

# Taller

## Introducción al análisis de Series de tiempo

### 1 Objetivo

Facilitar el análisis de series de tiempo tiene como herramienta para modelar y explicar secuencias de datos (mediciones) con respecto al tiempo.

Usar técnicas estadísticas para analizar secuencias de datos y extraer conclusiones o interpretaciones de los resultados.

### 2 Entrega

- Este taller se entrega de forma individual.
- Cada estudiante selecciona un conjunto de datos del Portal de Estadísticas Económicas del Banco de la República <https://suameca.banrep.gov.co/estadisticas-economicas/>. El conjunto de datos debe ser distinto al utilizado como ejemplo de Tasa de desempleo en Colombia - total nacional.
- Cargue un Python notebook .ipynb a la plataforma con su código y las respuestas del taller.
- Cargue el archivo de datos utilizado para el taller, tanto en el formato original como el modificado.

### 3 Taller

Los ejemplos que este documento se basan en la secuencia de datos de la Tasa de Desempleo en Colombia - Total Nacional entre los años 2000 y 2024. Este conjunto de datos se publica junto con el taller con propósito demostrativo.

Adjunto a este taller encontrará un documento de la Oficina Australiana de Estadísticas titulado *A Guide to Interpreting Time Series* donde se referencian las lecturas correspondientes.

## 4 Preguntas

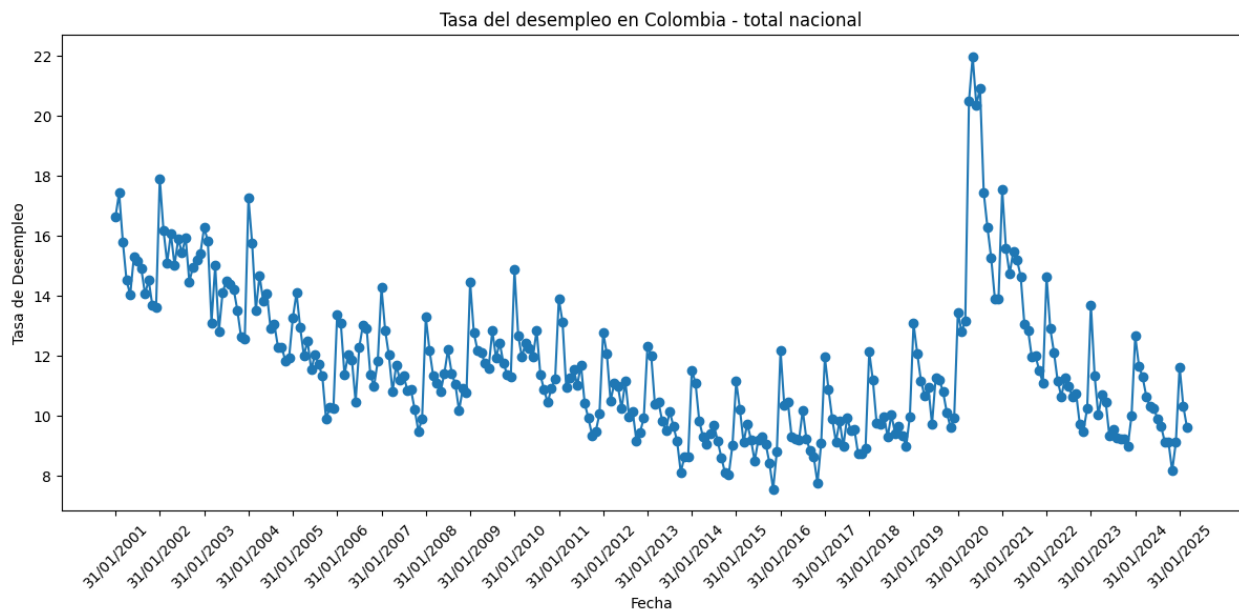
1. Seleccione y descargue un conjunto de datos del Portal de Estadísticas Económicas del Banco de la República.
2. Cargue la secuencia de datos como una **Series** de pandas, y asigne el índice la variable tiempo.

```
desempleo_df = pd.read_csv("./desempleo_colombia_total_nacional.csv")
desempleo_df['date'] = pd.to_datetime(desempleo_df['fecha'], format='%d/%m/%Y')
desempleo_ts = pd.Series(desempleo_df['desempleo'].values,
index=desempleo_df['date'], name='Desempleo')
```

