar, sino una estimación matemática basada únicamente en los extremos

En la Figura 14 se muestra un diagrama de cajas comparativo que indica cómo fue el cambio de la distribución de las variables después de aplicar la interpolación temporal.

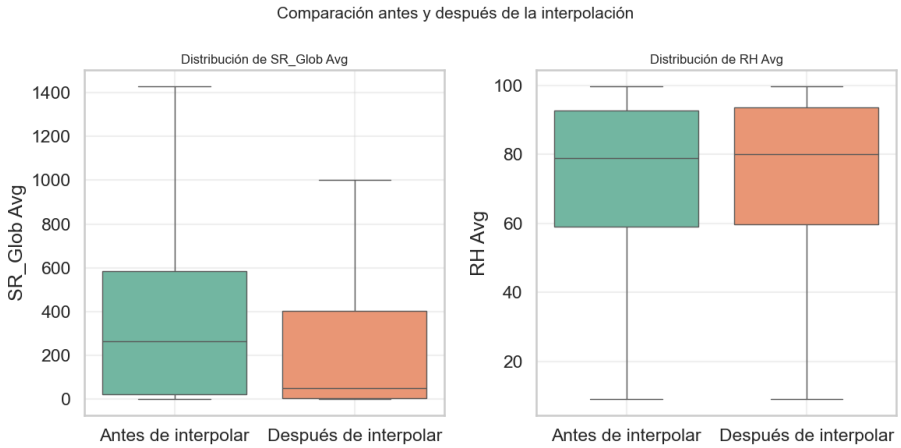


Figura 14. Efecto de la interpolación temporal en la distribución de los datos de las variables de Radiación Solar y Humedad Relativa.

SR_Glob Avg (Radiación Solar Global):

- **Antes de interpolar:**
  - Se observa una caja más alta y extendida, esto indica alta dispersión de valores y posiblemente huecos (NaN) que hacían variar mucho las estadísticas (mediana y cuartiles).

  - Los valores máximos llegan cerca de 1400 W/m², lo cual es típico en picos de radiación solar.

- **Después de interpolar:**

  - Ahora se observa que la caja tiene mayor compacidad y la mediana bajó un poco, esto ocurre porque la interpolación suavizó la variabilidad extrema, completando datos intermedios con valores intermedios y reduciendo los saltos.

  - El resultado es una señal más continua y con menos outliers, en otras palabras, los datos ahora representan mejor la tendencia promedio diaria de la radiación solar, sin grandes huecos.

RH Avg (Humedad Relativa Promedio)

- **Antes de interpolar:**
  - Se nota una distribución bastante estable la caja está alta y concentrada entre 70 % y 90 %, eso significa que había pocos vacíos o variaciones anormales.

- **Después de interpolar:**

  - Observamos que la caja casi no cambia, lo cual es muy bueno, esto significa que la interpolación no distorsionó la distribución original solo rellenó algunos vacíos sin alterar la estructura general de los datos.

  - La mediana (~80 %) se mantiene igual, lo que confirma una interpolación coherente.

Finalmente, después de haber realizado el proceso descrito se obtuvo un nuevo archivo procesado y depurado en extensión .csv como se muestra en la Figura 15, el cual constituyó la base principal para la siguiente fase del estudio: la selección, diseño y ejecución de los modelos de Machine Learning aplicados a la predicción de la temperatura en la provincia de Chimborazo.

	Tiempo	TA Avg	PA Avg	RH Avg	SR_Glob Avg	GenWind SpdAvg	hora	dia	mes	año
0	2013-12-18 08:00:00	13.844	729.122	74.868	245.090	0.000	8	18	12	2013
1	2013-12-18 09:00:00	15.462	729.141	65.533	423.202	0.562	9	18	12	2013
2	2013-12-18 10:00:00	17.386	729.057	54.983	749.175	1.602	10	18	12	2013
3	2013-12-18 11:00:00	19.478	728.352	45.664	945.447	1.625	11	18	12	2013
4	2013-12-18 12:00:00	21.115	727.407	40.346	1031.185	2.398	12	18	12	2013
5	2013-12-18 13:00:00	22.034	726.476	39.371	1018.196	3.578	13	18	12	2013
6	2013-12-18 14:00:00	22.926	725.487	35.747	828.595	5.406	14	18	12	2013
7	2013-12-18 15:00:00	18.405	725.909	45.521	79.187	3.398	15	18	12	2013
8	2013-12-18 16:00:00	16.249	726.143	56.756	66.434	1.117	16	18	12	2013
9	2013-12-18 17:00:00	16.421	725.946	59.715	141.241	0.992	17	18	12	2013

Figura 15. Modelo final de archivos .csv procesado.

3.8.3 Diseño y Calibración de los modelos Machine Learning

Anteriormente, en la Fase 1 se seleccionó 3 modelos ML para realizar las predicciones, a continuación, se detalla cómo se diseñaron y calibraron para cumplir el objetivo planteado.

Tabla 3: Proceso General de Construcción de los Modelos Predictivos.

N°	PASO	DESCRIPCIÓN
1	**Prepara datos**	Se carga datos, se convierte la columna de fecha y hora en tipo datetime , y se validan los valores.
2	**Generar variables**	Se crean variables temporales, cíclicas, lag y meteorológicas.
3	**Dividir dataset**	Separamos los datos para obtener 98% de valores para entrenar el modelo y el 0.02% para comparar con los datos predichos.
4	**Entrenar modelos**	Se entrena lo modelos RF, XGBoost y Prophet con sus hiperparámetros específicos de cada modelo.
5	**Predecir**	Se genera valores de temperatura predichos para obtener un conjunto de prueba.
6	**Evaluar el rendimiento**	Se calcula las métricas de error R^2 y MAE.
7	**Visualizar resultados obtenidos**	Se grafica valores reales vs predichos para comparar, además de generar tablas de datos las temperaturas reales y predichas.

Los modelos se estructuraron principalmente para predecir la temperatura promedio horaria (TA Avg) que fue considerada como variable objetivo a partir de un conjunto de variables meteorológicas y temporales previamente procesadas. Se seleccionó como variables predictoras:

- Factores temporales: hora, día, mes y año
- Presión atmosférica (PA Avg)
- Humedad relativa (RH Avg)
- Radiación solar global (SR_Glob Avg)

- Velocidad del viento (GenWind SpdAvg)

- **Variables cíclicas**

Para poder capturar patrones estacionales y cíclicos propios de los datos meteorológicos, se transformaron las variables de tiempo mediante funciones trigonométricas seno y coseno, obteniendo componentes como:

- hour_sin
- hour_cos
- month_sin
- month_cos

- **Variables Lag**

Adicionalmente, se incorporó variables de retardo (lag), que permiten representar la dependencia temporal de la temperatura con respecto a valores anteriores. En este caso, se crearon los siguientes retardos:

- Temperatura promedio con 1 y 24 horas de desfase (temp_lag1, temp_lag24).

