Presentación

**Entrega #1**

**Presentado por:**

Samuel Moncada Salazar

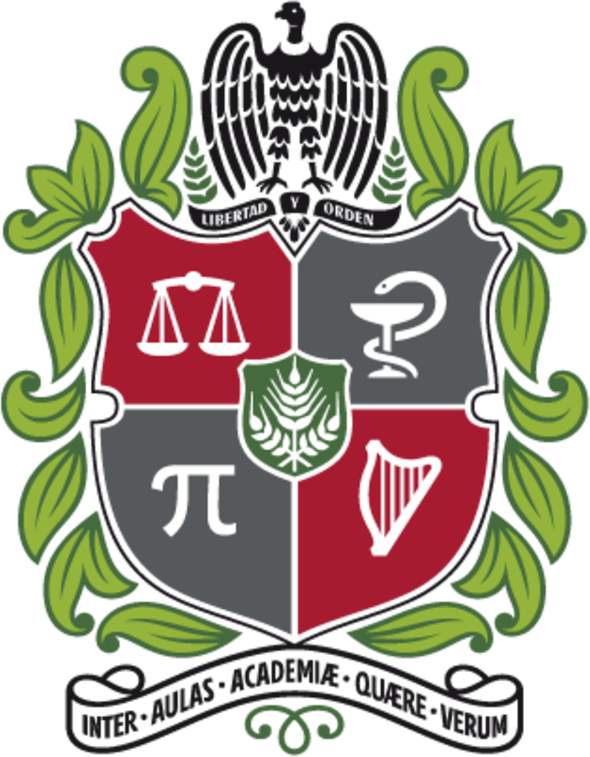
[Jeronimo Tabares Gallego](mailto:jetabaresg@unal.edu.co)

Santiago Arias Posada

**Profesor:**

[Nestor Dario Duque Mendez](mailto:ndduqueme@unal.edu.co)

Martes 9 de Diciembre

****

**Universidad Nacional de Colombia**

**Facultad de administración**

**Departamento de informática y computación**

**2025**

**Caracterización de Microclimas y Predicción de Precipitación en Zonas Urbanas Utilizando Algoritmos de Minería de Datos**

Las ciudades, debido a su topografía y desarrollo urbano, a menudo presentan variaciones climáticas significativas en distancias cortas, dando lugar a lo que se conoce como microclimas. Estos microclimas pueden influir directamente en fenómenos meteorológicos localizados, como la precipitación, lo que tiene implicaciones importantes para la gestión de riesgos, la planificación urbana y las actividades cotidianas de los ciudadanos. La capacidad de identificar estos microclimas y predecir la precipitación de manera diferenciada para cada uno de ellos es crucial, ya que los sistemas de pronóstico a gran escala a menudo no capturan estas particularidades. Este proyecto busca que se aborde esta situación mediante el uso de técnicas de minería de datos, aprovechando datos de estaciones meteorológicas distribuidas en diferentes zonas de la ciudad de Manizales

Modulo 1

**MÓDULO 1 Preprocesamiento y Exploración de Datos:**

**Libro de código :**.

<https://colab.research.google.com/drive/1fOk2qUqWqrREXoNCF1Qf5bQJkIJr0lXI?usp=sharing>

**1.Introducción**

El presente informe describe el proceso de limpieza, estandarización y transformación aplicado a un dataset meteorológico con más de 900.000 registros, cuyo objetivo es garantizar la consistencia, calidad y correcta tipificación de los datos para futuros análisis estadísticos y modelos predictivos.

El dataset original contenía múltiples columnas almacenadas como cadenas de texto, presencia de valores no numéricos, formatos inconsistentes y valores faltantes en diversas variables ambientales, lo cual requería un proceso de depuración sistemático.

1. **Revisión Inicial de DataSet**

El dataset originalmente contaba con 21 columnas en las cuales se identificaron algunas inconsistencias. Por ejemplo:

* Columnas de naturaleza numérica registradas como tipo String (object), lo cual dificulta el análisis estadístico y demás procesos posteriores (ej. Temperatura, presión, humedad, velocidad. etc).
* Del mismo modo que en el punto anterior, se registraron datos no numéricos dentro de estas columnas. (ej. Nan, –, datos con coma decimal, etc)

También se presentaron datos faltantes en varias de las columnas (la cantidad de datos faltantes variaron según la estación; por ejemplo, el dataset est\_19 presentó varias columnas con cantidades entre 50 y 200 valores faltantes, haciendo excepción de la columna observaciones, la cual presentó más de 750 mil datos faltantes.)

1. **Proceso de limpieza aplicado**

* **Conversión de Columnas numéricas:** Todas aquellas variables que naturalmente son numéricas pero están guardadas como tipo Object se les aplicaron los siguientes casos:
* Limpieza de la cadena de texto para extraer el primer valor numérico encontrado. (ej. ss15.6 → 15.6)
* Reemplazo décimas decimales por puntos (ej. 15,6 → 15.6)
* Eliminación de espacio dentro de la cadena (ej. 15. 6 → 15.6)
* Detección y corrección de valores extraños (ej. 15, → 15)
* **Estandarización de valores nulos:** Se normalizan valores vacíos (ej. ‘’, ‘ ‘, ‘nan’, ‘null’, etc). Todas estas variables tomaron el valores de ‘NaN’
* **Eliminación de columna:** Se decide eliminar la columna ‘estacion\_sk’ debido a que ya contamos con la division de estaciones según el dataset.
* **Se normalizan variables booleanas:** Columnas como ‘puerta\_Abierta’ o ‘bateria\_Baja’, contaban con formatos heterogéneos (ej. True o false, 1 o 0, si o no, etc). se homogeneizaron quedando todas con el mismo formate (True, false).
* **Eliminacion de registros duplicados:** En caso de que llegue a existir algun registro duplicado se decidió eliminar para que no haya sesgo.

1. **Outliers**

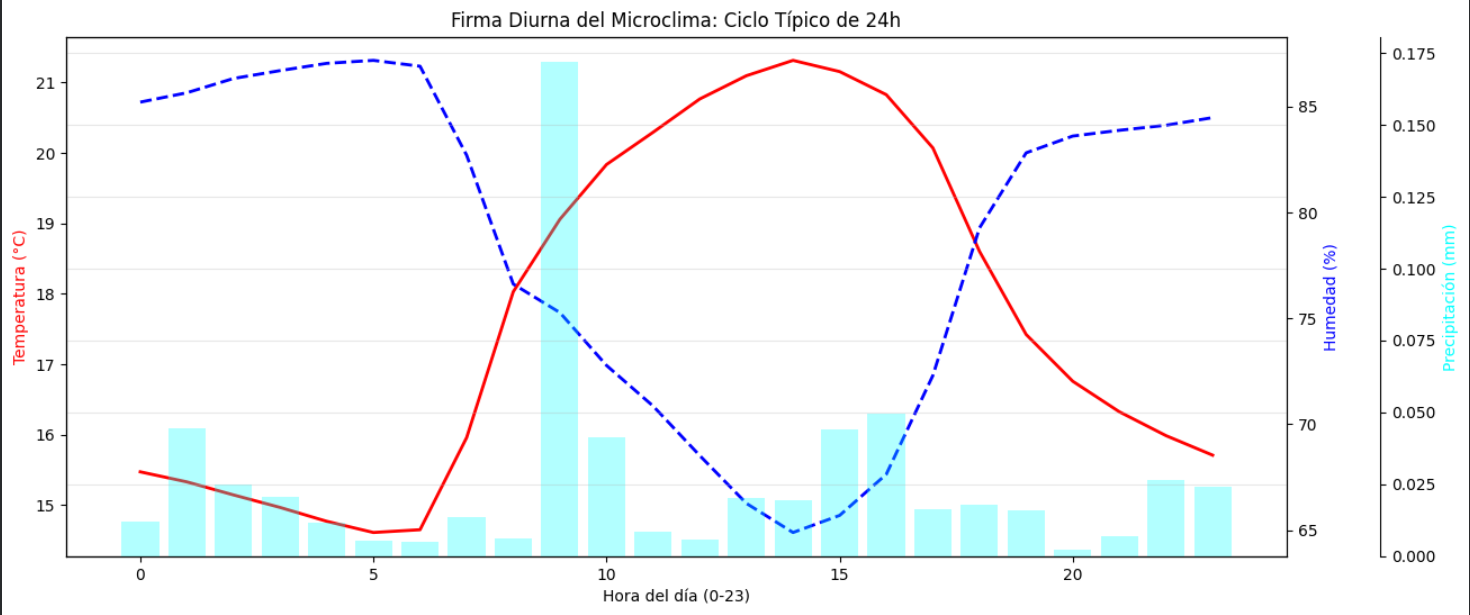
* Se decide hacer una búsqueda para definir rangos normales según la variable, teniendo en cuenta que los registros son de la ciudad de Manizales. (humedad: 0 - 100, temperatura: -10 - 40, etc).

1. **Imputación**

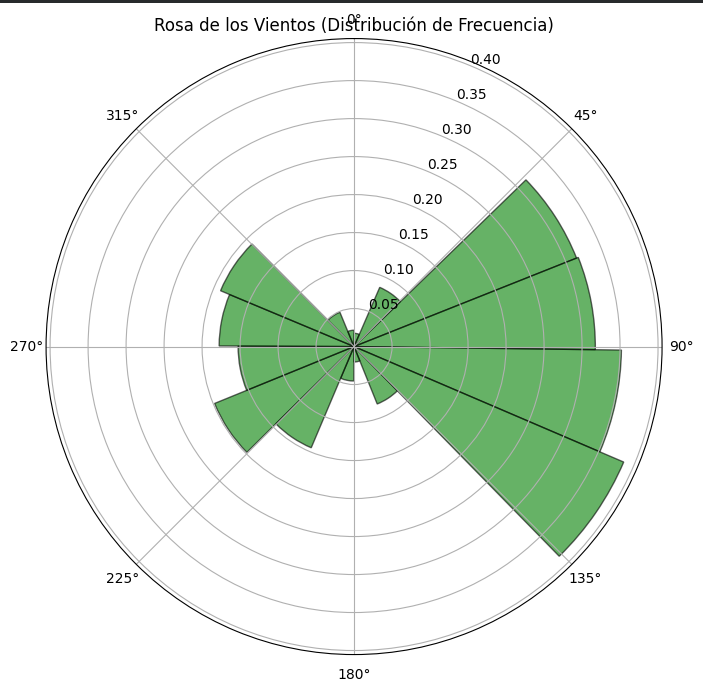
* **Extracción de datos de la columna observaciones:** Se observa que dentro de la columna ‘observaciones’ se tienen registro de algunas variables faltantes (ej. temperatura: 15, humedad: 22).
* **Por la mediana:** En casa de no hallarse rastro del valor faltante, se decidió imputar por la mediana en caso de las variables numéricas

1. **EDA**

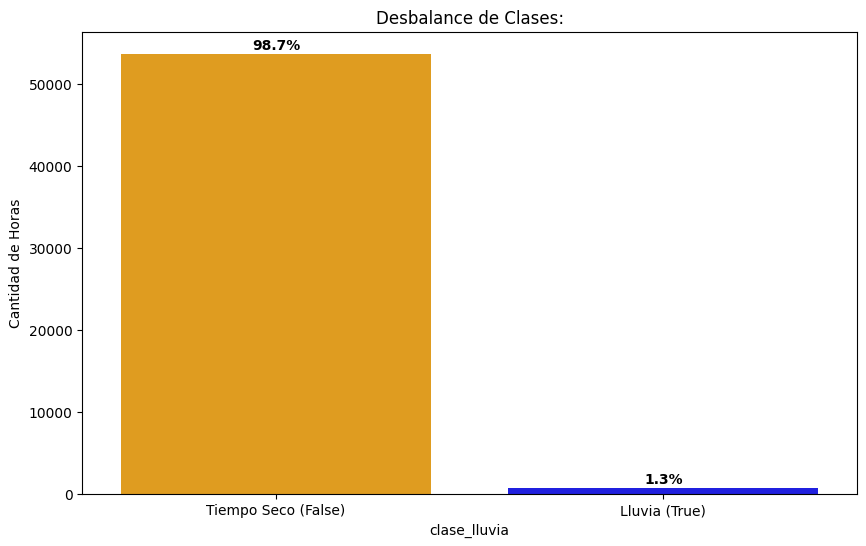
* **Firma diurna del microclima:** En esta gráfica podemos observar la precipitación, la temperatura ambiente y la humedad.



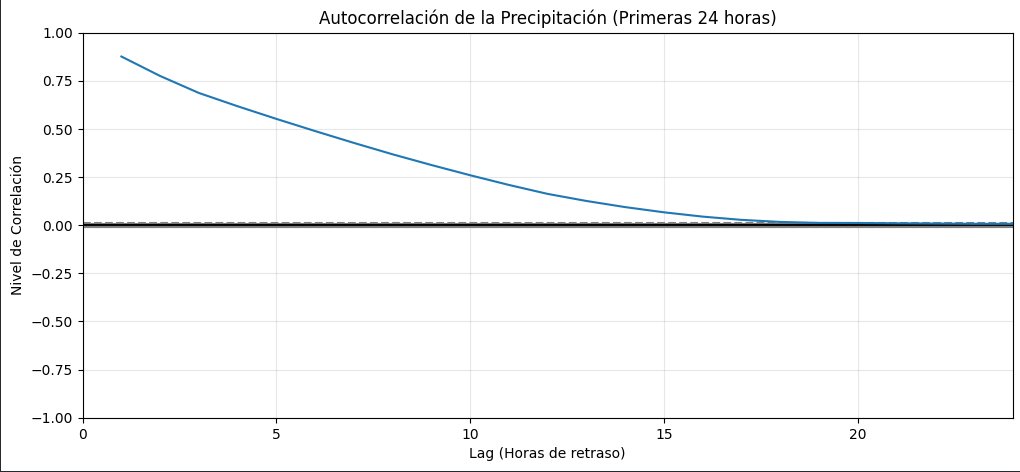
* **Rosa de los Vientos:** Esta gráfica la utilizamos para observar por donde entran los vientos en manizales



* **Desbalances de clases :** Esta gráfica la vamos a utilizar para mirar que tiempo (seco/lluvioso) obtenemos de cada dataset



* **Autocorrelación de precipitación:**  esta gráfica lo que nos hace saber es si la lluvia cuenta con memoria o procesos estadísticos



Modulo 2

**MÓDULO 2 Caracterización de Microclimas:**

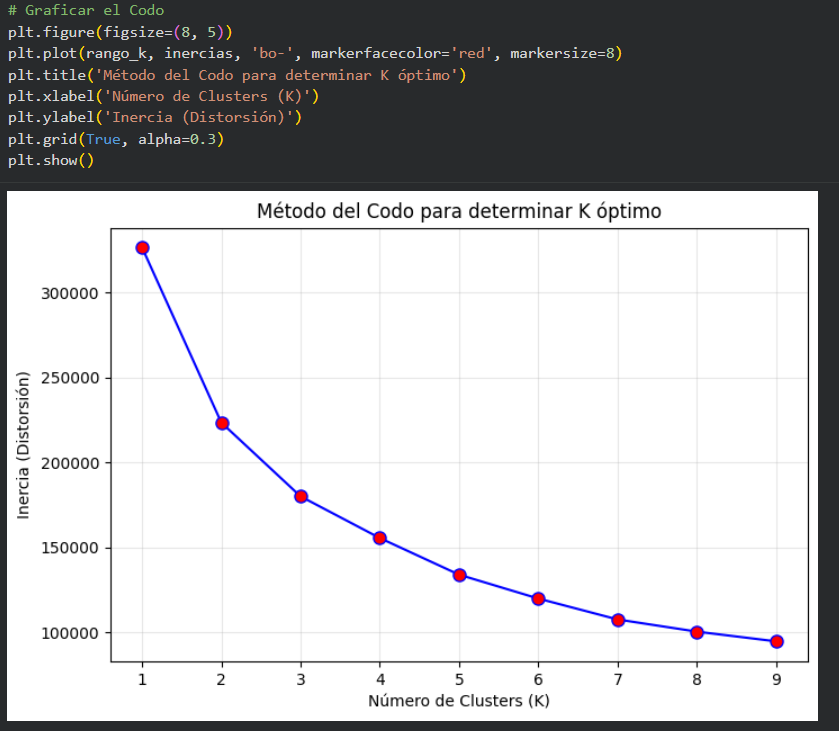
**Libro de código :**.