3. Grafique la serie con respecto al tiempo. Utilice las siguientes bibliotecas de Python

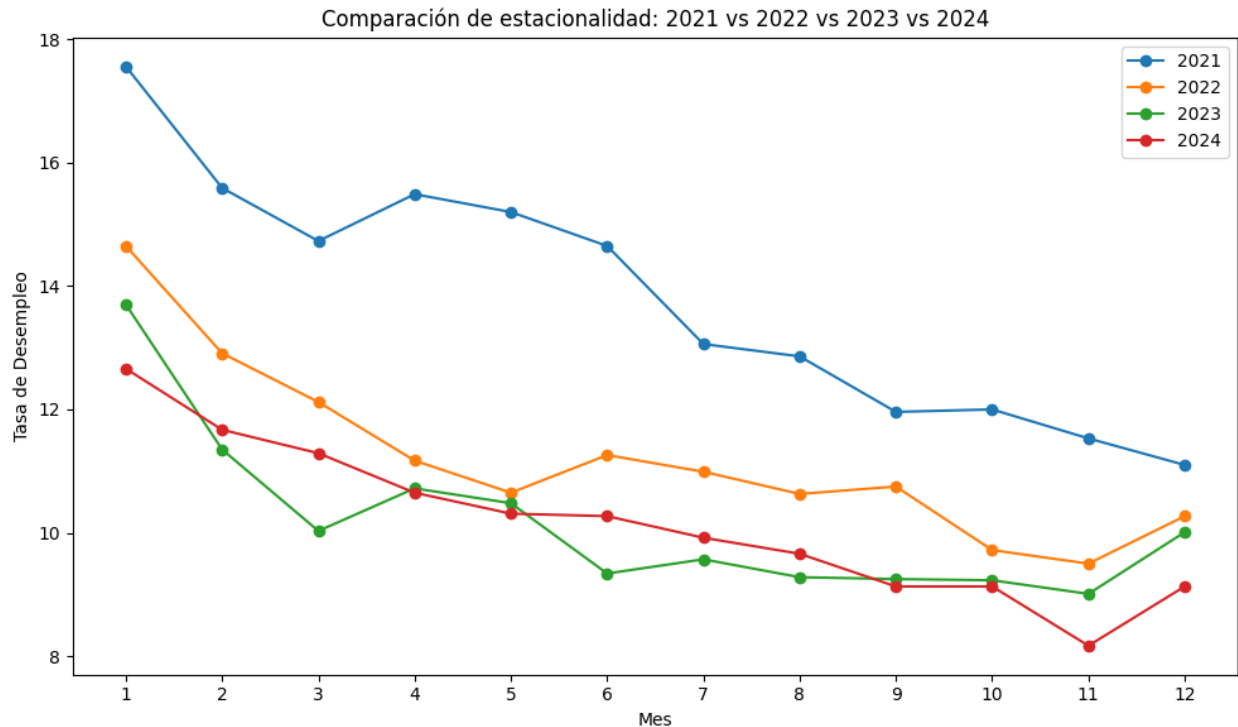
```
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.formula.api as sm
from statsmodels.tsa import tsatools
```

Ejemplo:



4. Lea la sección 2.1 acerca de los componentes de una serie de tiempo. Grafique segmentos de la serie con el propósito de observar estacionalidades.

### Ejemplo



5. Lea la documentación del uso de tsatools <https://www.statsmodels.org/stable/tsa.html> especialmente la función `add_trend` que agrega valores de tendencia a la serie de datos.
  - a. ¿qué significa parámetro 'ctt'?
  - b. ¿cómo funciona el objeto *formula* de la función *ols*?
  - c. ¿cuál es el significado del operador tilde (~) en la fórmula?
6. Cree un modelo a partir del API de formula y grafique la tendencia.