- **Split temporal**

El Split temporal se creó porque no se puede dividir el conjunto de datos aleatoriamente, eso rompería el orden cronológico y haría que el modelo "viera el futuro" durante el entrenamiento
.
Por esa razón:

- El Split temporal separa los datos según la fecha, no al azar.

Después de varias pruebas realizadas se llegó a un punto de mejor rendimiento:

- Los datos más antiguos se usan para entrenar al modelo ➔ 98% de datos.
- Los datos más recientes se reservan para validar o probar el modelo ➔ 0.02% de datos.

Se debe mencionar que este proceso de diseño se realizó principalmente para los modelos Random Forest y XGBoost, para Prophet se diseñó con una metodología similar exceptuando pequeños cambios y definiciones.

a) Random Forest

El diseño del modelo predictivo de temperatura que utiliza un enfoque de aprendizaje supervisado basado en el algoritmo Random Forest Regressor, se implementó en el lenguaje de programación Python por medio de la librería scikit-learn, inmediatamente después del diseño se procedió a su respectiva calibración para encontrar los valores ideales para encontrar el mejor rendimiento del modelo.

- **Hiperparámetros**

Para el diseño se seleccionaron 3 hiperparámetros básicos con el fin de entrenar al modelo correctamente.

  - **n_estimators,** corresponde a la cantidad de árboles de decisión que van a conformar el bosque aleatorio.

  - **random_state,** con esto se asegura la repetibilidad de los resultados en n ejecuciones.

  - **n_jobs,** optimiza el tiempo de entrenamiento al aprovechar todos los núcleos disponibles del procesador .

Los hiperparámetros de Random Forest se establecieron a partir de pruebas preliminares realizadas.

- Se seleccionó n_estimators = 200 porque al incrementar el número de árboles de decisión (por ejemplo, 300 - 500) no mejoraron las métricas de evaluación de una manera relevante, pero en cambio el costo computacional aumento de manera notable, es decir, el tiempo de entrenamiento aumento significativamente.

- Se fijó random_state = 42 con el fin de garantizar la reproducibilidad de los resultados, considerando la aleatoriedad propia del algoritmo como el muestreo bootstrap y selección de atributos.

- Finalmente, n_jobs = -1 se emplea para aprovechar todos los núcleos disponibles del CPU , lo que permitió reducir el tiempo de entrenamiento sin comprometer la calidad del ajuste.

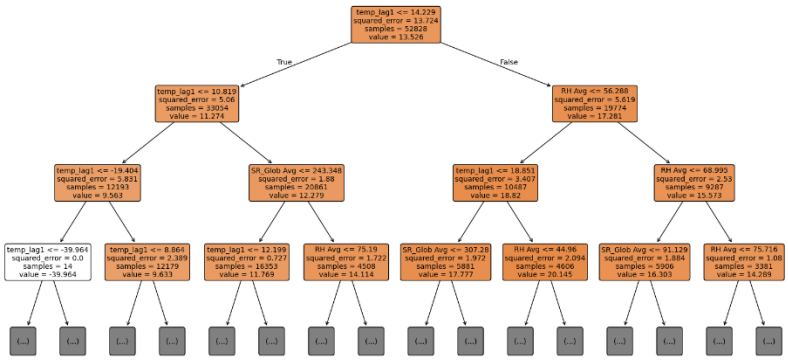


Figura 16. Árbol de decisión del modelo Random Forest (Profundidad 3 niveles).

Para realizar el ajuste del modelo se usó el método `fit()`, este método entrena el conjunto de árboles de decisión a partir de los datos de entrada, posteriormente se aplicó el método `predict()` con el fin de estimar los valores de temperatura correspondientes al conjunto de prueba.

b) XGBoost

Para la predicción de la temperatura promedio se diseñó un modelo de regresión basado en XGBRegressor.

- **Hiperparámetros**
  - **n_estimators = 200:** es el número de árboles que se crearon: proporciona estabilidad y buen rendimiento sin un costo computacional excesivo.
  - **learning_rate = 0.05:** Se adoptó un ritmo de aprendizaje moderado para garantizar una convergencia estable del modelo.
  - **max_depth = 8:** Este valor estableció una profundidad máxima de los árboles que permite modelar relaciones no lineales relevantes evitando un sobreajuste excesivo.
  - **subsample = 0.8, colsample_bytree = 0.8:** Se aplicó un muestreo de filas y columnas en cada árbol con el objetivo de reducir varianza y mejorar su capacidad de generalización.
  - **random_state = 42:** Valor de semilla fija para que se garantice la repetibilidad de los resultados en cada ejecución.
  - **n_jobs = -1:** paraleliza el entrenamiento usando todos los núcleos disponibles para acelerar la ejecución

Los valores de hiperparámetros seleccionados constituyen un compromiso entre robustez predictiva y costo computacional los valores seleccionados después de la calibración de n_estimators y learning_rate proporcionan suficiente capacidad de modelado con estabilidad; subsample y colsample_bytree reducen la varianza sin pérdida significativa de información y la semilla aleatoria y la paralelización se fijaron para reproducibilidad y eficiencia.

- **Entrenamiento y evaluación**
  - El modelo fue entrenado con (model.fit), aquí es donde ocurrió el aprendizaje ya que contiene las variables predictoras y la variable objetivo.

Durante el entrenamiento, XGBoost construye secuencialmente 200 árboles de decisión, cada nuevo árbol intenta corregir los errores de la combinación de árboles anteriores, siguiendo el principio de Gradient Boosting. El objetivo fue minimizar una función de pérdida (en regresión, generalmente el error cuadrático medio) ajustando las estructuras de los árboles.

- La predicción se llevó a cabo sobre el conjunto de prueba mediante el método `model.predict()`, el cual hace uso del modelo ya entrenado para estimar valores sobre un nuevo conjunto de datos.

La predicción de obtuvo a partir de la suma de las estimaciones generadas por los 200 árboles de decisión para cada instancia, las cuales fueron ponderadas por el learning_rate. Posteriormente se compara el resultado con la verdadera variable objetivo del conjunto de prueba para calcular métricas de error **(como el MAE o R^2).**

a) Prophet

En el diseño del modelo Prophet se incluyó variables meteorológicas adicionales y retardos temporales *(lags)* como regresores externos. Seguidamente, se detalla el proceso realizado para su diseño, la preparación de datos y también la calibración del modelo.

- **Estructura de Dato para Prophet**

El modelo requirió un formato de datos específico que incluyó determinadas columnas:

- **ds:** Corresponde a la marca temporal, es decir, el instante exacto en el que cada dato es registrado.
- **y:** Variable objetivo **(temperatura promedio)**

Además, se añadieron como regresores externos las variables meteorológicas relevantes:

- Humedad relativa (RH Avg),
- Radiación solar global (SR_Glob Avg),
- Velocidad del viento (GenWind SpdAvg),
- Presión atmosférica (PA Avg),
- Y las variables lag previamente generadas

- **Hiperparámetros**

```
yearly_seasonality=True,
weekly_seasonality=True,
daily_seasonality=True,
seasonality_mode='multiplicative',
changepoint_prior_scale=0.01
```

Figura 17. Calibración de hiperparámetros para Prophet.