<https://colab.research.google.com/drive/1fOk2qUqWqrREXoNCF1Qf5bQJkIJr0lXI?usp=sharing>

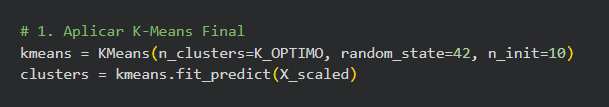
se realiza importaciones, se carga el dataset limpio, seleccionamos variables para el clustering y se realiza un escalado de datos.

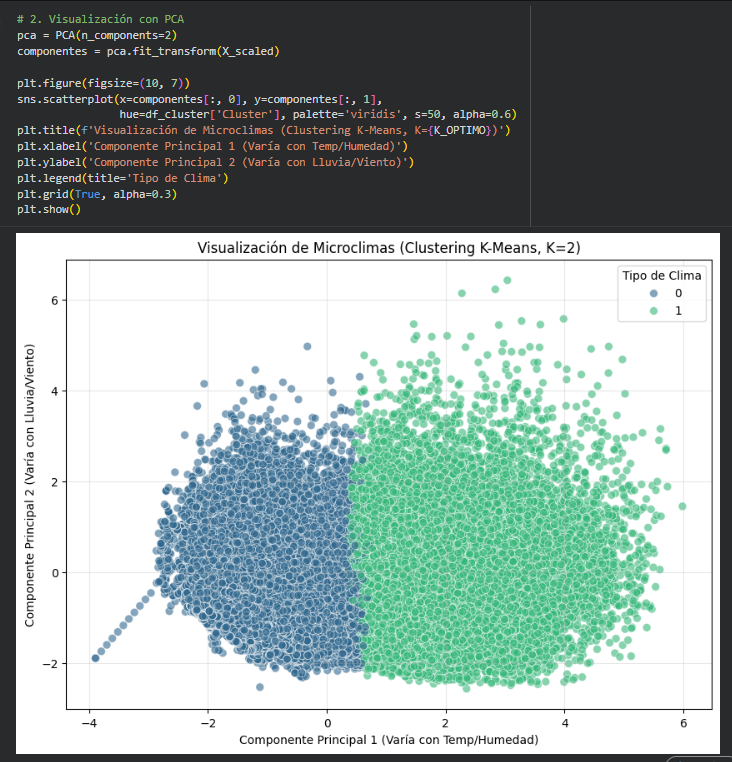


* **Análisis del Método del Codo (Elbow Method):** En este análisis se observa que podemos obtenemos donde se encuentra el quiebre de K, en la grafica observamos que el punto de quiebre fue en k=2, Esto nos da a entender que las variables meteorológicas (temperatura, humedad, lluvia) se van a organizar en dos estados opuestos y dominantes y no en múltiples estados.



* **Visualización PCA:** En esta grafica podemos observar por los colores, donde nos muestra que los cluster 0 son los dias frios/humedos, y en el cluster 1 los dias calidos,





* **Caracterización de los Clusters (La "Personalidad" del Clima):** En esta caracterización de Clusters podemos observar los procesos de precipitacion extrema.

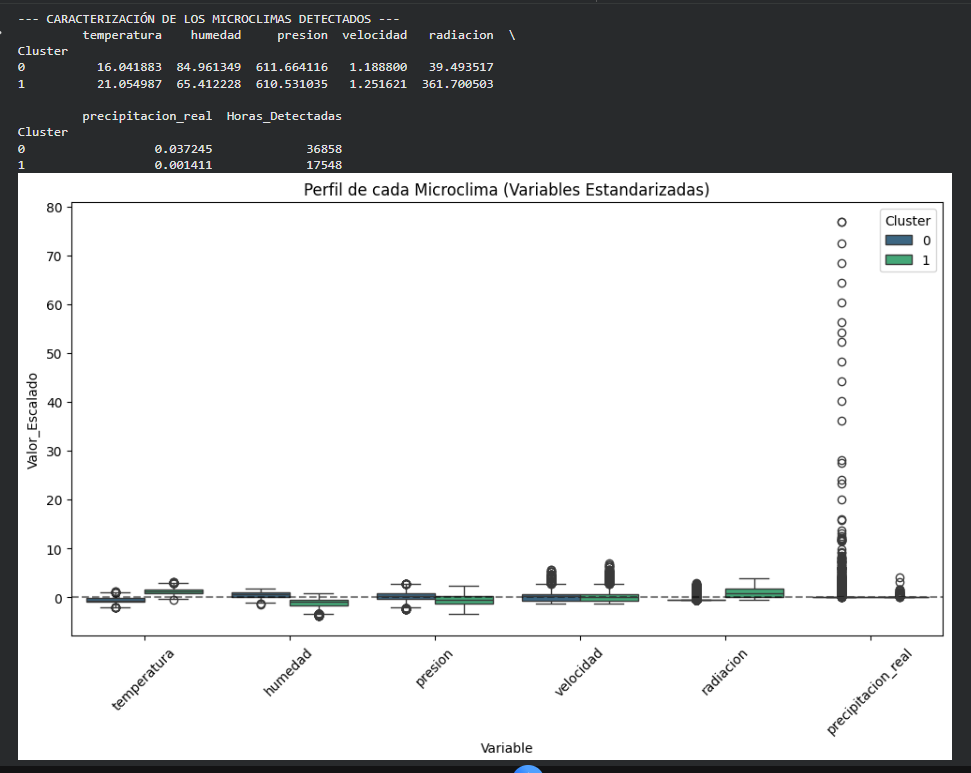
En la primera tabla muestra el promedio de cada variable para cada grupo, donde podemos ver los dos cluster, salen 2 cluster debido a la radiación ya que en una es demasiado alto, en el dia, es mayor y en la noche es casi nula. lo que quiere decir que el cluster 1 es calor y el cluster 0 es frio.

Primero se agrega por cluster y se saca el promedio, luego se agregan la cantidad de horas de cada clima y se normaliza. El gráfico visualiza qué tan diferentes son los grupos. El eje Y está "escalado" (desviaciones estándar), por lo que el 0 es el promedio global.

Mira la última columna a la derecha (precipitación real).

La caja verde (Cluster 1) está aplastada en cero esto quiere decir que no llueve en ese microclima.

La caja azul (Cluster 0) tiene muchos puntos negros hacia arriba (Outliers). Esto significa que todos los eventos de lluvia fuerte pertenecen al Cluster 0.



Modulo 3

0

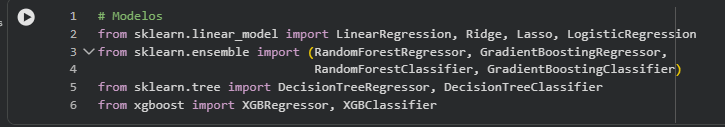
**MÓDULO 3 Predicción de Precipitación y Validación del Modelo**

**Enlace a Libro de código :**

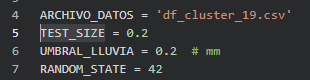
[MODULO3.ipynb](https://drive.google.com/file/d/1r-sb_z77Hh_HogOmXoswUCA1uIZp8CSt/view?usp=sharing)

ó

[**https://drive.google.com/file/d/1r-sb\_z77Hh\_HogOmXoswUCA1uIZp8CSt/view?usp=sharing**](https://drive.google.com/file/d/1r-sb_z77Hh_HogOmXoswUCA1uIZp8CSt/view?usp=sharing)



importamos los modelos que utilizaremos .



Test\_size representa el tamaño de datos de prueba , en este caso es 20 por ciento , consideramos que sigue un estándar y es coherente con la cantidad de datos .

El umbral de la lluvia es para saber desde que punto se considera que llovió o no , pusimos 0.2 que representa una llovizna leve , entonces menos de 0.2 representa no lluvia .

utilizamos una semilla para poder manejar mejor los datos , como estándar el numero 42.

**Features**

*codigo:*

hora\_sin = np.sin(2 \* π \* hora / 24) hora\_cos = np.cos(2 \* π \* hora / 24) mes\_sin = np.sin(2 \* π \* mes / 12) mes\_cos = np.cos(2 \* π \* mes / 12)

\mathrm{hora\\_ sin}=\sin \! \left( 2\pi \frac{\mathrm{hora}}{24}\right) ,\quad \mathrm{hora\\_ cos}=\cos \! \left( 2\pi \frac{\mathrm{hora}}{24}\right)

\mathrm{mes\\_ sin}=\sin \! \left( 2\pi \frac{\mathrm{mes}}{12}\right) ,\quad \mathrm{mes\\_ cos}=\cos \! \left( 2\pi \frac{\mathrm{mes}}{12}\right)

utilizamos estas fórmulas matemáticas para establecer valores enteros al tiempo , para que los modelos de predicción puedan relacionar mejor los datos

features\_calendario = [ 'hora', *# 0-23* 'dia\_semana', *# 0-6 (Lun-Dom)* 'dia\_mes', *# 1-31* 'mes', *# 1-12* 'dia\_año' *# 1-365* ]

**LAGS**

# Lags implementados: 1, 3, 6, 12, 24 horas

for variable in ['temperatura', 'humedad', 'presion',

'velocidad', 'radiacion', 'precipitacion\_real']:

for lag in [1, 3, 6, 12, 24]:

df[f'{variable}\_lag{lag}'] = df[variable].shift(lag)

implementamos estos lags , porque representan comportamientos típicos climáticos , entonces algunos modelos analizan que sucedió horas antes del acontecimiento para correlacionarlo , ejemplo , si llovio a las 11 se ve si a las 10 llovió y se correlacionan los datos , sirve para modelos no secuenciales(no procesan el tiempo de forma inherente) como el random forest

**Total de features lag:** 6 variables × 5 lags = 30 features

**ventanas**

ventanas= [3, 6, 12, 24] # horas

for variable in variables\_meteo:

for ventana in ventanas:

# Media móvil

df[f'{variable}\_rolling\_mean\_{ventana}h'] = (

df[variable].rolling(window=ventana).mean()

)

# Desviación estándar móvil

df[f'{variable}\_rolling\_std\_{ventana}h'] = (

df[variable].rolling(window=ventana).std()

)

Capturamos tendencias a corto plazo

**Para cambios rápidos :**

# Cambio en el tiempo (1h y 3h)

for variable in variables\_meteo:

df[f'{variable}\_diff\_1h'] = df[variable].diff(1)

df[f'{variable}\_diff\_3h'] = df[variable].diff(3)

ejemplo si baja mucho la presión , en el tiempo que establecimos podría llover

**Total de features creadas: 156**

Categorías:

- Temporales: 9 (hora, mes, cíclicas)

- Variables originales: 6 (temperatura, humedad, etc.)

- Lags: 30 (6 vars × 5 lags)

- Rolling means: 24 (6 vars × 4 ventanas)

- Rolling stds: 24 (6 vars × 4 ventanas)

- Diferencias: 12 (6 vars × 2 intervalos)

**Utilizamos loos targets para hacer este analisis**

**# Crear targets**

**df\_features['target\_precipitacion'] = df\_features['precipitacion\_real'].shift(-1)**

**df\_features['target\_lluvia\_binaria'] = (df\_features['target\_precipitacion'] > UMBRAL\_LLUVIA).astype(int)**

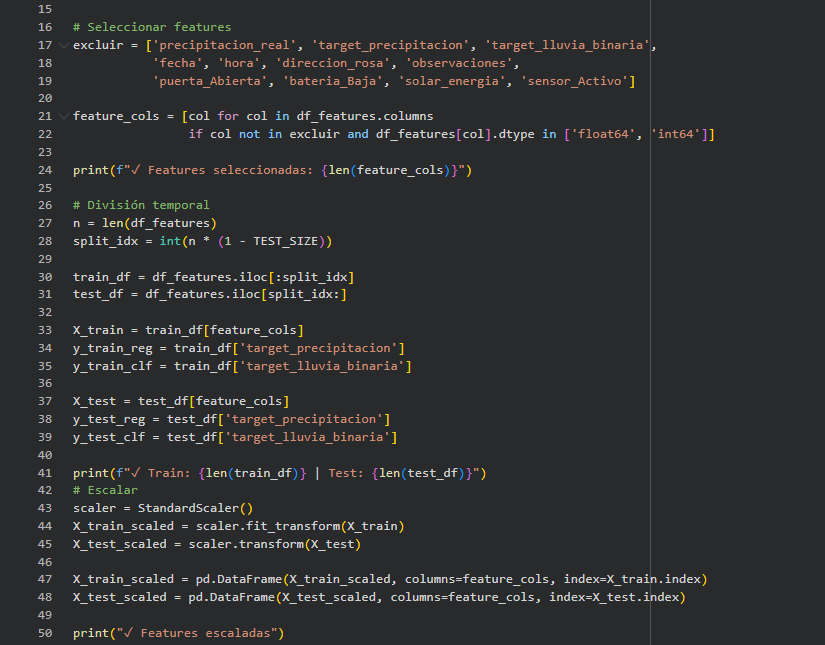
**df\_features = df\_features.dropna(subset=['target\_precipitacion'])**

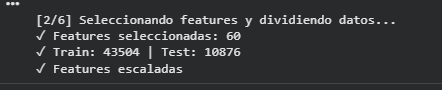
**print(f"✓ Features creadas: {df\_features.shape}")**

**print(f"✓ Distribución target: {df\_features['target\_lluvia\_binaria'].value\_counts().to\_dict()}")**

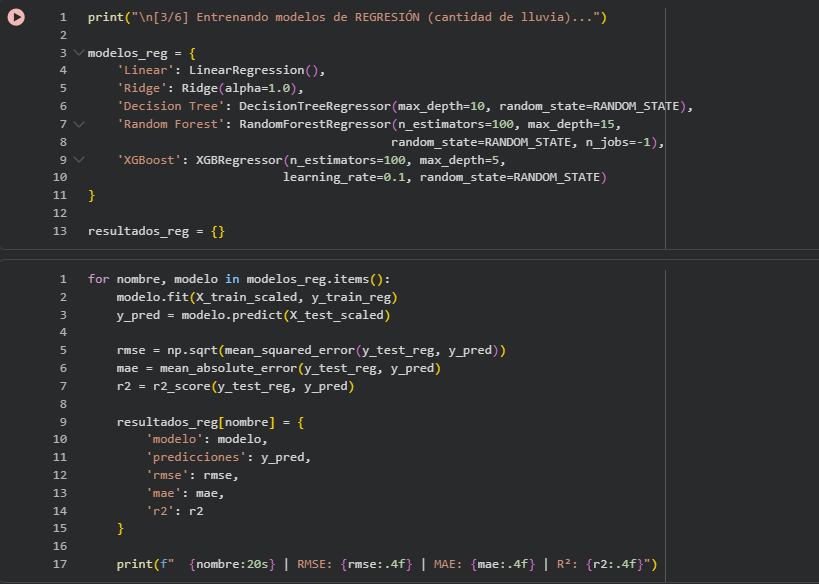
queremos predecir la precipitacion siguiente inmediata por eso el shift(-1)