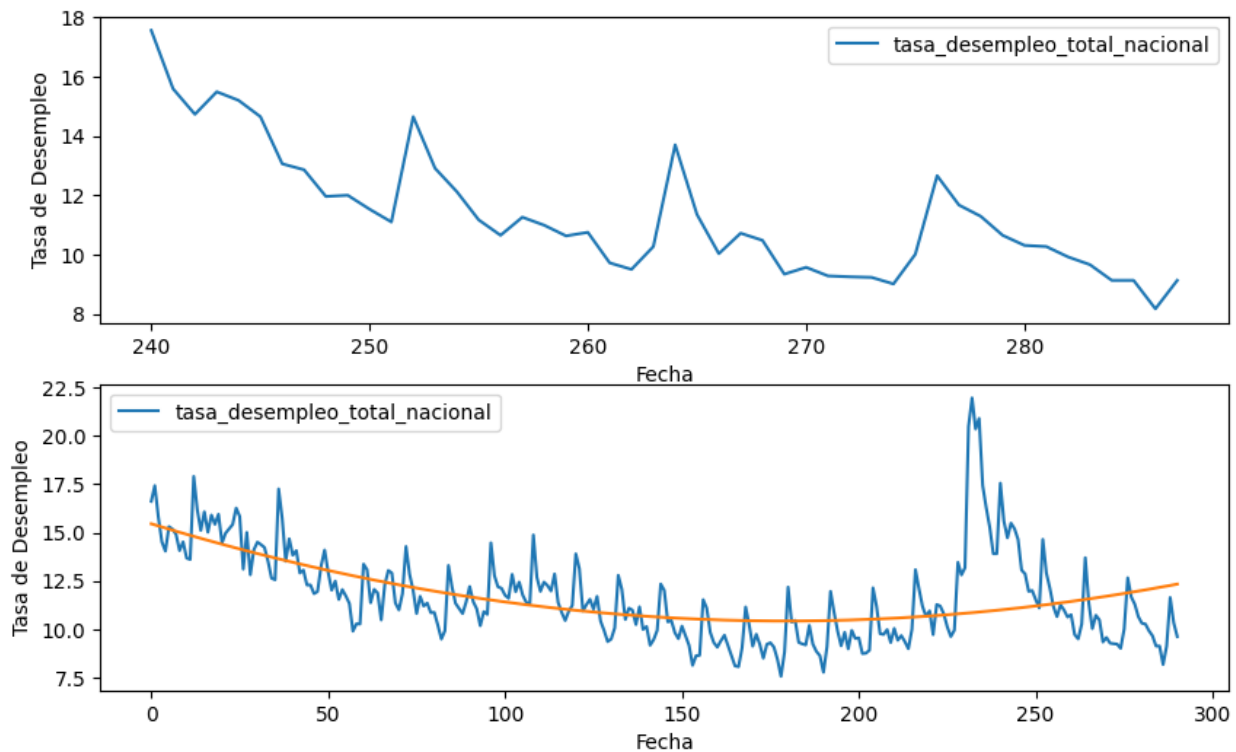
### Ejemplo

```
desempleo_trend_df =  
tsatools.add_trend(desempleo_df['tasa_desempleo_total_nacional'],  
trend='ctt')  
  
desempleo_lm = sm.ols(formula='tasa_desempleo_total_nacional ~ trend +  
trend_squared', data=desempleo_trend_df).fit()  
  
fig, axes = plt.subplots(nrows=2, ncols=1, figsize=(10,6))  
four_years_df[['tasa_desempleo_total_nacional']].plot(ax=axes[0])
```

```

desempleo_df[['tasa_desempleo_total_nacional']].plot(ax=axes[1])
for ax in axes:
    ax.set_xlabel('Fecha')
    ax.set_ylabel('Tasa de Desempleo')
desempleo_lm.predict(desempleo_trend_df).plot(ax=axes[1])
plt.show()

```



7. Calcule la estacionalidad. Consulte la documentación de la función `seasonal_decompose` y describa el comportamiento de sus datos con respecto a la estacionalidad y el tipo de descomposición aplicado.