Para calibrar el modelo Prophet, se evaluaron diversos valores de hiperparámetros con el fin de optimizar el desempeño predictivo en la estimación de la temperatura promedio. Como se observa en la figura 17:

- Se habilitaron las componentes anual, semanal y diaria debido a que las variaciones térmicas presentan comportamientos cíclicos en múltiples escalas temporales.

- Asimismo, se adoptó el modo de estacionalidad multiplicativo, ya que la amplitud de las oscilaciones de temperatura varía en función de la tendencia general; es decir, en periodos más cálidos las fluctuaciones son mayores y en periodos fríos, menores, comportamiento que el modo aditivo no reproduce adecuadamente.

- Finalmente, se definió el parámetro `changepoint_prior_scale = 0.01`, con el fin de restringir la flexibilidad de la tendencia y así reducir el riego de sobreajuste.

La configuración de los hiperparámetros seleccionados permitieron obtener un modelo estable, capaz de representar las principales variaciones estacionales y temporales de la temperatura, manteniendo al mismo tiempo un nivel adecuado de generalización en los datos de prueba.

CAPÍTULO IV. RESULTADOS Y DISCUSIÓN

4.1. RESULTADOS

4.1.1. Ejecución del modelo ML para descubrir patrones de temperatura

- **Random Forest**

En la Figura 18 se observa la comparación temporal entre la temperatura real y la temperatura predicha por el modelo Random Forest, como se observa ambas series mantienen una tendencia similar, lo que indica que el modelo logró capturar adecuadamente los patrones temporales de variación de la temperatura. Las diferencias puntuales pueden deberse a factores atmosféricos no contemplados en las variables de entrada o a la naturaleza aleatoria de las condiciones meteorológicas.

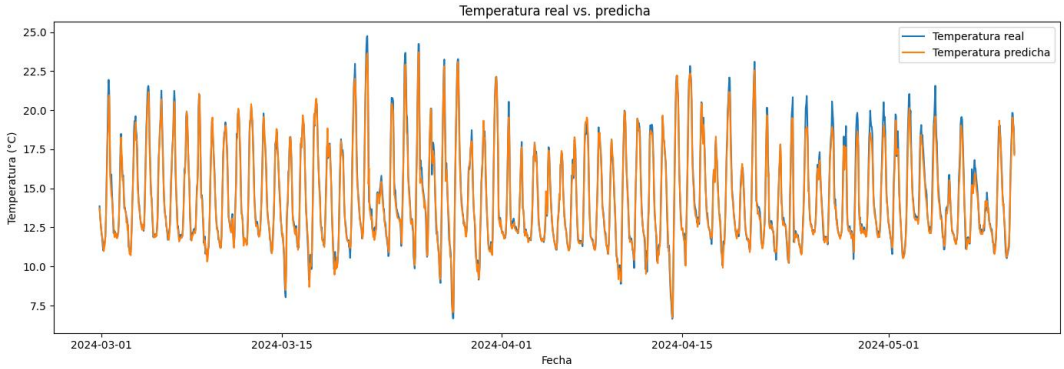


Figura 18. Gráfica de líneas temporales Temperatura Real vs Predicha del modelo Random Forest para la estación ESPOCH.

La Figura 19 se presenta una comparación directa entre los valores reales y los valores predichos por el modelo. El análisis permitió apreciar tanto la magnitud como la dirección de los errores, además de permitir evaluar la consistencia del modelo en distintos puntos del tiempo.

	Tiempo	temp_real	temp_predicha
1679	2024-05-09 20:00:00	14.555	14.592083
1680	2024-05-09 21:00:00	14.202	13.902289
1681	2024-05-09 22:00:00	13.850	14.021157
1682	2024-05-09 23:00:00	13.059	13.112792
1683	2024-05-10 00:00:00	12.480	12.669613
1684	2024-05-10 01:00:00	11.552	12.038648
1685	2024-05-10 02:00:00	10.937	11.356405
1686	2024-05-10 03:00:00	10.624	10.856029
1687	2024-05-10 04:00:00	10.529	10.597334
1688	2024-05-10 08:00:00	11.244	11.473945

Figura 19. Valores de temperatura reales vs. predichos del modelo Random Forest para la estación ESPOCH.

- **XGBoost**

La figura 20 muestra un alto nivel de rendimiento del modelo, ya que la línea de Temperatura real *(azul)* y la línea de Temperatura predicha por el modelo XGBoost *(naranja)* se superponen significativamente y siguen el mismo patrón de fuerte estacionalidad diaria *(picos de calor durante el día y valles de frío durante la noche*) a lo largo de los meses de marzo, abril y principios de mayo de 2024. Se observa que el modelo tiende a suavizar la curva, subestimando ligeramente los picos más altos y sobreestimando los valles más bajos, lo que indica una ligera falta de sensibilidad a los valores extremos y una predicción más conservadora de la volatilidad.

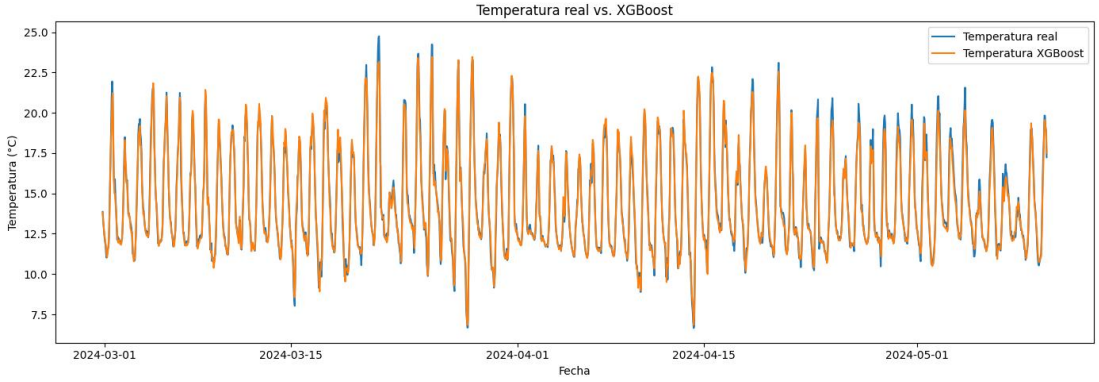


Figura 20. Gráfica de líneas temporales Temperatura Real vs Predicha del modelo XGBoost para la estación ESPOCH.

