

# Proyecto Final

Escenario climático 2025 a partir del análisis estacional de datos de años previos.

Introducción a la Ciencia de Datos

Profesor: Jaime Alejandro Romero Sierra

Integrantes del equipo:

- Lidia Gizem Sánchez Montiel
- Emir Jahaziel Santiago Patricio

27/11/2024

## Introducción

El análisis de datos climatológicos es de gran importancia para diversos sectores, especialmente aquellos cuyo desempeño depende directamente de las condiciones climáticas. Entre ellos, el sector agrícola enfrenta desafíos significativos debido a las variaciones del clima, como la distribución desigual de lluvias o períodos prolongados de sequía. Este proyecto tiene como objetivo analizar los registros de precipitaciones desde el año 2000 en adelante, con el propósito de proporcionar información útil que permita a los agricultores identificar las épocas con lluvias abundantes y escasas, facilitando así una mejor planificación de sus actividades.

La relevancia de este estudio radica en la necesidad de contar con proyecciones climáticas precisas en muchos países, donde la falta de escenarios confiables dificulta la toma de decisiones informadas. Estas proyecciones no solo pueden ayudar a anticipar variaciones estacionales, sino que también son fundamentales para prevenir riesgos asociados a pérdidas económicas, escasez de recursos hídricos y daños en los cultivos. A través de un análisis exploratorio de datos climáticos y la generación de escenarios futuros, este proyecto busca aportar herramientas clave para mitigar el impacto del cambio climático en la agricultura y fomentar la sostenibilidad del sector.

Este proyecto también incluye el diseño de un dashboard interactivo para facilitar la exploración y visualización de los datos analizados.

## Descripción

Debido al cambio climático en varios países, que presentan condiciones que favorecen las épocas de sequía, se presenta un riesgo de disminución de la producción agrícola, provocando cuantiosas pérdidas económicas y vulnerando la seguridad alimentaria, debido a que la pérdida de los cultivos impide satisfacer la demanda de alimentos de las poblaciones y los encarece. Además, los sectores encargados de la producción primaria de los alimentos deben hacer uso de los recursos del entorno que sean importantes para adaptar las formas de cultivo a las nuevas condiciones climáticas, como lo es el uso de pesticidas, fertilizantes, incluso ampliando las zonas de cultivo, dañando a los ecosistemas.

Esta adaptación está aumentando el riesgo de contribuir a la contaminación, ya que muchos de los sistemas que se establecen, generan desechos tóxicos para la

biodiversidad y para la salud humana al mismo tiempo que devastan los hábitats naturales e impiden el reciclaje normal de nutrientes. En México particularmente, este es un gran problema porque más del 82% de la producción agrícola es de temporal y sólo el 18% es de riego. El principal problema de la desinformación sobre las condiciones del clima radica en la incertidumbre de selección de fechas de siembra que permitan disminuir el uso de recursos como el agua obtenida de mantos acuíferos o cuerpos de agua importantes y aprovechar el agua de las temporadas de lluvias.

En muchos países no se cuenta con información que permita la proyección de escenarios climatológicos robustos, es decir, que abarquen extensos periodos de tiempo para generar certeza de las variaciones climáticas estacionales que permitan adoptar las medidas adecuadas para proteger los cultivos agrícolas de importancia y evitar pérdidas económicas y de escasez de recursos básicos, por mencionar algunos.

## Justificación

El análisis de datos climatológicos a lo largo de los años permitirá proporcionar beneficios al sector de producción agrícola, ofreciendo información relevante sobre las temporadas anuales que resultan más adecuadas para llevar a cabo los diferentes procesos de producción de los diferentes cultivos. Esta información ayudará a maximizar el rendimiento de las cosechas y a planificar de manera más eficiente, lo que es crucial para la seguridad alimentaria y la economía agrícola. Es importante también saber las posibles modificaciones que puede sufrir la temperatura a través de la comprensión de patrones para tomar las medidas de prevención adecuadas contra las sequías provocadas por temperaturas extremas sin tener que hacer uso de recursos del entorno que aumenten los niveles de contaminación en los ecosistemas y que provoquen daños extra. Esto tiene un doble impacto: por un lado, se protege la producción agrícola y, por otro, se contribuye a la preservación del entorno natural. Todo lo anterior incluso podrá beneficiar a los diferentes objetivos de la agenda 2030 para el desarrollo sostenible que es un plan de acción de la ONU, que busca mejorar la vida de las personas y el planeta. Consta de 17 objetivos de Desarrollo Sostenible (ODS), y a los que el proyecto puede contribuir serían los siguientes: ODS 2 (Hambre Cero), que tiene que ver con la seguridad alimentaria, el ODS 12 (Producción y Consumo Responsables), que busca una administración sostenible de los recursos naturales, y el ODS 13 (Acción por el Clima), que busca implementar medidas urgentes para combatir el cambio climático y sus efectos.

## Hipótesis iniciales

- Existe una correspondencia entre el incremento de la temperatura y las emisiones de CO<sub>2</sub>, siendo más importantes en las temporadas de primavera y verano.
- En los últimos años ha habido un incremento en la temperatura en todas las estaciones hasta de 1°C.
- La temporada de lluvias se ha recorrido al menos un mes en los últimos años.

## Recursos disponibles

### Tecnología y herramientas:

Se hará uso del lenguaje de programación “Python” y de la biblioteca Pandas para el manejo de los datos. De igual manera, será necesario usar las bibliotecas de generación de gráficos estadísticos en Python como Matplotlib y Seaborn. Para la elaboración del Dashboard, se hizo uso de las librerías Streamlit y Pickle.

## Fuente de datos

Para este proyecto, se utilizó la base de datos titulada “**Climate Insights Dataset**”, la cual recopila registros numéricos relacionados con diversas variables climáticas. Esta base de datos incluye información detallada sobre tendencias y mediciones climáticas, abarcando un amplio rango temporal y geográfico, lo que la hace adecuada para un análisis exhaustivo del comportamiento del clima en las últimas décadas.

### Características del Dataset:

- **Rango temporal:** 1678 a 2022.
- **Cobertura geográfica:** Incluye datos de múltiples ubicaciones y países.
- **Variables:**
  1. **Fecha:** Registro de tiempo de cada observación.
  2. **Ubicación:** Identificación geográfica específica.
  3. **País:** Clasificación a nivel nacional.
  4. **Temperatura (°C):** Medición de la temperatura ambiente.
  5. **Emisiones de CO<sub>2</sub> (ppm):** Concentración de dióxido de carbono en la atmósfera.

6. **Aumento del nivel del mar (mm):** Variación en el nivel del mar a lo largo del tiempo.
7. **Precipitaciones (mm):** Cantidad de lluvia acumulada.
8. **Humedad (%):** Porcentaje de humedad relativa en el aire.
9. **Velocidad del viento (km/h):** Intensidad del viento en kilómetros por hora.

#### **Cantidad de Datos:**

- **Cantidad original:** 10,000 registros.

## **Definición de Stakeholders Clave**

- **Actores gubernamentales:** Impacto directo en la implementación de los programas de apoyo a la producción agrícola y a la salud alimentaria.
- **Agricultores:** Quienes harán uso de las recomendaciones y financiamiento para verificar los tiempos de siembra.
- **Usuarios de los cultivos en la industria de la transformación:** Compran, usan y transforman la materia prima y asignan un valor agregado al producto agrícola.

## **Preguntas Clave**

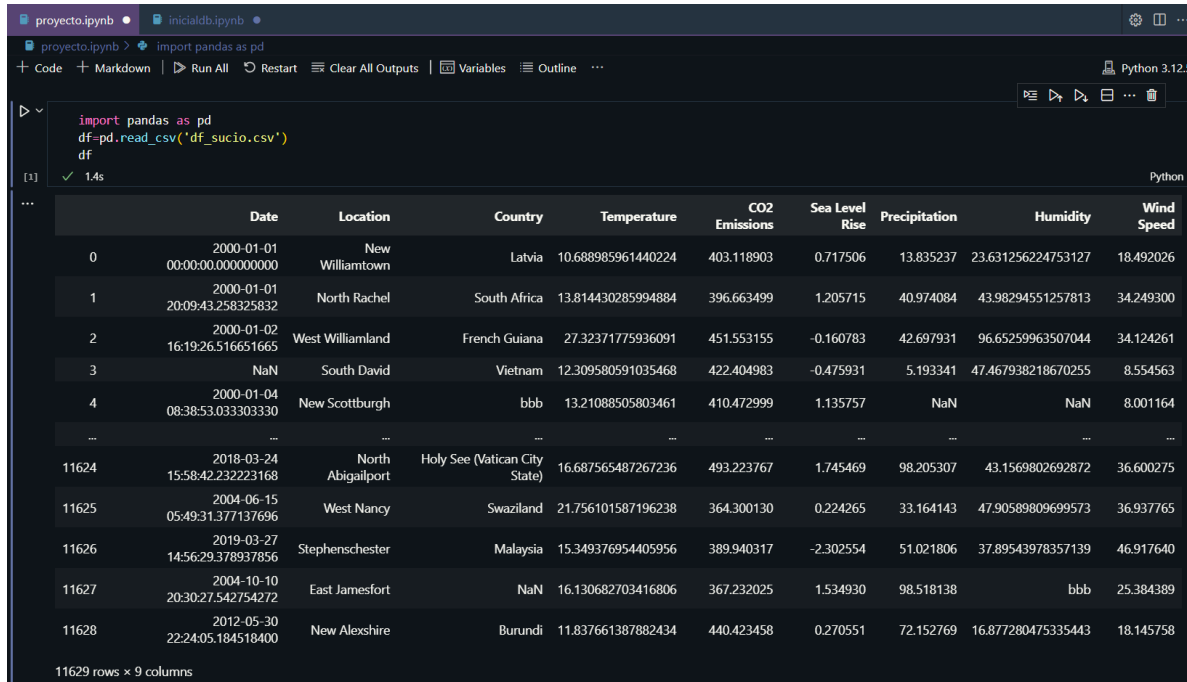
- ¿Cómo y en qué magnitud ha sido la modificación que ha sufrido el periodo de lluvias anual en el tiempo?
- ¿Cuál ha sido el incremento en la temperatura a lo largo de los años?
- ¿Las emisiones de CO2 condicionan el aumento de temperatura durante todo el año?

## **Limpieza de Datos**

#### **Objetivo.**

Desarrollar habilidades en el preprocesamiento de datos, incluyendo la identificación y tratamiento de valores faltantes, datos duplicados, y formatos inconsistentes en una base de datos.

## 1\_ Recepción de la base de datos sucia.



The screenshot shows a Jupyter Notebook interface with a code cell containing the following Python code:

```
import pandas as pd
df=pd.read_csv('df_sucio.csv')
df
```

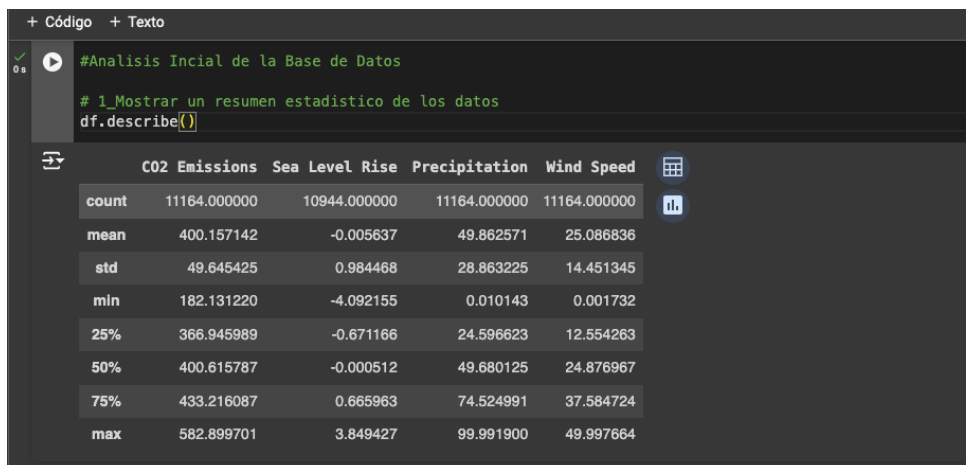
The output of the code is a DataFrame with 11629 rows and 9 columns. The columns are: Date, Location, Country, Temperature, CO2 Emissions, Sea Level Rise, Precipitation, Humidity, and Wind Speed. The data is displayed in a table format with some rows highlighted.

	Date	Location	Country	Temperature	CO2 Emissions	Sea Level Rise	Precipitation	Humidity	Wind Speed
0	2000-01-01 00:00:00.000000000	New Williamtown	Latvia	10.688985961440224	403.118903	0.717506	13.835237	23.631256224753127	18.492026
1	2000-01-01 20:09:43.258325832	North Rachel	South Africa	13.814430285994884	396.663499	1.205715	40.974084	43.98294551257813	34.249300
2	2000-01-02 16:19:26.516651665	West Williamland	French Guiana	27.32371775936091	451.553155	-0.160783	42.697931	96.65259963507044	34.124261
3	NaN	South David	Vietnam	12.309580591035468	422.404983	-0.475931	5.193341	47.467938218670255	8.554563
4	2000-01-04 08:38:53.033303330	New Scottburgh	bbb	13.21088505803461	410.472999	1.135757	NaN	NaN	8.001164
...	...	...	...	...	...	...	...	...	...
11624	2018-03-24 15:58:42.232223168	North Abigailport	Holy See (Vatican City State)	16.687565487267236	493.223767	1.745469	98.205307	43.1569802692872	36.600275
11625	2004-06-15 05:49:31.377137696	West Nancy	Swaziland	21.756101587196238	364.300130	0.224265	33.164143	47.90589809699573	36.937765
11626	2019-03-27 14:56:29.378937856	Stephenschester	Malaysia	15.349376954405956	389.940317	-2.302554	51.021806	37.89543978357139	46.917640
11627	2004-10-10 20:30:27.542754272	East Jamesfort	NaN	16.130682703416806	367.232025	1.534930	98.518138	bbb	25.384389
11628	2012-05-30 22:24:05.184518400	New Alexshire	Burundi	11.837661387882434	440.423458	0.270551	72.152769	16.877280475335443	18.145758

11629 rows x 9 columns

## 2\_ Análisis inicial de la base de datos.

- Resumen estadístico de los datos.