### Ejemplo

```

from statsmodels.tsa.seasonal import seasonal_decompose

multiplicative_decomposition = seasonal_decompose(desempleo_ts,
model='multiplicative', period=30)

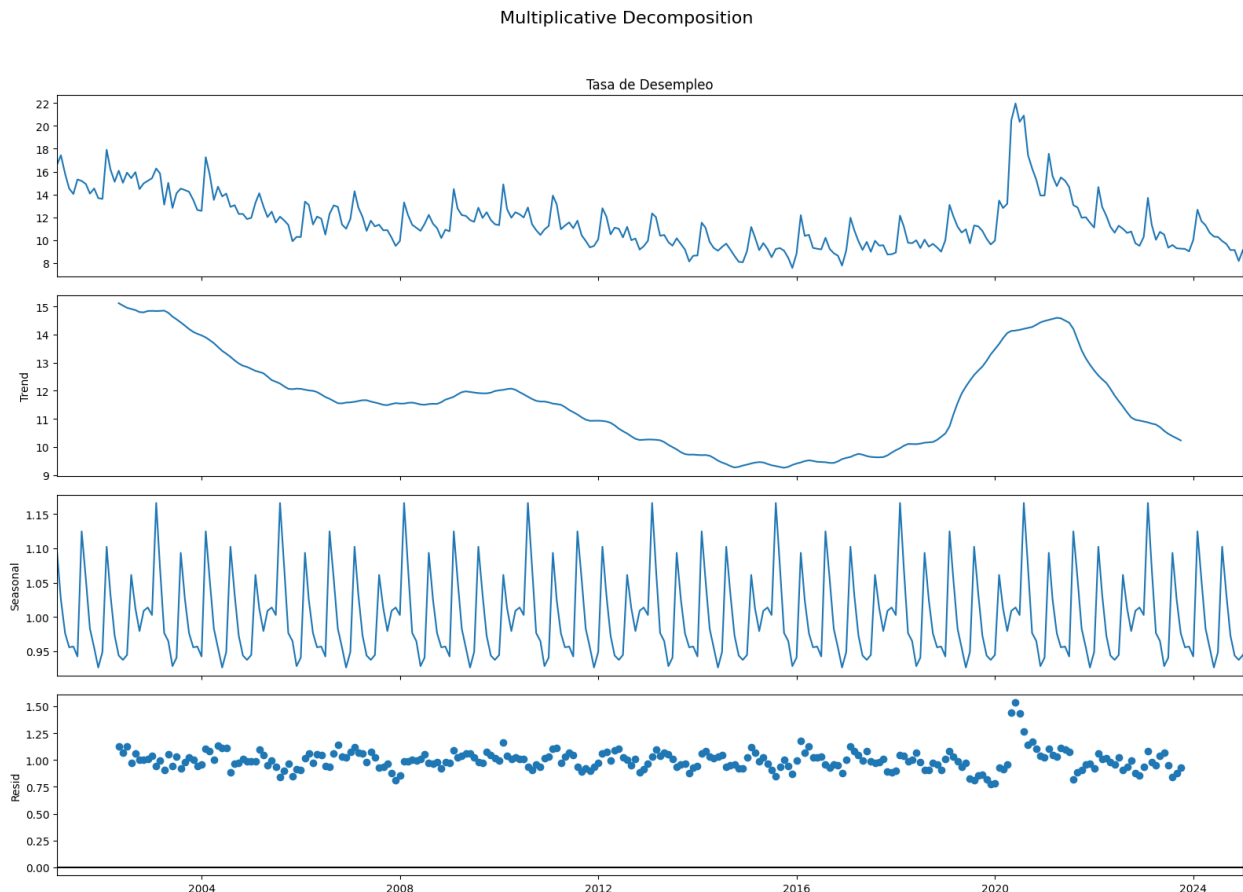
additive_decomposition = seasonal_decompose(desempleo_ts, model='additive',
period=30)

plt.rcParams.update({'figure.figsize': (16,12)})
multiplicative_decomposition.plot().suptitle('Multiplicative
Decomposition', fontsize=16)

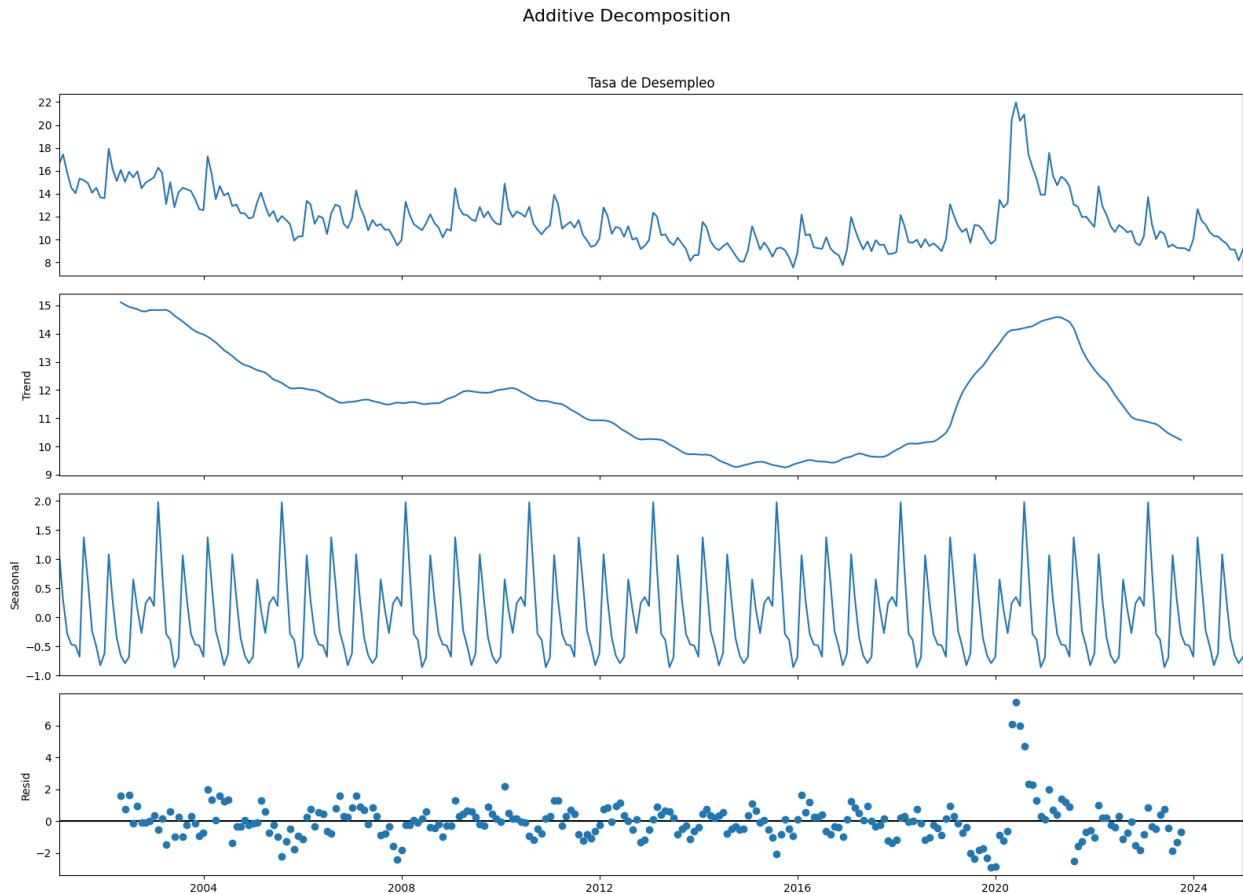
```

```
plt.tight_layout(rect=[0, 0.03, 1, 0.95])

additive_decomposition.plot().suptitle('Additive Decomposition',
fontsize=16)
plt.tight_layout(rect=[0, 0.03, 1, 0.95])
plt.show()
```