La tabla de datos que se observa en la figura 21, muestra una porción de las predicciones cuantifica la precisión del modelo en un día específico, revelando que el error absoluto es generalmente bajo. Se puede notar una subestimación consistente de la temperatura real durante las horas pico de calor, donde la diferencia entre el valor real y el predicho es máxima *(por ejemplo, a las 14:00 la temperatura real es 19.831 °C y la predicha es 19.2057 °C),* sin embargo, a las 18:00, el modelo sobreestima ligeramente la temperatura, prediciendo 17.4955 °C frente a un valor real de 17.239 °C, lo que sugiere que el modelo transita rápidamente de subestimar el pico a sobreestimar el descenso de la curva de temperatura.

	Tiempo	temp_real	temp_predicha
1689	2024-05-10 09:00:00	12.431	12.177548
1690	2024-05-10 10:00:00	14.264	14.068316
1691	2024-05-10 11:00:00	16.217	15.635718
1692	2024-05-10 12:00:00	17.586	16.726795
1693	2024-05-10 13:00:00	18.503	18.255857
1694	2024-05-10 14:00:00	19.831	19.205742
1695	2024-05-10 15:00:00	19.816	19.552197
1696	2024-05-10 16:00:00	19.148	19.065361
1697	2024-05-10 17:00:00	18.857	18.850014
1698	2024-05-10 18:00:00	17.239	17.495527

Figura 21. Valores de temperatura reales vs. predichos del modelo XGBoost para la estación ESPOCH.

- **Prophet**

La Figura 22 demuestra que el modelo Prophet es altamente efectivo para el pronóstico de esta serie de tiempo, ya que la línea de Temperatura predicha (naranja) se superpone y sigue muy de cerca a la línea de Temperatura real (azul) a lo largo de los tres meses, capturando la fuerte estacionalidad diaria con gran precisión, aunque el modelo logra ajustarse mejor a la magnitud de los picos y valles que el XGBoost, todavía presenta un ligero suavizado de las variaciones extremas. La principal fortaleza de Prophet reside en su capacidad para modelar la estructura temporal subyacente (tendencia y estacionalidad) de la serie, lo cual resulta en una predicción consistentemente cercana a los valores observados durante el período.

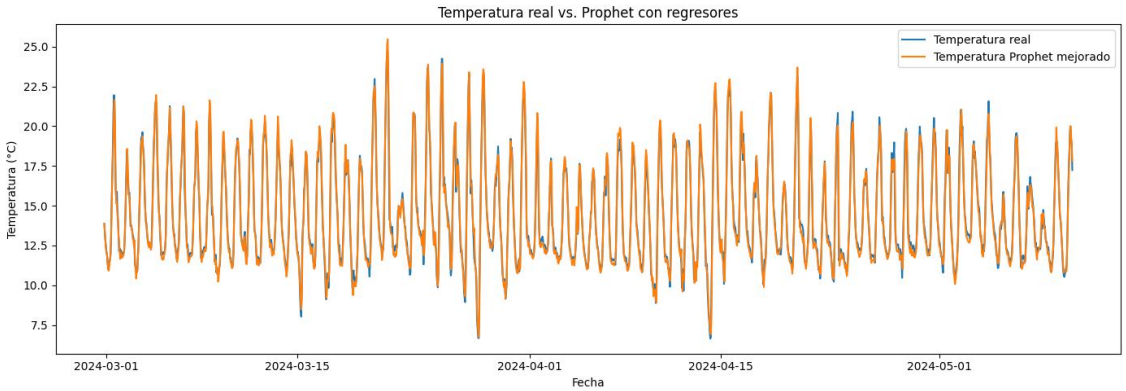


Figura 22. Gráfica de líneas temporales Temperatura Real vs Predicha del modelo Prophet para la estación ESPOCH.

Los valores que se presentan en la figura 23, que se enfoca en las predicciones, confirma la alta precisión del modelo, mostrando errores absolutos bajos, especialmente durante las primeras horas de la mañana (ej. 0.126 °C a las 9:00). No obstante, la tabla revela un cambio en el sesgo del modelo durante la tarde: el modelo tiende a subestimar ligeramente el pico de temperatura máxima (ej. 19.426 °C predicho vs. 19.831°C real a las 14:00), pero inmediatamente después comienza a sobreestimar la temperatura real, prediciendo una caída de temperatura más lenta de lo que ocurre en realidad (ej. 17.818 °C predicho vs. 17.239 °C real a las 18:00), lo que sugiere que Prophet podría estar demorando la predicción del descenso vespertino de la temperatura.

	fecha	temperatura_real	temperatura_predicha
1689	2024-05-10 09:00:00	12.431	12.304711
1690	2024-05-10 10:00:00	14.264	14.029973
1691	2024-05-10 11:00:00	16.217	15.841222
1692	2024-05-10 12:00:00	17.586	17.032775
1693	2024-05-10 13:00:00	18.503	18.508753
1694	2024-05-10 14:00:00	19.831	19.426049
1695	2024-05-10 15:00:00	19.816	20.009196
1696	2024-05-10 16:00:00	19.148	19.383066
1697	2024-05-10 17:00:00	18.857	18.893133
1698	2024-05-10 18:00:00	17.239	17.818779

Figura 23: Valores de temperatura reales vs. predichos del modelo Prophet para la estación ESPOCH.

4.1.2. Comparación de Modelos Machine Learning

Para evaluar el desempeño de los modelos de predicción de temperatura diseñados se emplearon métricas ampliamente utilizadas en predicciones climáticas o en series de tiempo: Coeficiente de Determinación (R^2) y Error Absoluto Medio (MAE). El R^2 cuantifica la proporción de la varianza en la variable predictora que es predecible a partir de las variables exógenas utilizadas para entrenar lo modelos, mientras que el MAE representa el promedio de las magnitudes de los errores en las unidades originales de la variable (temperatura en °C).

El rendimiento de cada algoritmo para las 11 estaciones se presenta en las Tablas 4, 5 y 6.

- **Random Forest**

Tabla 4: Métricas de Evaluación del Modelo Random Forest: Coeficiente de Determinación y Error Absoluto Medio.

Estación	R^2	MAE
Espoch	98%	0.313 °C
Alao	96%	0.559 °C
Atillo	92.2%	0.681 °C
Cumandá	97.2%	0.231 °C
Matus	95.6%	0.454 °C
Multitud	90.8%	0.383 °C
Quimiag	90%	0.753 °C
San Juan	96.7%	0.451 °C
Tixán	95%	0.605 °C
Tunshi	96.7%	0.422 °C
Urbina	96.9%	0.355 °C

- **XGBoost**

Tabla 5: Métricas de Evaluación del Modelo XGBoost: Coeficiente de Determinación y Error Absoluto Medio.

Estación	R^2	MAE
Espoch	98.3%	0.291 °C
Alao	96.5%	0.531 °C
Atillo	91.3%	0.691 °C
Cumandá	98%	0.200 °C
Matus	96.6%	0.405 °C
Multitud	90.9%	0.375 °C
Quimiag	87.8%	0.821 °C
San Juan	97.2%	0.417 °C
Tixán	96.1%	0.539 °C

Tunshi	97.2%	0.392 °C
Urbina	97.3%	0.325 °C

- **Prophet**

Tabla 6:Métricas de Evaluación del Modelo Prophet: Coeficiente de Determinación y Error Absoluto Medio.

