

Bogotá D.C., 12 de mayo de 2025

## REPORTE FASE 2

### TRABAJO FINAL DE MAESTRÍA “PRONÓSTICO DE LA PRECIPITACIÓN ACUMULADA EN COLOMBIA, UTILIZANDO TÉCNICAS DE APRENDIZAJE AUTOMÁTICO”

<b>NOMBRE:</b> JUAN DIEGO MOGOLLÓN OVIEDO	
<b>OBJETIVO:</b> Desarrollar un modelo para el pronóstico de la precipitación acumulada en Colombia, utilizando técnicas de aprendizaje automático.	
<b>OBJETIVOS ESPECÍFICOS:</b> <ul style="list-style-type: none"><li>• Seleccionar un conjunto de datos abiertos de precipitación acumulada en Colombia, verificando su acceso y calidad mediante algoritmos de análisis de datos.</li><li>• Implementar un proceso para la extracción, transformación y carga de los datos, aplicando técnicas de ingeniería de características.</li><li>• Evaluar el desempeño de diferentes algoritmos de aprendizaje automático para el pronóstico de precipitación acumulada, utilizando los datos procesados y métricas de evaluación empleadas en predicción climática.</li><li>• Diseñar una arquitectura para la implementación del modelo en un entorno de prueba, aplicando prácticas y herramientas de operaciones de aprendizaje automático.</li></ul>	
<b>PLAZO DE EJECUCIÓN:</b> 16 semanas	
<b>INICIO:</b> 31 de marzo de 2025	<b>FINALIZACIÓN:</b> 26 de julio de 2025

Con el objetivo de dar cumplimiento al cronograma estipulado en la “Propuesta de Trabajo Final de Maestría” en el periodo 21 de abril de 2025 al 12 de mayo de 2025, se presenta a continuación el reporte de la fase 2:

<p><b>RESULTADO OBTENIDO:</b> Se implementó un proceso para extraer, transformar y cargar los datos de precipitación acumulada de la red de estaciones meteorológicas del IDEAM, utilizando los métodos de descomposición de señales Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAM) y Variational Mode Decomposition (VMD).</p> <p>Para la implementación del proceso, se utilizó el lenguaje de programación Python mediante el entorno Jupyter Notebooks, lo cual facilita el seguimiento detallado de la ejecución de los scripts. El flujo de trabajo está conformado por dos pipelines principales:</p> <ol style="list-style-type: none"><li>1. Extracción de datos: este primer pipeline se conecta al servicio <a href="#">API REST</a> de la plataforma DHIME del IDEAM, realiza solicitudes en bloques de 20 estaciones meteorológicas, decodifica las respuestas del servidor y almacena los datos obtenidos en un archivo en formato .csv.</li><li>2. Transformación y carga de datos: el segundo pipeline lee el archivo generado, aplica los filtros de estaciones definidos en la fase de análisis exploratorio de datos, normaliza los valores y genera</li></ol>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

secuencias temporales continuas por estación. Posteriormente, se aplica el método de descomposición CEEMDAN y, sobre el primer modo de frecuencia resultante, se realiza una segunda descomposición utilizando el método VMD, con el objetivo de refinar la señal principal. Finalmente, los datos transformados se cargan en formato .parquet, optimizando así el almacenamiento y la velocidad de lectura para el posterior entrenamiento de modelos.

Este procedimiento fue aplicado tanto a datos de precipitación acumulada diaria como mensual, con el objetivo de evaluar el desempeño de los modelos seleccionados sobre dos conjuntos de datos diferentes y observar las variaciones en los resultados obtenidos.

A continuación, se presentan las gráficas que soportan los resultados mencionados anteriormente.

```
def main():
    download_cne()
    stations_catalog = pd.read_excel(os.path.join(DATA_FOLDER, 'CNE_IDEAM.xls'))
    parameters_labels = {
        "Precipitacion Acumulada": {"IdParameter": "PRECIPITACION", "Label": "PTPM_CON"}
    }
    for parameter_name, values in parameters_labels.items():
        try:
            get_station_data(
                stations_catalog['CODIGO'].tolist(),
                '2000-01-01',
                '2024-12-31',
                values['IdParameter'],
                values['Label']
            )
        except Exception as e:
            print(f"Error processing {parameter_name}: {e}")

if __name__ == '__main__':
    main()
```

Fig 1. Pipeline de extracción de los datos.

```
# Pipeline principal
def main():
    data_folder = setup_workspace()
    df = load_and_clean_data(data_folder)
    df = filter_stations(df, data_folder)
    df_windows = normalize_and_create_windows(df)
    df_windows = calculate_longest_window(df_windows, data_folder)
    df_windows = decompose_windows(df_windows, data_folder)
    save_processed_data(df_windows, data_folder)

if __name__ == "__main__":
    main()
```

Fig. 2. Pipeline de transformación y carga de los datos.

