

# Imágenes Contexto Económico - Python

October 7, 2022



UNIDAD DE ACTUARIA - IHSS

Lic. Juan Isaula

Código 1

Tema: Generación de Imágenes del contexto económico para EA - 2021

Creado: 2022-Jun-05

Actualizado: 2022-Oct-07

## 1 Librerías Utilizadas

```
[ ]: import matplotlib
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import io
import math
from matplotlib.ticker import FuncFormatter
import seaborn as sns
```

## 2 Funciones utilizadas

A lo largo del código notarán que cada gráfico hace el llamado a una función para poder nombrar las etiquetas de los mismos. Es por ello, que se crean las diferentes funciones para dicho fin.

Como los gráficos son de estilo de barra y líneas, entonces la función la llamamos *label\_comma\_bar\_porcentaje\_line* haciendo referencia a que algunas etiquetas van separadas por comas y otras son de tipo porcentaje.

```

[397]: def label_comma_bar_porcentaje_line(ax, typ, spacing=5):
    space = spacing
    va = 'bottom'

    if typ == 'bar':
        for i in ax.patches:
            y_value = i.get_height()
            x_value = i.get_x() + i.get_width() / 2

            label = f'{math.ceil(y_value):,}'
            ax.annotate(label, (x_value, y_value), xytext=(0, space),
                        textcoords="offset points", ha='center', va=va)

    if typ == 'bar_porcentaje':
        for i in ax.patches:
            y_value = i.get_height()
            x_value = i.get_x() + i.get_width() / 2

            label = '{:.1%}'.format(y_value/100)
            ax.annotate(label, (x_value, y_value), xytext=(0, space),
                        textcoords="offset points", ha='center', va=va,
                        color = 'white').set_backgroundcolor('darkorange')

    if typ == 'bar_porcentaje_orange':
        for i in ax.patches:
            y_value = i.get_height()
            x_value = i.get_x() + i.get_width() / 2

            label = '{:.1%}'.format(y_value/100)
            ax.annotate(label, (x_value, y_value), xytext=(0, space),
                        textcoords="offset points", ha='center', va=va,
                        color = 'orange')

    if typ == 'bar_comma_orange':
        for i in ax.patches:
            y_value = i.get_height()
            x_value = i.get_x() + i.get_width() / 2

            label = f'{math.ceil(y_value):,}'
            ax.annotate(label, (x_value, y_value), xytext=(0, space),
                        textcoords="offset points", ha='center', va=va,
                        color = 'orange')

    if typ == 'line':
        line = ax.lines[0]
        for x_value, y_value in zip(line.get_xdata(), line.get_ydata()):
            label = '{:.1%}'.format(y_value/100)

```

```

        ax.annotate(label,(x_value, y_value), xytext=(19, -14),
                    textcoords="offset points", ha='right',
                    va=va,color = 'white').set_backgroundcolor('green') ##965786

if typ == 'line_color_#965786':
    line = ax.lines[0]
    for x_value, y_value in zip(line.get_xdata(), line.get_ydata()):
        label = '{:.1%}'.format(y_value/100)
        ax.annotate(label,(x_value, y_value), xytext=(19, -14),
                    textcoords="offset points", ha='right',
                    va=va,color = 'white').set_backgroundcolor('#965786')

```

### 3 Grafico Producto Interno Bruto a Precios Constantes

```

[ ]: import seaborn as sns
sns.set_context('talk')
#sns.set_style("whitegrid")
sns.set(style="white", rc={"lines.linewidth": 3})

#sns.set(style="white", rc={"lines.linewidth": 3})

fig, ax1 = plt.subplots(figsize=(12,9))

ax2 = ax1.twinx()
sns.barplot(x=['2019', '2020', '2021'],
            y=[220728,200940,226126],
            color='#004488',
            ax=ax1,label = 'PIB')
sns.lineplot(x=['2019', '2020', '2021'],
             y=[2.7,-9.0,12.5],
             color='green',
             marker="o",
             ax=ax2,label = '% crec interanual')

ax1.set_title('Producto Interno Bruto a Precios Constantes\n (en millones de_
↳lempiras)', pad=15)

label_comma_bar_porcentaje_line(ax1, typ = 'bar')
label_comma_bar_porcentaje_line(ax2, typ='line')
fig.tight_layout()

ax1.yaxis.set_major_formatter(FuncFormatter(lambda y, _: f'{math.ceil(y):,}'.
↳format(y)))
ax2.yaxis.set_major_formatter(FuncFormatter(lambda y, _: '{:.0%}'.format(y/100)))