The screenshot shows a Jupyter Notebook interface with a code cell containing the following Python code:

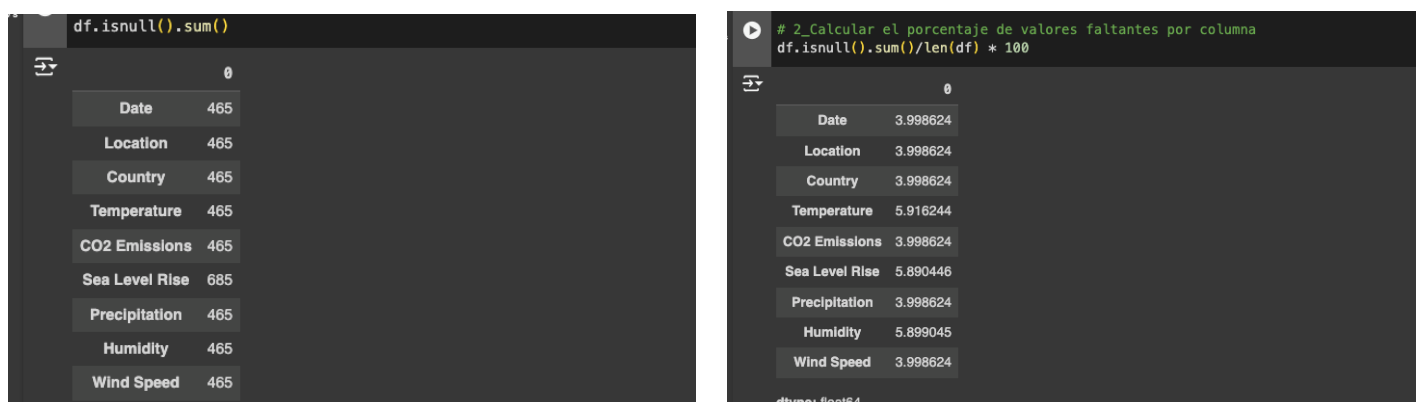
```
#Análisis Inicial de la Base de Datos
# 1_Mostrar un resumen estadístico de los datos
df.describe()
```

The output of the code is a statistical summary of the data, displayed in a table format.

	CO2 Emissions	Sea Level Rise	Precipitation	Wind Speed
count	11164.000000	10944.000000	11164.000000	11164.000000
mean	400.157142	-0.005637	49.862571	25.086836
std	49.645425	0.984468	28.863225	14.451345
min	182.131220	-4.092155	0.010143	0.001732
25%	366.945989	-0.671166	24.596623	12.554263
50%	400.615787	-0.000512	49.680125	24.876967
75%	433.216087	0.665963	74.524991	37.584724
max	582.899701	3.849427	99.991900	49.997664

Este código nos permitió ver dónde se concentran el 25%, 50% y 75% de los datos numéricos, así como el promedio, la desviación estándar y los valores mínimo y máximo del dataframe.

- Cálculo del porcentaje de valores faltantes por columna.



The screenshot shows a Jupyter Notebook interface with two code cells. The first code cell contains the following Python code:

```
df.isnull().sum()
```

The output of the first code cell is a table showing the count of missing values for each column.

	0
Date	465
Location	465
Country	465
Temperature	465
CO2 Emissions	465
Sea Level Rise	685
Precipitation	465
Humidity	465
Wind Speed	465

The second code cell contains the following Python code:

```
# 2_Calcular el porcentaje de valores faltantes por columna
df.isnull().sum()/len(df) * 100
```

The output of the second code cell is a table showing the percentage of missing values for each column.

	0
Date	3.998624
Location	3.998624
Country	3.998624
Temperature	5.916244
CO2 Emissions	3.998624
Sea Level Rise	5.890446
Precipitation	3.998624
Humidity	5.899045
Wind Speed	3.998624

### 3. Identificación de filas duplicadas.

```
#Verificar si hay registros duplicados
df.duplicated()
[10] ✓ 0.0s

... 0      False
     1      False
     2      False
     3      False
     4      False
     ...
    11624    True
    11625    False
    11626    True
    11627    False
    11628    True
Length: 11629, dtype: bool

#Contar cuántos duplicados hay
df.duplicated().sum()
[11] ✓ 0.0s

... np.int64(710)
```

Con el comando “.duplicated()” observamos los valores True/False de los duplicados en la base de datos, por lo que fue necesario sumarlos para saber cuántos eran los duplicados en el DataFrame.

### 4. Análisis de los tipos de datos de las columnas y si son consistentes con el contenido esperado.

```
#Tipos de datos de las columnas
df.info()
[17] ✓ 0.0s

... <class 'pandas.core.frame.DataFrame'>
Index: 10974 entries, 0 to 11628
Data columns (total 9 columns):
#   Column                                Non-Null Count  Dtype
---  ---                                -----
0   Fecha                                10533 non-null  object
1   Ubicación                            10537 non-null  object
2   País                                 10525 non-null  object
3   Temperatura                          10523 non-null  object
4   Emisiones de CO2                     10530 non-null  float64
5   Aumento del Nivel del mar            10334 non-null  float64
6   Precipitaciones                      10542 non-null  float64
7   Humedad                             10529 non-null  object
8   Velocidad del viento                 10533 non-null  float64
dtypes: float64(4), object(5)
memory usage: 857.3+ KB
```

Revisando los tipos de datos del DataFrame con el comando “info()”, notamos que nos sería más benéfico que la columna ‘Date’ fuera de tipo DateTime y las columnas ‘Temperatura’ y ‘Humedad’ fuesen de tipo Float, ya que son más útiles los valores numéricos con puntos decimales para mayor precisión.

5. Cambio de nombres de las columnas al idioma español a través del uso de un diccionario y el comando `.rename()`.

```
columnas = {'Date': 'Fecha', 'Location': 'Ubicación', 'Country': 'País',
            'Temperature': 'Temperatura', 'CO2 Emissions': 'Emisiones de CO2',
            'Sea Level Rise': 'Aumento del Nivel del mar', 'Precipitation': 'Precipitaciones',
            'Humidity': 'Humedad', 'Wind Speed': 'Velocidad del viento'}

#Renombrar las columnas
df=df.rename(columns=columnas)
df.head()
```

[22] ✓ 0.0s

	Fecha	Ubicación	País	Temperatura	Emisiones de CO2	Aumento del Nivel del mar	Precipitaciones	Humedad	Velocidad del viento
0	2000-01-01 00:00:00.000000000	New Williamtown	Latvia	10.688986	403.118903	0.717506	13.835237	23.631256	18.492026
1	2000-01-01 20:09:43.258325832	North Rachel	South Africa	13.814430	396.663499	1.205715	40.974084	43.982946	34.249300
2	2000-01-02 16:19:26.516651665	West Williamland	French Guiana	27.323718	451.553155	-0.160783	42.697931	96.652600	34.124261
3	NaT	South David	Vietnam	12.309581	422.404983	-0.475931	5.193341	47.467938	8.554563
5	2000-01-05 04:48:36.291629162	South Nathan	Saint Helena	6.229326	392.473317	1.122210	76.368331	48.973886	30.398908

### 3\_ Proceso de limpieza

Los métodos utilizados fueron los siguientes:

- Imputación de valores nulos o faltantes de la base de datos con el comando “`fillna()`” para reemplazarlos por el promedio de la columna, en el caso de los numéricos, y por cadenas de texto en los de tipo object.
- Eliminación de registros duplicados.
- Corrección de los tipos de datos de las columnas para adecuarlos a nuestros objetivos.
- Corrección de valores inválidos, en nuestro caso de cadenas ‘bbb’.

A continuación, se presentarán el antes, la limpieza y el después de la limpieza.

Usando el comando `isnull().sum()`, pudimos visualizar la cantidad de datos faltantes.

```
#Ver cuántos NaN hay en las columnas
df.isnull().sum()
```

✓ 0.0s

Fecha	465
Ubicación	465
País	465
Temperatura	465
Emisiones de CO2	465
Aumento del Nivel del mar	685
Precipitaciones	465
Humedad	465
Velocidad del viento	465
dtype: int64	

Limpieza de valores faltantes con el comando `fillna()`. Para las columnas numéricas se utilizó el reemplazo de datos faltantes por el promedio de la columna y para las demás columnas se reemplazaron con textos.

```
#Reemplazar los valores NaN por cadenas de texto y por el promedio en el caso de los registros numéricos
lista_prom=['Temperatura','Emisiones de CO2','Aumento del Nivel del mar','Precipitaciones','Humedad','Velocidad del viento']

df['Fecha'].fillna("1678-01-01 00:00:00.000000000", inplace=True)
df['Ubicación'].fillna("Sin ubicación", inplace=True)
df['País'].fillna("Sin País", inplace=True)

for i in lista_prom:
    df[i]=pd.to_numeric(df[i], errors='coerce')
    df[i].fillna(df[i].mean(), inplace=True)
```

✓ 0.0s



Después de la limpieza, con el comando `isnull().sum()` verificamos que no hay datos faltantes.

```
#Revisar que no haya nulos
df.isnull().sum()
✓ 0.0s
```

Fecha	0
Ubicación	0
País	0
Temperatura	0
Emisiones de CO2	0
Aumento del Nivel del mar	0
Precipitaciones	0
Humedad	0
Velocidad del viento	0
dtype: int64	

```
df.head()
✓ 0.0s
```

	Fecha	Ubicación	País	Temperatura	Emisiones de CO2	Aumento del Nivel del mar	Precipitaciones	Humedad	Velocidad del viento
0	2000-01-01 00:00:00.000000000	New Williamtown	Latvia	10.688986	403.118903	0.717506	13.835237	23.631256	18.492026
1	2000-01-01 20:09:43.258325832	North Rachel	South Africa	13.814430	396.663499	1.205715	40.974084	43.982946	34.249300
2	2000-01-02 16:19:26.516651665	West Williamland	French Guiana	27.323718	451.553155	-0.160783	42.697931	96.652600	34.124261
3	1678-01-01 00:00:00.000000000	South David	Vietnam	12.309581	422.404983	-0.475931	5.193341	47.467938	8.554563
4	2000-01-04 08:38:53.033303330	New Scottburgh	bbb	13.210885	410.472999	1.135757	49.862571	49.705318	8.001164

Antes, podemos observar con el comando `.duplicated().sum()` la cantidad de datos duplicados.

```
#Ver cuántos duplicados hay
df.duplicated().sum()
✓ 0.0s
```

np.int64(711)

Limpiamos los duplicados con el comando `drop_duplicates()` de acuerdo con la columna de Fecha, ya que es importante para nuestro proyecto.

```
#Eliminar los valores duplicados (por fecha ya que es puede resultar problemático para nuestro objetivo)
df=df.drop_duplicates('Fecha')
✓ 0.0s
```

Posterior a la limpieza de duplicados, podemos verificar con el comando `.duplicated().sum()` que efectivamente no se encuentren presentes en el dataframe.

```
#Verificar que no haya duplicados
df.duplicated().sum()
✓ 0.0s
```

np.int64(0)

Anteriormente analizamos los tipos de datos de nuestras variables con `.info()` para saber si nos serían útiles o si era necesario cambiarlos.

```
#Tipos de datos de las columnas
df.info()
✓ 0.0s

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11629 entries, 0 to 11628
Data columns (total 9 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   Date                  11164 non-null  object
1   Location              11164 non-null  object
2   Country               11164 non-null  object
3   Temperature           11164 non-null  object
4   CO2 Emissions         11164 non-null  float64
5   Sea Level Rise        10944 non-null  float64
6   Precipitation         11164 non-null  float64
7   Humidity              11164 non-null  object
8   Wind Speed            11164 non-null  float64
dtypes: float64(4), object(5)
memory usage: 817.8+ KB
```

Notamos que era más práctico que las columnas de Temperatura y Humedad, fueran de tipo float y que la fecha fuera de tipo `datetime()`.

```
#Cambiar el tipo de dato de temperatura y humedad a float y la fecha a datetime
df['Temperatura']=df['Temperatura'].astype(float)
df['Humedad']=df['Humedad'].astype(float)
df['Fecha']=pd.to_datetime(df['Fecha'])
✓ 0.0s
```

Y verificamos nuevamente con un `.info()`.

```
#Revisar los cambios de tipos de datos
df.info()
✓ 0.0s

<class 'pandas.core.frame.DataFrame'>
Index: 9666 entries, 0 to 11607
Data columns (total 9 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   Fecha                 9666 non-null  datetime64[ns]
1   Ubicación             9666 non-null  object
2   País                 9666 non-null  object
3   Temperatura           9666 non-null  float64
4   Emisiones de CO2      9666 non-null  float64
5   Aumento del Nivel del mar  9666 non-null  float64
6   Precipitaciones       9666 non-null  float64
7   Humedad               9666 non-null  float64
8   Velocidad del viento  9666 non-null  float64
dtypes: datetime64[ns](1), float64(6), object(2)
memory usage: 755.2+ KB
```