Estación	R^2	MAE
Espoch	98.2%	0.298 °C
Alao	94.6%	0.652 °C
Atillo	89.8%	0.740 °C
Cumandá	96.9%	0.247 °C
Matus	97.1%	0.384 °C
Multitud	50.2%	0.568 °C
Quimiag	89.4%	0.696 °C
San Juan	96.3%	0.417 °C
Tixán	96%	0.548 °C
Tunshi	96.1%	0.456 °C
Urbina	95.1%	0.417 °C

Para determinar el modelo con el mejor desempeño para cada ubicación, se aplicó el siguiente criterio: se seleccionó el modelo con el MAE más bajo (precisión mínima de error) y con el R^2 más alto (máximo ajuste). Esto aseguró que el modelo elegido no solo sea el más preciso, sino también el que mejor explica la varianza de la temperatura.

La figura 24 resume visualmente esta clasificación final, ubicando cada una de las 11 estaciones en el círculo del modelo que les otorgó la mejor métrica combinada.

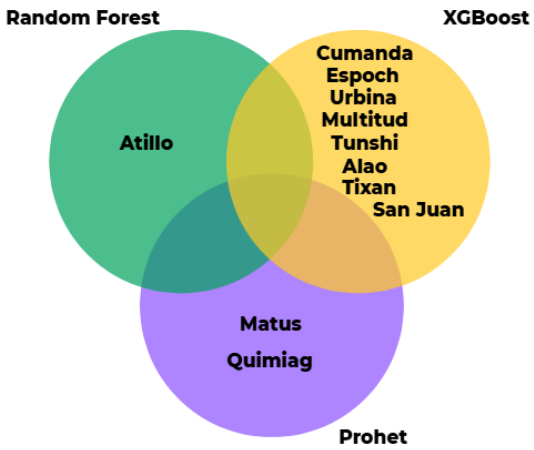


Figura 24. Clasificación de las 11 Estaciones según el Modelo con la Mejor Métrica Combinada MAE Mínimo y R^2 Máximo.

La interpretación revela tendencias claras en la superioridad de los modelos para diferentes microclimas:

- El modelo XGBoost resultó ser el que mejor desempeño mostro en 8 de las 11 estaciones, su dominio incluye las estaciones con el mejor rendimiento absoluto: Cumandá (MAE = 0.200 °C, R^2 = 98%) y Espoch (MAE = 0.291 °C, R^2 = 98.3%). Esto subraya que la arquitectura de boosting es la más efectiva para capturar la compleja relación no lineal de las variables climáticas.

- El modelo Prophet presento un mejor desempeño en las estaciones de Matus (MAE = 0.384 °C, R^2 = 97.1%) y Quimiag (MAE = 0.696 °C). En particular se observa un valor elevado de R^2 en Matus, esto sugiere que la modelación de la estacionalidad en esta ubicación resultó ser el enfoque más adecuado para representar el comportamiento climático.

- El modelo Random Forest fue la mejor opción para la estación Atillo (MAE=0.681 °C), si bien el valor del MAE es más alto que en otras estaciones, el modelo logró en este caso el mejor equilibrio entre las métricas de desempeño evaluadas.

4.1.3. Validación: Análisis ANOVA

Para validar las diferencias en el sesgo (la tendencia promedio a subestimar o sobreestimar) de los modelos, se aplicó un Análisis de Varianza (ANOVA) a los residuos o errores simples de la estación Cumandá (el mejor caso de rendimiento). Los residuos se definieron como la resta directa entre la temperatura real y la predicha (R = Tem_Real – Temp_Predicha).

El objetivo fue evaluar si la media del error simple, entendida como el sesgo promedio, presentaba diferencias significativamente diferente entre los modelos R1 (Prophet), R2 (Random Forest), y R3 (XGBoost).

- **Resultados del Análisis ANOVA**

Tabla 7: Resumen de la prueba ANOVA para la diferencia de medias de error simple en la estación Cumandá.

ANOVA

Temp

	Suma de cuadrados	gl	Media cuadrática	F	Sig.
Entre grupos	,319	2	,159	1,694	,184
Dentro de grupos	327,780	3486	,094		
Total	328,098	3488			

En la tabla 7 se observa el valor de significancia (p-valor) obtenido es 0.184. Dado que $p > \alpha$ (donde $\alpha = 0.05$), se concluye que no existe una diferencia estadísticamente significativa

en el sesgo promedio entre los tres modelos. Es decir, a pesar de las diferencias en precisión (MAE), el promedio de los errores de subestimación y sobreestimación de los tres modelos es estadísticamente el mismo.

Tabla 8: Pruebas Post Hoc HSD de Tukey, estación Cumandá.

HSD Tukey[a]

Modelos	N	Subconjunto para alfa = 0.05 1
R3	**1163**	**,0763**
R2	**1163**	**,0910**
R1	**1163**	**,0995**
Sig.		**,164**

La tabla 8 muestra que las pruebas HSD de Tukey confirman que, aunque los modelos no son estadísticamente diferentes, XGBoost (R3) presenta la menor media de error simple (0.0763). Esto indica que el sesgo de XGBoost es marginalmente más cercano a cero, lo que lo convierte en el modelo menos propenso a subestimar o sobreestimar consistentemente, aunque la diferencia no sea significativa.

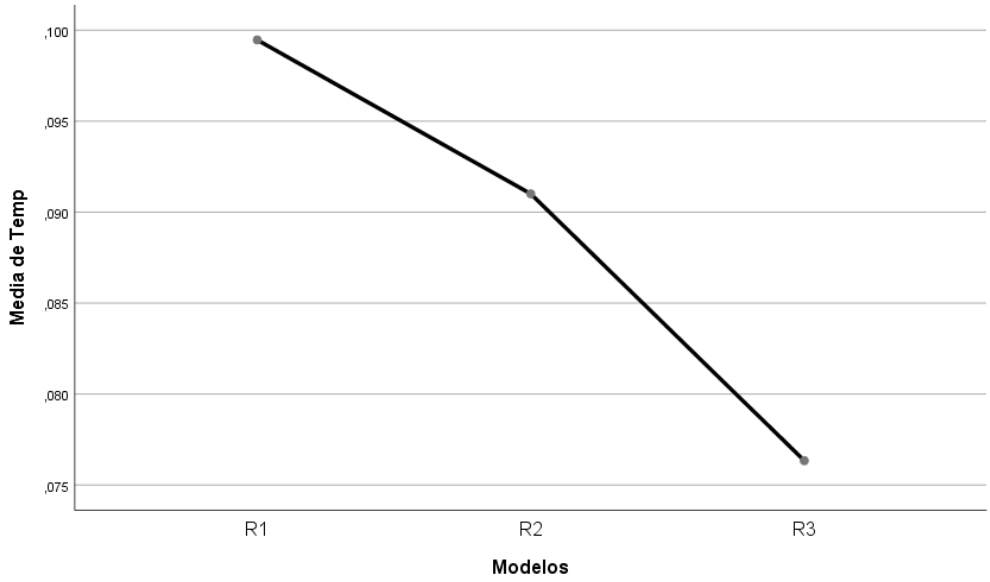


Figura 25. Media del Error Simple (Sesgo) de los Modelos (R1, R2, R3).