```

```

#ax2.legend(bbox_to_anchor=(1.01, 1.07))
#ax1.legend(bbox_to_anchor=(0.11, 1.07))

ax2.legend(bbox_to_anchor=(0.7, -0.1))
ax1.legend(bbox_to_anchor=(0.43, -0.1))

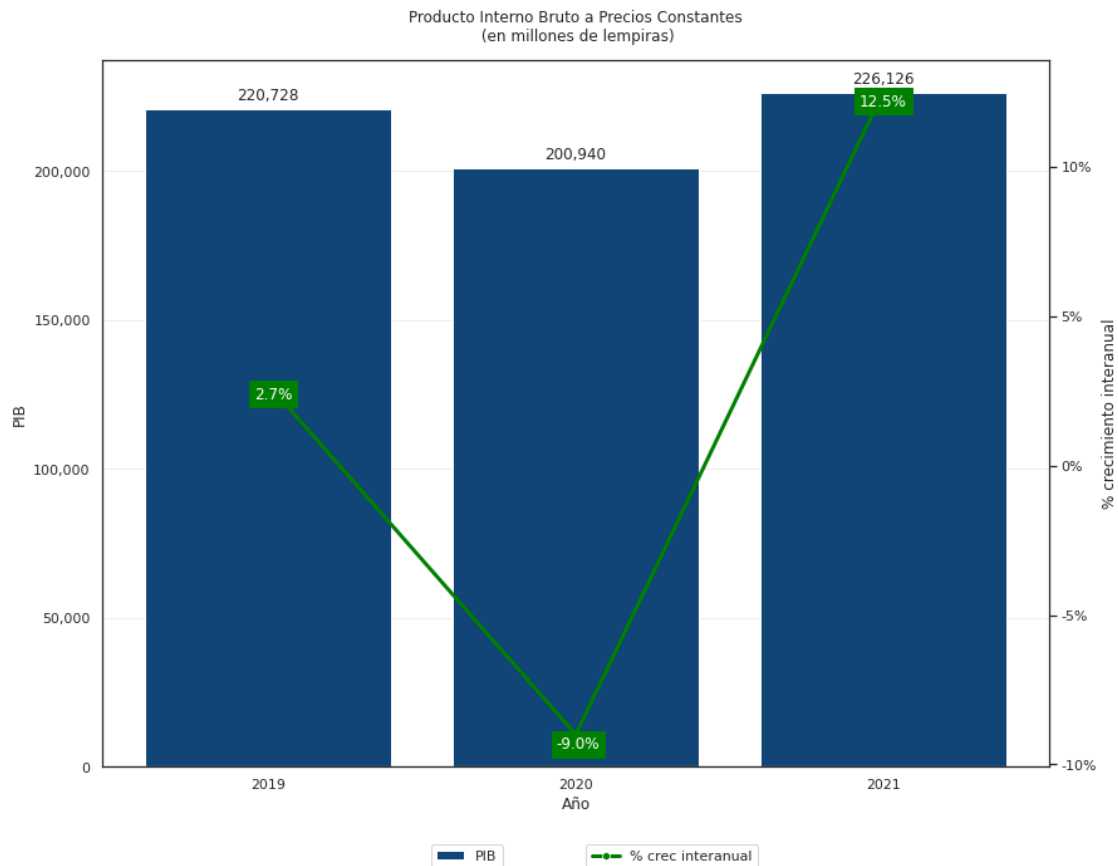
ax1.spines['top'].set_visible(False)
ax1.spines['right'].set_visible(False)
ax1.spines['left'].set_visible(False)
ax1.spines['bottom'].set_color('#DDDDDD')
ax1.tick_params(bottom=False, left=False)
ax1.set_axisbelow(True)
ax1.yaxis.grid(True, color='#EEEEEE')
ax1.xaxis.grid(False)

ax1.set_xlabel("Año")
ax1.set_ylabel("PIB")
ax2.set_ylabel("% crecimiento interanual")

plt.savefig('PIB_Final_2.eps',format='eps',pdi=1000,bbox_inches="tight")

plt.show()
sns.set()

```



## 4 Grafico Tasas Activas y Pasivas

```
[ ]: from google.colab import files
```

```
uploaded = files.upload()
```

Saving tasas\_activas\_pasivas\_2021.csv to tasas\_activas\_pasivas\_2021.csv

```
[ ]: df = pd.read_csv(io.BytesIO(uploaded['tasas_activas_pasivas_2021.csv']))
print(df)
```

	mes	activa	pasiva
0	ene	13.47	5.94
1	feb	13.70	5.86
2	mar	12.84	5.76
3	apr	13.35	4.66
4	may	13.12	4.28
5	jun	13.06	4.24

6	jul	13.17	4.34
7	aug	12.89	4.01
8	sep	12.72	4.04
9	oct	12.74	4.07
10	nov	12.73	3.67
11	dec	12.58	4.19

```
[ ]: from __future__ import barry_as_FLUFL
# 1. Estilo del Grafico
sns.set_context("talk")
sns.set_style("whitegrid")
sns.set(style="white", rc={"lines.linewidth": 3})
# 2. Instanciamos el grafico
fig, ax = plt.subplots(figsize = (12,7))
x = np.arange(len(df.mes.unique()))
bar_width = 0.47

b1 = ax.bar(x, df["activa"], bar_width, label = "Activa")
b2 = ax.bar(x+bar_width, df["pasiva"], bar_width, label = "Pasiva")

# 3. Etiqueta en el eje x
ax.set_xticks(x + bar_width / 2)
ax.set_xticklabels(df.mes.unique())

# 4. Diseño del Plano xy
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
ax.spines['left'].set_visible(False)
ax.spines['bottom'].set_color('#DDDDDD')
ax.tick_params(bottom=False, left=False)
ax.set_axisbelow(True)
ax.yaxis.grid(True, color='#EEEEEE')
ax.xaxis.grid(False)

# 5. Titulo y nombres de ejes
ax.set_xlabel('Mes', labelpad=15)
ax.set_ylabel('', labelpad=15)
ax.set_title('Tasas de Interés Promedio Anual en MN sobre Operaciones Nuevas\n'
'(en porcentajes)', pad=15)

# 6. Convertimos la escala del eje y en %
ax.yaxis.set_major_formatter(FuncFormatter(lambda y, _: '{:.0%}'.format(y/100)))

# 7. Etiquetas de las barras
for bar in ax.patches:

    bar_value = bar.get_height()
```

```

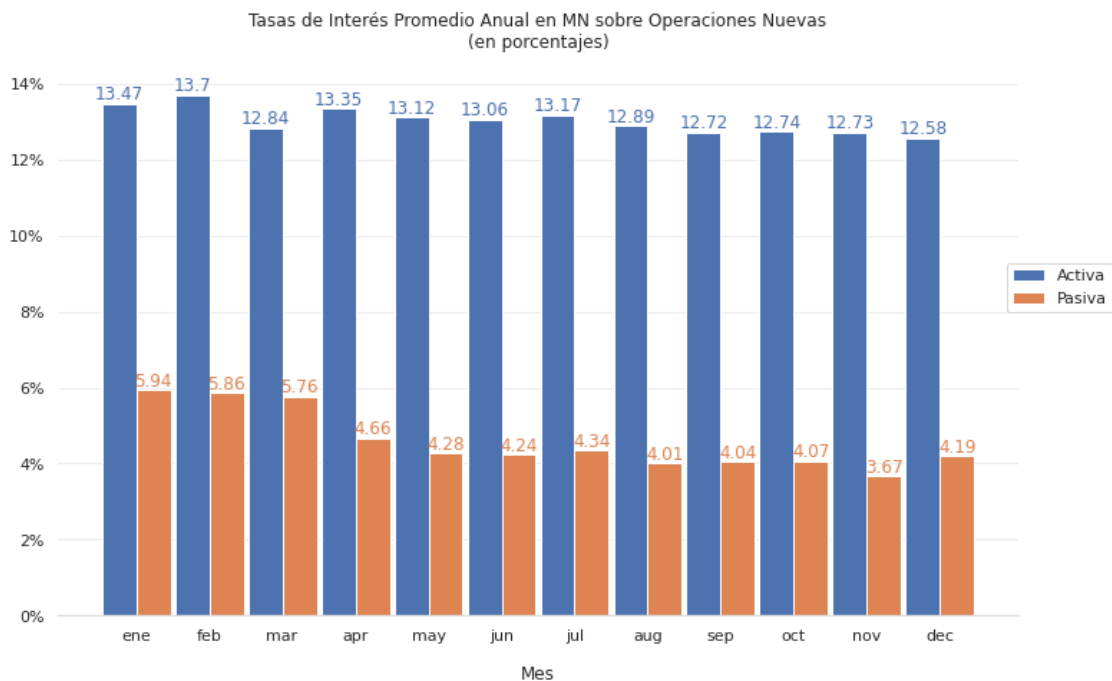
text = f'{bar_value:,}'
text_x = bar.get_x() + bar.get_width() / 2
text_y = bar.get_y() + bar_value
bar_color = bar.get_facecolor()
ax.text(text_x, text_y, text, ha='center', va='bottom', color=bar_color,
        size=12)

# 8. Leyenda
ax.legend(bbox_to_anchor=(0.98, 0.66))

# 9. Salvamos la el grafioo en calida .eps
plt.savefig('Tasa_Ac_Pa.eps', format = 'eps', pdi = 1000, bbox_inches = "tight")

plt.show()
sns.set()