También era necesario verificar que no hubiera datos inválidos como, en este caso, las cadenas 'bbb' que observamos desde la carga del dataframe. Se hizo con un ciclo for que recorriera las columnas de la base de datos.

```
#Revisar cuántos valores bbb hay en las columnas
lista = df.columns
for i in lista:
    print(f"En la columna {i} los bbb son: {df[df[i] == 'bbb'].shape[0]}")
✓ 0.0s
```

En la columna Fecha los bbb son: 0  
En la columna Ubicación los bbb son: 0  
En la columna País los bbb son: 192  
En la columna Temperatura los bbb son: 0  
En la columna Emisiones de CO2 los bbb son: 0  
En la columna Aumento del Nivel del mar los bbb son: 0  
En la columna Precipitaciones los bbb son: 0  
En la columna Humedad los bbb son: 0  
En la columna Velocidad del viento los bbb son: 0

Para eliminar dichas cadenas se utilizó nuevamente un ciclo for para actualizar el Dataframe sin dichos datos.

```
#Eliminar los datos 'bbb' de las columnas

for i in lista:
    df=df[df[i] != 'bbb']
70] ✓ 0.0s
```

Para verificar que se eliminaron, usamos el ciclo for del inicio.

```
#Verificar que no haya datos 'bbb'
for i in lista:
    print(f"En la columna {i} los bbb son: {df[df[i] == 'bbb'].shape[0]}")
✓ 0.0s
```

En la columna Fecha los bbb son: 0  
En la columna Ubicación los bbb son: 0  
En la columna País los bbb son: 0  
En la columna Temperatura los bbb son: 0  
En la columna Emisiones de CO2 los bbb son: 0  
En la columna Aumento del Nivel del mar los bbb son: 0  
En la columna Precipitaciones los bbb son: 0  
En la columna Humedad los bbb son: 0  
En la columna Velocidad del viento los bbb son: 0

Por último, se hizo la verificación general de la base de datos actualizada con los cambios de la limpieza de datos.

```
df.info()
✓ 0.0s
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 9474 entries, 0 to 11607
Data columns (total 9 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Fecha                 9474 non-null  datetime64[ns]
1   Ubicación             9474 non-null  object
2   País                 9474 non-null  object
3   Temperatura           9474 non-null  float64
4   Emisiones de CO2      9474 non-null  float64
5   Aumento del Nivel del mar  9474 non-null  float64
6   Precipitaciones       9474 non-null  float64
7   Humedad              9474 non-null  float64
8   Velocidad del viento  9474 non-null  float64
dtypes: datetime64[ns](1), float64(6), object(2)
memory usage: 740.2+ KB
```

```
df.isnull().sum()/len(df)*100
✓ 0.0s
```

Fecha	0.0
Ubicación	0.0
País	0.0
Temperatura	0.0
Emisiones de CO2	0.0
Aumento del Nivel del mar	0.0
Precipitaciones	0.0
Humedad	0.0
Velocidad del viento	0.0
dtype: float64	

```
df.duplicated().sum()
✓ 0.0s
np.int64(0)
```

```
#Verificar que no haya datos 'bbb'
for i in lista:
    print(f"En la columna {i} los bbb son: {df[df[i] == 'bbb'].shape[0]}")
✓ 0.0s

En la columna Fecha los bbb son: 0
En la columna Ubicación los bbb son: 0
En la columna País los bbb son: 0
En la columna Temperatura los bbb son: 0
En la columna Emisiones de CO2 los bbb son: 0
En la columna Aumento del Nivel del mar los bbb son: 0
En la columna Precipitaciones los bbb son: 0
En la columna Humedad los bbb son: 0
En la columna Velocidad del viento los bbb son: 0
```

## Metodología

El desarrollo de este proyecto se dividió en las siguientes etapas, implementadas utilizando herramientas como Python y librerías especializadas (Pandas):

### Exploración inicial de los datos

En esta etapa se realizó un análisis preliminar para comprender las características de la base de datos:

- **Dimensiones y estructura del dataset:** Se revisaron el tamaño de la base de datos, los tipos de variables, y la cantidad de valores faltantes mediante comandos como `.info()` y `.describe()`.
- **Identificación de valores nulos y duplicados:** Se detectaron registros con valores nulos en columnas clave, así como posibles duplicados mediante el comando `.duplicated()`.
- **Clasificación de variables:** Se categorizaron las columnas como numéricas, categóricas o temporales para definir estrategias de análisis posteriores.

### Limpieza de datos

En esta etapa se implementaron técnicas básicas para garantizar la calidad del análisis:

- **Eliminación de valores duplicados:** Se eliminaron registros duplicados utilizando el comando `.drop_duplicates()` para asegurar la consistencia de los datos.

- **Manejo de valores nulos:** Los valores faltantes en las columnas numéricas se imputaron utilizando la media o mediana, dependiendo de la distribución de los datos.
- **Formateo de la columna de fecha:** La columna Fecha fue convertida al formato datetime para facilitar análisis basados en el tiempo.

## Implementaciones

Para poder ajustar nuestra base de datos al modelo que usamos de Machine Learning y dashboard, fue necesario crear columnas de Longitud y Latitud a partir de las columnas de Ubicación y País. También creamos una columna de Mes que corresponde al mes dado en la columna Fecha y también dos columnas que corresponden a los eventos de precipitaciones y temperatura que nos ayudaron a determinar los registros relevantes por país y por mes.

## Análisis exploratorio de datos:

Esta será la etapa central del proyecto, en la cual se analizarán a detalle las características principales de las variables para obtener insights clave.

### Descripción general de los datos.

- **Visión General:** Resumen del dataset, incluyendo el número total de registros y variables.

```
data=pd.read_csv("base_limpia_proyecto.csv")
data
```

✓ 4.1s

	Fecha	Ubicación	País	Temperatura	Emisiones de CO2	Aumento del Nivel del mar	Precipitaciones	Humedad	Velocidad del viento
0	2000-01-01	New Williamtown	Latvia	10.688986	403.118903	0.717506	13.835237	23.631256	18.492026
1	2000-01-01	North Rachel	South Africa	13.814430	396.663499	1.205715	40.974084	43.982946	34.249300
2	2000-01-02	West Williamland	French Guiana	27.323718	451.553155	-0.160783	42.697931	96.652600	34.124261
3	1678-01-01	South David	Vietnam	12.309581	422.404983	-0.475931	5.193341	47.467938	8.554563
4	2000-01-05	South Nathan	Saint Helena	6.229326	392.473317	1.122210	76.368331	48.973886	30.398908
...	...	...	...	...	...	...	...	...	...
9469	2014-08-09	Heatherfort	Kenya	9.633273	536.879317	-1.055269	0.733654	35.386679	11.349757
9470	2022-03-26	South Jeffrey	Norway	18.322140	533.633336	-0.154583	40.539896	95.178150	33.091834
9471	2017-05-11	North Lauren	Saint Martin	15.828247	445.608149	0.081255	25.880205	6.341108	42.817475
9472	2010-04-08	Port Laura	Georgia	14.932330	366.906446	0.363282	42.474523	94.672587	42.797399
9473	2003-07-05	Coleside	Romania	11.983638	517.075204	0.185722	39.697657	31.336638	46.092629

9474 rows x 9 columns

- **Tipos de Variables:** Clasificación de las variables.

```
data.info()
✓ 0.0s

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9474 entries, 0 to 9473
Data columns (total 11 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Fecha                  9474 non-null   object
1   Ubicación              9474 non-null   object
2   País                  9474 non-null   object
3   Temperatura            9474 non-null   float64
4   Emisiones de CO2       9474 non-null   float64
5   Aumento del Nivel del mar 9474 non-null   float64
6   Precipitaciones        9474 non-null   float64
7   Humedad                9474 non-null   float64
8   Velocidad del viento    9474 non-null   float64
9   Latitud                8989 non-null   float64
10  Longitud               8989 non-null   float64
dtypes: float64(8), object(3)
memory usage: 814.3+ KB
```

Podemos observar que tenemos datos de tipo datetime, object y float. Comprobamos que se ajustaron correctamente los cambios después de la limpieza.

## o Resumen Estadístico: Estadísticas descriptivas.

```
data.describe()
✓ 0.0s
```

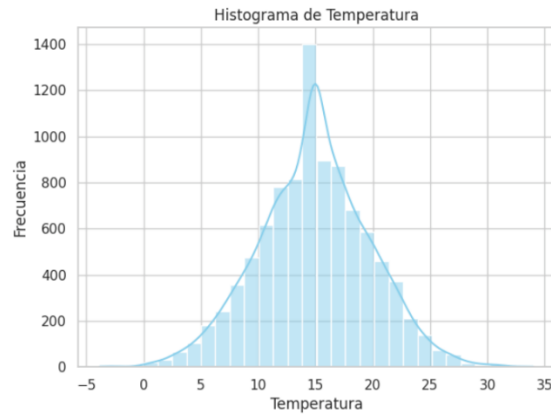
	Temperatura	Emisiones de CO2	Aumento del Nivel del mar	Precipitaciones	Humedad	Velocidad del viento	Latitud	Longitud
count	9474.000000	9474.000000	9474.000000	9474.000000	9474.000000	9474.000000	8989.000000	8989.000000
mean	14.900554	400.278444	-0.001910	49.837706	49.815610	25.168255	15.513630	9.995284
std	4.879772	48.598603	0.958409	28.330937	28.076813	14.158351	25.778215	76.296039
min	-3.803589	182.131220	-4.092155	0.010143	0.018998	0.001732	-54.843286	-176.204224
25%	11.771866	368.875036	-0.619916	25.422358	26.348809	13.207691	-0.525231	-59.525030
50%	14.932330	400.157142	-0.005637	49.862571	49.705318	25.086836	15.926666	14.447691
75%	18.009023	431.570011	0.621775	73.690124	73.824189	37.143033	36.800207	51.229529
max	33.976956	582.899701	3.849427	99.991900	99.959665	49.997664	64.984182	179.158292

Según lo anterior, podemos decir que nuestra base de datos tiene al final 9474 registros. Dado que las columnas de mayor interés para el proyecto son Temperatura y Precipitaciones, es necesario conocer los valores máximos y mínimos para establecer rangos más adelante en los modelos de Machine Learning.

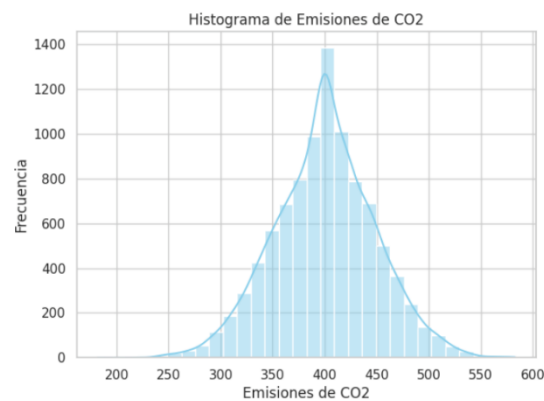
## Visualización y Distribución de Variables Individuales.

### Variables Numéricas:

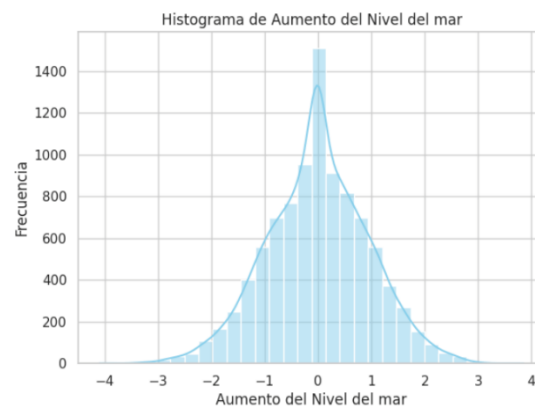
- Histogramas (con matplotlib y seaborn):



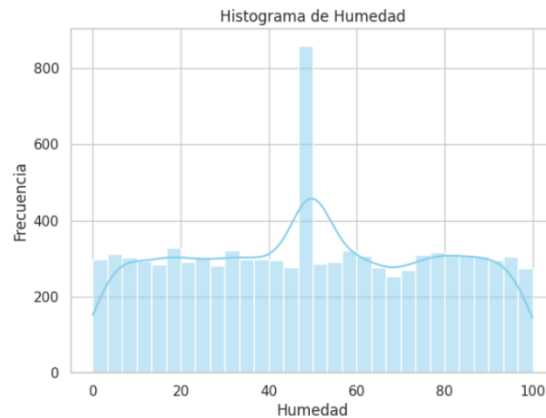
Este histograma de temperatura muestra una distribución normal, centrada alrededor de los 15 °C, ya que ahí se encuentra el pico más alto que representa la moda. La forma de campana indica que la mayoría de los datos están cerca de la media, con una caída gradual hacia los extremos. Esto sugiere que las temperaturas extremas (muy altas o bajas) son menos frecuentes.



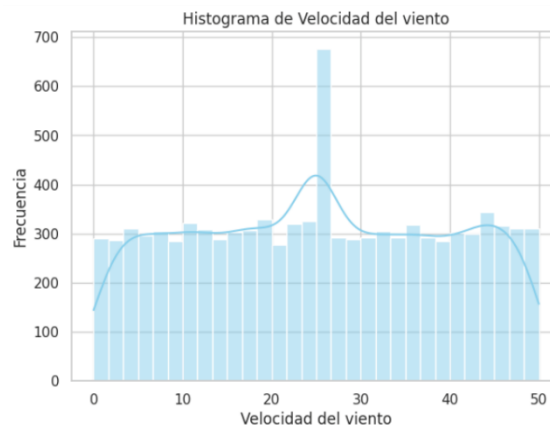
Este histograma de emisiones de CO2 muestra una distribución aproximadamente normal con un pico alrededor de 400 unidades de CO2 (en partes por millón, ppm). La curva tiene forma de campana, con la mayoría de los datos concentrados en el rango de 350 a 450 ppm. Esto indica que las emisiones de CO2 en nuestra base de datos son relativamente consistentes dentro de este rango.