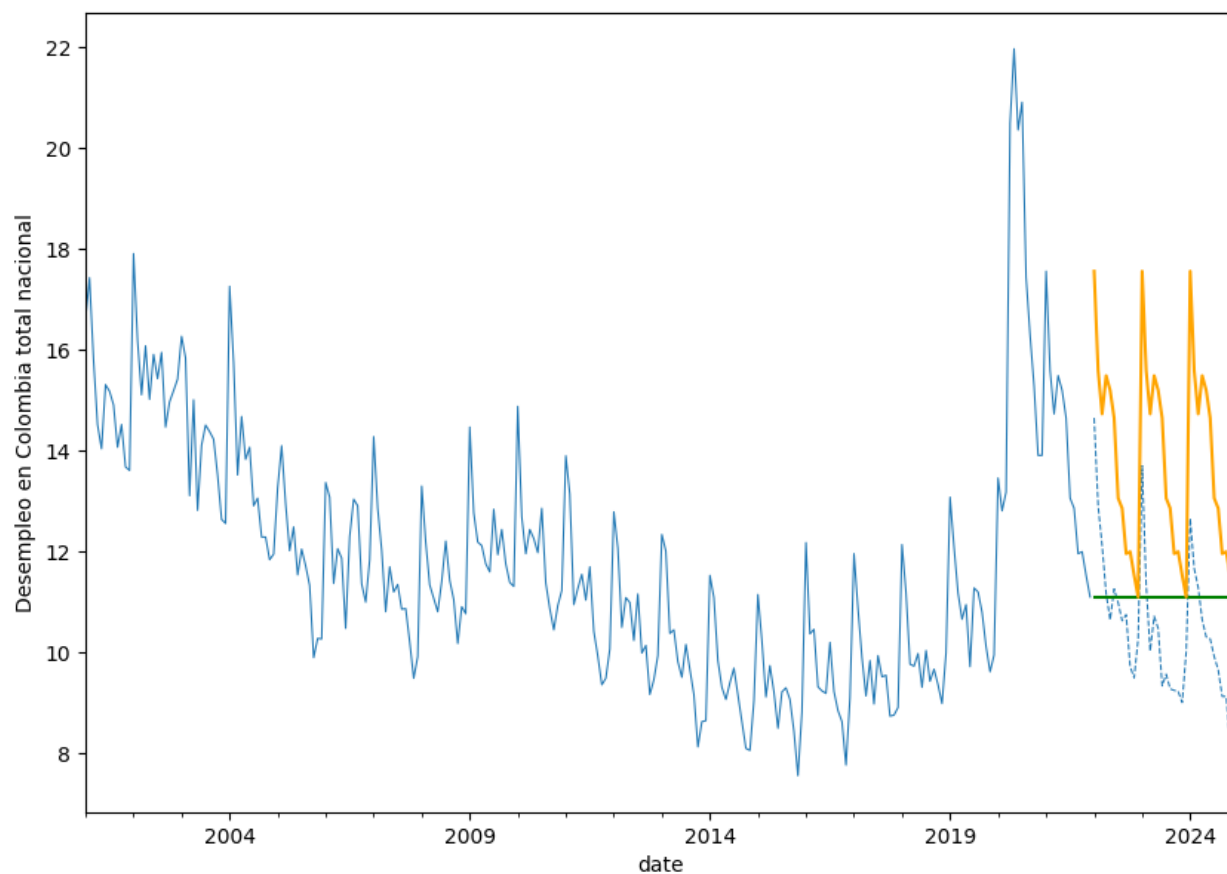
8. ¿cuáles son las diferencias entre la descomposición aditiva y la multiplicativa y cómo afecta el tipo de descomposición en la interpretación de su serie de datos?
9. Explique la razón por la cual son distintos los componentes estacionales de la descomposición aditiva de la multiplicativa.
10. ¿Su serie tiene picos o valles muy prolongados como los del ejemplo? Si es así, ¿cómo se expresan estos valores atípicos en el residuo de la descomposición?