El análisis ANOVA determinó que las diferencias no son estadísticamente significativas, si se observa la figura 25 se certifica la conclusión basada en las métricas: XGBoost (R3) presenta el mejor desempeño consistente en la estación Cumandá, tanto en precisión (MAE) como en la minimización del sesgo promedio.

CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

- Estudiar el estado de arte permitió seleccionar los algoritmos Machine Learning más relevantes para la predicción climática, este análisis guio la decisión de utilizar algoritmos de regresión basados en árboles de decisión como Random Forest y XGBoost, además modelos de series de tiempo como Prophet, reconociéndolos como las herramientas adecuadas para la estimación precisa de la temperatura. La evaluación práctica demostró que los modelos de Boosting superan a los modelos de series de tiempo específicos y de Bagging en la mayoría de los casos de uso para la predicción de temperatura.

- El desarrollo exitoso de los modelos de predicción se basó en un riguroso proceso metodológico que aseguró la calidad y el entrenamiento efectivo de los algoritmos. Este proceso incluyó la limpieza y el preprocesamiento exhaustivo de los datos meteorológicos históricos, la implementación de técnicas clave de Ingeniería de Características como las variables lag y cíclicas, y el papel fundamental de las variables exógenas en el entrenamiento; todo esto resultó esencial para que los algoritmos de Machine Learning pudieran capturar con precisión tanto los patrones de estacionalidad como la dependencia temporal inherente a la serie de temperaturas.

- La evaluación de los modelos mediante métricas de análisis MAE, R^2 y ANOVA demostró que: El desempeño de XGBoost es significativamente alta, con un R^2 promedio superior al 96% en la mayoría de las estaciones meteorológicas .El Análisis de Varianza (ANOVA) en la estación Cumandá indicó que no existe una diferencia estadísticamente significativa en el sesgo promedio de los tres modelos, no obstante, las pruebas post hoc confirmaron que XGBoost presenta el mejor desempeño en la predicción de temperatura en la ya mencionada estación.

5.2. RECOMENDACIONES

- Una vez optimizado los modelos, desarrollarlos de manera online e implementarlo en un entorno de producción (como una plataforma web o un sistema de monitoreo automatizado). Esto permitirá a las autoridades locales, agrícolas y de gestión de riesgos utilizar el sistema de predicción como una herramienta de alerta temprana para tomar decisiones informadas ante variaciones extremas de temperatura.

- Para aumentar la precisión predictiva, se recomienda integrar variables exógenas adicionales, estas pueden incluir datos geográficos como la latitud, longitud y altitud, además tratar de no perder grandes cantidades de datos climatológicos, si se daña algún sensor tratar de reponerlo de manera inmediata para así no afectar la continuidad de las series temporales.

- Es fundamental establecer un protocolo de validación y reentrenamiento periódico de los modelos especialmente del que obtuvo el mejor desempeño. A medida que cambian los patrones climáticos o se dispone de nuevos datos, el modelo debe ser ajustado con la información más reciente para asegurar que mantenga su capacidad de generalización y evite la degradación del rendimiento.

BIBLIOGRAFIA

[1] G. Batista Mendoza, E. J. Cedeño Herrera y G. Cedeño Batista, «Machine learning aplicado al análisis de un set de datosde parámetros ambientales en galpones de pollos de engorde,» Visión Antataura, vol. 7, nº 2, pp. 1-2, 2023.

[2] CLEAN, «CLEAN, committed to climate and energy education,» cleanet.org, 4 Noviembre 2020. [En línea]. Available: https://cleanet.org/clean/literacy/climate/spanish/principle_6.html. [Último acceso: 10 Noviembre 2024].

[3] G. Wang, X. Hao, X. Yao, J. Wang, H. Li, R. Chen y Z. Liu, «Simulations of Snowmelt Runoff in a High-Altitude Mountainous Area Based on Big Data and Machine Learning Models: Taking the Xiying River Basin as an Example,» MDP, vol. 15, nº 4, pp. 1-16, 2023.

[4] M. Reichstein, G. Camps-Valls, B. Stevens, M. Jung, J. Denzler, N. Carvalhais y Prabhat, «Deep learning and process understanding for data-driven Earth system science,» Nature, vol. 1, nº 566, p. 195–204 , 2019.

[5] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar y P.-A. Muller , «Deep learning for time series classification: a review,» Springer Nature Link, vol. 33, nº 4, pp. 917-963, 2019.

[6] IPCC, «Climate Change 2022: Impacts, Adaptation and Vulnerability, Contribution of Working Group II to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change,» Cambridge University Press, Cambridge, 2022.

[7] J. M. Citar, J. Paz, S. Navas, E. Turienzo, J. Diez-Sierra y N. Peña, «Climate change impacts on the water resources of Andean countries,» Ingeniería del agua, vol. 24, pp. 219-233, 2020.

[8] W. F. M. Patrick, «Hydrological Patterns of the Chimborazo Reserve: Streamflow, climate, and glacier recession data show a loss of glacial influence on the southwestern aspect of the Chimborazo volcano, Ecuador,» SIT Study Abroad, Brattleboro, 2023.

[9] M. Barooni, K. Ziarati y A. Barooni, «Frost Prediction Using Machine Learning Methods in Fars Province,» arXiv, 2024.

[10] L. Tuaza Castro, C. Johnson, M. McBurney y A. Isla, El CAMBIO CLIMÁTICO Y LAS COMUNIDADES INDÍGENAS EN LOS ANDES DEL ECUADOR, Riobamba: Unach, 2021.

[11] B. Urquizo Tenesaca, «Cambio climático y migración en los pueblos indígenas de la provincia de Chimborazo - Ecuador,» Conciencia Digital, vol. 4, nº 1.2, p. 470–488, 2021.

[12] V. Monego, J. Anochi y H. Campos Velho, « South America Seasonal Precipitation Prediction by Gradient-Boosting Machine-Learning Approach,» Atmosphere, vol. 13, nº 2, p. 243, 2022.

[13] A. T, C. K, A. K y L. V, «Temperature Prediction using Machine Learning,» IEEE, vol. 1, pp. 1264-1268, 2019.

[14] B. Bochenek y Z. Ustrnul, «Machine Learning in Weather Prediction and Climate Analyses—Applications and Perspectives,» atmosphere, vol. 13, nº 2, p. 180, 2022.