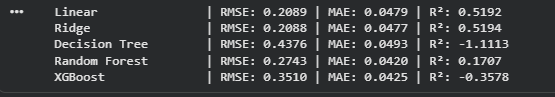
descartamos los targets nulos y con el umbral de lluvia declaramos si llueve o no





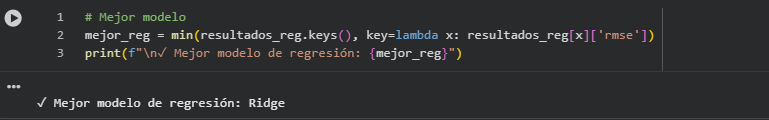
Dividimos usando los features claves , luego escalamos para realmente hacer la division de datos de prueba y reales



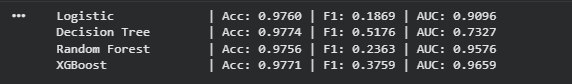


Aca definimos los modelos de REGRESION (variables continuas ) que vamos a utilizar y los corremos para compararlos , tambien definimos las medidas de error con las que vamos a comparar los modelos , son r2 , rmse y mae

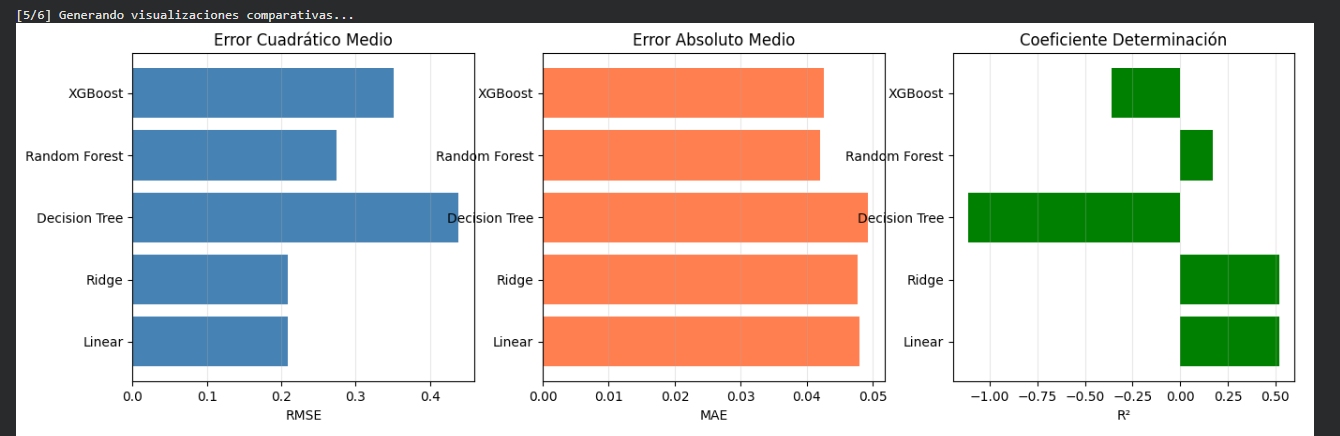
* MAE bajo (~0.04–0.05): en términos absolutos, los errores promedio son pequeños. Si tu precipitación está en mm/hora, esto es bastante bueno.
* RMSE y R²: muestran que solo los modelos lineales captaron bien la variabilidad. Los árboles y boosting no están generalizando.
* R² negativo: significa que el modelo es peor que un baseline trivial (la media). No es aceptable para producción.



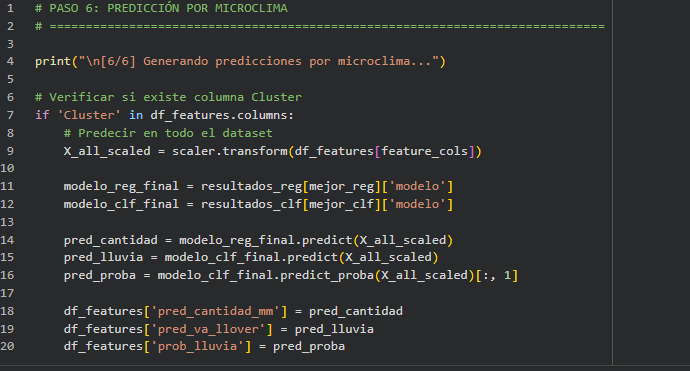


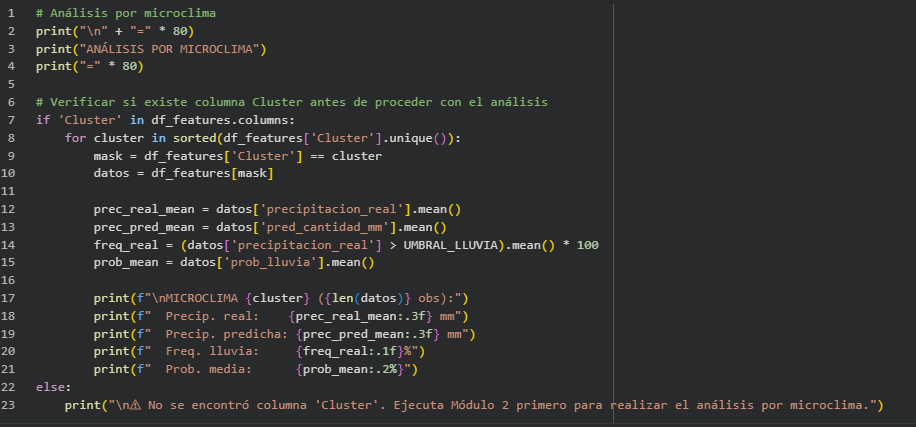


Los entrenamos modelos de CLASIFICACION (variables categóricas ) para predecir la variable lluvia , previamente hayamos los mejores hiperparametros

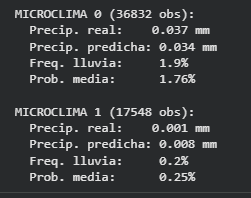


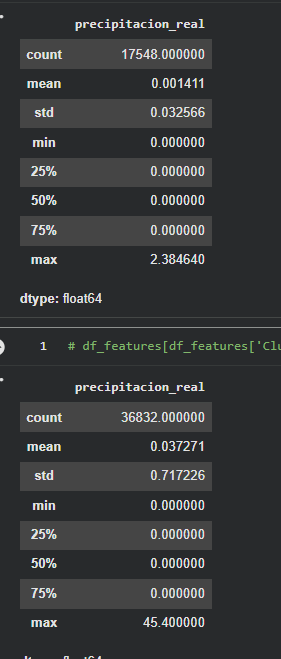
En esta grafica Podemos comparar con la metricas que establecimos , los errores de cada modelo , descartamos los que tengan coeficiente de determinación negativo , es decir , XGBost y árboles de decisión , podríamos escoger el de menor RMSE que este acorde con un MAE entre .04 y .05 o el menor en el mejor caso



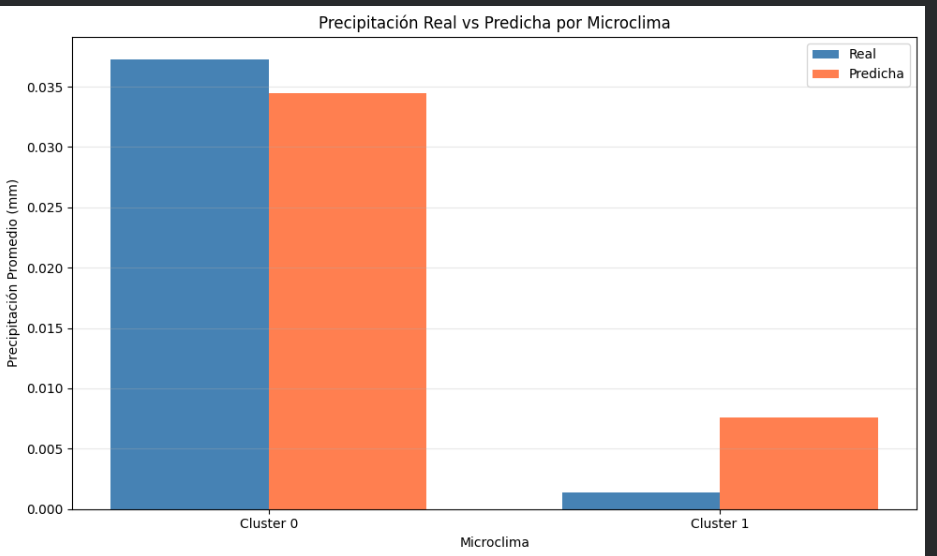
Predecimos por microclima , es decir , utilizamos los cluster del modulo 2 . 

Analizamos el modelo



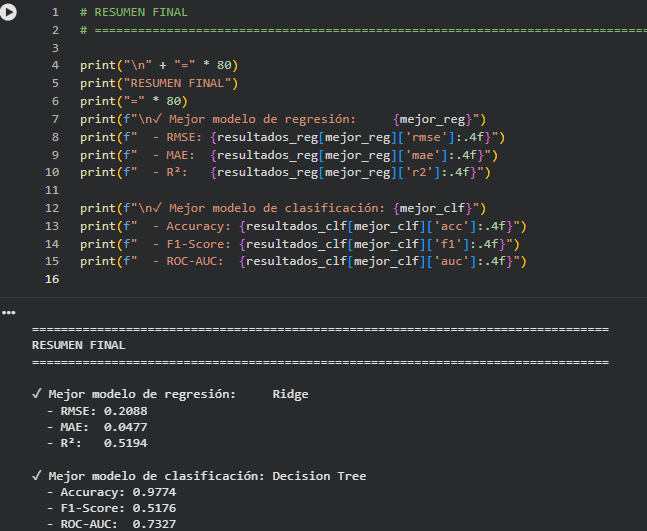


En este caso, con los cluster construidos , podemos analizar estos dos microclimas , en el que básicamente uno divide datos en los que “llovió” y el otro no , tenemos variables métricas de frecuencias en estas dos etapas



Presentamos un comportamiento extraño en el cluster 1 , hay un pequeño sesgo y se predice más milímetros de lluvia en una condición seca de lo que se tiene como real , a pesar de no estar tan erróneo procedemos a evaluar .

Podríamos hacer el entrenamiento desde el principio con referencia en cada microclima o utilizar otro modelo pero bajo las condiciones que tenemos y pruebas realizados , preferimos dejar el modelo así .



Vemos el mejor modelo de regresión y clasificación en la estación evaluada.

Integración de Modulos y analisis de estaciones

**MÓDULO 3 Predicción de Precipitación y Validación del Modelo**

**Enlace a Libro de código :**

<https://colab.research.google.com/drive/1Nli99etLTBH7Cu-BkPRqZcY0HKAMPhGo?usp=sharing>

**Conclusiones !**

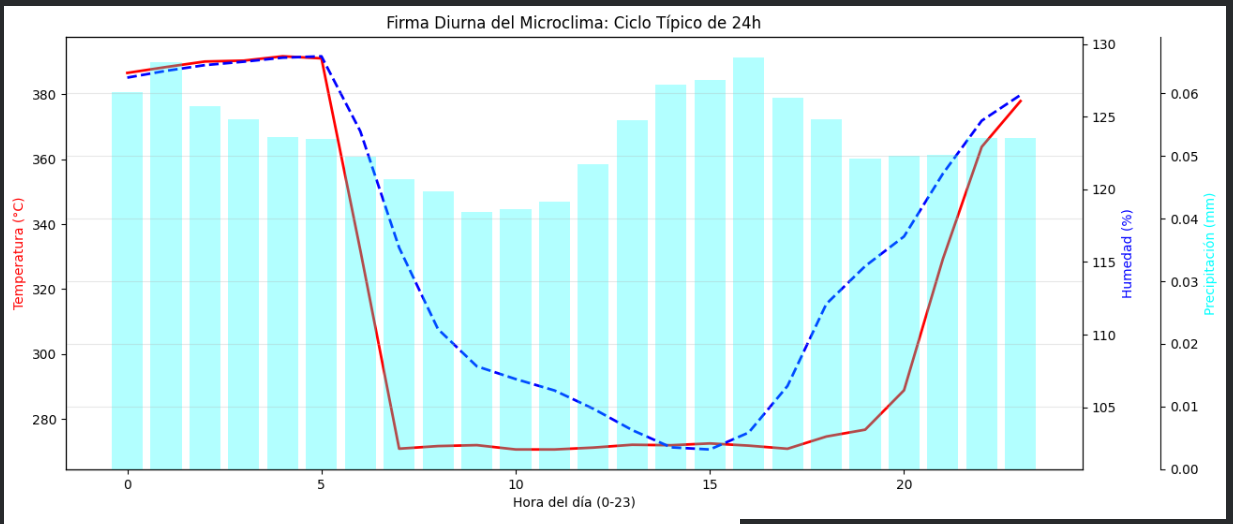
**Encontramos algunos casos de sobreestimación , pero no hay un caso total de overfitting ni errores de predicción demasiado grandes , los modelos parecen bien aplicados , a futuro conviene escalar los cluster para que el entrenamiento sea más específico.**

**Nuestros modelos utilizan la humedad como factor predominante de predicción , nos enfocamos también en la búsqueda de errores en la recolección de datos , como sensores dañados , la correción de esto pudo haber sesgado un poco los modelos .**

**La Estación 20 demoró aproximadamente 50 minutos de ejecución , 21 1 hora y 15 min , 22 1 hora y 20 min, 23 1 hora y 40 minutos, 28 → 40 minutos, 29 → 37 minutos, 30 → alrededor de 10 a 15 minutos, 31 → 52 minutos.**