```



## 5 Plot - Deficit Administracion Central

```
[ ]: from google.colab import files
```

```
uploaded = files.upload()
```

Saving deficit\_AC.csv to deficit\_AC.csv

```
[ ]: df = pd.read_csv(io.BytesIO(uploaded['deficit_AC.csv']))
print(df)
```

```
   Date  dac
0  2017  2.7
1  2018  2.1
2  2019  2.5
3  2020  7.0
4  2021  5.0
```

```
[ ]: import seaborn as sns
sns.set_context("talk")
sns.set_style("whitegrid")
sns.set(style="white", rc={"lines.linewidth": 3})

fig, ax1 = plt.subplots(figsize=(12,9))

b1 = ax1.bar(df['Date'], df["dac"], 0.4, label = "Activa",color = '#004488')

label_comma_bar_porcentaje_line(ax1, typ = 'bar_porcentaje')
fig.tight_layout()

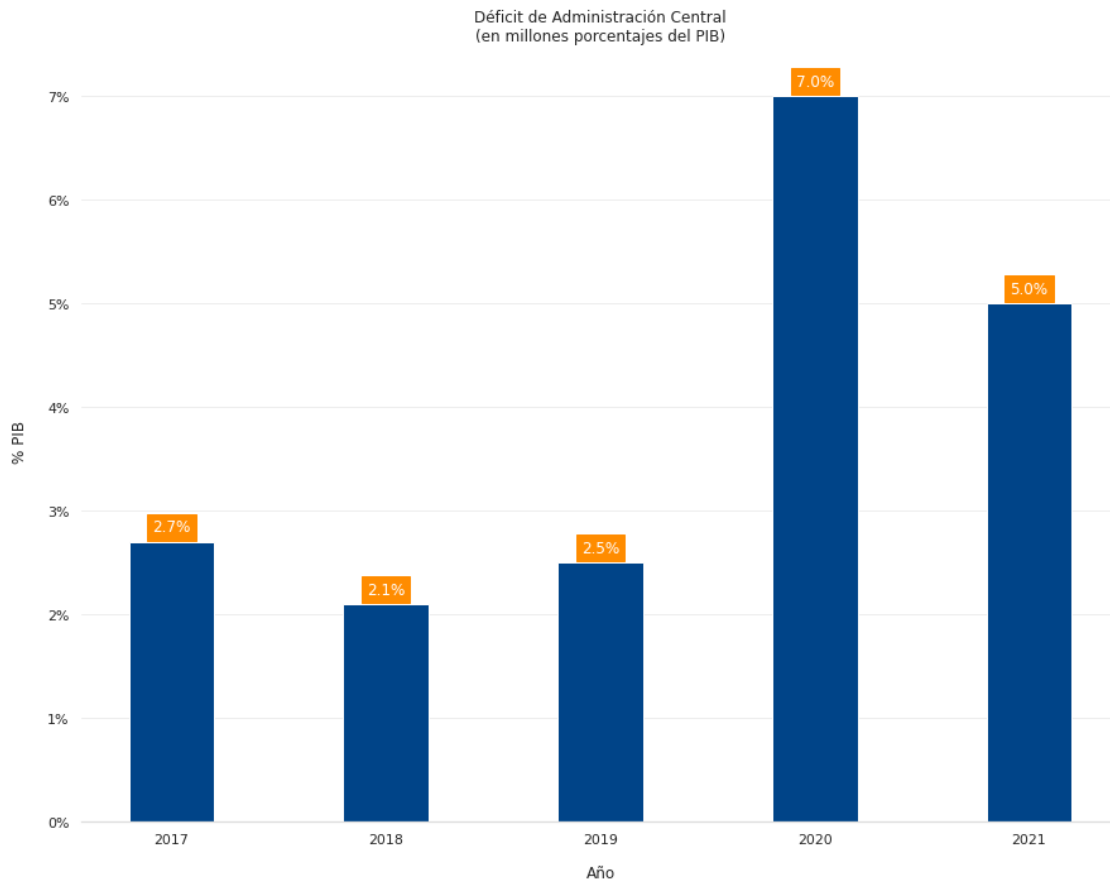
ax1.yaxis.set_major_formatter(FuncFormatter(lambda y, _: '{:.0%}'.format(y/100)))

ax1.spines['top'].set_visible(False)
ax1.spines['right'].set_visible(False)
ax1.spines['left'].set_visible(False)
ax1.spines['bottom'].set_color('#DDDDDD')
ax1.tick_params(bottom=False, left=False)
ax1.set_axisbelow(True)
ax1.yaxis.grid(True, color='#EEEEEE')
ax1.xaxis.grid(False)

ax1.set_xlabel('Año', labelpad=15)
ax1.set_ylabel('% PIB', labelpad=15)
ax1.set_title('Déficit de Administración Central\n'
'(en millones porcentajes del PIB)', pad=15)

plt.savefig('deficit_AC.eps', format = 'eps', pdi = 1000, bbox_inches = "tight")
plt.show()
sns.set()
```