11. Seleccione un punto en el tiempo para dividir la serie. El ejemplo mostrado tiene 288 puntos (periodos), se utilizó como punto de división los últimos 36 periodos, es decir 3 años. Predicción Naive simplemente mantiene el mismo valor final para los demás puntos. Cuando hay estacionalidad, se tiene simplemente el valor del periodo anterior ajustado a la estacionalidad.

- a. La primera serie, desde el inicio hasta el punto de corte es utilizada para entrenamiento. Línea azul continua en el ejemplo.
- b. La serie posterior, desde el punto de corte hasta el final, serie se utiliza para validación. Línea azul punteada en el ejemplo.
- c. Asuma que la predicción corresponde a una copia del último segmento estacional (para el ejemplo 12 periodos) predicción. Línea naranja en el ejemplo.

12. Grafique la serie de prueba, hasta el punto de corte, y luego la serie real (utilizada como validación) y la serie estimada



13. Construya una función que determine el error entre dos series y muestre el ME, RMSE, MAE, MPE y MAPE.

- Cuales son los valores de error entre la serie de validación (real) la estimación simple de los valores de la última estación.
- Escriba el análisis de las medidas de error calculadas.

Ejemplo de la medida de error entre el pronóstico Naive, línea verde del gráfico, con respecto a los valores reales, línea punteada.