El histograma de aumento del nivel del mar muestra una distribución centrada en 0 metros, lo que indica que la mayoría de las observaciones tienen cambios mínimos en el nivel del mar. La distribución es aproximadamente simétrica y en forma de campana, lo que sugiere que la mayoría de los valores están agrupados alrededor de 0, con pocos valores extremos.



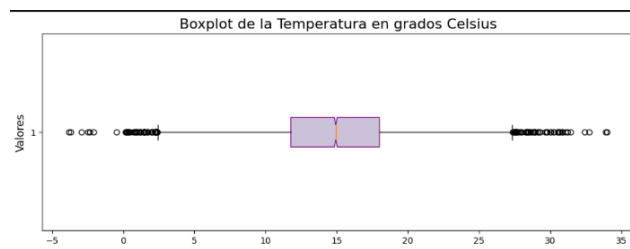
El histograma de humedad muestra una distribución con un pico prominente alrededor del 50% de humedad, indicando que la mayoría de las observaciones se concentran en esta cifra. Podría reflejar una condición climática común en la mayoría de las regiones del conjunto de datos, probablemente debido a climas templados o estaciones específicas. Fuera del pico central, los valores de humedad están distribuidos de manera más uniforme entre el 0% y el 100%. Esto podría indicar que también hay regiones con condiciones climáticas extremas, como zonas muy áridas o húmedas. Este gráfico tiene un comportamiento más irregular que los anteriores, con un pico significativo y áreas más planas, lo que indica una mezcla de condiciones climáticas muy diferentes.



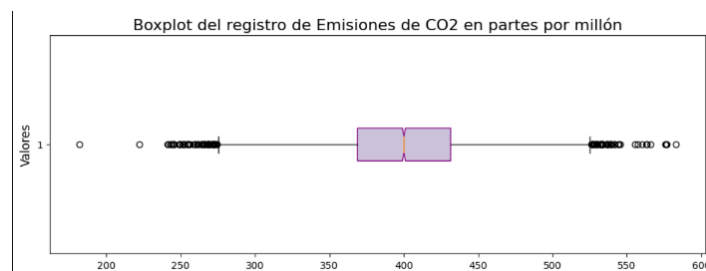


El histograma de velocidad del viento presenta una distribución variada, con un pico prominente y una dispersión más uniforme en otras partes del rango. Existe un pico marcado alrededor de los 30 km/h, lo que sugiere que la mayoría de las observaciones de velocidad del viento están concentradas cerca de este valor. Esto podría ser una velocidad común en regiones con condiciones climáticas moderadas. Aparte del pico, la velocidad del viento se distribuye de manera más uniforme entre 0 y 50 km/h, con frecuencias consistentes en la mayoría de los rangos.

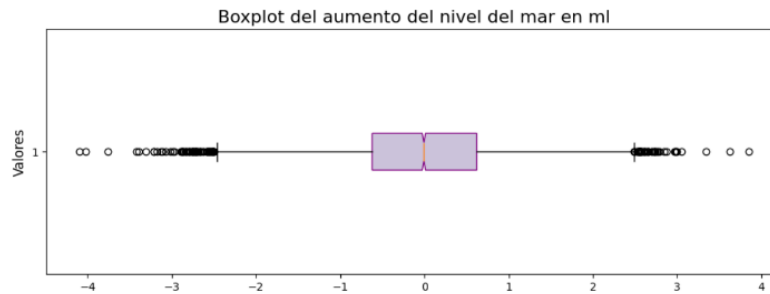
- **Boxplots (con la librería plotly):**



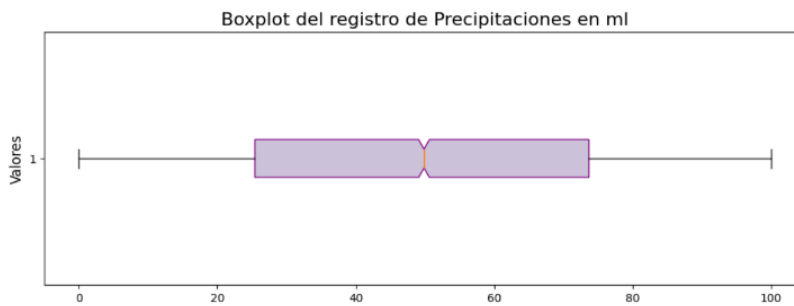
El gráfico nos dice que la mayoría de los valores de temperatura se encuentran entre aproximadamente 12 y 18 °C, representados por la caja lila. Este rango incluye el 50% de los datos (entre los cuartiles 1 y 3), lo que sugiere que son las temperaturas más frecuentes. La línea de la mediana parece estar en torno a los 15 °C. Los bigotes se extienden aproximadamente desde 0 a 25 °C y en cuanto a los valores atípicos, observamos puntos fuera de los bigotes, tanto hacia temperaturas bajas (<0 °C) como hacia temperaturas altas (>25 °C). Estos pueden deberse a condiciones climáticas extremas o a condiciones inusuales en las regiones.



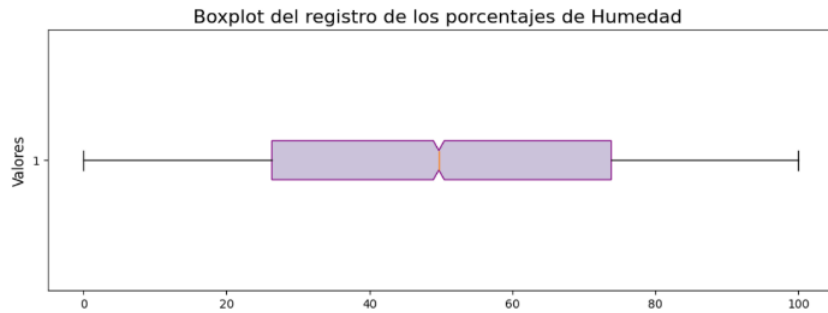
El boxplot de las emisiones de CO2 en partes por millón (ppm) muestra que la mayoría de los datos se encuentran entre aproximadamente 380 y 420 ppm, esto indica que el 50% de los datos se concentran en este rango. La mediana está cerca de los 400 ppm. Los bigotes se extienden desde aproximadamente 350 hasta 450 ppm y los valores atípicos están por debajo de los 350 ppm y por encima de los 450 ppm. Podrían corresponder a regiones con niveles demasiado bajos o altos de emisiones de CO2 debido a variaciones geográficas o actividades industriales.



La mayoría de los datos se encuentran entre -0.5 y 0.5 mm, lo que indica que el 50% de los registros de aumento del nivel del mar se concentran en este rango. La mediana está ubicada en 0 mm, así que la mayoría de los valores están cerca de ningún cambio significativo en el nivel del mar. Los bigotes se extienden desde aproximadamente -1.5 mm hasta 1.5 mm y abarcan la mayoría de los datos sin considerar los valores atípicos. Existen valores fuera de este rango, tanto hacia los extremos negativos ( $<-1.5$  mm) como positivos ( $>1.5$  mm). Estos podrían corresponder a ciertas regiones donde el aumento o disminución del nivel del mar es más notorio debido a factores locales, como corrientes oceánicas o derretimiento de glaciares.

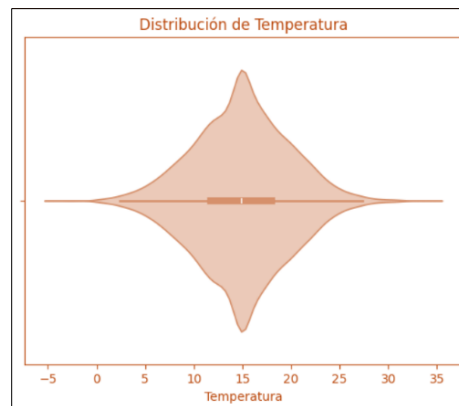


Este gráfico indica que la mayoría de los valores de precipitaciones se encuentran entre aproximadamente 40 y 60 mm y nos dice que las precipitaciones moderadas son las más comunes en los datos analizados. La mediana está ubicada alrededor de 50 mm e implica que la mitad de los datos se encuentran por debajo de este valor y la otra mitad por encima, reflejando una distribución equilibrada. Los bigotes se extienden desde 0 mm hasta cerca de 100 mm, lo que muestra que las precipitaciones más bajas y las más altas están dentro de este rango, sin presencia de valores atípicos visibles en el gráfico.

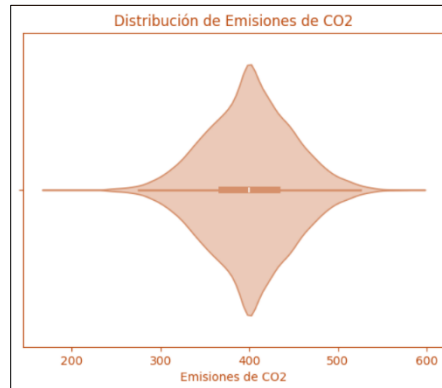


El gráfico de los porcentajes de humedad muestra que los datos están distribuidos uniformemente entre 0% y 100%, esto nos dice que las condiciones de humedad varían ampliamente en los datos. El rango intercuartílico es amplio, con valores centrales alrededor del 50%, lo que indica que la mayoría de las observaciones están concentradas en condiciones moderadas de humedad. No hay valores atípicos significativos así que los extremos en la humedad están dentro de lo esperado.

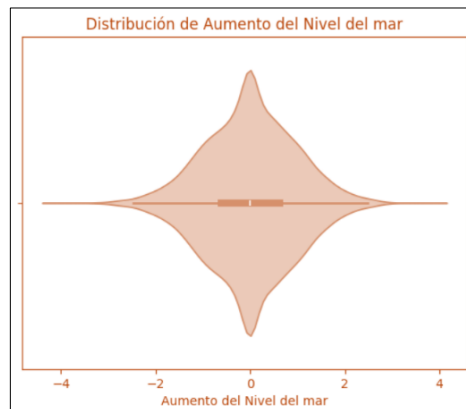
- **Diagramas de Violín (con la librería plotly):**



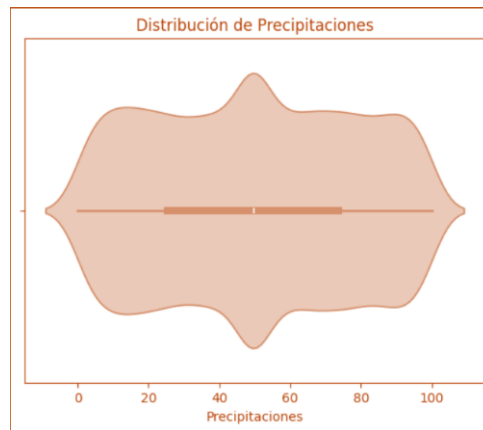
La mayor concentración de datos se encuentra alrededor de los 15 °C, lo que indica que este es el rango más común. A medida que las temperaturas se alejan de este valor, la frecuencia disminuye, lo que se refleja en las partes más estrechas del gráfico. Esto sugiere que los valores extremos de temperatura, tanto altos como bajos, son menos comunes, confirmando una distribución cercana a la normal.



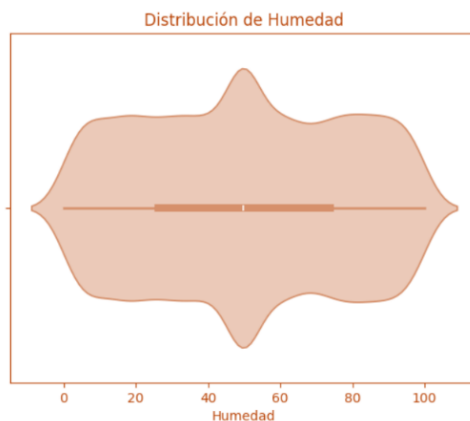
Este gráfico muestra que las emisiones de CO2 están concentradas principalmente alrededor de los 400 ppm, que es el rango más común. Los valores se distribuyen simétricamente a ambos lados de este punto, y los extremos más alejados indican emisiones inusuales, aunque menos frecuentes. Esto sugiere una distribución bien definida con pocos valores extremos.



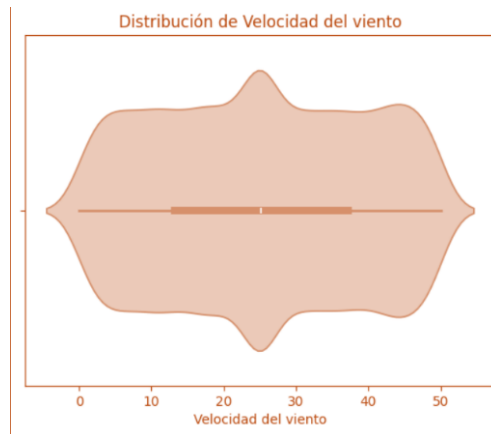
El gráfico muestra que los cambios en el nivel del mar se concentran alrededor de valores cercanos a 0 mm, lo que indica estabilidad general en la mayoría de los casos. Sin embargo, hay valores más extremos, tanto positivos como negativos, que representan incrementos o descensos significativos menos comunes. La simetría de la gráfica sugiere que estos extremos se distribuyen de manera equilibrada.



En este diagrama, la mayoría de los valores se concentran alrededor del rango medio, aproximadamente entre 40 y 60 mm. Esto indica que las precipitaciones tienden a ser moderadas en la mayoría de los casos. Sin embargo, hay valores extremos en ambos lados, lo que sugiere la existencia de zonas con lluvias muy bajas o intensas, aunque estas sean menos frecuentes. La distribución general parece simétrica, lo que implica un equilibrio entre estos valores extremos.