## 6 Plot - Crecimiento Deduda Externa

```
[ ]: from google.colab import files
```

```
uploaded = files.upload()
```

Saving `crec_deuda_externa_hn.csv` to `crec_deuda_externa_hn (2).csv`

```
[ ]: df = pd.read_csv(io.BytesIO(uploaded['crec_deuda_externa_hn.csv']))
print(df)
```

	date	deuda_externa
0	2007	2026.0
1	2008	2358.0
2	2009	2458.0
3	2010	2847.0
4	2011	3218.0
5	2012	3664.0

6	2013	5202.0
7	2014	5566.0
8	2015	5927.0
9	2016	6108.0
10	2017	7022.0
11	2018	7259.0
12	2019	7609.0
13	2020	8507.0
14	2021	8558.0

```
[ ]: import seaborn as sns
sns.set_context("talk")
sns.set_style("whitegrid")
sns.set(style="white", rc={"lines.linewidth": 3})

fig, ax1 = plt.subplots(figsize=(12,9))

x = np.arange(len(df.date.unique()))

b1 = ax1.bar(x, df["deuda_externa"], 0.6, label = "Activa",color = '#004488')

label_comma_bar_porcentaje_line(ax1, typ = 'bar_comma_orange')
fig.tight_layout()

ax1.yaxis.set_major_formatter(FuncFormatter(lambda y, _: f'{math.ceil(y):,}'))

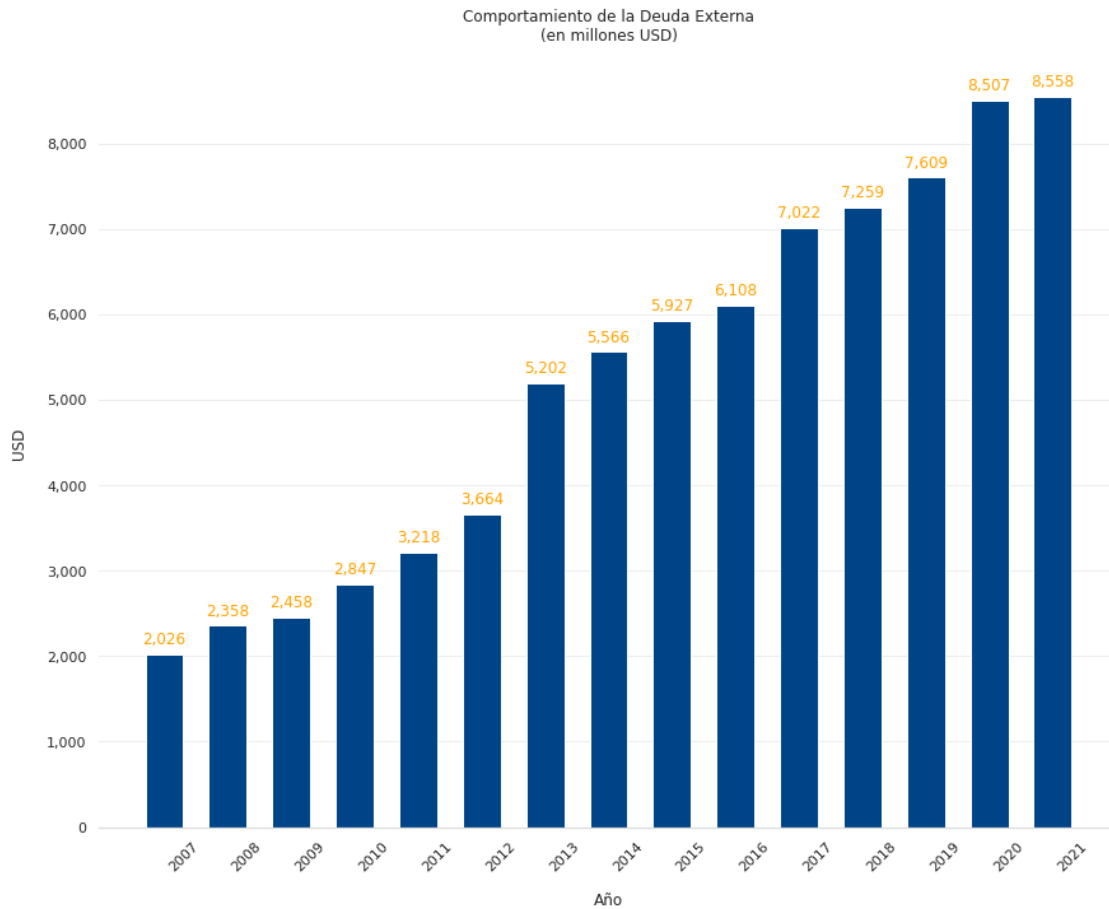
ax1.set_xticks(x + 0.6 / 2)
ax1.set_xticklabels(df.date.unique(),rotation=45)

ax1.spines['top'].set_visible(False)
ax1.spines['right'].set_visible(False)
ax1.spines['left'].set_visible(False)
ax1.spines['bottom'].set_color('#DDDDDD')
ax1.tick_params(bottom=False, left=False)
ax1.set_axisbelow(True)
ax1.yaxis.grid(True, color='#EEEEEE')
ax1.xaxis.grid(False)

ax1.set_xlabel('Año', labelpad=15)
ax1.set_ylabel('USD', labelpad=15)
ax1.set_title('Comportamiento de la Deuda Externa\n'
'(en millones USD)', pad=15)

plt.savefig('deuda_externa.eps', format = 'eps', pdi = 1000,bbox_inches =_
->"tight")
```

```
plt.show()
sns.set()
```