**Tiempo aproximado de ejecución por estación es de 40 minutos**

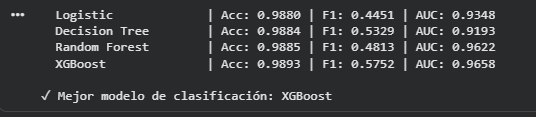
**Estación 20:**

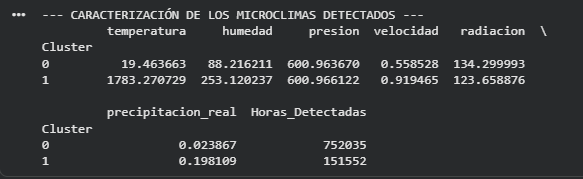
****

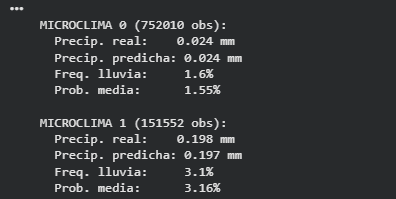
**modelos de Regresión :**

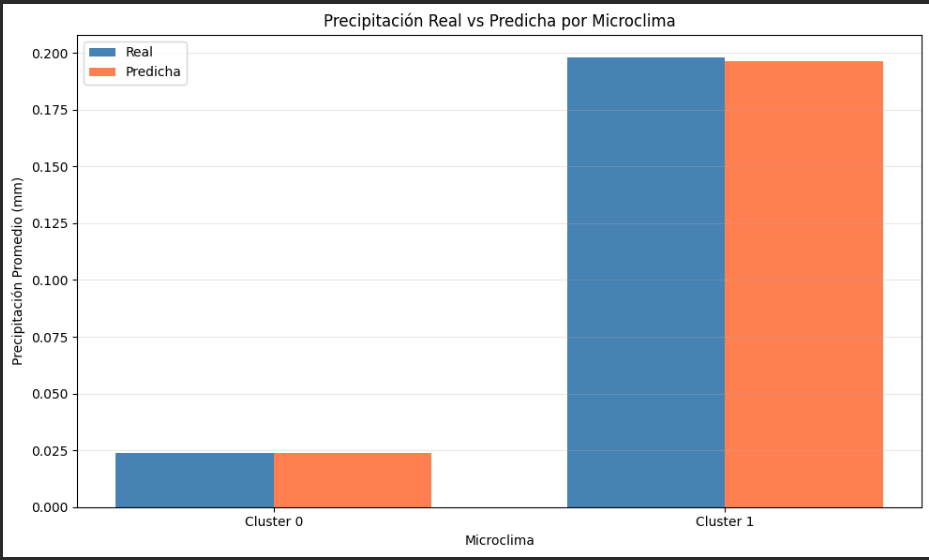
****

**modelos de clasificación :**

****

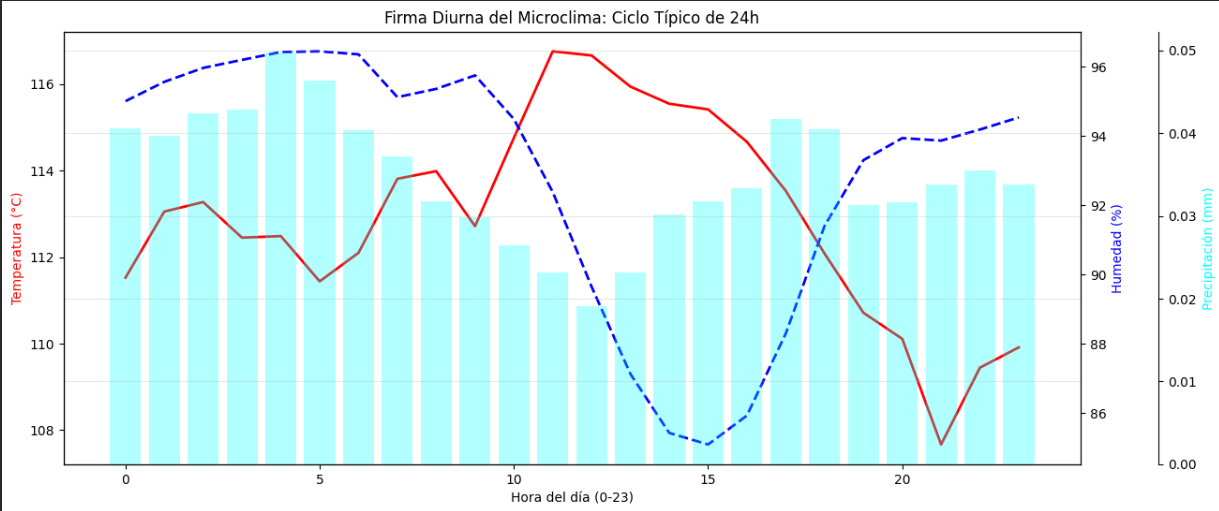
****

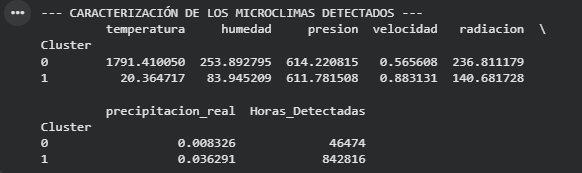
****

****

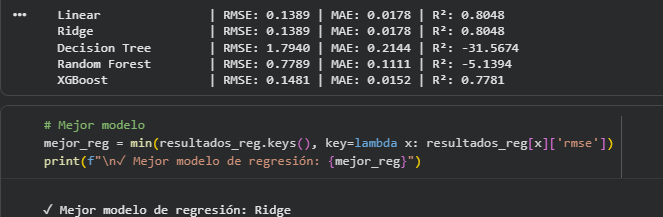
Modelo tan bien entrenado que puede tender al overfitting como se ve en el cluster 0 , la diferencia es muy pequeña

**Estación 21:**

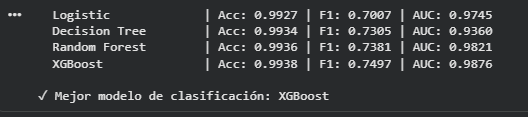
****

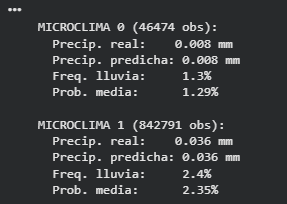
****

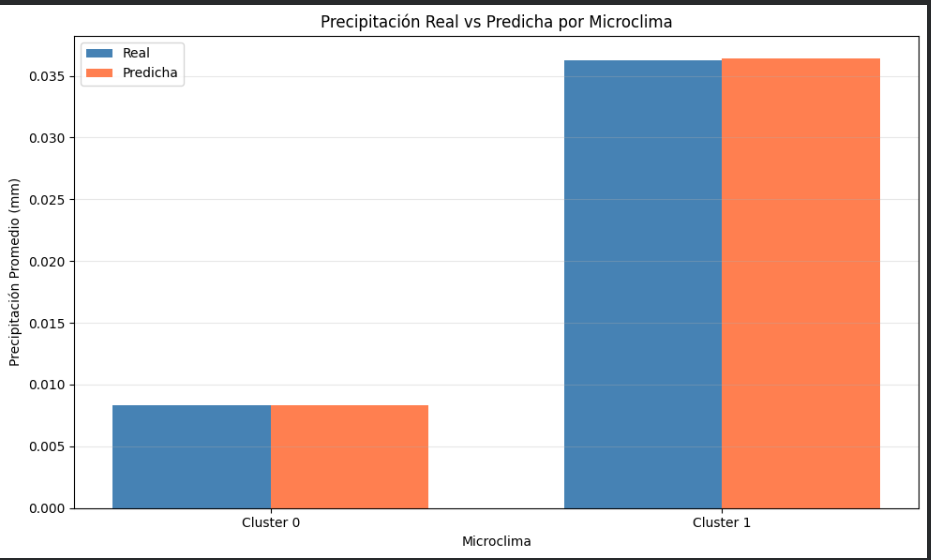
**modelos de Regresión :**

****

**modelos de Clasificación :**

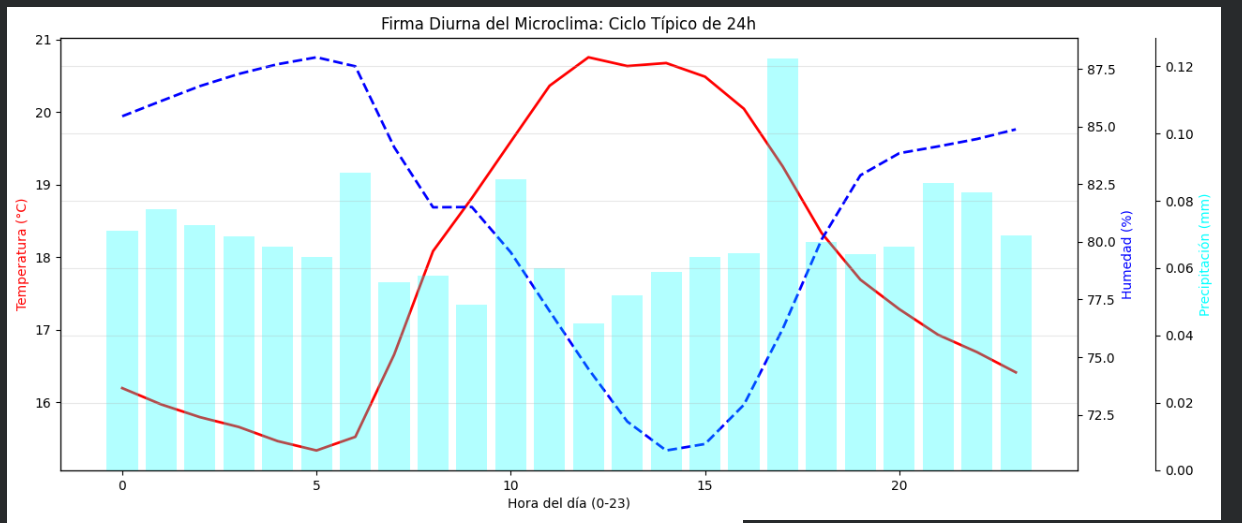
****

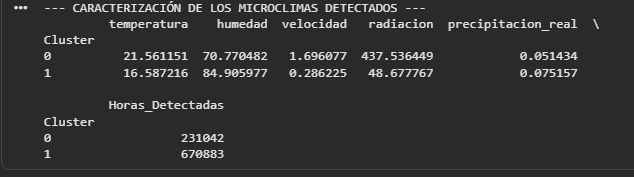
****

****

Modelo tan bien entrenado que puede tender al overfitting como se ve en el cluster 0 y 1 , la diferencia es muy pequeña

**Estación 22:**

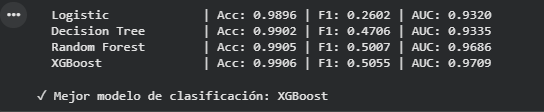
****

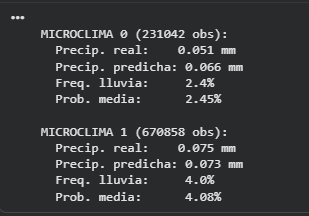
****

**modelos de Regresión :**

****

**modelos de clasificación :**

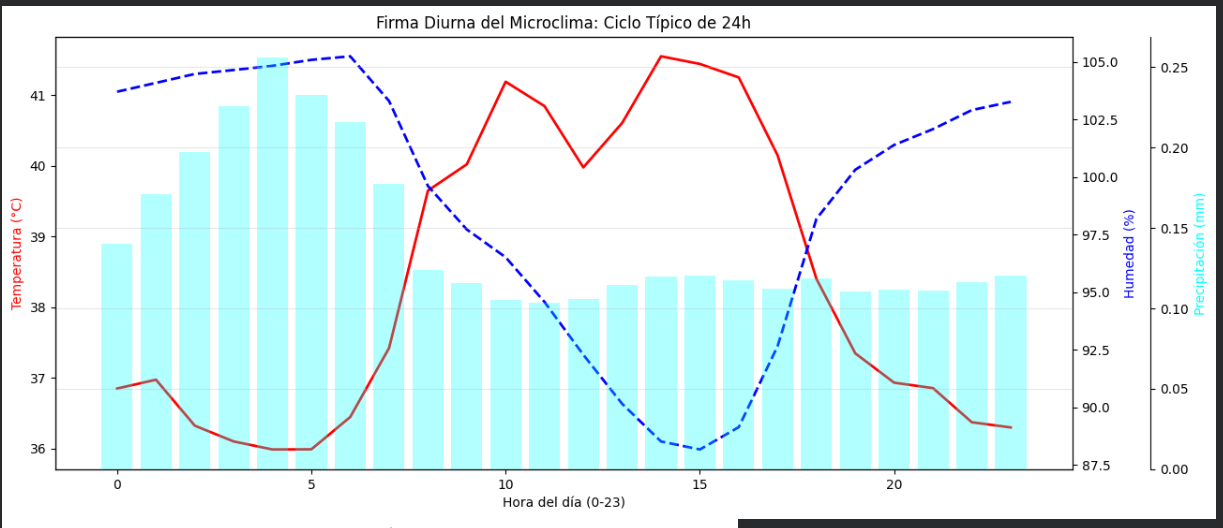
****

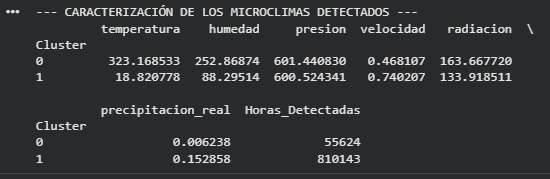
****

****

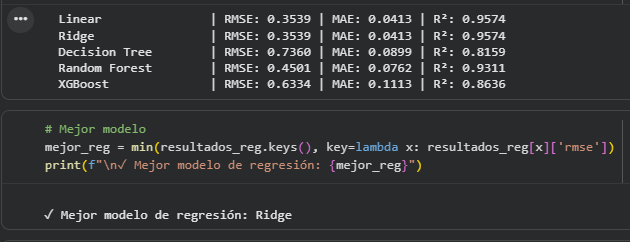
Modelos bien entrenados , un leve sesgo en el cluster 0 , sería conveniente ajustarlo para una mayor precisión