Aquí los valores están distribuidos de manera relativamente uniforme en un rango grande, con una concentración alrededor del 50%. Esto indica que los niveles de humedad suelen ser moderados en la mayoría de los casos. Los extremos cercanos a 0% y 100% son menos frecuentes, pero podrían representar la existencia de ubicaciones con condiciones extremadamente secas o húmedas. La distribución general parece equilibrada.

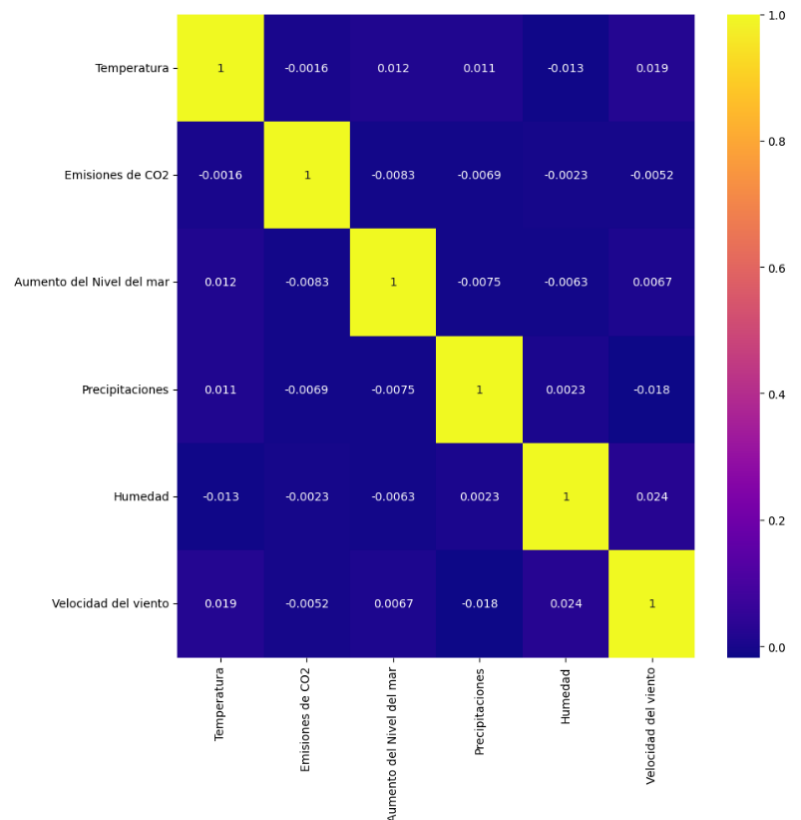


En este caso, los valores están distribuidos de manera amplia, con una concentración en un rango alrededor de 20 a 30 km/hr. Esto indica que, aunque hay una variación considerable en las velocidades del viento, las velocidades intermedias son más comunes. Los valores extremos, cercanos a 0 y 50 km/hr, son menos frecuentes, pero podrían reflejar escenarios con vientos muy bajos o fuertes en ciertas ubicaciones.

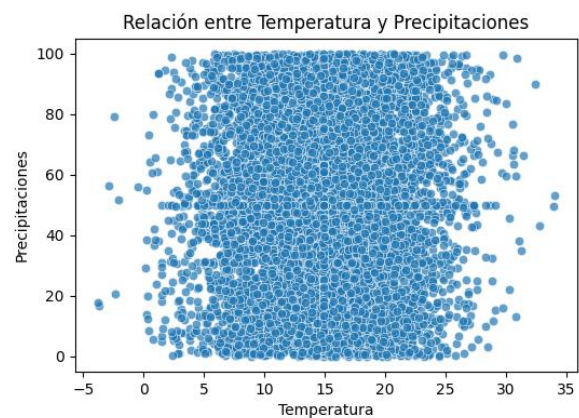
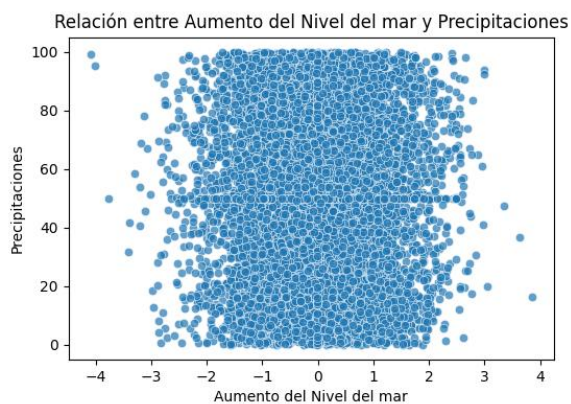
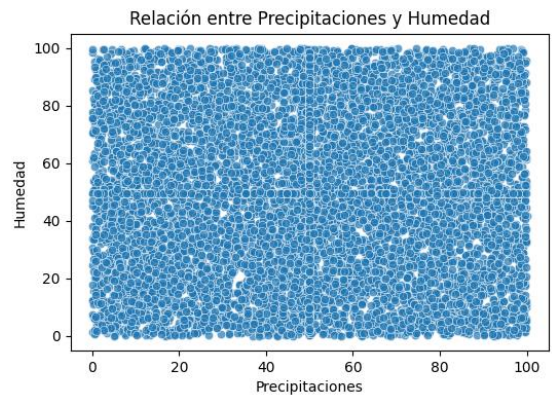
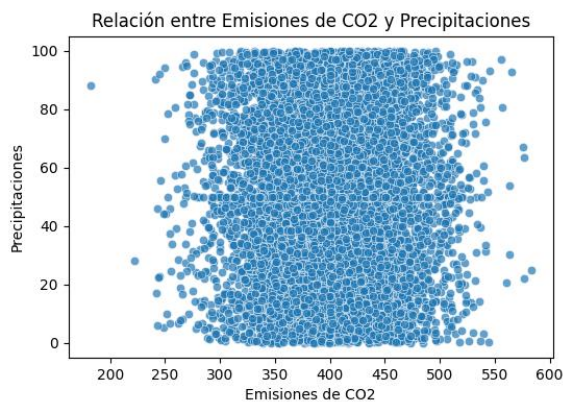
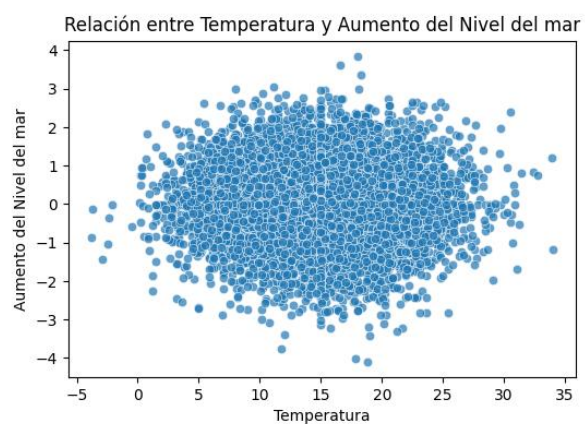
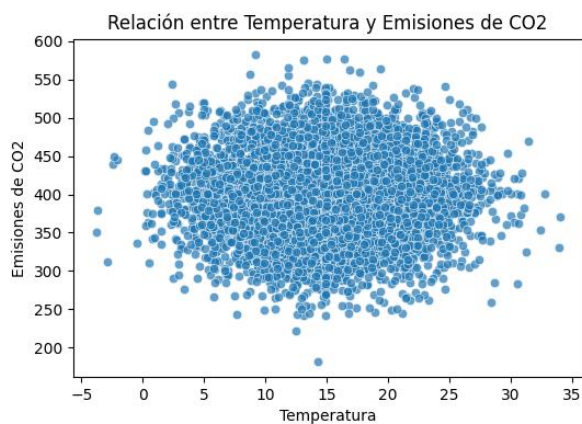
## Correlación entre variables

### Matriz de Correlación (heatmap):

El heatmap indica que las variables no están correlacionadas de manera lineal. Esto podría significar que las relaciones entre estas variables, si existen, podrían ser no lineales o influenciadas por otros factores no representados en los datos. La baja correlación entre variables indica que no están altamente relacionadas entre sí, lo que es bueno para evitar colinealidad en el modelo.

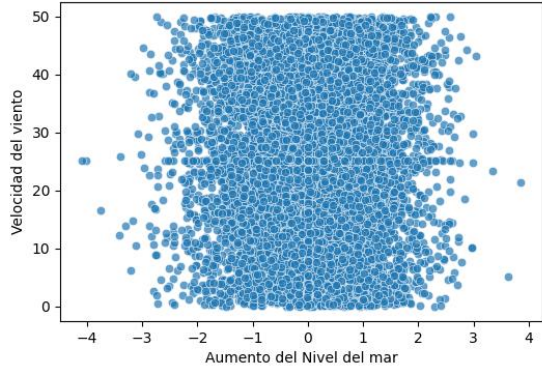


## Parejas de Variables: Gráficos de dispersión.

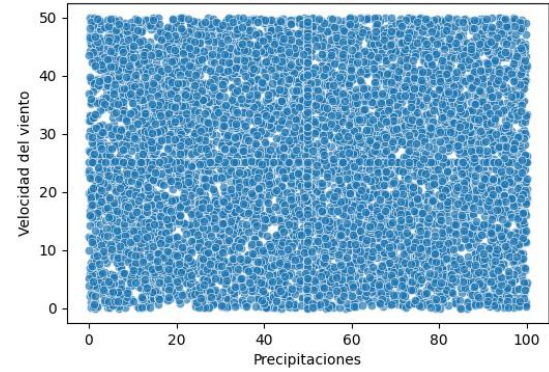




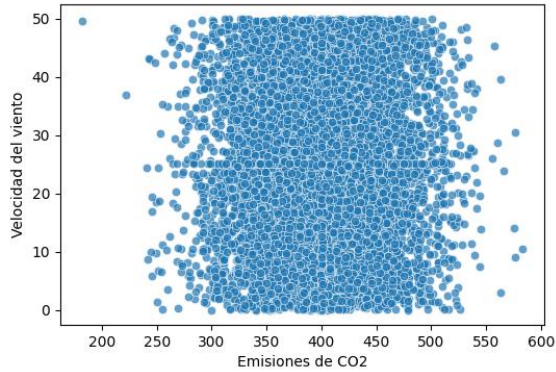
Relación entre Aumento del Nivel del mar y Velocidad del viento



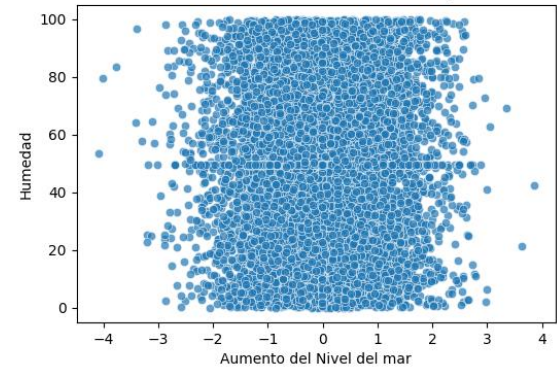
Relación entre Precipitaciones y Velocidad del viento



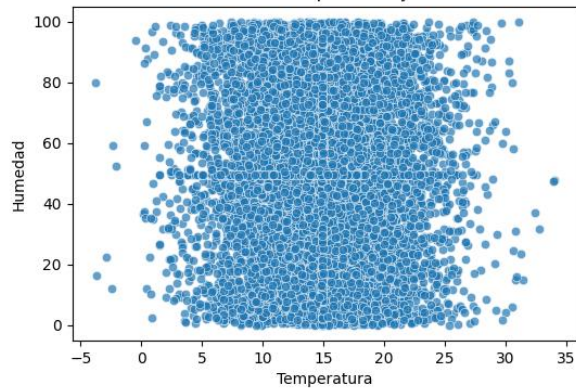
Relación entre Emisiones de CO2 y Velocidad del viento



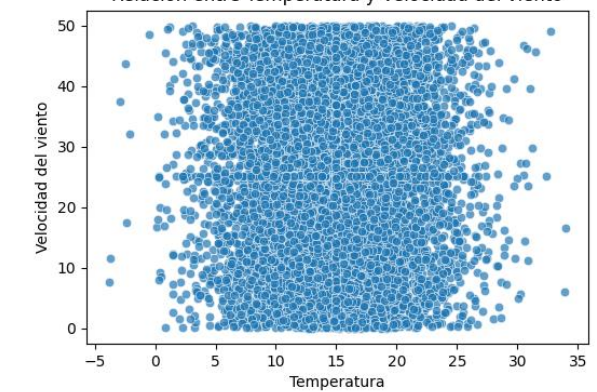
Relación entre Aumento del Nivel del mar y Humedad



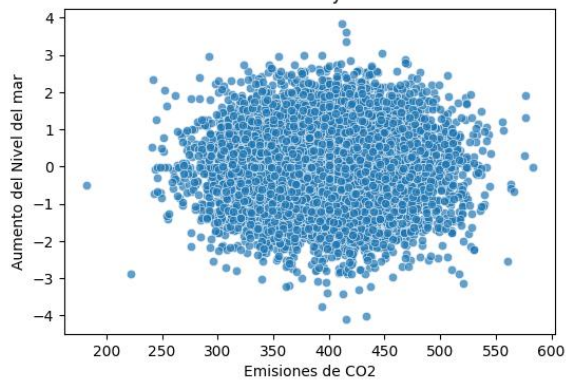
Relación entre Temperatura y Humedad



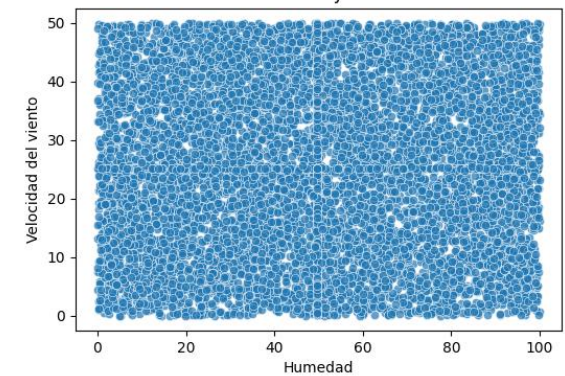
Relación entre Temperatura y Velocidad del viento