## 7 Plot - Comportamiento del GNI

```
[ ]: from google.colab import files
```

```
uploaded = files.upload()
```

<IPython.core.display.HTML object>

Saving GNI.csv to GNI.csv

```
[ ]: df = pd.read_csv(io.BytesIO(uploaded['GNI.csv']))
      print(df)
```

	date	gni
0	2001	0.56
1	2002	0.56
2	2003	0.58
3	2004	0.58
4	2005	0.60
5	2006	0.59
6	2007	0.59
7	2008	0.55
8	2009	0.52
9	2010	0.54
10	2011	0.55
11	2012	0.57
12	2013	0.54
13	2014	0.52
14	2015	0.51
15	2016	0.52
16	2017	0.52
17	2018	0.53
18	2019	0.52
19	2021	0.55

```
[ ]: sns.set_context("talk")
sns.set_style("whitegrid")
sns.set(style="white", rc={"lines.linewidth": 3})

fig, ax1 = plt.subplots(figsize=(12,9))

x = np.arange(len(df.date.unique()))

b1 = ax1.bar(x, df["gni"], 0.6, label = "Activa",color = '#004488')

label_comma_bar_porcentaje_line(ax1, typ = 'bar_porcentaje_orange')
fig.tight_layout()

ax1.yaxis.set_major_formatter(FuncFormatter(lambda y, _: '{:.1%}'.format(y/100)))

ax1.set_xticks(x + 0.6 / 2)
ax1.set_xticklabels(df.date.unique(),rotation=45)

ax1.spines['top'].set_visible(False)
ax1.spines['right'].set_visible(False)
ax1.spines['left'].set_visible(False)
ax1.spines['bottom'].set_color('#DDDDDD')
ax1.tick_params(bottom=False, left=False)
```

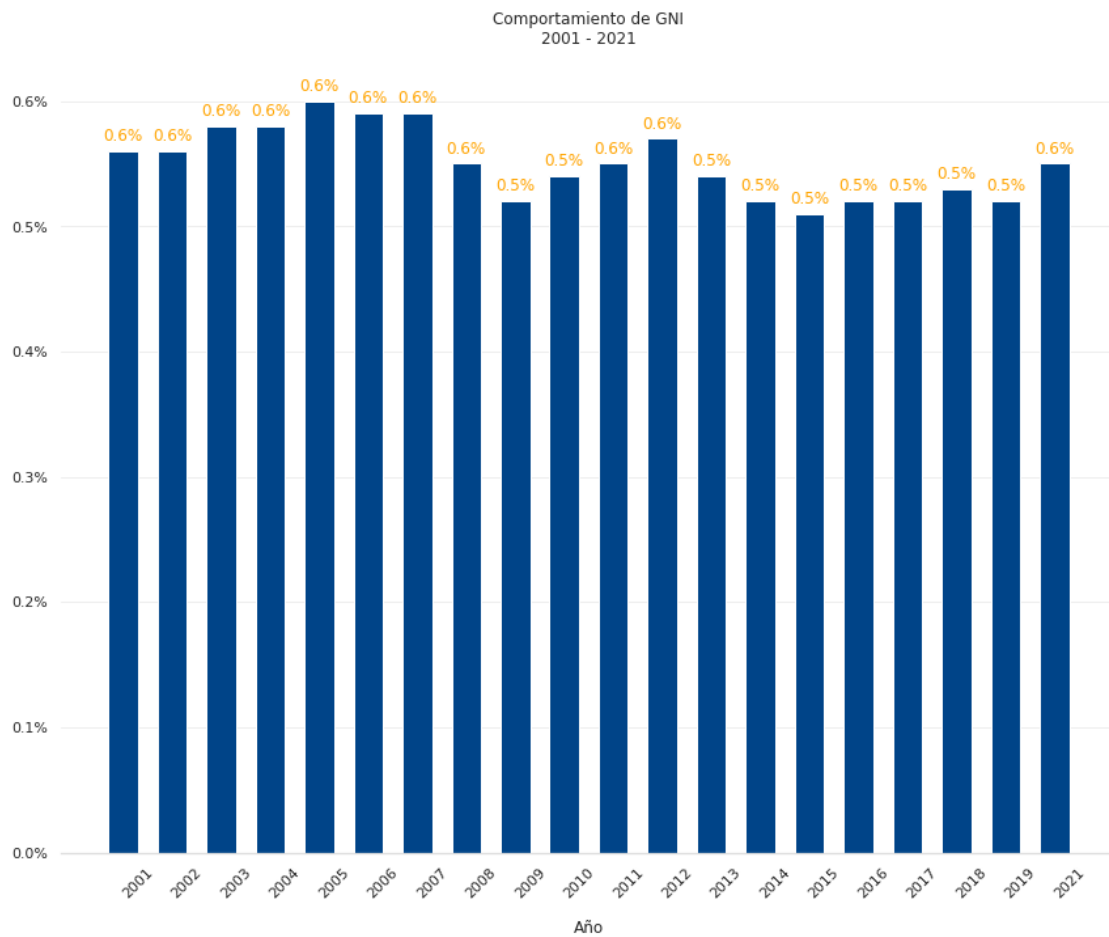
```

ax1.set_axisbelow(True)
ax1.yaxis.grid(True, color='#EEEEEE')
ax1.xaxis.grid(False)

ax1.set_xlabel('Año', labelpad=15)
ax1.set_ylabel('', labelpad=15)
ax1.set_title('Comportamiento de GNI\n'
'2001 - 2021', pad=15)

plt.savefig('GNI.eps', format = 'eps', pdi = 1000, bbox_inches = "tight")
plt.show()
sns.set()