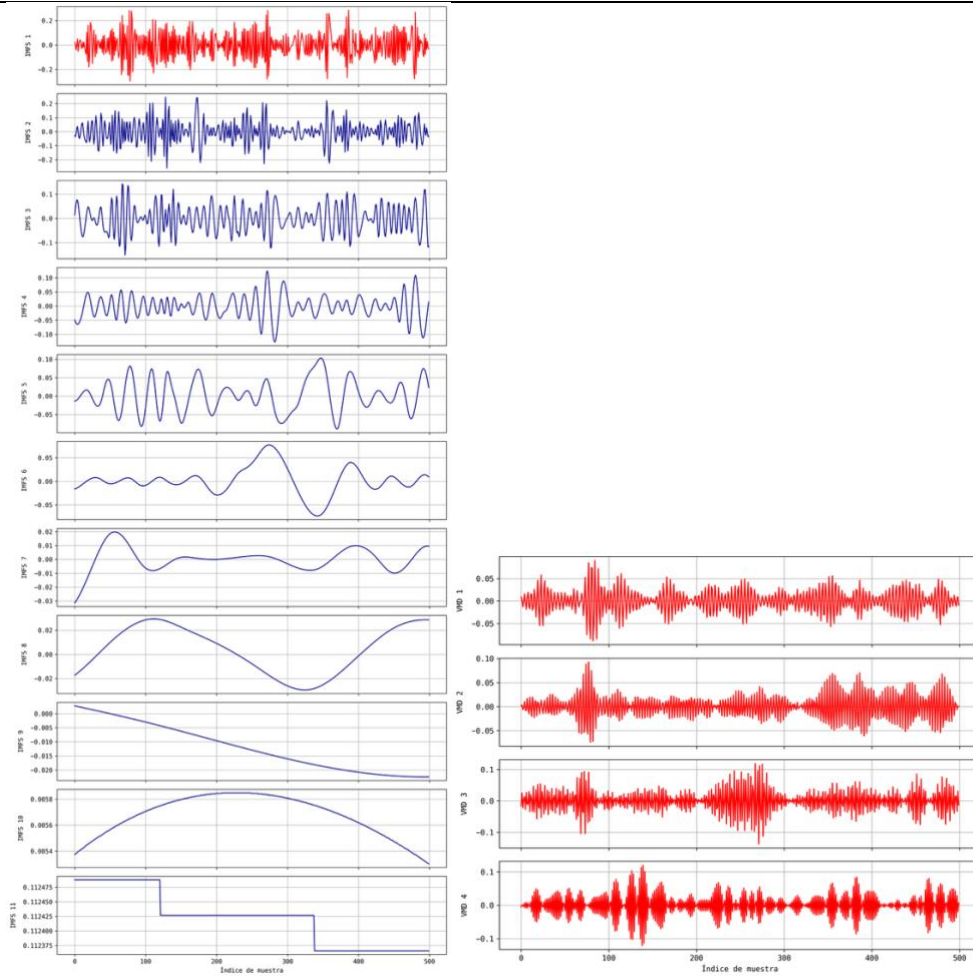


Fig. 3. Izquierda: Modos de descomposición CEEMDAM de una estación de muestra. Derecha: Modos de descomposición VMD aplicado sobre el primer modo de la descomposición CEEMDAM.

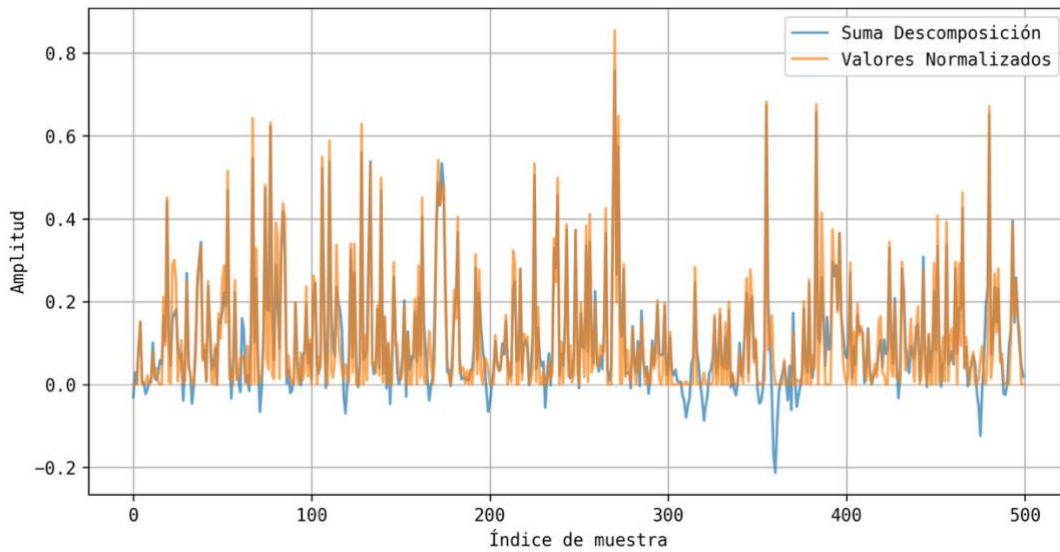


Fig 4. Comparación entre la suma de los modos CEEMDAM – VMD y los datos normalizados de precipitación acumulada de una estación de muestra.

**ACTIVIDADES REALIZADAS:**

1. Se desarrolló el script “data\_load.ipynb” para la extracción de los datos.
2. Se desarrolló el script “processing.ipynb” para la transformación y carga de los datos.

**ENLACE:** [Repositorio - GitHub](#)

Jorge Eliecer Camargo Mendoza  
**Director**

Juan Diego Mogollón Oviedo  
**Estudiante**