**Estación 23:**

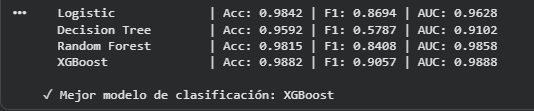
****

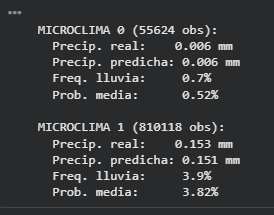
****

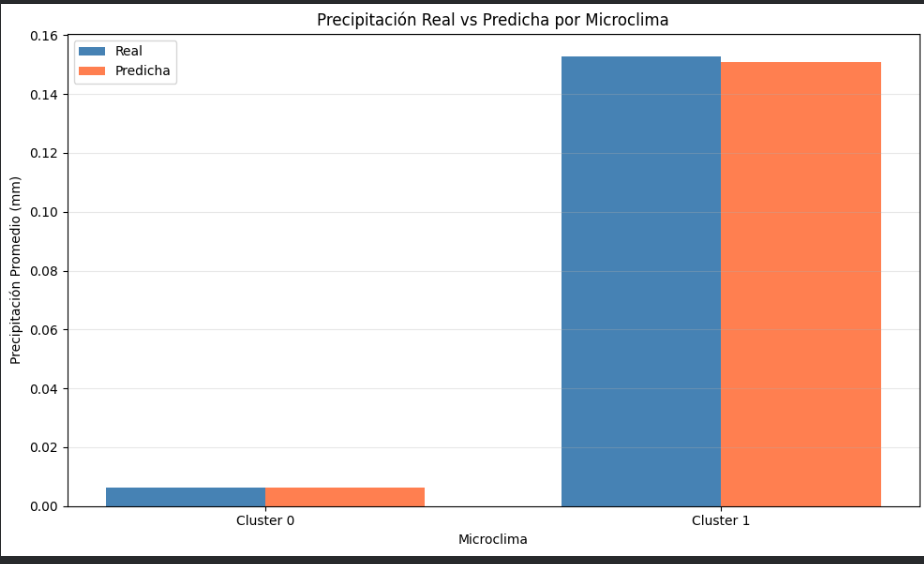
**modelos de Regresión :**

****

**modelos de clasificación :**

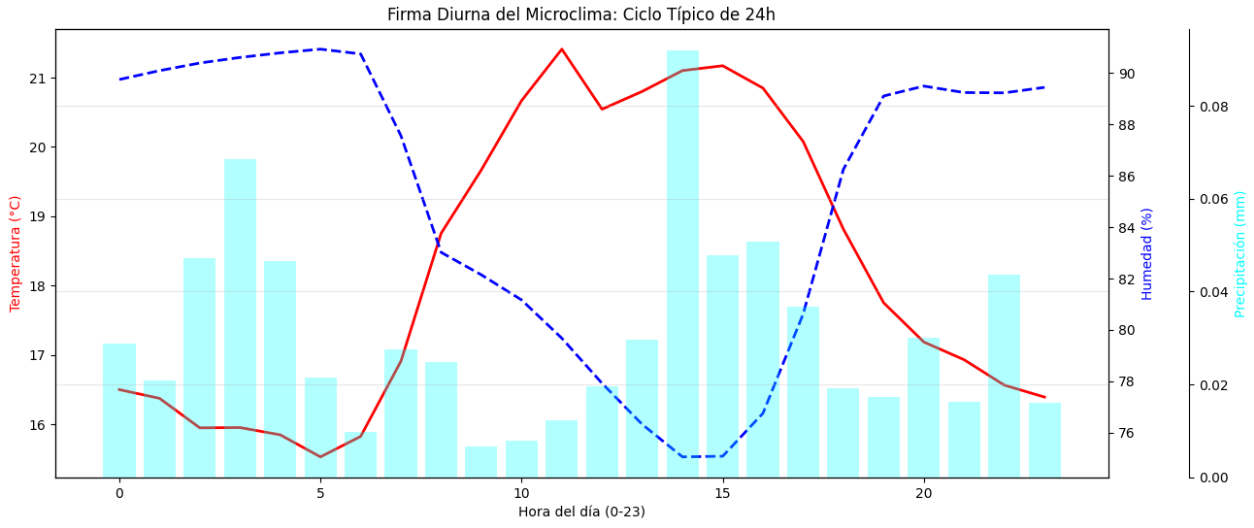
****

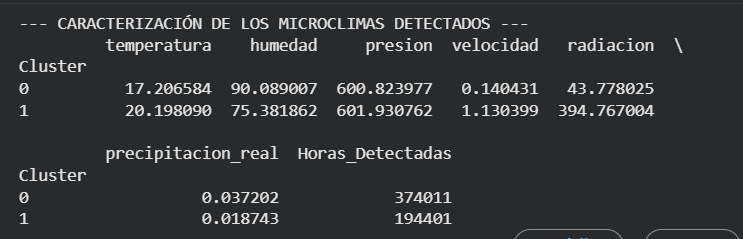
****

****

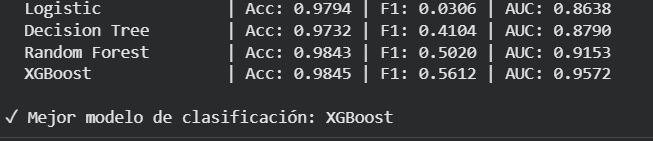
Modelo tan bien entrenado que puede tender al overfitting como se ve en el cluster 0 , la diferencia es muy pequeña

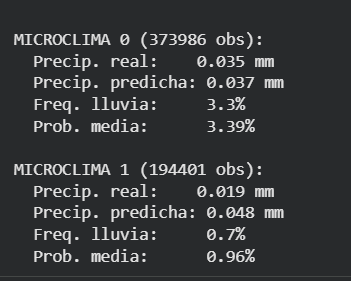
**Estacion 24**

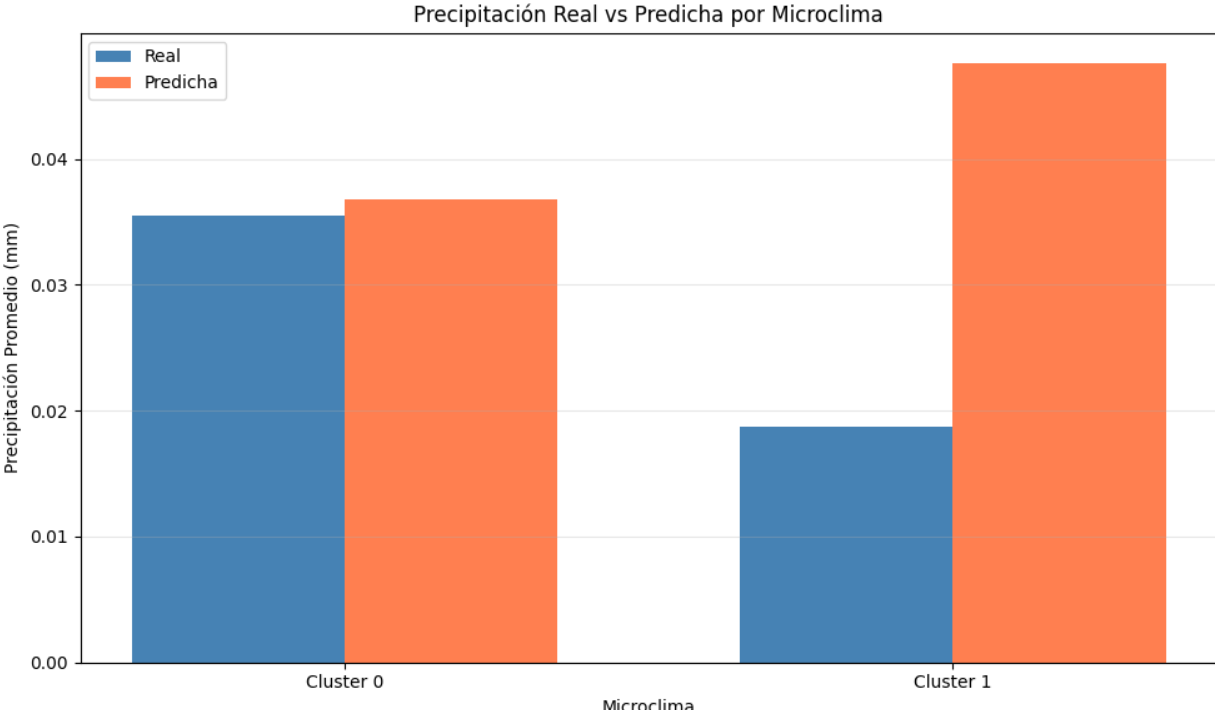
****

****

****

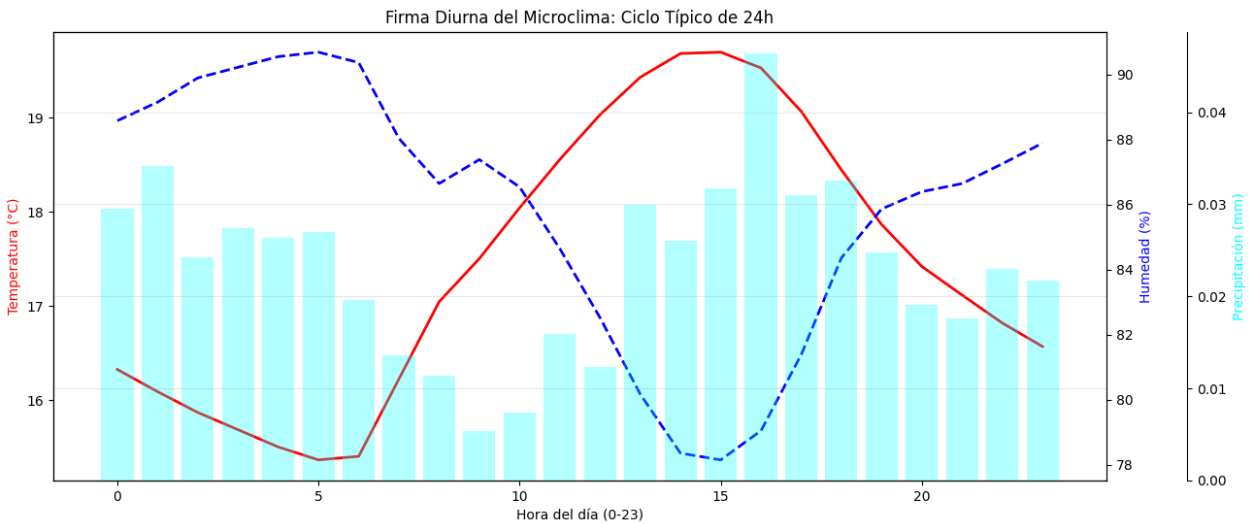
****

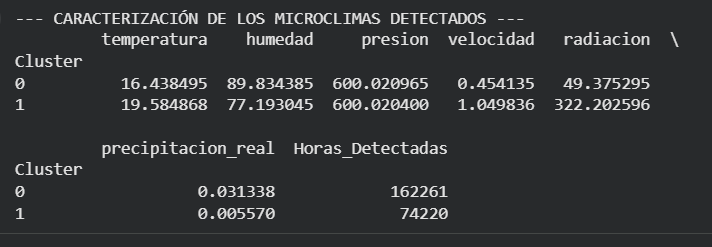
****

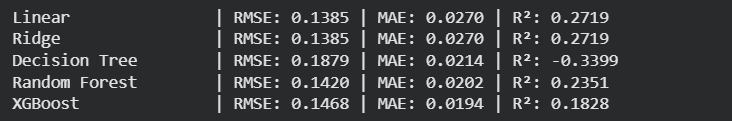
****

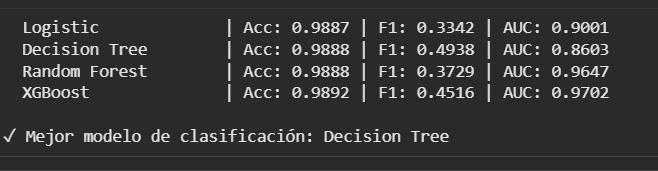
Modelos bien entrenados en detección de eventos, presentan un leve sesgo positivo en el Cluster 0 y una mayor dispersión en el Cluster 1; se sugiere aplicar técnicas de balanceo de datos o regresión regularizada para afinar la precisión en ambos regímenes

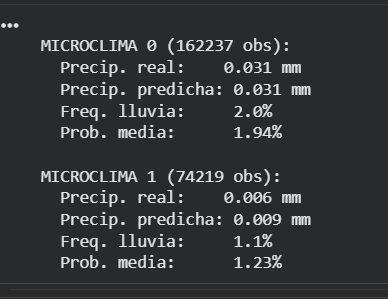
**Estacion 25:**

****

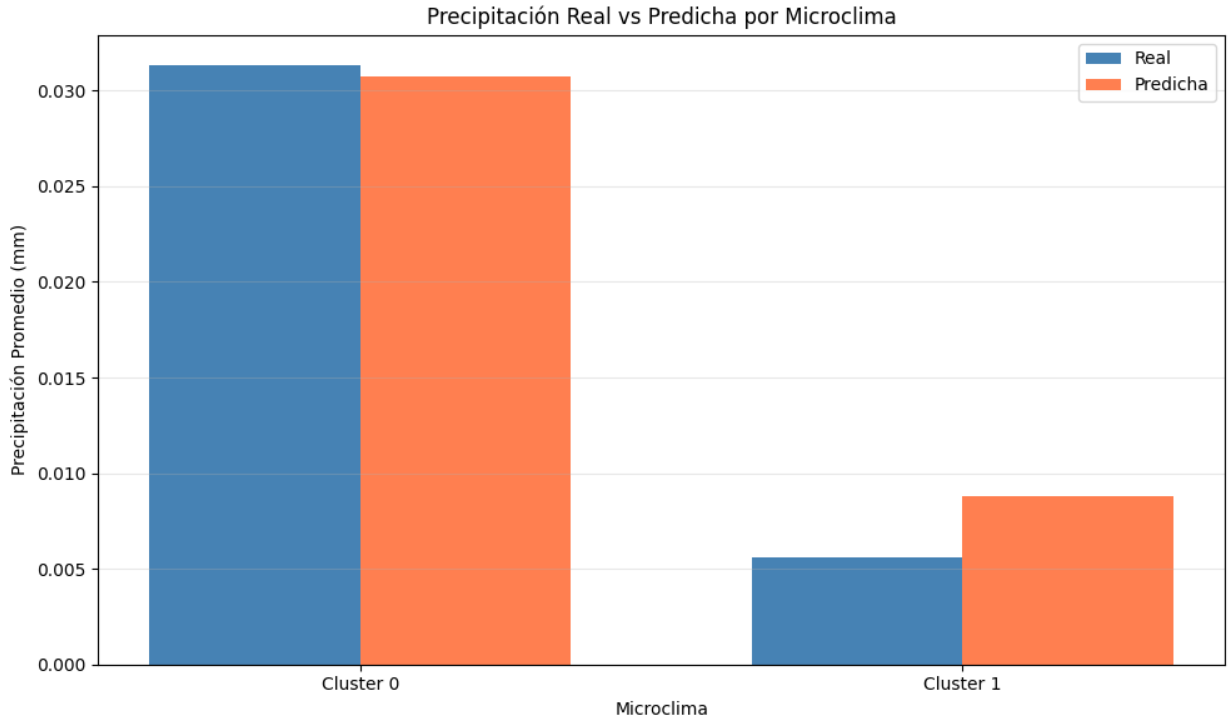
****

****

****

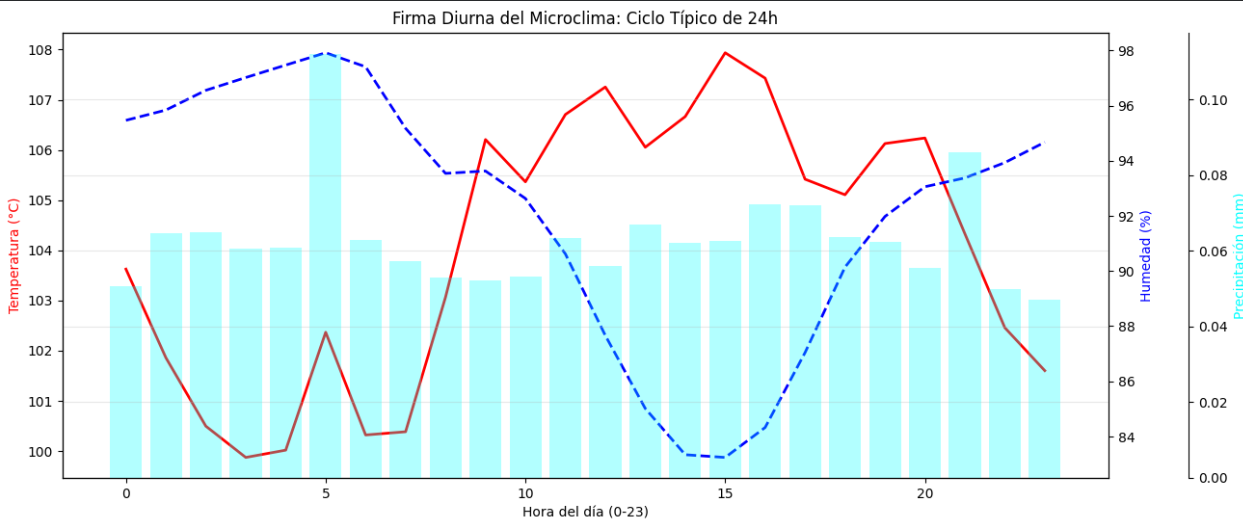
****

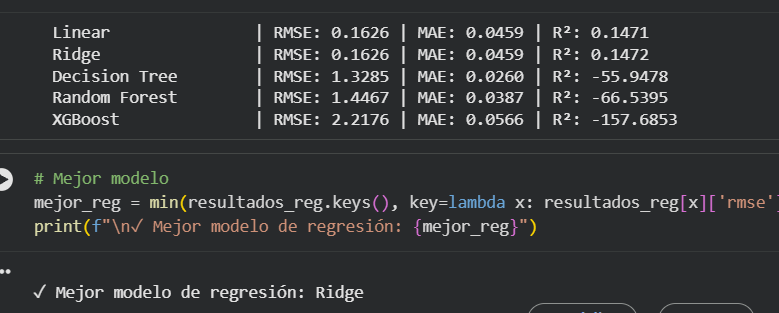
****

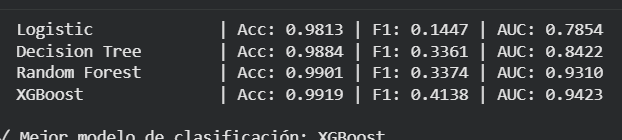
****

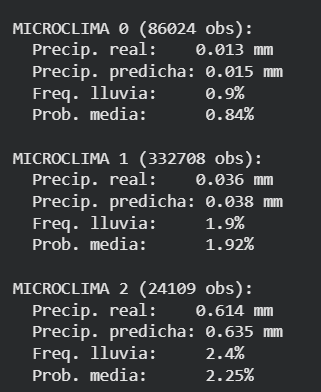
Modelos de clasificación con excelente desempeño (XGBoost, AUC 0.95), aunque los modelos de regresión presentan inestabilidad; se observa una sobreestimación de lluvia en el Cluster 1 (Régimen Seco), por lo que sería conveniente aplicar un filtro previo de clasificación para reducir falsas alarmas.