```



## 8 Plot - Tasa de Ocupacion Informal

```
[398]: from google.colab import files
```

```
uploaded = files.upload()
```

Saving Tasa\_Informalidad\_2010\_2017.csv to Tasa\_Informalidad\_2010\_2017 (2).csv

```
[399]: df = pd.read_csv(io.BytesIO(uploaded['Tasa_Informalidad_2010_2017.csv']))
print(df)
```

	Date	Tasa
0	2010	80.4
1	2011	81.9
2	2012	83.7
3	2013	82.7
4	2014	79.0
5	2015	80.9
6	2016	79.8
7	2017	82.6

```
[400]: import seaborn as sns
import math
sns.set_context('talk')

sns.set(style="white", rc={"lines.linewidth": 3})

fig, ax1 = plt.subplots(figsize=(12,9))

sns.lineplot(x = df['Date'],
             y = df['Tasa'],
             color='#965786',
             marker="o",
             ax=ax1)

label_comma_bar_porcentaje_line(ax1, typ = 'line_color_#965786')
fig.tight_layout()

ax1.spines['top'].set_visible(False)
ax1.spines['right'].set_visible(False)
ax1.spines['left'].set_visible(False)
ax1.spines['bottom'].set_color('#DDDDDD')
ax1.tick_params(bottom=False, left=False)
ax1.set_axisbelow(True)
ax1.yaxis.grid(True, color='#EEEEEE')
```

```

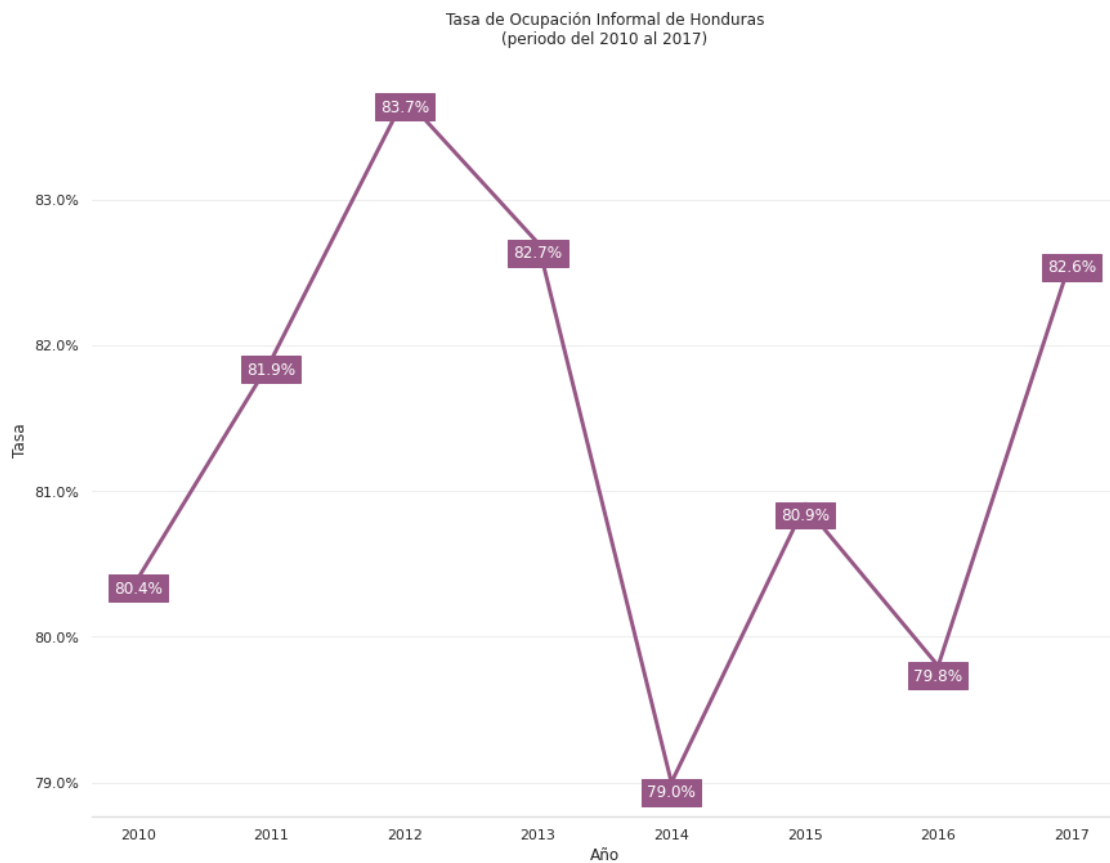
ax1.xaxis.grid(False)

ax1.set_xlabel('Año')
ax1.set_title('Tasa de Ocupación Informal de Honduras\n'
'(periodo del 2010 al 2017)', pad=15)

ax1.yaxis.set_major_formatter(FuncFormatter(lambda y, _: '{:.1%}'.format(y/100)))

plt.savefig('Tasa_Informal.eps', format = 'eps', pdi = 1000, bbox_inches = _
→"tight")
plt.show()