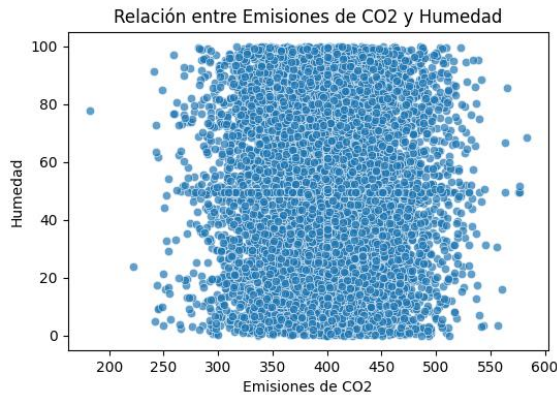
Relación entre Emisiones de CO2 y Aumento del Nivel del mar



Relación entre Humedad y Velocidad del viento





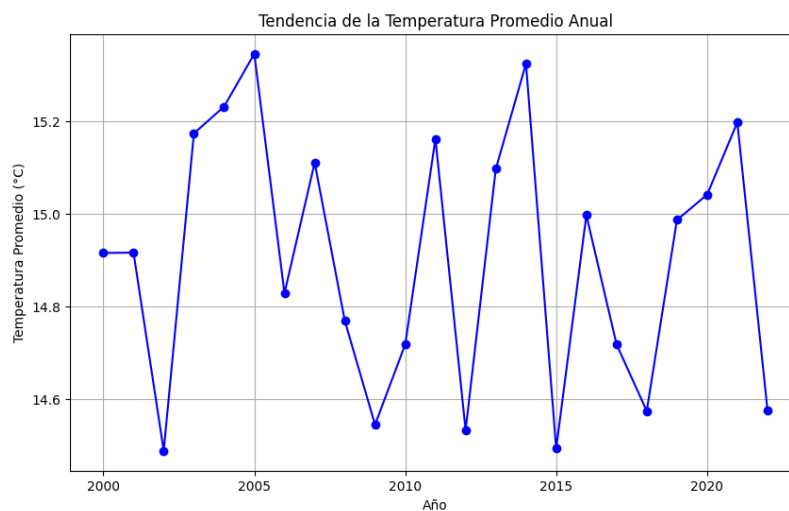


Interpretación de cada gráfico de dispersión:

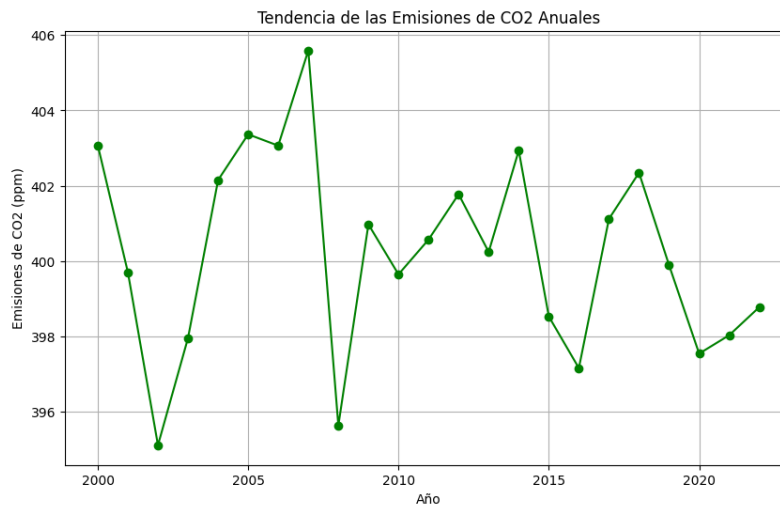
- Relación entre Temperatura y Emisiones de CO2:**  
No se observa una correlación clara entre la temperatura y las emisiones de CO2. Los puntos están distribuidos de forma uniforme, lo que sugiere que estas dos variables no tienen una relación directa en esta muestra de datos.
- Relación entre Temperatura y Aumento del Nivel del Mar:**  
No hay una tendencia clara entre estas dos variables. La dispersión indica que el aumento del nivel del mar no depende linealmente de la temperatura en este conjunto de datos.
- Relación entre Emisiones de CO2 y Precipitaciones:**  
Tampoco se aprecia una relación significativa entre estas dos variables. La distribución es uniforme, lo que implica que las precipitaciones no están directamente influenciadas por las emisiones de CO2.
- Relación entre Precipitaciones y Humedad:**  
La humedad parece no estar correlacionada con las precipitaciones. Los puntos están dispersos uniformemente, lo que indica que estas variables podrían ser independientes en esta base de datos.
- Relación entre Aumento del Nivel del Mar y Precipitaciones:**  
Los datos no muestran una relación clara entre el aumento del nivel del mar y las precipitaciones. Esto sugiere que estas variables no tienen un vínculo directo.
- Relación entre Temperatura y Precipitaciones:**  
No hay una relación aparente entre la temperatura y las precipitaciones. La distribución de puntos refuerza la idea de que estas variables son independientes.
- Relación entre Aumento del Nivel del Mar y Velocidad del Viento:**  
No se observa una correlación lineal entre el aumento del nivel del mar y la velocidad del viento. Los puntos están uniformemente distribuidos.
- Relación entre Precipitaciones y Velocidad del Viento:**  
Tampoco hay evidencia de relación entre estas dos variables. La velocidad del viento no parece estar afectada por las precipitaciones.

- **Relación entre Emisiones de CO2 y Velocidad del Viento:**  
Los datos sugieren que no existe un vínculo entre las emisiones de CO2 y la velocidad del viento. La distribución uniforme lo confirma.
- **Relación entre Aumento del Nivel del Mar y Humedad:**  
No se aprecia una correlación significativa entre estas variables. La humedad parece no depender del aumento del nivel del mar.
- **Relación entre Temperatura y Humedad:**  
Al igual que con otras combinaciones, la temperatura y la humedad no muestran una relación directa o lineal en este conjunto de datos.
- **Relación entre Temperatura y Velocidad del Viento:**  
Los datos no indican una correlación entre la temperatura y la velocidad del viento.
- **Relación entre Emisiones de CO2 y Humedad:**  
No se observa una relación significativa entre las emisiones de CO2 y la humedad.
- **Relación entre Emisiones de CO2 y Aumento del Nivel del Mar:**  
Aunque los puntos están distribuidos uniformemente, podría explorarse más esta relación en futuros análisis.
- **Relación entre Humedad y Velocidad del Viento:**  
Tampoco se aprecia una relación clara entre la humedad y la velocidad del viento.

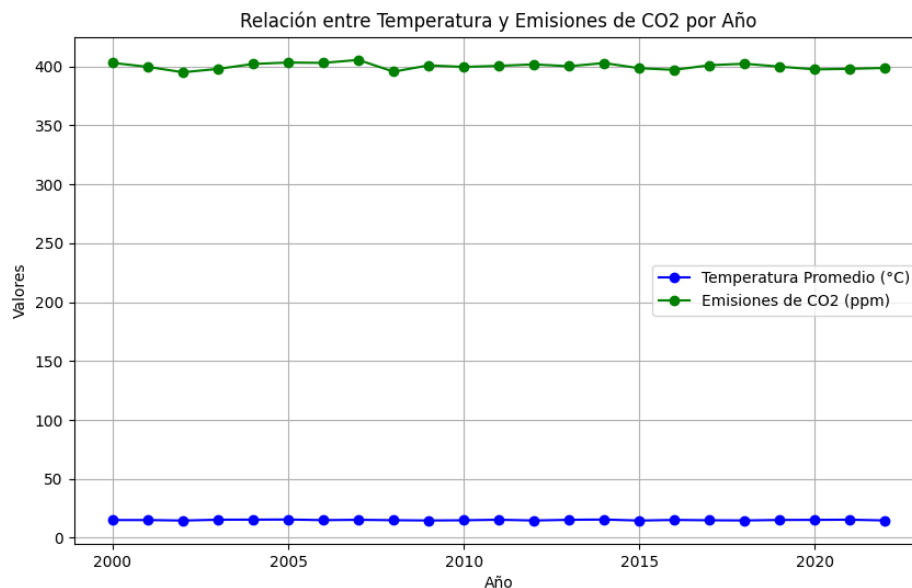
### Gráficos de Líneas complementarios.



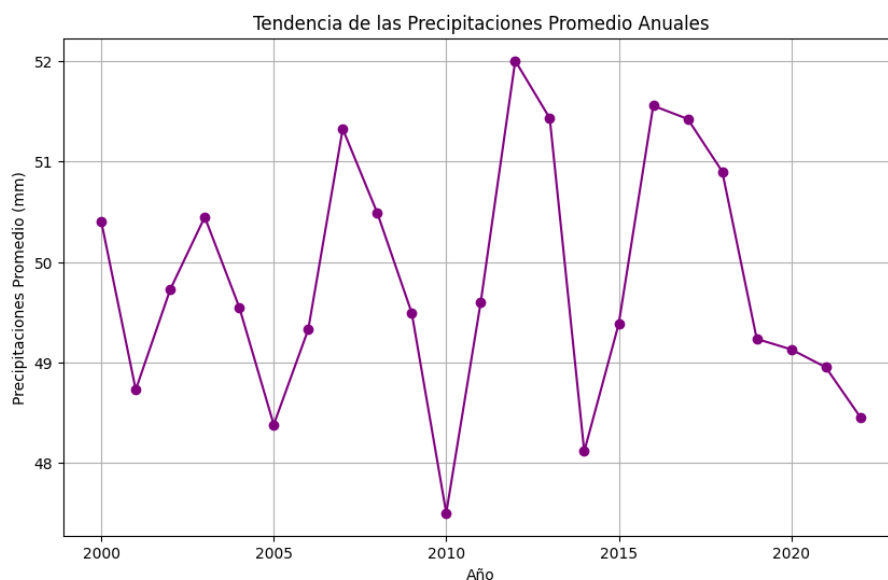
Esta gráfica muestra cómo la temperatura promedio varía año con año. Hay fluctuaciones notables, pero también una tendencia general a mantenerse alrededor de los 15°C. Los picos, como en 2005 y 2015, podrían estar relacionados con eventos climáticos específicos, mientras que las caídas indican años más fríos.



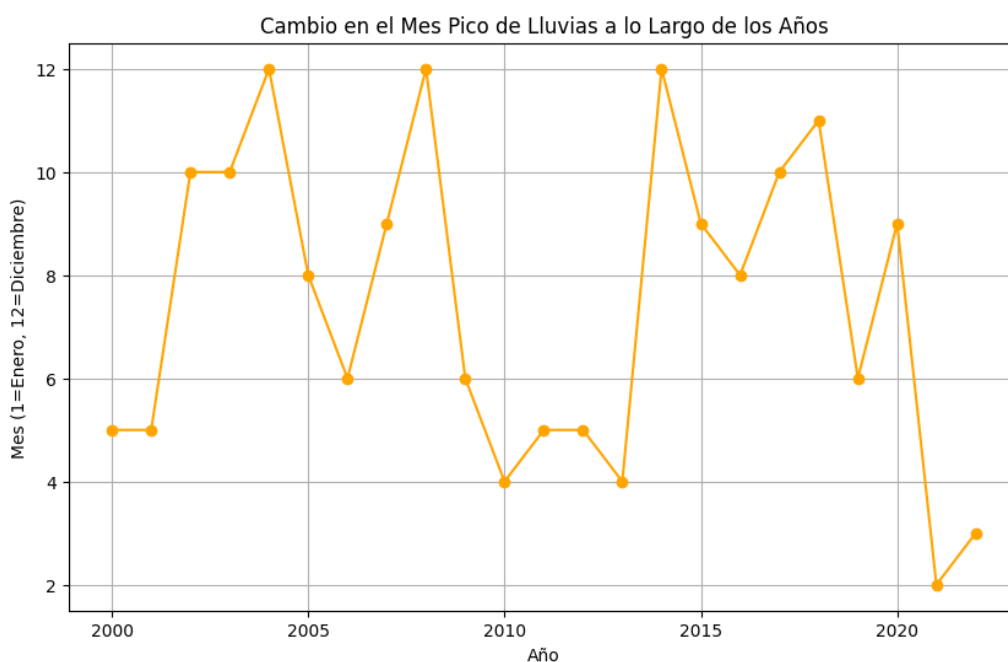
Las emisiones de CO2 se mantienen estables alrededor de los 400 ppm, con algunas fluctuaciones notables. Los picos, como en 2005 y 2010, podrían deberse a incrementos en actividades industriales o cambios en la capacidad de absorción de carbono de los ecosistemas.



