

```
#Sección API INEGI--BANXICO
!pip install INEGIPy
# =====
from INEGIPy import Indicadores
#from fredapi import Fred
import requests
import json
import requests
import json
import datetime
import plotly.graph_objects as go
#import geopandas
# =====
#Librerías básicas
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
plt.style.use('fivethirtyeight') #ggplot
import statsmodels.api as sm
import shapely
# =====
# Gráficos fancys
import plotly.express as px
import plotly.graph_objects as go
from statsmodels.graphics.gofplots import qqplot
#import plotly.figure_factory as ff
# =====
# Otras cosas útiles
import warnings
warnings.filterwarnings('ignore')
```

Token INEGI

```
# Define tu token de acceso
token = "tu-token-aquí"
# Crea una instancia de la clase Indicadores
inegi = Indicadores(token)
# Obtiene el catálogo de indicadores para 'BIE'
indicadores = inegi.catalogo_indicadores('BIE')

# Obtiene un DataFrame con los datos de los indicadores seleccionados
df = inegi.obtener_df(indicadores = ["628194","628195","628196","628197","628198","628199","628200"],
                        nombres = ['INPC general',
                                   'INPC subyacente',
                                   'INPC subyacente mercancías',
                                   'INPC subyacente servicios',
                                   'INPC no subyacente',
                                   'INPC no agropecuario',
                                   'INPC no energeticos'],
                        inicio = '2000',
                        fin = '2023')

# Calcula la inflación anualizada y otros valores
df['Inflacion'] = df['INPC general'].pct_change(12) #Inflación anualizada
df['Core'] = df['INPC subyacente'].pct_change(12) #Inflación subyacente anualizada
df['No Sub'] = df['INPC no subyacente'].pct_change(12) #Inflación no subyacente anualizada
df["Target"] = 0.03 #Objetivo de inflación de Banxico
df["Gap"] = df['Inflacion']-0.03 #Diferencial de inflación
df['In Target'] = np.where(df['Inflacion'].between(0.02,0.04), 1, 0) #Crea columna para saber si la inflación está dentro del objetivo

# Elimina filas con valores faltantes y reinicia el índice
df = df.dropna(axis=0)
df = df.reset_index(drop=True)
# Crea una columna de fechas
df['Fecha'] = pd.date_range('2001-01-01', periods = len(df), freq='MS')
# Establece la columna de fechas como índice
df.set_index('Fecha',inplace=True)
df
```

	INPC general	INPC subyacente	INPC subyacente mercancias	INPC subyacente servicios	INPC no subyacente	INPC no agropecuario	INPC no energeticos	Inflacion	Core	No Sub	Target	
Fecha												
2001-01-01	48.575476	52.524223	51.154392	53.318948	37.591134	38.398598	36.923007	0.081117	0.072598	0.105713	0.03	0.051
2001-02-01	48.543328	52.920347	51.541543	53.719405	36.702649	35.905804	37.312734	0.070902	0.068926	0.076652	0.03	0.040
2001-03-01	48.850888	53.237143	51.789193	54.115363	36.972223	36.417318	37.389455	0.071745	0.068415	0.081473	0.03	0.041
2001-04-01	49.097309	53.493597	52.026419	54.391015	37.182900	37.165237	37.172150	0.071058	0.067950	0.080111	0.03	0.041
2001-05-01	49.209970	53.686532	52.182914	54.625233	37.127908	37.568271	36.752439	0.069517	0.067134	0.076488	0.03	0.039
...
2023-05-01	128.084000	128.621130	136.267873	120.364996	126.424217	141.915608	115.300315	0.058353	0.073865	0.012373	0.03	0.028
2023-06-01	128.214000	129.007619	136.648971	120.757305	125.762088	141.578285	114.404952	0.050556	0.068861	-0.003558	0.03	0.020
2023-07-01	128.832000	129.511404	137.071687	121.348621	126.733883	144.273773	114.139015	0.047858	0.066400	-0.006687	0.03	0.017
2023-08-01	129.545000	129.863773	137.400333	121.726603	128.561281	146.785023	115.475359	0.046380	0.060829	0.003728	0.03	0.016
2023-09-01	130.120000	130.333655	137.822881	122.247592	129.459067	148.011892	116.136841	0.044545	0.057567	0.005985	0.03	0.014

```
# Crea la columna 'Fecha' utilizando pd.date_range
df['Fecha'] = pd.date_range(start='2001-01-01', periods=len(df), freq='M')

# Crear el gráfico interactivo con Plotly
fig = go.Figure()
fig.add_trace(go.Scatter(x=df['Fecha'], y=df['Inflacion'], name='Inflacion', mode='lines+markers'))
fig.add_trace(go.Scatter(x=df['Fecha'], y=df['Core'], name='Core', mode='lines+markers'))
fig.add_trace(go.Scatter(x=df['Fecha'], y=df['No Sub'], name='No Sub', mode='lines+markers'))
fig.add_trace(go.Scatter(x=df['Fecha'], y=df['Target'], name='Target', mode='lines+markers'))

# Actualizar el diseño del gráfico
fig.update_layout(
    title=f"Evolución de la inflación desde {df['Fecha'].min().strftime('%B %Y')} hasta {df['Fecha'].max().strftime('%B %Y')}",
    xaxis_title='Fecha',
    yaxis_title='Valor (%)',
    legend=dict(x=0, y=1, traceorder='normal')
)

# Mostrar el gráfico interactivo
fig.show()
```

Evolución de la inflación desde January 2001 hasta September 2023