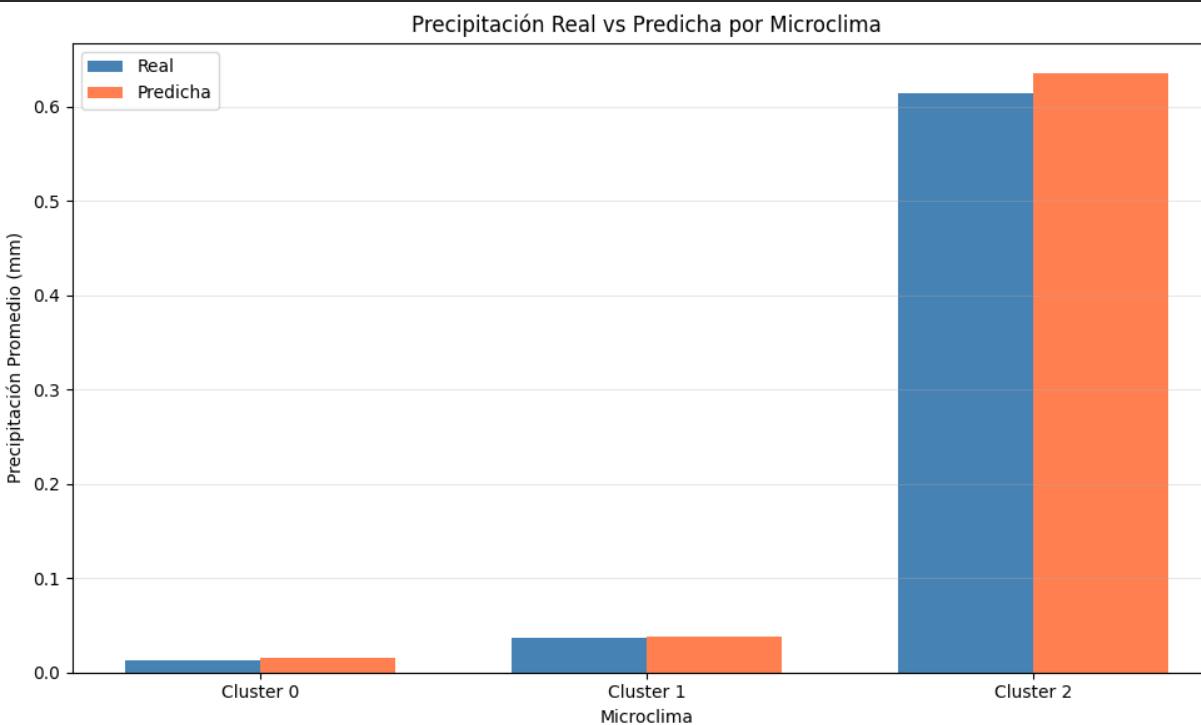
**Estacion 26:**

****

****

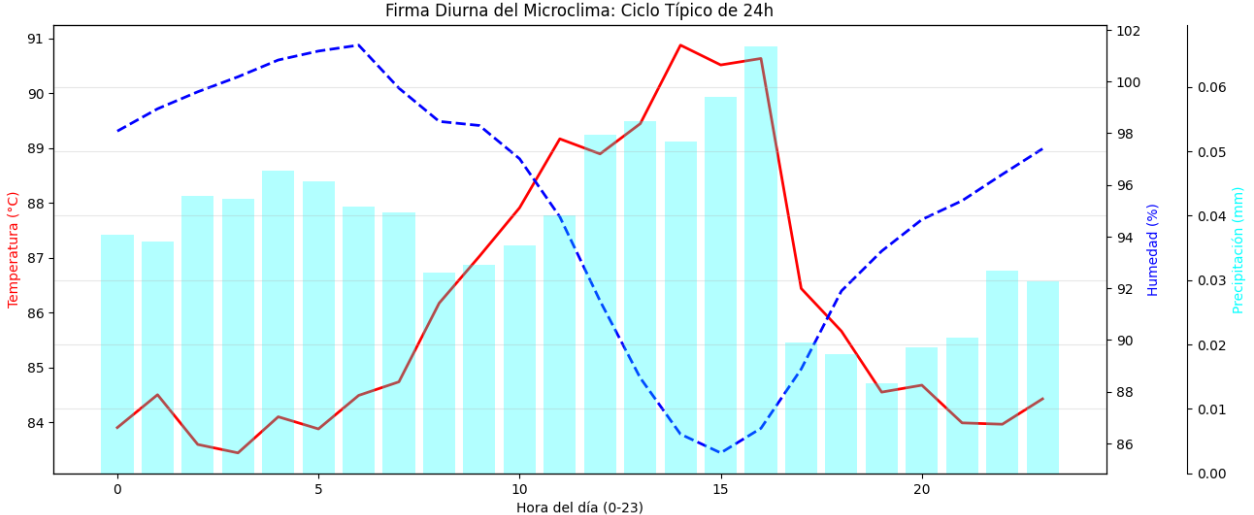
****

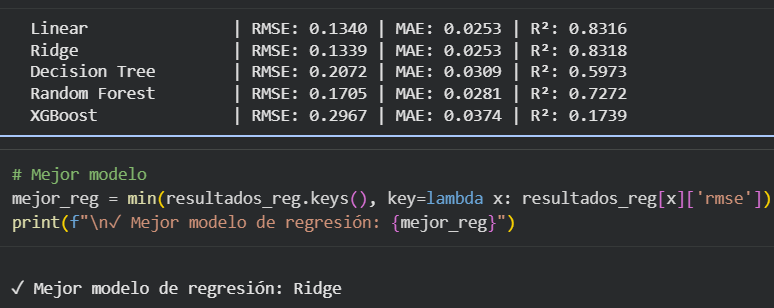
****

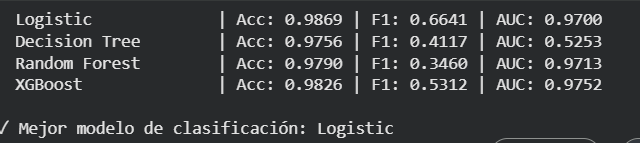
****

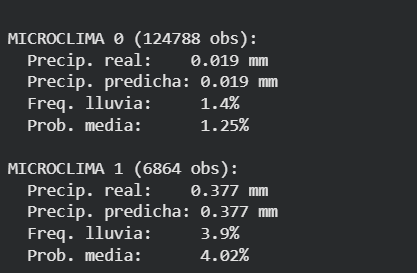
Matriz de confusión sólida con alta capacidad de detección de eventos secos (True Negatives), aunque presenta una tasa de Falsos Positivos que confirma el sesgo del Cluster 1; sería conveniente calibrar el umbral de probabilidad (threshold tuning) para reducir las falsas alarmas, se corrige K=3

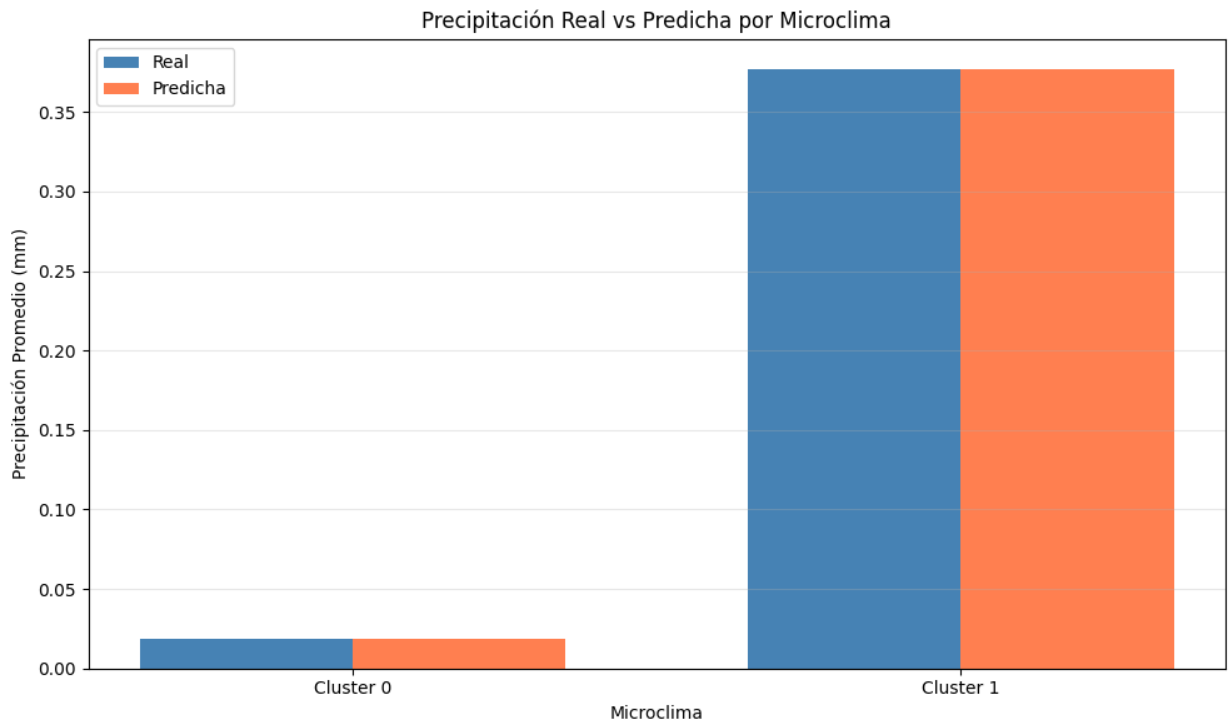
**Estación 27**

****

****

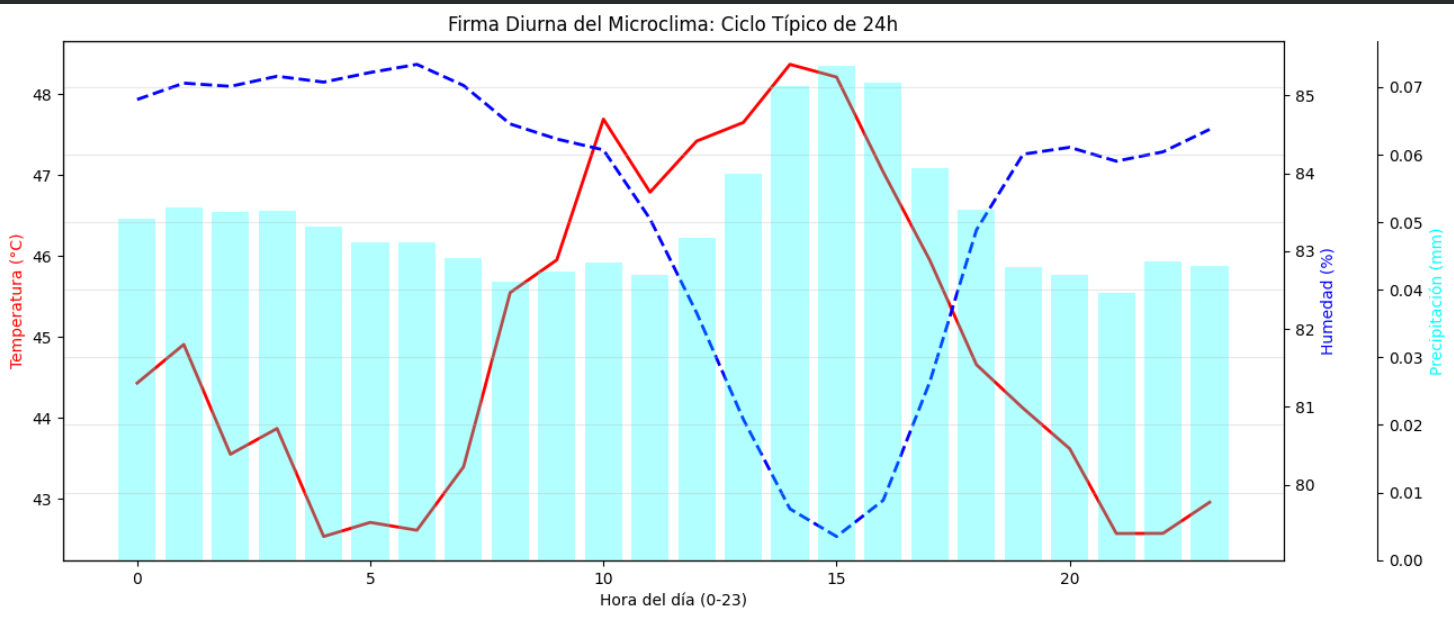
****

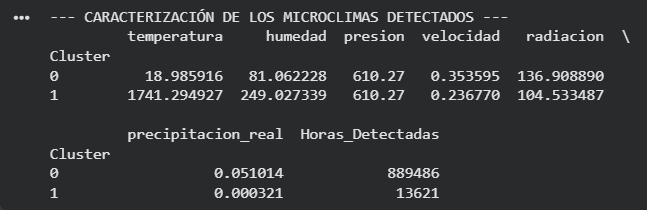
****

****

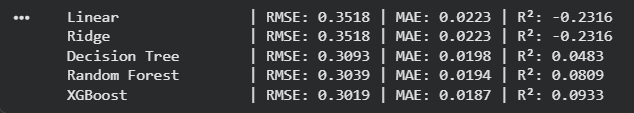
La variable 'Humedad' domina la estructura predictiva del modelo (>40% de importancia), seguida por sus valores históricos ($t-1$), lo cual valida la coherencia física del sistema; sería conveniente asegurar la calibración de los sensores higrométricos, ya que cualquier error en esta variable degradaría drásticamente la predicción.