```



## 9 Plot - Tasa Informal Segun Sexo

```
[401]: from google.colab import files
```

```
uploaded = files.upload()
```

Saving Tasa\_Informa\_Sex.csv to Tasa\_Informa\_Sex (1).csv

```
[402]: df = pd.read_csv(io.BytesIO(uploaded['Tasa_Informa_Sex.csv']))  
print(df)
```

	Date	Femenino	Masculino
0	2010	76.9	82.3
1	2011	77.5	84.3
2	2012	80.0	85.7
3	2013	78.6	85.1
4	2014	75.4	81.2
5	2015	78.5	82.4
6	2016	77.6	81.3
7	2017	81.0	83.6

```
[403]: import seaborn as sns  
import math  
sns.set_context('talk')  
  
sns.set(style="white", rc={"lines.linewidth": 3})  
  
fig, ax = plt.subplots(figsize = (12,7))  
  
x = np.arange(len(df.Date.unique()))  
  
bar_width = 0.47  
  
b1 = ax.bar(x, df["Femenino"], bar_width, color='darkolivegreen',label =  
    →"Femenino")  
  
b2 = ax.bar(x+bar_width, df["Masculino"], bar_width, color =  
    →'darkgoldenrod',label = "Masculino")  
  
ax.set_xticks(x + bar_width / 2)  
ax.set_xticklabels(df.Date.unique())  
  
ax.yaxis.set_major_formatter(FuncFormatter(lambda y, _: '{:.0%}'.format(y/100)))  
  
ax.spines['top'].set_visible(False)  
ax.spines['right'].set_visible(False)  
ax.spines['left'].set_visible(False)  
ax.spines['bottom'].set_color('#DDDDDD')  
ax.tick_params(bottom=False, left=False)  
ax.set_axisbelow(True)  
ax.yaxis.grid(True, color='#EEEEEE')  
ax.xaxis.grid(False)
```



```

for bar in ax.patches:
    bar_value = bar.get_height()
    text = '{:.1%}'.format(bar_value/100)
    text_x = bar.get_x() + bar.get_width() / 2
    text_y = bar.get_y() + bar_value
    bar_color = bar.get_facecolor()
    ax.text(text_x, text_y, text, ha='center', va='bottom', color=bar_color,
            size=12)

ax.set_title('Tasa de Ocupación Informal Según Sexo\n'
            '(periodo del 2010 al 2017)', pad=15)

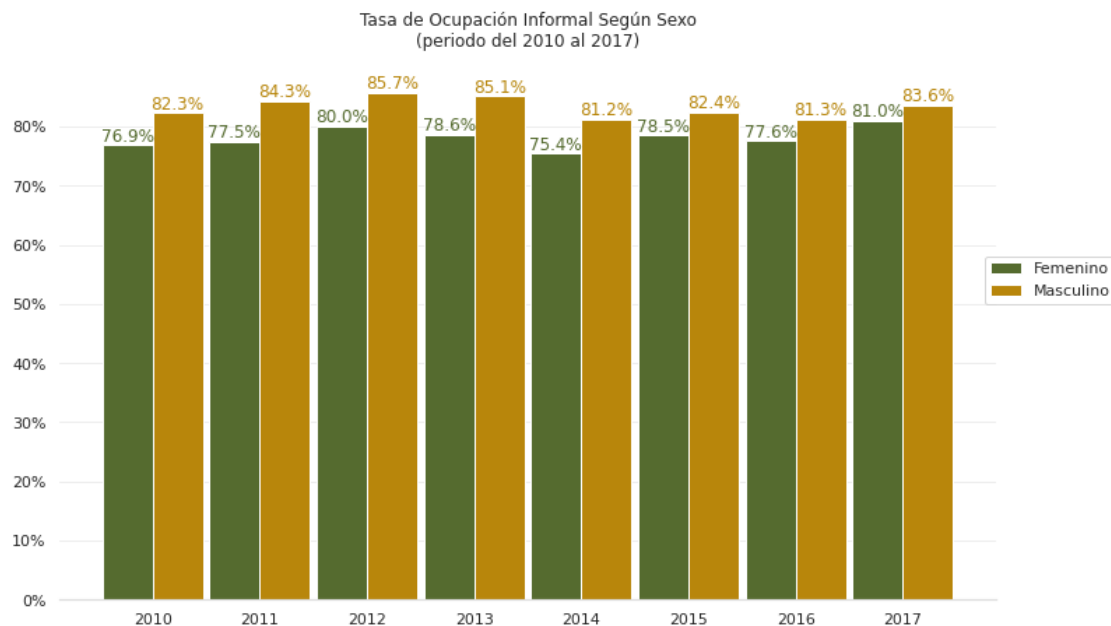
ax.legend(bbox_to_anchor=(0.98, 0.66))

plt.savefig('Tasa_Informal_Sexo.eps', format = 'eps', pdi = 1000, bbox_inches =
    →"tight")
plt.show()

```

WARNING:matplotlib.backends.backend\_ps:The PostScript backend does not support transparency; partially transparent artists will be rendered opaque.

WARNING:matplotlib.backends.backend\_ps:The PostScript backend does not support transparency; partially transparent artists will be rendered opaque.