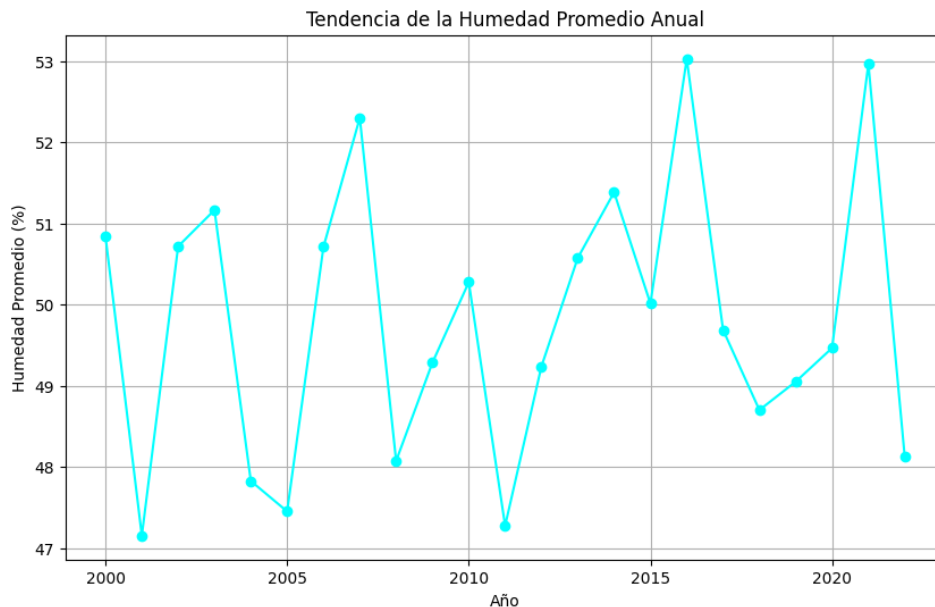
Al graficar juntas las temperaturas promedio y las emisiones de CO2, no se observa una relación visual evidente. Esto implica que, aunque ambos factores son importantes para el clima, su relación podría ser más compleja y requerir un análisis estadístico para detectar correlaciones significativas.



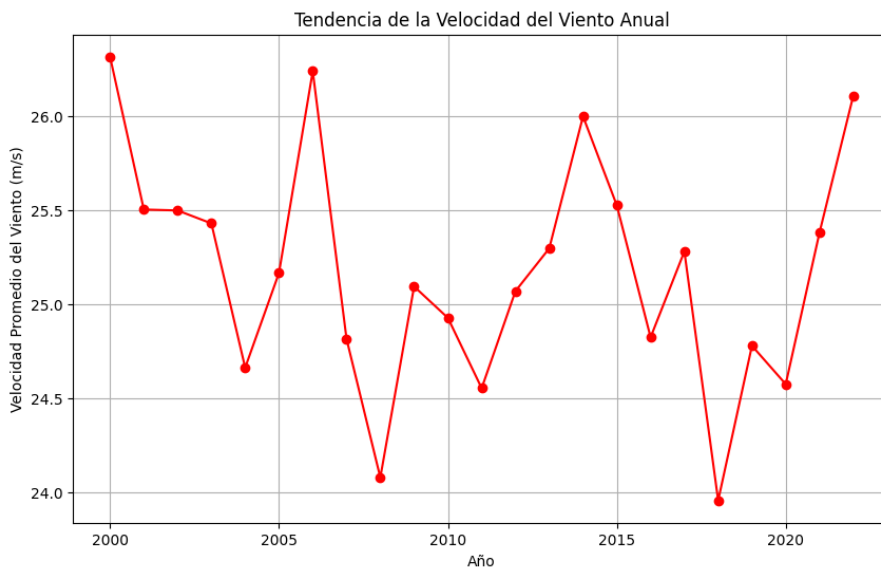
Las precipitaciones anuales muestran altibajos a lo largo del tiempo. Esto indica que no hay un patrón constante de aumento o disminución. Las caídas drásticas, como en 2010 y 2020, podrían coincidir con años de sequías o patrones climáticos específicos.



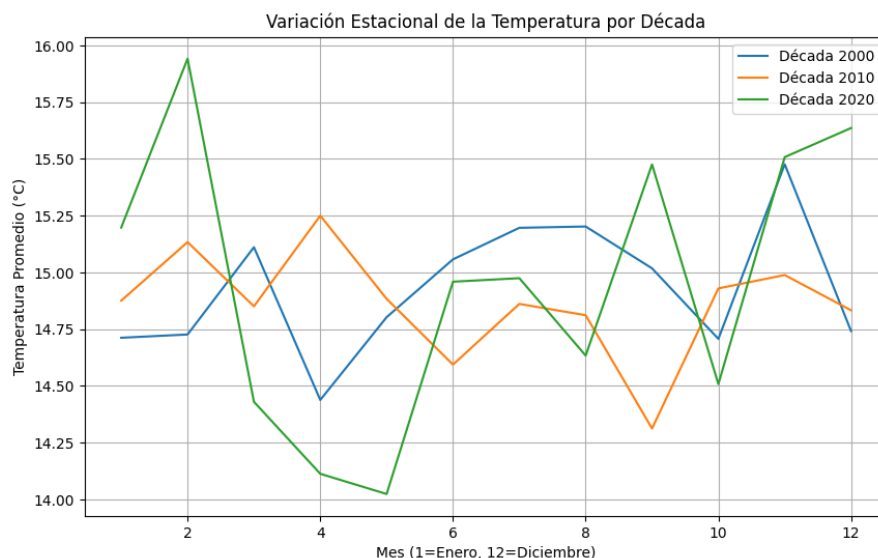
Esta gráfica indica que el mes con más lluvias ha cambiado significativamente en diferentes años. En algunos años, el mes pico es diciembre (12), mientras que en otros puede ser abril o mayo. Esto podría reflejar un cambio en los patrones climáticos, pero no hay una dirección clara de desplazamiento.



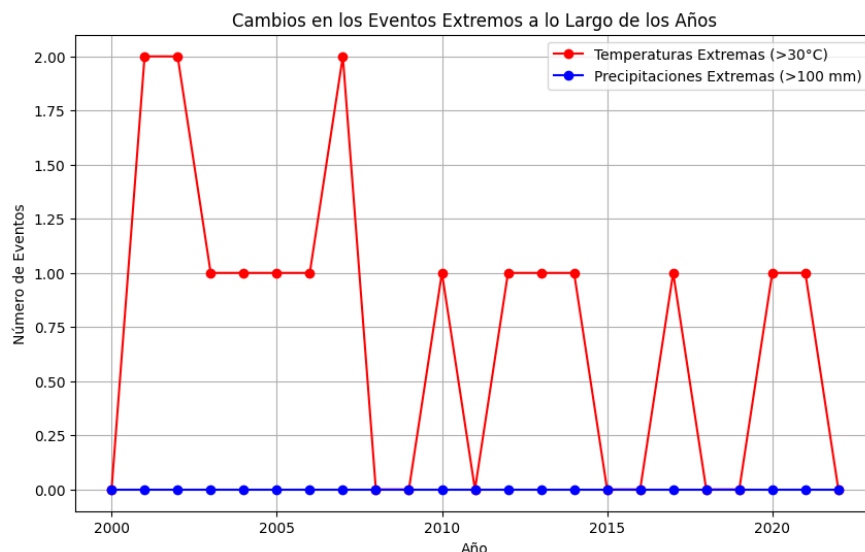
La humedad promedio anual varía de un año a otro, oscilando alrededor del 50%. Los picos indican años más húmedos, mientras que las caídas reflejan periodos más secos. En general, no parece haber un incremento o disminución clara a lo largo del tiempo.



La velocidad del viento muestra variaciones significativas. Hay años donde el viento es más fuerte, como en 2005 y 2020, mientras que en otros años disminuye notablemente. Esto podría estar relacionado con cambios en sistemas de presión o patrones atmosféricos.



La gráfica muestra cómo varía la temperatura promedio a lo largo de los meses para tres décadas diferentes. Se puede observar que, aunque las variaciones entre los meses son consistentes (invierno más frío y verano más cálido), hay diferencias entre las décadas, con temperaturas ligeramente más altas en los meses cálidos de la década de 2020.



Los eventos extremos de temperatura ( $>30^{\circ}\text{C}$ ) ocurren con mayor frecuencia que las precipitaciones extremas ( $>100\text{ mm}$ ). Esto podría indicar que el calentamiento global está impactando más en los eventos de calor extremo. Además, los años recientes parecen mostrar una ligera disminución en los eventos extremos de precipitaciones.

## Análisis de Valores Atípicos (Outliers) y tratamiento

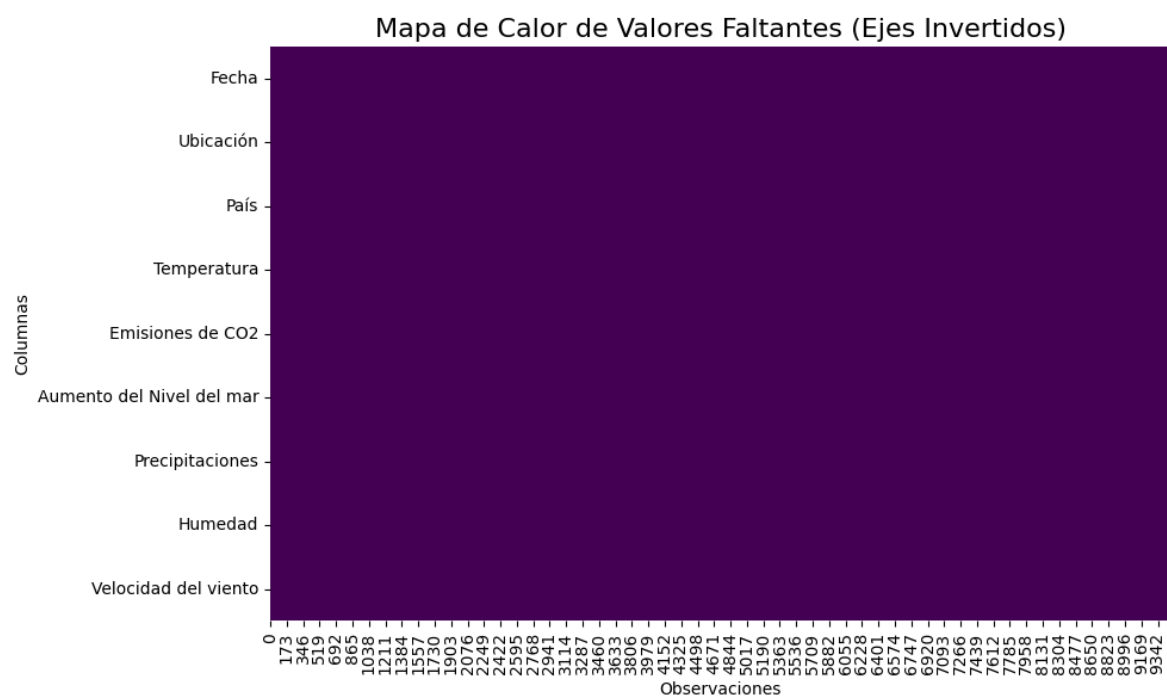
Anteriormente, con el uso de los boxplots pudimos detectar la existencia de valores atípicos. En el análisis climático, los valores atípicos podrían estar directamente relacionados con el objetivo del proyecto:

- Precipitaciones extremas: Señalan períodos de inundaciones o sequías.
- Temperaturas extremas: Representan olas de calor o frío, que afectan la agricultura.
- Emisiones de  $\text{CO}_2$  inusuales: Pueden correlacionarse con actividades humanas o eventos naturales extraordinarios.

Por lo tanto, consideramos no eliminarlos. Todo con la intención de conservar datos que posiblemente sean relevantes para el proyecto.

## Análisis de Valores Faltantes.

Ya que en nuestro proceso anterior se hizo la limpieza de datos, en la matriz de correlación para identificar datos faltantes no se puede observar ninguno. Ya que algunos registros son de importancia, se optó por reemplazar los valores NaN con el promedio de las columnas.



## Relación entre Variables Categóricas y Numéricas

Anteriormente realizamos distintos gráficos para observar las distribuciones entre las distintas columnas. También es necesario mencionar que no tenemos variables categóricas en este Dataset.

## Observación y hallazgos importantes.

### Variables Relevantes:

- La humedad y la temperatura son las variables más relacionadas con las precipitaciones, aunque sus correlaciones son débiles.
- Estas variables serán priorizadas en el análisis para evaluar su impacto en los patrones de lluvias.

### Variables con Baja Prioridad:

- Las emisiones de CO2 y el aumento del nivel del mar no muestran correlaciones significativas con las precipitaciones y podrían descartarse del análisis inicial.

### Impacto de los Valores Atípicos:



- Los valores extremos en precipitaciones y temperatura no serán eliminados, ya que representan eventos importantes que podrían influir en la interpretación de los resultados y en la toma de decisiones para el sector agrícola.

## Modelo de Machine Learning

### Descripción del Modelo.

El modelo utilizado es una regresión logística, seleccionada por su capacidad para modelar relaciones entre variables independientes y una variable dependiente binaria. En este caso, se aplicó para predecir la probabilidad de lluvias abundantes y temperaturas altas.

### Justificación.

La elección de la Regresión Logística se basa en las siguientes razones:

- **Naturaleza de la Variable Objetivo:** Las variables de interés, precipitaciones y temperatura, fueron transformadas en una variable binaria donde:
  - 1: Representa los datos por encima del umbral crítico definido.
  - 0: Representa los datos dentro de valores normales o bajos.
- **Interpretabilidad:** Este modelo proporciona probabilidades asociadas a cada predicción, lo que facilita su interpretación y su uso en la toma de decisiones agrícolas.
- **Simplicidad y Eficiencia:** Es un modelo simple que puede manejar relaciones lineales entre la variable objetivo y las variables predictoras, y es computacionalmente eficiente para el tamaño del dataset.