**Estación 28**

****

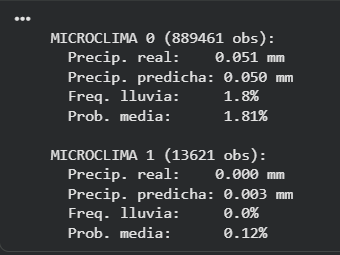
****

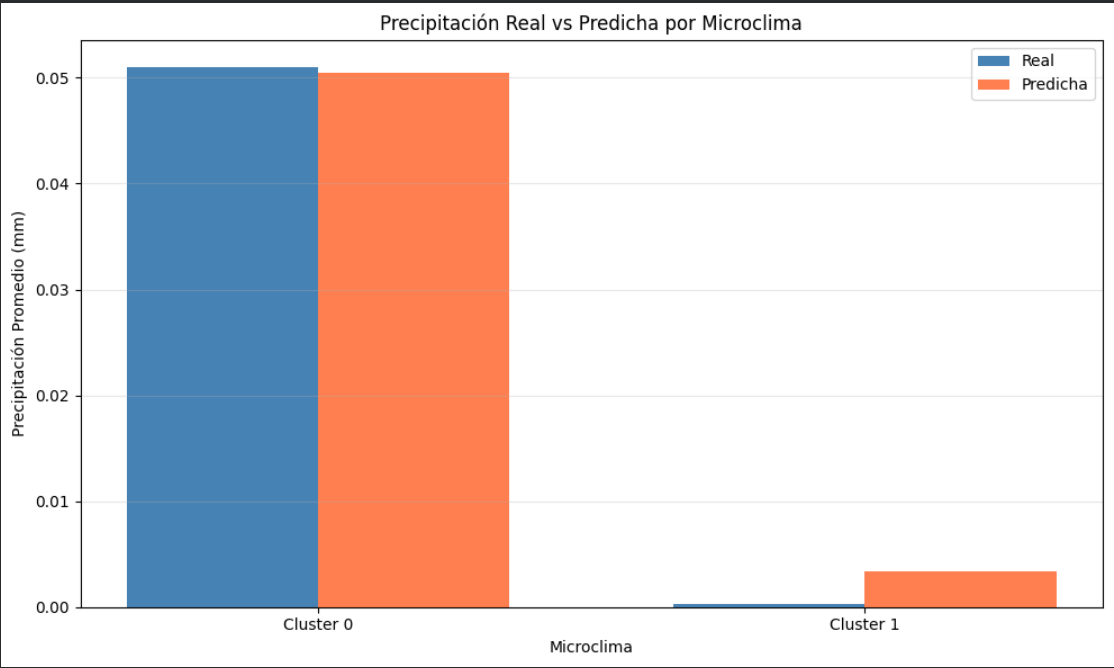
**Modelos de regresión:**

****

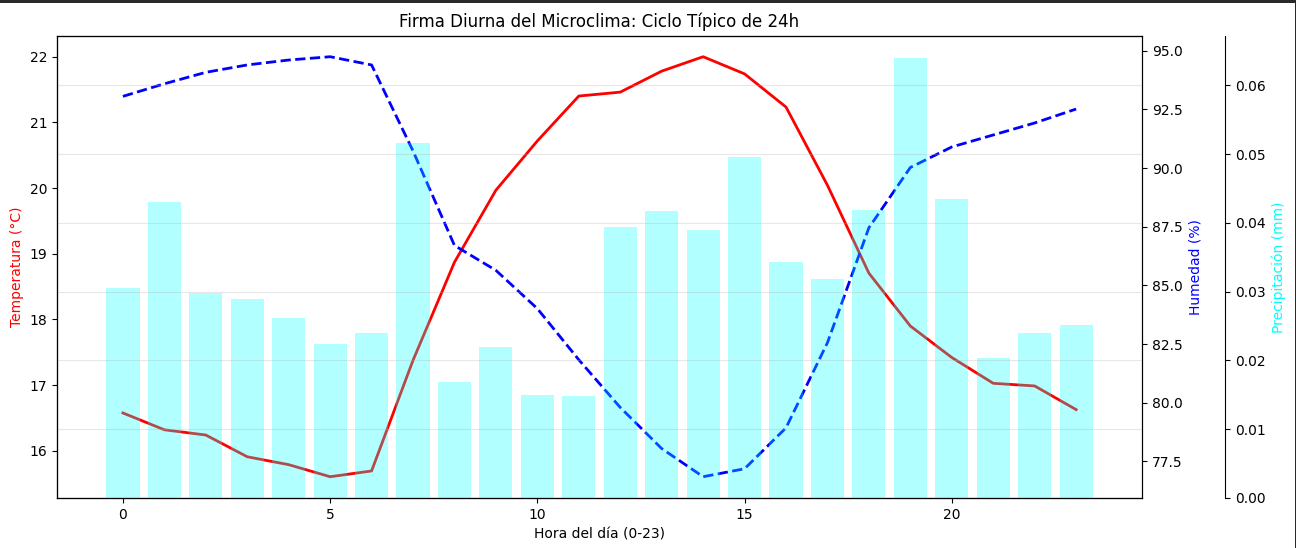
**Modelos de Clasificación:**

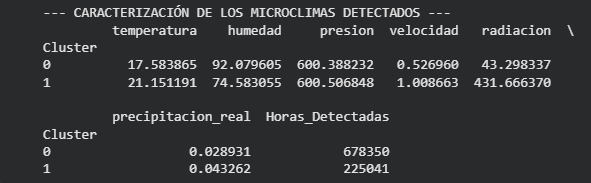
****

****

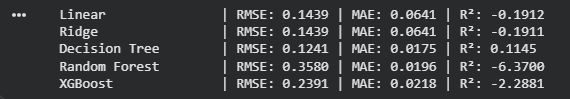
****

**Estacion 29:**

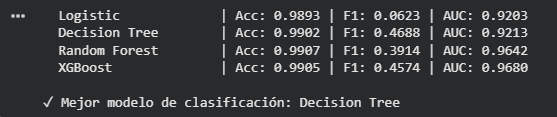
****

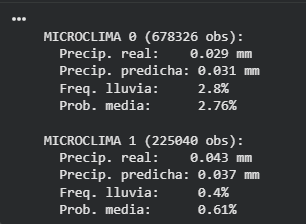
****

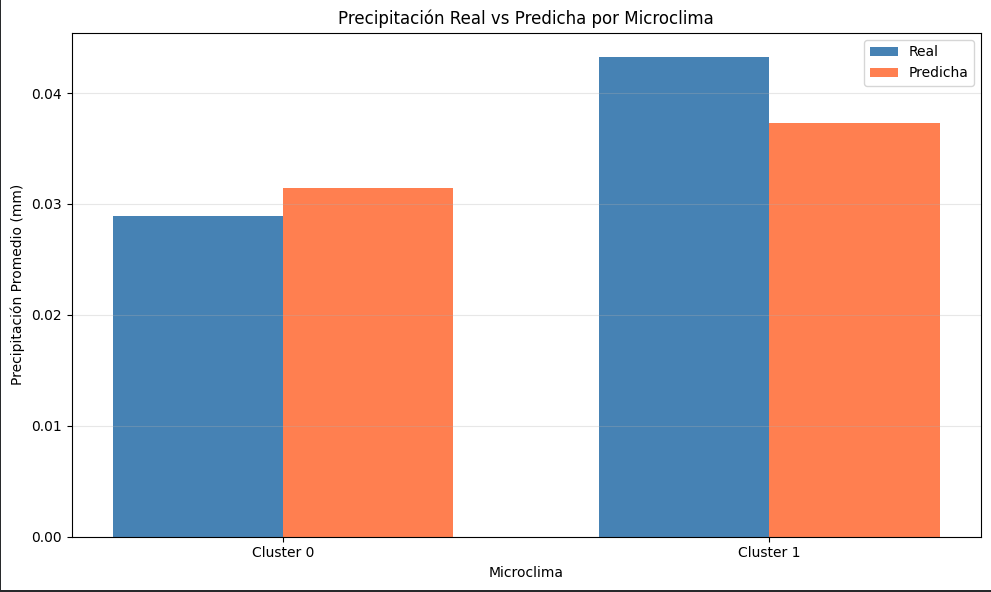
**Modelos de Regresión:**

****

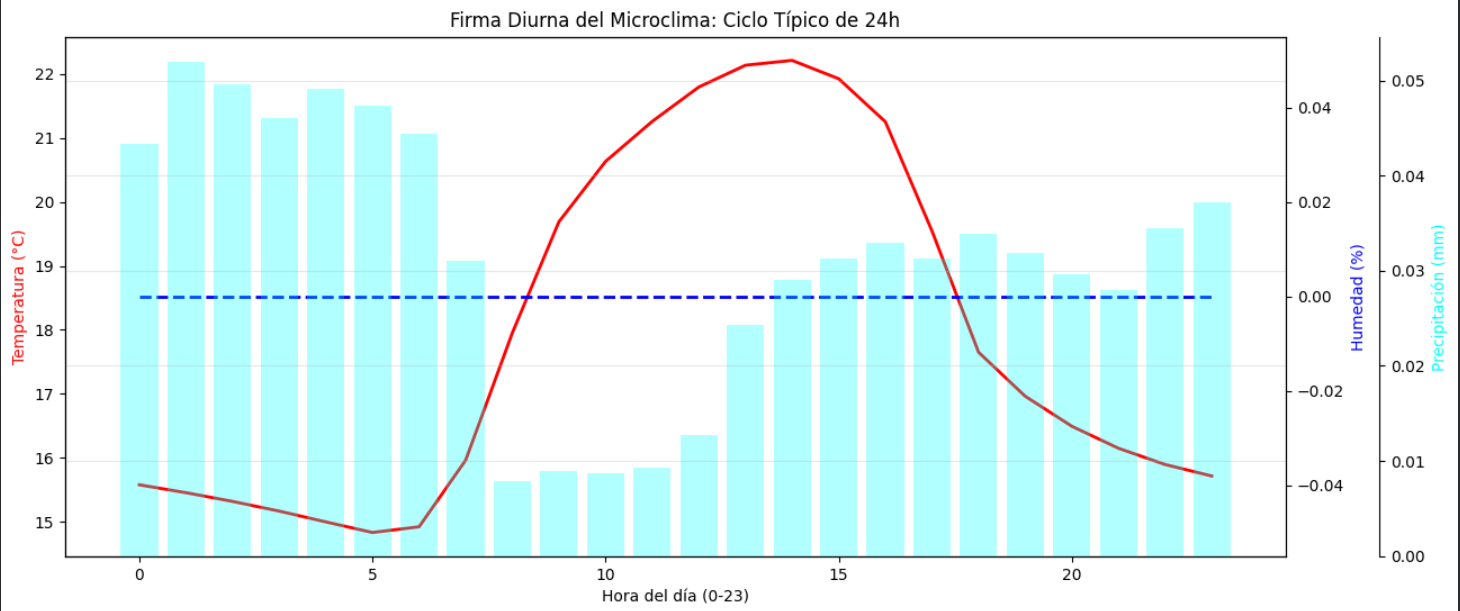
**Modelos de Clasificación:**

****

****

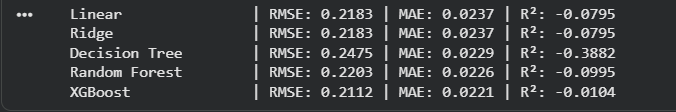
****

**Estacion 30**

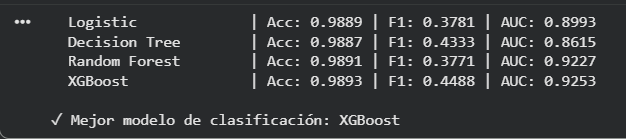
****

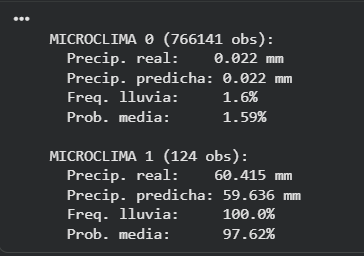
****

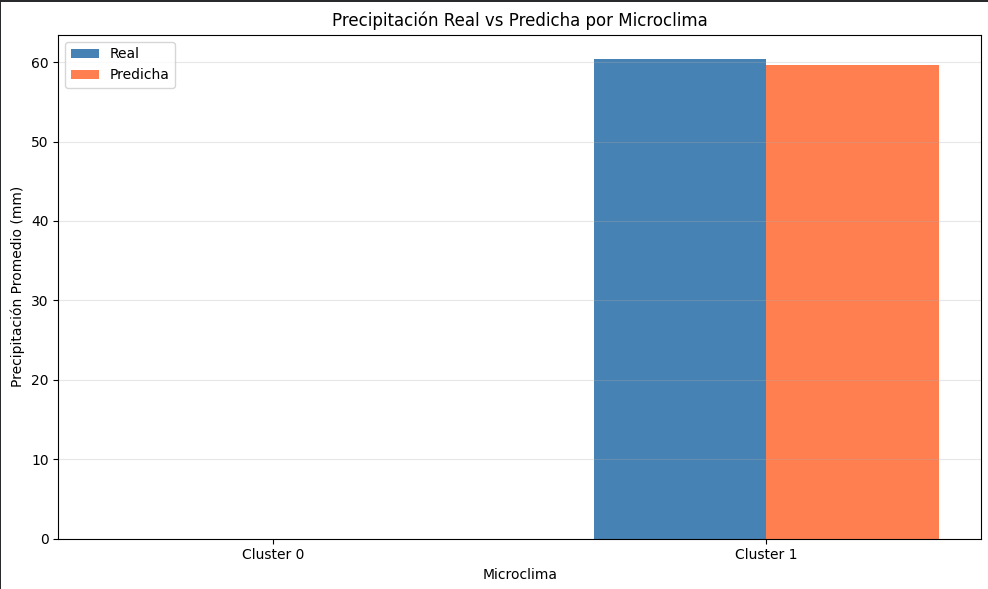
**Modelos de regresión:**

****

**Modelos de Clasificación:**

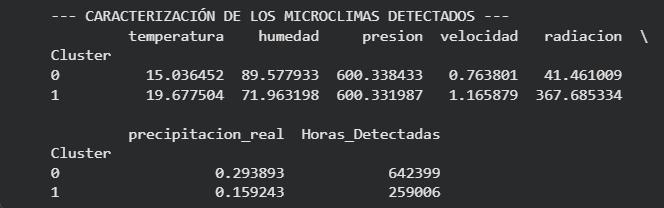
****

****

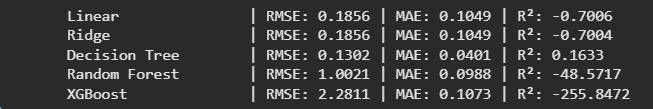
****

**Estacion 31**

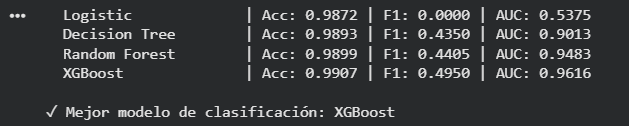
****

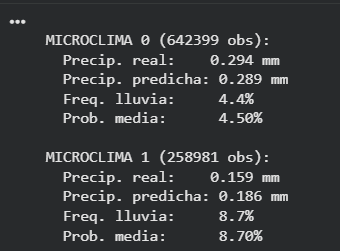
****

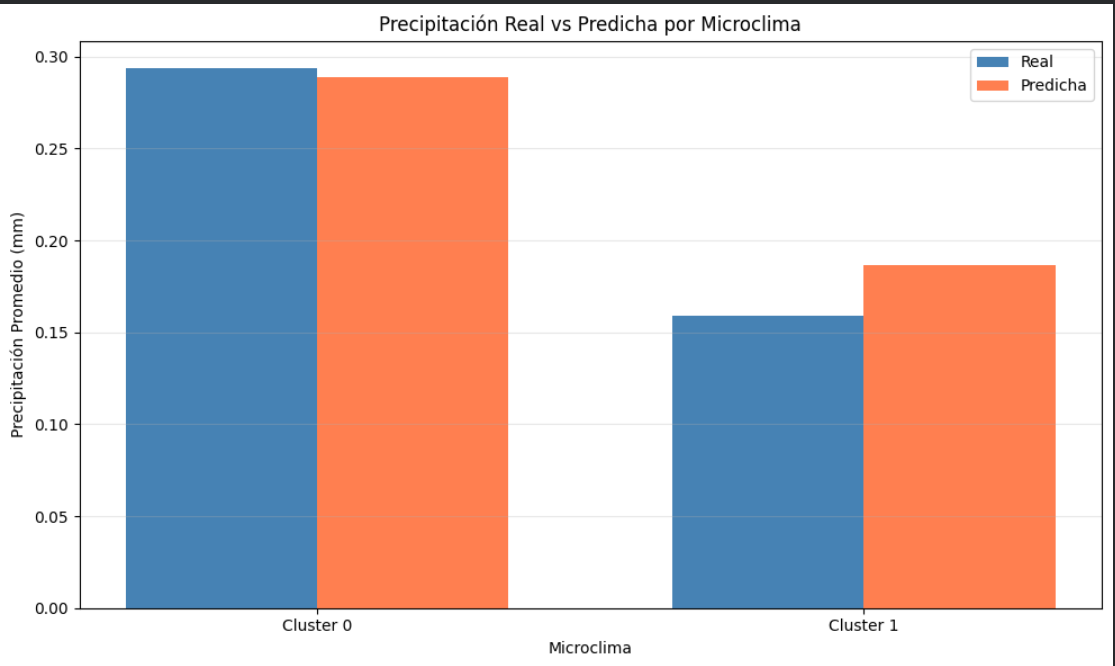
**Modelo de regresión:**

****

**Modelos de Clasificación:**

****

****

****

Informe Estadistico Por Estación

Con el uso de IA generativa , quisimos hacer un informe estadístico final :

Con el script anexado al final de esta pestaña , obtuvimos estos resultados (como ejemplo la estación 19) :

Medidas de tendencia central

* **Media** Promedio aritmético de todas las observaciones. Resume el valor típico, pero es sensible a valores extremos.
* **Mediana** Valor central cuando ordenas los datos. Divide la distribución en dos mitades iguales. Es más robusta frente a outliers.

Medidas de dispersión

* **Std** Desviación estándar: mide cuánto se alejan los valores de la media. Una desviación de ~2.8 indica variabilidad moderada.
* **Q1 y Q3** Primer y tercer cuartil: valores que delimitan el 25% inferior y el 75% superior de los datos.
* **IQR** Rango intercuartílico: diferencia entre Q3 y Q1. Captura la dispersión del 50% central de los datos, menos sensible a outliers.

Forma de la distribució

* **Skewness** Asimetría: mide si la distribución se inclina hacia la izquierda o derecha.
  + Valor positivo → cola hacia la derecha (más valores altos aislados).
  + Aquí 0.53 indica **ligera asimetría positiva**: algunos valores de temperatura más altos que elevan la media.
* **Kurtosis** Curtosis: mide la “aplanamiento” o “concentración” de la distribución respecto a una normal.
* Valor negativo → distribución más plana, con colas menos pesadas.
* Aquí -0.70 indica que la temperatura está más uniformemente distribuida, sin extremos muy frecuentes.

ESTADÍSTICAS DETALLADAS POR VARIABLE

================================================================================

📊 TEMPERATURA:

Media : 17.66

Mediana : 17.00

Std : 2.79

Q1 : 15.39

Q3 : 19.83

IQR : 4.44

Skewness : 0.53

Kurtosis : -0.70

📊 HUMEDAD:

Media : 78.66

Mediana : 81.00

Std : 11.71

Q1 : 72.00

Q3 : 87.00

IQR : 15.00

Skewness : -0.78

Kurtosis : 0.27

📊 PRESION:

Media : 611.30

Mediana : 611.36

Std : 1.50

Q1 : 610.26

Q3 : 612.40

IQR : 2.14

Skewness : -0.19

Kurtosis : -0.34

📊 VELOCIDAD:

Media : 1.21

Mediana : 1.34

Std : 0.98

Q1 : 0.45

Q3 : 1.79

IQR : 1.34

Skewness : 0.81

Kurtosis : 1.31

📊 PRECIPITACION\_REAL:

Media : 0.03

Mediana : 0.00

Std : 0.59

Q1 : 0.00

Q3 : 0.00

IQR : 0.00

Skewness : 55.43

Kurtosis : 3532.12

📊 RADIACION:

Media : 143.42

Mediana : 0.00

Std : 223.82

Q1 : 0.00

Q3 : 232.00

IQR : 232.00

Skewness : 1.76

Kurtosis : 2.51

================================================================================

TABLA MARKDOWN PARA EL INFORME (copia esto):

================================================================================

| Variable | Media | Mediana | Std | Q1 | Q3 | IQR | Skewness | Kurtosis |

|----------|-------|---------|-----|-----|-----|-----|----------|----------|

| Temperatura | 17.66 | 17.00 | 2.79 | 15.39 | 19.83 | 4.44 | 0.53 | -0.70 |

| Humedad | 78.66 | 81.00 | 11.71 | 72.00 | 87.00 | 15.00 | -0.78 | 0.27 |

| Presion | 611.30 | 611.36 | 1.50 | 610.26 | 612.40 | 2.14 | -0.19 | -0.34 |

| Velocidad | 1.21 | 1.34 | 0.98 | 0.45 | 1.79 | 1.34 | 0.81 | 1.31 |

| Precipitacion\_real | 0.03 | 0.00 | 0.59 | 0.00 | 0.00 | 0.00 | 55.43 | 3532.12 |

| Radiacion | 143.42 | 0.00 | 223.82 | 0.00 | 232.00 | 232.00 | 1.76 | 2.51 |

✓ Tabla guardada en: estadisticas\_anexo\_a.csv

================================================================================

INFORMACIÓN ADICIONAL DEL DATASET

================================================================================

📅 Período: 2012-03-28 01:06:28 a 2012-10-24 07:03:23

📅 Duración: 210 días

🌧️ Precipitación Total: 1397.5 mm

🌧️ Promedio por hora: 0.026 mm

🌧️ Máximo: 45.4 mm

🌧️ Horas con lluvia (>0.2mm): 724 (1.3%)

🗺️ Microclimas identificados:

Cluster 0: 36,858 obs (67.7%)

Cluster 1: 17,548 obs (32.3%)

import pandas as pd

import numpy as np

def calculate\_descriptive\_stats(df, columns):

stats = {}

for col in columns:

if col in df.columns:

series = df[col].dropna() # Eliminar NaN para cálculos correctos

q1 = series.quantile(0.25)

q3 = series.quantile(0.75)

iqr = q3 - q1

stats[col] = {

'Media': series.mean(),

'Mediana': series.median(),

'Std': series.std(),

'Q1': q1,

'Q3': q3,

'IQR': iqr,

'Skewness': series.skew(),

'Kurtosis': series.kurtosis()

}

else:

stats[col] = f"Columna '{col}' no encontrada"

return stats

if \_\_name\_\_ == "\_\_main\_\_":

file\_path = 'df\_cluster\_19.csv'

print("=" \* 80)

print("GENERANDO ESTADÍSTICAS PARA EL INFORME")

print("=" \* 80)

try:

df = pd.read\_csv(file\_path)

print(f"\n✓ Archivo cargado: {file\_path}")

print(f"✓ Filas: {len(df):,} | Columnas: {df.shape[1]}")

except FileNotFoundError:

print(f"❌ Error: El archivo '{file\_path}' no se encontró.")

exit()

target\_columns = ['temperatura', 'humedad', 'presion', 'velocidad',

'precipitacion\_real', 'radiacion']

print("\n" + "=" \* 80)

print("ANEXO A: ESTADÍSTICAS DETALLADAS POR VARIABLE")

print("=" \* 80)

stats\_results = calculate\_descriptive\_stats(df, target\_columns)

# Mostrar en formato de tabla

print("\nFormato para visualización:")

print("-" \* 80)

for col, data in stats\_results.items():

if not isinstance(data, str):

print(f"\n📊 {col.upper()}:")

for stat\_name, stat\_value in data.items():