[15] «Elementos y factores del clima - Meteo Navarra,» Navarra.es, s.f.. [En línea]. Available: https://meteo.navarra.es/definiciones/elementosfactores.cfm. [Último acceso: 04 08 2025].

[16] A. P. Joshi y B. V. Patel, «Data Preprocessing: The Techniques for Preparing Clean and Quality Data for Data Analytics Process,» Oriental Journal of Computer Science and Technology, vol. 13, nº 2-3, pp. 78-81, 2020.

[17] K. Rojas, «CienciaDatos,» Bookdown, 26 09 2022. [En línea]. Available: https://bookdown.org/keilor_rojas/CienciaDatos/an%C3%A1lisis-de-series-de-tiempo.html#:~:text=Una%20serie%20de%20tiempo%20es,la%20tendencia%20en%20los%20datos.. [Último acceso: 07 08 2025].

[18] IBM, «Interpolación de valores en Series temporales de Netezza,» IBM, 03 01 2025. [En línea]. Available: https://www.ibm.com/docs/es/spss-modeler/18.5.0?topic=series-interpolation-values-in-netezza-time. [Último acceso: 09 08 2025].

[19] C. Janiesch, P. Zschech y K. Heinrich, «Machine learning and deep learning,» arXiv, 2021.

[20] B. Romero Rojas, «Una introducción a los modelos de Machine Learning,» Benemérita Universidad Autónoma de Puebla, Puebla, 2020.

[21] IBM, «¿Qué es un algoritmo de machine learning?,» IBM Think, [En línea]. Available: https://www.ibm.com/mx-es/think/topics/machine-learning-algorithms. [Último acceso: 10 08 2025].

[22] R. Sharma, K. Sharma y A. Khanna, «Estudio del aprendizaje supervisado y no supervisado Aprendiendo,» International Journal for Research in Applied Science & Engineering Technology (IJRASET), vol. 8, nº 6, pp. 588-593, 2020.

[23] L. J. Sandoval, «Algoritmos de aprendizaje automático para análisis y predicción de datos,» Repositorio Digital de la Ciencia y Cultura de El Salvador (REDICCES), Santa Tecla, 1018.

[24] S. Sah, «Machine Learning: A Review of Learning Types,» Preprints, 2020.

[25] P. Taser, «Application of Bagging and Boosting Approaches Using Decision Tree-Based Algorithms in Diabetes Risk Prediction,» Proceedings, vol. 74, p. 6, 2021.

[26] F. López, «Towards Data Science,» Medium, 11 Enero 2021. [En línea]. Available: https://towardsdatascience.com/ensemble-learning-bagging-boosting-3098079e5422/. [Último acceso: 11 Agosto 2025].

[27] H. A. Salman, A. Kalakech y A. Steiti, «Random Forest Algorithm Overview,» Babylonian Journal of Machine Learning, vol. 2024, pp. 69-79, 2024.

[28] A. ,. T. D. ,. K. S. Raut, «Random Forest Regressor Model for Rainfall Prediction,» de 1109/OCAMS60111 2073 10526085, 2023.

[29] C. Bentéjac, A. Csörgő y G. Martínez-Muñoz, «A Comparative Analysis of XGBoost,» arXiv Preprint, 2019.

[30] R. Das, A. Raut, S. Mahadik y M. Behl, «A New AGE Forecasting Model PROPHET,» International Journal of Scientific Research in Engineering and Management (IJSREM), vol. 6, nº 7, 2023.

[31] J. Amat Rodrigo y J. Escobar Ortiz, «Skforecast Docs,» 2023. [En línea]. Available: https://skforecast.org/0.8.1/faq/cyclical-features-time-series. [Último acceso: 19 Agosto 2025].

[32] T. i. Data, «Feature-engine Docs,» Lag Features — Feature-engine Documentation, 2025. [En línea]. Available: https://feature-engine.trainindata.com/en/1.8.x/user_guide/timeseries/forecasting/LagFeatures.html. [Último acceso: 19 Agosto 2025].

[33] J. Jácome, «Técnicas de predicción aplicada a la demanda,» Dpto. Ingeniería Eléctrica Escuela Técnica Superior de Ingeniería, Sevilla, 2023.

[34] D. Chicco, M. Warrens y G. Jurman, «The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation,» PeerJ Computer Science, vol. 7, p. e623, 2021.