Mean Error (ME) : -0.5839

Root Mean Squared Error (RMSE) : 1.4943

Mean Absolute Error (MAE) : 1.2383

Mean Percentage Error (MPE) : -7.2090

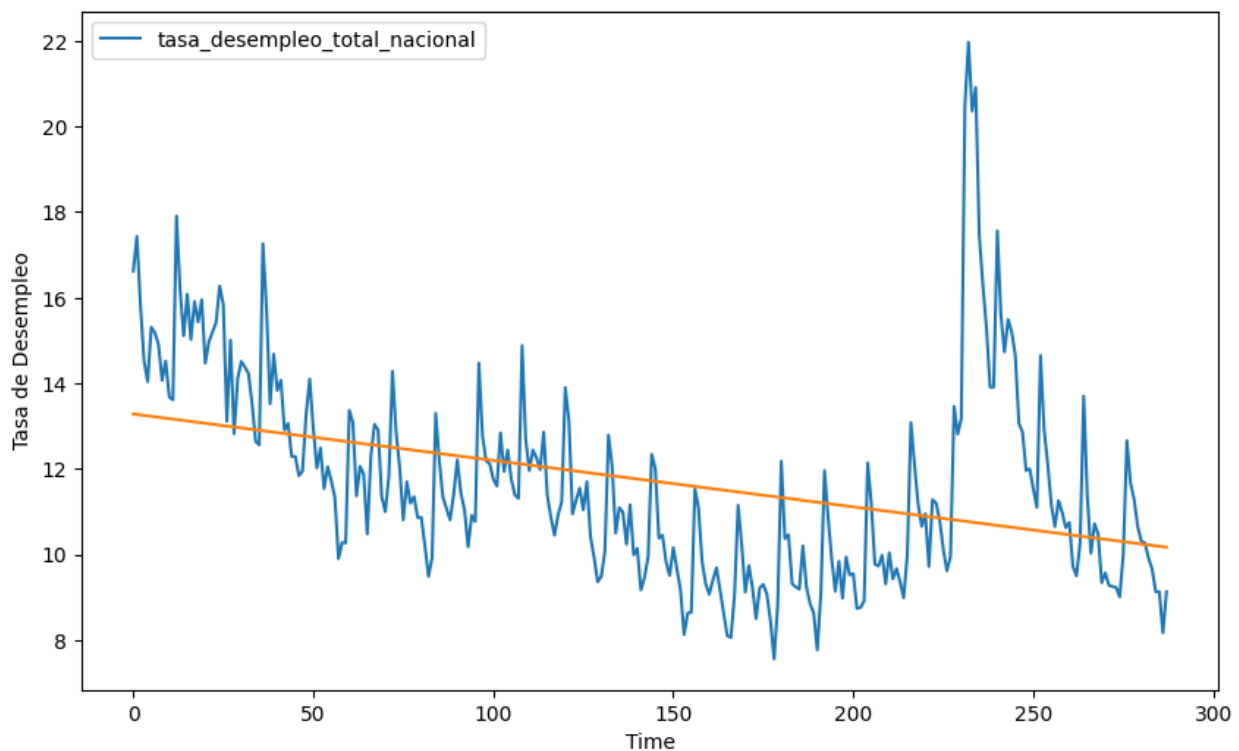
Mean Absolute Percentage Error (MAPE) : 12.1415

14. Aplique un modelo de regresión lineal para capturar la tendencia y estacionalidad. Describa las ventajas y limitaciones de la regresión lineal para explicar la tendencia y estacionalidad de la serie.

Ejemplo

```
import matplotlib.pyplot as plt
import statsmodels.formula.api as sm
from statsmodels.tsa import tsatools

desempleo_df = tsatools.add_trend(desempleo_ts, trend='ct')
ridership_lm = sm.ols(formula='tasa_desempleo_total_nacional ~ trend',
data=desempleo_df).fit()
# shorter and longer time series
ax = desempleo_ts.plot(figsize=(10, 6))
ax.set_xlabel('Time')
ax.set_ylabel('Tasa de Desempleo')
ridership_lm.predict(desempleo_df).plot(ax=ax)
plt.show()
```



15. BONO

- Investigue sobre los modelos de autocorrelación y ARIMA.
- Investigue sobre el suavizado con promedio móvil (MA)