print(f" {stat\_name:20s}: {stat\_value:>10.2f}")

# Crear DataFrame para tabla Markdown

print("\n" + "=" \* 80)

print("TABLA MARKDOWN PARA EL INFORME (copia esto):")

print("=" \* 80)

df\_stats = pd.DataFrame(stats\_results).T

# Formatear valores

for col in df\_stats.columns:

df\_stats[col] = df\_stats[col].apply(lambda x: f"{x:.2f}" if isinstance(x, (int, float)) else x)

# Imprimir tabla Markdown

print("\n| Variable | Media | Mediana | Std | Q1 | Q3 | IQR | Skewness | Kurtosis |")

print("|----------|-------|---------|-----|-----|-----|-----|----------|----------|")

for idx, row in df\_stats.iterrows():

print(f"| {idx.capitalize()} | {row['Media']} | {row['Mediana']} | {row['Std']} | "

f"{row['Q1']} | {row['Q3']} | {row['IQR']} | {row['Skewness']} | {row['Kurtosis']} |")

# Guardar en CSV

output\_file = 'estadisticas\_anexo\_a.csv'

df\_stats.to\_csv(output\_file)

print(f"\n✓ Tabla guardada en: {output\_file}")

# Información adicional del dataset

print("\n" + "=" \* 80)

print("INFORMACIÓN ADICIONAL DEL DATASET")

print("=" \* 80)

if 'datetime' in df.columns:

df['datetime'] = pd.to\_datetime(df['datetime'])

print(f"\n📅 Período: {df['datetime'].min()} a {df['datetime'].max()}")

print(f"📅 Duración: {(df['datetime'].max() - df['datetime'].min()).days} días")

if 'precipitacion\_real' in df.columns:

prec = df['precipitacion\_real']

print(f"\n🌧️ Precipitación Total: {prec.sum():.1f} mm")

print(f"🌧️ Promedio por hora: {prec.mean():.3f} mm")

print(f"🌧️ Máximo: {prec.max():.1f} mm")

lluvia = (prec > 0.2).sum()

print(f"🌧️ Horas con lluvia (>0.2mm): {lluvia} ({lluvia/len(prec)\*100:.1f}%)")

if 'Cluster' in df.columns:

print(f"\n🗺️ Microclimas identificados:")

for cluster in sorted(df['Cluster'].unique()):

count = (df['Cluster'] == cluster).sum()

pct = count / len(df) \* 100

print(f" Cluster {cluster}: {count:,} obs ({pct:.1f}%)")

print("\n" + "=" \* 80)

print("✅ GENERACIÓN COMPLETADA")

print("=" \* 80)

Referencias

pandas# Manipulación de datos

numpy # Cálculos numéricos

scikit-learn # Machine learning

xgboost # Gradient boosting

matplotlib # Visualización

seaborn # Visualización estadística

Fuente: Estación Meteorológica 19-31, Manizales

Frecuencia: Horaria

Variables: 18 columnas meteorológicas y operacionales

Semilla :

RANDOM\_STATE = 42

Todas las operaciones aleatorias en el módulo 3 utilizan esta semilla

-Uso de distintas IA generativas para principalmente el informe estadístico y pruebas de calidad de código ( claude , chatgpt , copilot y deep seek )

Enunciado Del Proyecto

**Minería de Datos.**

El tema central del proyecto es Caracterización de Microclimas y Predicción de Precipitación en Zonas Urbanas Utilizando Algoritmos de Minería de Datos.

Las ciudades, debido a su topografía y desarrollo urbano, a menudo presentan variaciones climáticas significativas en distancias cortas, dando lugar a lo que se conoce como microclimas. Estos microclimas pueden influir directamente en fenómenos meteorológicos localizados, como la precipitación, lo que tiene implicaciones importantes para la gestión de riesgos, la planificación urbana y las actividades cotidianas de los ciudadanos. La capacidad de identificar estos microclimas y predecir la precipitación de manera diferenciada para cada uno de ellos es crucial, ya que los sistemas de pronóstico a gran escala a menudo no capturan estas particularidades.

Este proyecto busca que se aborde esta situación mediante el uso de técnicas de minería de datos, aprovechando datos de estaciones meteorológicas distribuidas en diferentes zonas de la ciudad de Manizales. Los estudiantes deben hacer revisión de trabajos relacionados y proponer y validar un modelo que permita caracterizar estos microclimas y generar predicciones de precipitación. Se deben realizar todas las tareas de preparación de datos posibles y proponer modelos acordes con este tipo de datos.

- Módulo 1: Preprocesamiento y Exploración de Datos: • Recolección y Estructuración: Recuperar y estructurar series de tiempo de variables meteorológicas (ej., Humedad, Precipitación, Radiación Solar, Temperatura) provenientes de múltiples estaciones. La columna estacion\_sk hace referencia al numero de la estación y es su identificador. • Las variables estado,\_Alarma, puerta\_Abierta,bateria\_Baja, solar\_energia, sensor\_Activo, observaciones se refieres a condiciones de la estación y no a medidas meteorológicas. Es posible que las puedan aprovechar para entender algunas situaciones pero no para el análisis de series de tiempo. • Limpieza e Imputación: Identificar y manejar datos faltantes o inconsistencias en las series de tiempo. Proponer, aplicar y evaluar un método de imputación. • Realizar las demás tareas requeridas. • Análisis Exploratorio de Datos (EDA): Realizar un análisis estadístico descriptivo para entender las características de las variables meteorológicas. Aplicar técnicas de correlación, para identificar dependencias temporales e identificar patrones de frecuencia en las series temporales

- Módulo 2: Caracterización de Microclimas: • Agrupamiento (Clustering): Aplicar algoritmos de agrupamiento (ej., K-Means, DBSCAN u otros) a los datos preprocesados para identificar zonas geográficas con patrones climáticos homogéneos, es decir, microclimas. • Determinación del Número Óptimo de Grupos: Implementar y justificar métodos para determinar el número óptimo de clústeres (ej., método del codo, coeficiente de silueta). • Visualización: Generar visualizaciones que permitan interpretar los microclimas identificados.

- Módulo 3: Predicción de Precipitación y Validación del Modelo: • Definir las características relevantes para la predicción. Usar varios métodos • Desarrollo del Modelo de Predicción: Seleccionar y aplicar varios algoritmos de machine learning para predecir la precipitación, considerando la caracterización de microclimas. El modelo debe ser capaz de generar predicciones diferenciadas por microclima. • Entrenamiento y Pruebas: Dividir el conjunto de datos en conjuntos de entrenamiento y prueba para entrenar y evaluar el modelo de predicción. • Evaluación del Modelo: Evaluar el rendimiento del modelo de predicción utilizando métricas adecuadas. • Análisis de Resultados: Interpretar los resultados de las predicciones, identificando posibles tendencias, estacionalidades o anomalías en la precipitación de cada microclima.