ANEXOS

- Código en Python del preprocesamiento datos obtenidos de la estación meteorológica

```
import xarray as xr
import pandas as pd
import numpy as np

# cargo el archivo de las estaciones meteorologicas
archivo = 'DatosEstaciones.nc'

# Abro el archivo .nc
ds = xr.open_dataset(archivo)

# imprimo
ds.head()

# Imprimo estaciones disponibles
print("Estaciones:", ds['estacion'].values)

# elejijo la estacion y las fechas de los datos
estacion = 'ESPOCH'
inicio = '2013-01-01'
fin = '2024-12-31'

# 4. Filtro por estación y rango de tiempo
ds_filtrado = ds.sel(estacion=estacion, Tiempo=slice(inicio, fin))

# 5. Convierto a DataFrame
df_datos = ds_filtrado.to_dataframe()

# df_datos.to_csv(f'{estacion}_{inicio[:4]}_{fin[:4]}.csv')
# print("Exportado:", f'{estacion}_{inicio[:4]}_{fin[:4]}.csv')

print("Columnas de datos de las estaciones:
",list(df_datos.columns))

df_datos.head(2)

# Si no existe la columna 'Tiempo', la recupero desde el índice
if 'Tiempo' not in df_datos.columns:
    df_datos = df_datos.reset_index()
    if 'index' in df_datos.columns:
        df_datos = df_datos.rename(columns={'index': 'Tiempo'})
```

```
# esta es la estructura esperada de las variables
estructura_esperada = [
    'Tiempo', 'TA Avg','TA Max','TA Min','RH Avg','RH Max','RH Min',
     'PA Avg','PA Max','PA Min','SR_Dif Avg','SR_Dif Max','SR_Dif
Min','Sum_SR_Dif',
    'SR_Glob Avg','SR_Glob Max','SR_Glob Min','Sum_SR_Glob',
    'TS_TG1 Avg','TS_TG1 Max','TS_TG1 Min',
    'TS_TG2 Avg','TS_TG2 Max','TS_TG2 Min',
    'TS_TG3 Avg','TS_TG3 Max','TS_TG3 Min',
    'TS_TG4 Avg','TS_TG4 Max','TS_TG4 Min',
    'TS_TG5 Avg','TS_TG5 Max','TS_TG5 Min',
    'TS_TG6 Avg','TS_TG6 Max','TS_TG6 Min',
    'TS_TG7 Avg','TS_TG7 Max','TS_TG7 Min',
      'GenWind  SpdMin','GenWind  WRun','GenWind  DirAvg','GenWind
DirMax',
    'GenWind GustDir','GenWind GustH','GenWind GustM',
    'GenWind SpdAvg','GenWind SpdMax',
    'WindChill Avg','WindChill Max','WindChill Min',
    'QMBATT meas','Sum_PR','estacion','latitud','longitud'
]  # Total 55

# valido la estructura
enc = list(df_datos.columns)
if len(enc) != len(estructura_esperada):
                      print(f"Columnas:          actual={len(enc)},
esperado={len(estructura_esperada)}")
else:
         difer   =   [(i,   e,   a)   for   i,   (e,   a)   in
enumerate(zip(estructura_esperada, enc)) if e != a]
    if difer:
        print("Columnas diferentes en posiciones:")
        for i, e, a in difer:
            print(f"  pos {i}: esperado '{e}' pero '{a}'")
    else:
        print("Estructura correcta (55 columnas en orden)")
        iguales = [e == a for e, a in zip(estructura_esperada, enc)]
        print(f"Columnas bien: {sum(iguales)}, mal: {len(iguales) -
sum(iguales)}")

# Filtro posibles columnas para ML
columnas_utiles = ["Tiempo", "TA Avg", "PA Avg", "RH Avg", "SR_Glob
Avg", "GenWind SpdAvg"]
```

```
df_datos = df_datos[columnas_utiles]

print("\n DF con las columnas filtradas:")

df_datos.head(2)

# la columna tiempo debe ser tipo datetime
df_datos["Tiempo"]      =      pd.to_datetime(df_datos["Tiempo"],
errors="coerce")

# columna con el año
df_datos["año"] = df_datos["Tiempo"].dt.year

variables = ["TA Avg", "PA Avg", "RH Avg", "SR_Glob Avg", "GenWind
SpdAvg"]

# Calculo el porcentaje de datos válidos
porcentaje_validos = {
    var: df_datos.groupby("año")[var].apply(lambda x: ((x.notna())
& (x != 0)).mean() * 100)
    for var in variables
}

df_porcentajes = pd.DataFrame(porcentaje_validos)
df_porcentajes.head(12)

# Filtro filas donde la temperatura ('TA Avg') sea válida (no NaN y
distinta de 0)
df_datos  =  df_datos[(df_datos['TA  Avg'].notna())  &  (df_datos['TA
Avg'] != 0)]

# Variables a analizar
variables_con_0_invalido = ["TA Avg", "PA Avg", "RH Avg", "SR_Glob
Avg"]
variable_con_0_valido = "GenWind SpdAvg"
todas_las_variables      =      variables_con_0_invalido      +
[variable_con_0_valido]

# Reemplazar ceros por NaN solo en las variables donde 0 es inválido
df_datos[variables_con_0_invalido]                                =
df_datos[variables_con_0_invalido].replace(0, np.nan)
```

```
# Asegurarse de que 'Tiempo' esté en formato datetime y crear columna
'año'
df_datos["Tiempo"]       =       pd.to_datetime(df_datos["Tiempo"],
errors="coerce")
df_datos["año"] = df_datos["Tiempo"].dt.year

# Calculo el porcentaje de datos válidos por año y por variable
porcentaje_validos = {
           var:    df_datos.groupby("año")[var].apply(lambda    x:
x.notna().mean() * 100)
    for var in todas_las_variables
}

# Creo DF con resultados
df_porcentajes = pd.DataFrame(porcentaje_validos).round(2)

print("Porcentaje de datos válidos por año y por variable (0 tratados
como inválidos solo donde corresponde):")
print(df_porcentajes.head(12))

# Hago una copia del valor original antes de interpolar
original = df_datos[["Tiempo", "SR_Glob Avg", "RH Avg"]].copy()
original = original.set_index("Tiempo")

# 2. Interpolo solo las variables indicadas
df_datos = df_datos.set_index("Tiempo")
df_datos["SR_Glob         Avg"]         =         df_datos["SR_Glob
Avg"].interpolate(method="time")
df_datos["RH Avg"] = df_datos["RH Avg"].interpolate(method="time")

# 3. Detecto qué valores fueron interpolados
df_datos["fue_interpolado_SR"]         =         df_datos["SR_Glob
Avg"].ne(original["SR_Glob Avg"])
df_datos["fue_interpolado_RH"] = df_datos["RH Avg"].ne(original["RH
Avg"])

# 4. Volver a tener "Tiempo" como columna
df_datos = df_datos.reset_index()

# 5. Creo columna de hora
df_datos["hora"] = df_datos["Tiempo"].dt.hour

# 6. Calculo media real por hora sin valores interpolados
```

```
media_sr                                                    =
df_datos[~df_datos["fue_interpolado_SR"]].groupby("hora")["SR_Glob
Avg"].mean()
media_rh                                                    =
df_datos[~df_datos["fue_interpolado_RH"]].groupby("hora")["RH
Avg"].mean()

# 7. Reemplazo valores interpolados anómalos por la media horaria
umbral_sr = 5    # Puedes ajustarlo
umbral_rh = 10

df_datos.loc[
     (df_datos["fue_interpolado_SR"]) & (df_datos["SR_Glob Avg"] <
umbral_sr),
    "SR_Glob Avg"
] = df_datos["hora"].map(media_sr)

df_datos.loc[
      (df_datos["fue_interpolado_RH"])  &  (df_datos["RH  Avg"]  <
umbral_rh),
    "RH Avg"
] = df_datos["hora"].map(media_rh)

# 8. Comparación opcional
comparacion_sr = df_datos[df_datos["fue_interpolado_SR"]].copy()
comparacion_sr["Media_hora_real_SR"]                            =
comparacion_sr["hora"].map(media_sr)

comparacion_rh = df_datos[df_datos["fue_interpolado_RH"]].copy()
comparacion_rh["Media_hora_real_RH"]                            =
comparacion_rh["hora"].map(media_rh)

print("Comparación SR_Glob Avg")
print(comparacion_sr[["Tiempo",     "hora",     "SR_Glob     Avg",
"Media_hora_real_SR"]].head(15))

print("\n Comparación RH Avg")
print(comparacion_rh[["Tiempo",       "hora",       "RH       Avg",
"Media_hora_real_RH"]].head(15))

# Conservo solo las variables principales
df_datos  =  df_datos[["Tiempo",  "TA  Avg",  "PA  Avg",  "RH  Avg",
"SR_Glob Avg", "GenWind SpdAvg"]]
```

```
# Aseguro que 'Tiempo' sea tipo datetime
df_datos["Tiempo"]       =       pd.to_datetime(df_datos["Tiempo"],
errors="coerce")

# Extraigo hora, día, mes y año
df_datos["hora"] = df_datos["Tiempo"].dt.hour
df_datos["dia"] = df_datos["Tiempo"].dt.day
df_datos["mes"] = df_datos["Tiempo"].dt.month
df_datos["año"] = df_datos["Tiempo"].dt.year

df_datos.to_csv("espoch_procesado.csv", index=False)

df_datos.head()
```