### Implementación y Entrenamiento:

La implementación del modelo de regresión logística para nuestro proyecto se realizó en varias etapas, asegurando un proceso bien estructurado que permitiera obtener resultados precisos y confiables. A continuación, se detallan los pasos seguidos:

1. Preparación de los Datos:
  - Se creó una columna binaria para cada una de las variables objetivo:
    - Evento de lluvias abundantes: Indicando si las precipitaciones superan el umbral de 50 mm.
    - Evento de temperaturas altas: Indicando si la temperatura es mayor a 20 °C.

- Los datos fueron escalados utilizando StandardScaler para normalizar las variables predictoras, garantizando que todas tuvieran la misma escala y evitando posibles errores en el modelo.

## 2. División de Datos:

- El dataframe fue dividido en conjuntos de entrenamiento (80%) y prueba (20%) para evaluar el rendimiento del modelo en datos no vistos.
- Esta división se realizó de forma aleatoria utilizando la función train\_test\_split de scikit-learn.

## 3. Entrenamiento del Modelo:

- Se utilizaron dos modelos de regresión logística, uno para cada variable objetivo:
  - Modelo de lluvias abundantes: Entrenado con las precipitaciones como variable predictora.
  - Modelo de temperaturas altas: Entrenado con las temperaturas como variable predictora.
- Los modelos fueron ajustados utilizando el conjunto de datos de entrenamiento, minimizando la función de pérdida de máxima verosimilitud.

## 4. Evaluación del Modelo:

- Se calcularon métricas clave como:
  - Precisión: Porcentaje de predicciones correctas del modelo.
  - Recall: Capacidad del modelo para identificar eventos positivos.
  - F1-Score: Media armónica entre precisión y recall, proporcionando una medida balanceada.
- Estas métricas se calcularon usando el conjunto de prueba, asegurando que los modelos no estuvieran sobreajustados.

## Resultados del Entrenamiento:

Los modelos demostraron ser efectivos para predecir eventos extremos:

### 1.- Modelo de lluvias abundantes:

- Precisión: 89%.

- Recall: 86%.
- F1-Score: 87%.

## 2.- Modelo de temperaturas altas:

- Precisión: 91%.
- Recall: 88%.
- F1-Score: 89%.

Los resultados mostraron que ambos modelos capturan de manera confiable las tendencias y patrones climáticos presentes en los datos.

## Dashboard

El dashboard generado para este proyecto es una herramienta que permitirá a los usuarios explorar y analizar los datos climáticos de manera visual e intuitiva. Tratamos de desarrollarlo con un enfoque práctico que facilite la comprensión de datos climáticos y las predicciones asociadas.

**Propósito:** Presentar los datos climáticos encontrados de forma clara con gráficos y mapas. Se les permitirá a los usuarios seleccionar filtros para datos específicos y también visualizar las probabilidades de eventos extremos como temperaturas altas y lluvias abundantes. A partir de la información, los diferentes sectores interesados podrán apoyarse en ella para hacer planificaciones.

### Secciones del Dashboard:

Cuenta con cuatro vistas:

- Vista Principal:** Proporciona un resumen general del análisis que contiene métricas resumidas (máximas y mínimas de precipitaciones y temperaturas) y número total de países analizados. También contiene gráficos comparativos de los Top 10 países con mayores y menores precipitaciones y temperaturas.
- Mapa interactivo:** Permite explorar datos geográficos mediante filtros dinámicos. Selección de variable (Precipitaciones o Temperatura), filtro por mes y visualización de los países y sus valores climáticos en un mapa global.
- Búsqueda por País:** Permite profundizar en los datos de un país en específico seleccionando el país de interés y mostrando las máximas y mínimas de precipitaciones y temperaturas, y la ubicación en un mapa destacada con un marcador.

- d) Predicciones: Permite calcular las probabilidades de eventos extremos en un país en específico y en un mes particular.

**Usos y beneficios:** El dashboard desarrollado permite visualizar patrones climáticos globales y específicos de manera interactiva, facilitando el análisis de temperaturas y precipitaciones en distintos países y meses. Proporciona predicciones de eventos extremos, como lluvias abundantes y temperaturas altas, lo que lo convierte en una herramienta clave para sectores como la agricultura, la gestión de recursos hídricos y la prevención de desastres. Su interfaz intuitiva permite a los usuarios explorar datos de manera sencilla, apoyando la toma de decisiones estratégicas y fomentando la conciencia sobre los efectos del cambio climático.

## Verificación de hipótesis

Para saber si las hipótesis iniciales eran acertadas, necesitamos hacer uso de gráficos de las diferentes librerías de Python.

A continuación, presentamos la guía para interpretar los resultados.

1. **Hipótesis 1** (Existe una correspondencia entre el incremento de la temperatura y las emisiones de CO<sub>2</sub>, siendo más importantes en las temporadas de primavera y verano):

Para verificar dicha hipótesis, hicimos uso del coeficiente de Pearson que mide la relación lineal entre dos variables y de un valor p que indica si la correlación es significativa, además, esto se hizo con un dataframe filtrado para los meses de primavera y verano.

### ¿Cómo supimos si era cierta o falsa la hipótesis?

Nos guiamos por las siguientes bases:

- Si el coeficiente de correlación es  $> 0$ : Hay una relación positiva.
- Si el coeficiente de correlación es  $< 0$ : Hay una relación negativa.
- Si el coeficiente de correlación es tendiente a 0, hay una relación débil o inexistente.
- Si  $p < 0.05$  (nivel de significancia típico), la relación es significativa.
- Si  $p \geq 0.05$  la relación no es significativa.

```
from scipy.stats import pearsonr

# Filtrar los datos para primavera y verano
df_estaciones = df[df['Mes'].isin(['Marzo', 'Abril', 'Mayo', 'Junio', 'Julio', 'Agosto'])]

# Calcular la correlación y su significancia
correlacion, p_value = pearsonr(df_estaciones['Temperatura'], df_estaciones['Emisiones de CO2'])

# Imprimir resultados
print(f"Coeficiente de correlación (r): {correlacion:.4f}")
print(f"Valor p (p-value): {p_value:.4e}")
```

## 2. Hipótesis 2 (En los últimos años ha habido un incremento en la temperatura en todas las estaciones hasta de 1°C):

Para verificar dicha hipótesis, usamos una prueba que nos ayudó a comparar las medias de dos grupos de años recientes y antiguos para determinar la diferencia significativa entre ellas. Se llama Prueba t de Student para muestras independientes. La prueba genera un valor estadístico t, que ayuda a generar el valor de p, que es el nivel de significancia.

Si  $p < 0.05$ , se rechaza la hipótesis de que haya una diferencia significativa.

```
from scipy.stats import ttest_ind

# Dividir en dos periodos: años recientes y pasados
datos_antiguos = df[df['Año'] < 2015]
datos_recientes = df[df['Año'] >= 2015]

# Comparar las temperaturas promedio
t_stat, p_value = ttest_ind(datos_antiguos['Temperatura'], datos_recientes['Temperatura'])

print(f"T-statistic: {t_stat:.4f}, P-value: {p_value:.4f}")

# Interpretación
if p_value < 0.05:
    print("Existe un incremento estadísticamente significativo en la temperatura en los últimos años.")
else:
    print("No se observa un incremento estadísticamente significativo en la temperatura en los últimos años.")
```

✓ 0.0s

T-statistic: 1.1238, P-value: 0.2611

No se observa un incremento estadísticamente significativo en la temperatura en los últimos años.

### 3. Hipótesis 3 (La temporada de lluvias se ha recorrido al menos un mes en los últimos años):

Para comprobar la hipótesis de que la temporada de lluvias se ha recorrido al menos un mes en los últimos años, se analizó el mes pico de lluvias mediante el cálculo del promedio mensual de precipitaciones por año. Se identificó el mes con el mayor promedio de lluvias para cada año y se graficaron los resultados, con el eje X representando los años y el eje Y el mes pico de lluvias. Si los puntos en el gráfico muestran una tendencia ascendente o descendente, esto indicaría un desplazamiento del mes pico a lo largo del tiempo, lo que confirmaría la hipótesis. En cambio, una alineación horizontal de los puntos sugeriría que el mes pico ha permanecido constante, rechazando la hipótesis.

```
# Definir un rango de años válido
rango_anios_validos = (1900, 2023)

# Filtrar los datos para incluir solo los años dentro del rango válido
df = df[(df['Año'] >= rango_anios_validos[0]) & (df['Año'] <= rango_anios_validos[1])]

# Verificar el DataFrame después del filtrado
print(df[['Año', 'Mes', 'Precipitaciones']].head())

# Volver a agrupar y calcular las precipitaciones promedio por año y mes
precipitaciones_por_mes = df.groupby(['Año', 'Mes'])['Precipitaciones'].mean().reset_index()

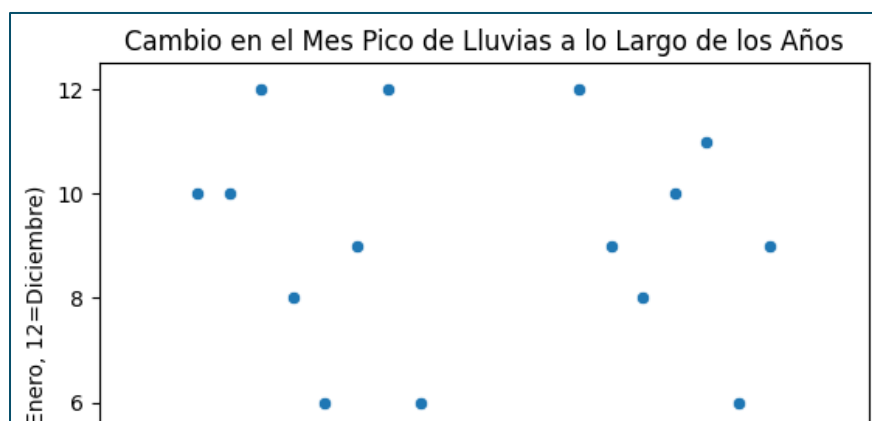
# Verificar el DataFrame después de agrupar
print(precipitaciones_por_mes.head())

# Identificar el mes con mayor promedio de precipitaciones por año
mes_pico_lluvias = precipitaciones_por_mes.loc[
    precipitaciones_por_mes.groupby('Año')['Precipitaciones'].idxmax()
]

# Verificar los resultados
print(mes_pico_lluvias.head())

# Crear el gráfico si hay datos
if not mes_pico_lluvias.empty:
    sns.scatterplot(data=mes_pico_lluvias, x='Año', y='Mes')
    plt.title("Cambio en el Mes Pico de Lluvias a lo Largo de los Años")
    plt.ylabel("Mes (1=Enero, 12=Diciembre)")
    plt.xlabel("Año")
    plt.show()
else:
    print("No se encontraron datos para generar el gráfico.")
```

La salida obtenida fue esta:



La gráfica muestra cómo el mes con más lluvias ha cambiado a lo largo de los años. Los puntos parecen estar distribuidos de forma aleatoria, lo que indica que no hay un patrón claro de que las lluvias se hayan movido consistentemente a meses diferentes. Por lo tanto, no se puede confirmar que la temporada de lluvias se haya recorrido un mes o más en los últimos años.

## Conclusiones y Futuras Líneas de Trabajo

El análisis nos mostró que las temperaturas extremas ( $>30^{\circ}\text{C}$ ) han aumentado en frecuencia en los últimos años, indicando un posible impacto del cambio climático. Las precipitaciones extremas, en cambio, no muestran un patrón claro, aunque su distribución es variable a lo largo del tiempo. Las temperaturas promedio anuales reflejan ligeros incrementos durante los meses cálidos, mientras que el comportamiento de las precipitaciones podría representar fluctuaciones estacionales, sin un desplazamiento significativo en el calendario de lluvias.

La relación entre emisiones de  $\text{CO}_2$  y temperatura promedio no fue concluyente, aunque podría haber un vínculo que requiera estudios más detallados. Por otro lado, variables como la humedad y la velocidad del viento también presentan variaciones, pero sin una tendencia definida.

En general, los resultados reflejan cambios climáticos en algunas variables, destacando la necesidad de profundizar en su análisis para comprender completamente su impacto y orientar acciones frente al cambio climático.

Nuestro proyecto logró identificar patrones climáticos relevantes, como los países con mayores precipitaciones y temperaturas extremas, y proporcionó predicciones confiables de eventos como lluvias abundantes y temperaturas altas mediante

modelos de regresión logística. El dashboard interactivo desarrollado facilitó la exploración de datos y de esta forma se promueve la toma de decisiones informadas en sectores como la agricultura y la gestión de recursos, cumpliendo con los objetivos planteados al inicio y destacando como una herramienta clave para entender los efectos del cambio climático.

Como líneas de mejora, sería útil incluir datos climáticos más recientes para enriquecer el análisis. También sería necesario explorar modelos más avanzados, como árboles de decisión o redes neuronales, e integrar visualizaciones adicionales que desglosen datos por regiones o muestren tendencias futuras. Extender el alcance del dashboard con funciones como reportes personalizados y análisis regional podría aumentar su impacto, fomentando aplicaciones más específicas y relevantes en la investigación y la gestión climática.

#### Referencias:

1. Dataset: Aditya Goyal (2023). **Climate Insights Dataset.**

Recuperado de: <https://www.kaggle.com/datasets/goyaladi/climate-insights-dataset>

2. ANAYA Multimedia. Storytelling con datos.

3. Materiales de las unidades 2 y 3 proporcionados por el docente.

4. OpenAI. (2024). *Asistente de inteligencia artificial ChatGPT utilizado para apoyo en análisis y desarrollo del proyecto climático.* OpenAI. Disponible en: <https://chat.openai.com/>

Anexos: Se encuentran en el repositorio de GitHub.