## 10 Plot - Crecimiento Economico Mundial

```
[404]: from google.colab import files
```

```
uploaded = files.upload()
```

Saving `crec_eco_mundial_prueba.csv` to `crec_eco_mundial_prueba (6).csv`

```
[405]: df = pd.read_csv(io.BytesIO(uploaded['crec_eco_mundial_prueba.csv']))
print(df)
```

	Año	Economias	tasa
0	2020	Mundial	-0.031
1	2021	Mundial	0.059
2	2020	Avanzadas	-0.045
3	2021	Avanzadas	0.050
4	2020	Emergentes	-0.020
5	2021	Emergentes	0.065

```
[406]: # Funcion a necesitar
```

```
import math
def add_value_labels2(ax, typ, spacing=5):
    space = spacing
    va = 'bottom'

    if typ == 'bar':
        for i in ax.patches:
            y_value = i.get_height()
            x_value = i.get_x() + i.get_width() / 2

            label = '{:.1%}'.format(y_value)
            ax.annotate(label, (x_value, y_value), xytext=(-3, 3),
                        textcoords="offset points", ha='center', va=va,
                        color = 'olive')
```

```
[407]: sns.set_theme(style="whitegrid")

plt.figure(figsize = (10,10))

ax = sns.barplot(
    data = df,
    x = 'Economias', y = 'tasa', hue = 'Año', alpha=1,
    palette = 'dark'
)
```

```

ax.yaxis.set_major_formatter(FuncFormatter(lambda y, _: '{:.0%}'.format(y)))

ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
ax.spines['left'].set_visible(False)
ax.spines['bottom'].set_color('#DDDDDD')
ax.tick_params(bottom=False, left=False)
ax.set_axisbelow(True)
ax.yaxis.grid(True, color='#EEEEEE')
ax.xaxis.grid(False)

add_value_labels2(ax, typ = 'bar')
fig.tight_layout()

ax.set_xlabel('')
ax.set_ylabel('')
ax.set_title('Crecimiento Economico Mundial\n'
'(variación Porcentual)', pad=15)

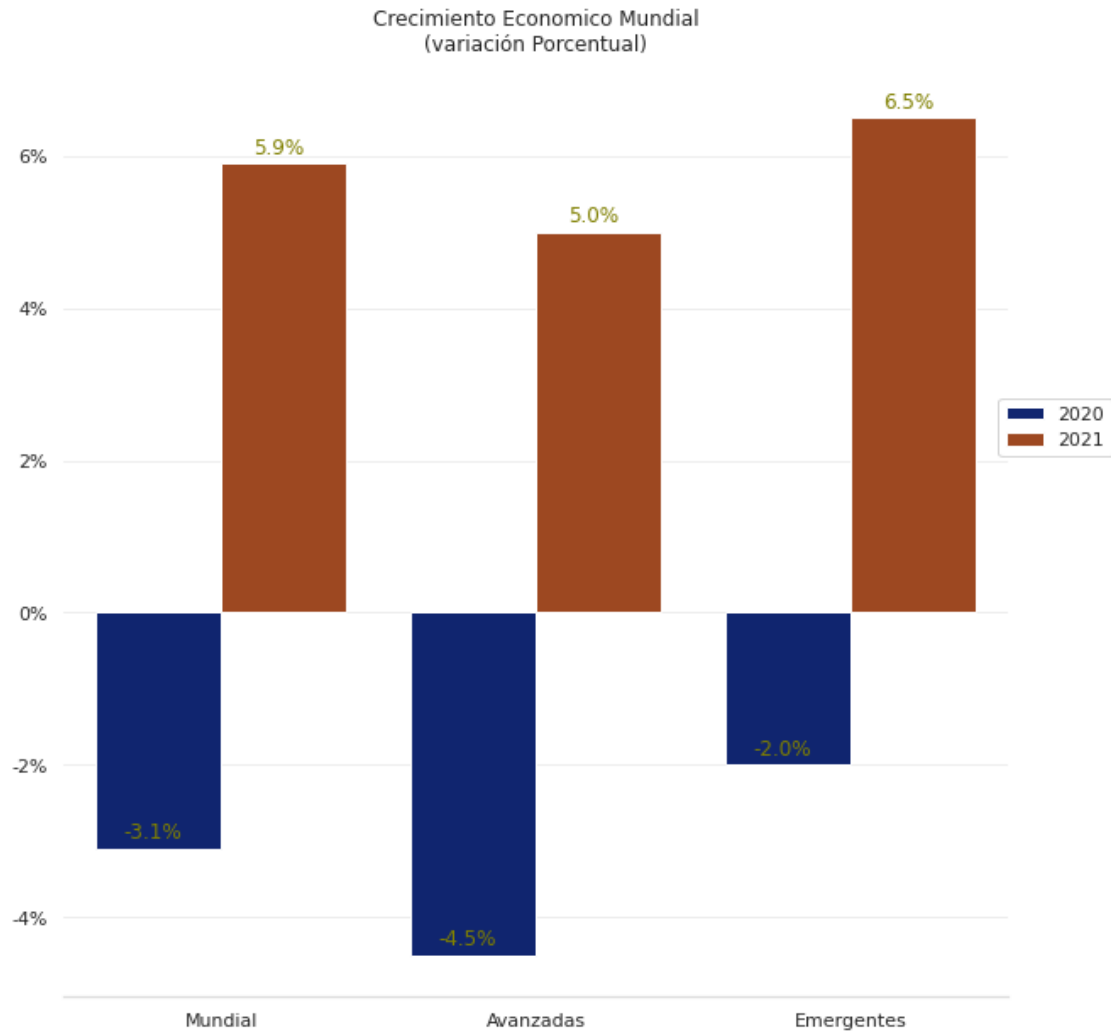
ax.legend(bbox_to_anchor=(0.98, 0.66))

plt.savefig('Crecim_Econo_Mund.eps', format = 'eps', pdi = 1000, bbox_inches =
    →"tight")
plt.show()
sns.set()

```

WARNING:matplotlib.backends.backend\_ps:The PostScript backend does not support transparency; partially transparent artists will be rendered opaque.

WARNING:matplotlib.backends.backend\_ps:The PostScript backend does not support transparency; partially transparent artists will be rendered opaque.



## 11 Plot - Inflacion Mundial

```
[409]: from google.colab import files
```

```
uploaded = files.upload()
```

Saving Inflacion\_mundial.csv to Inflacion\_mundial (4).csv

```
[410]: df = pd.read_csv(io.BytesIO(uploaded['Inflacion_mundial.csv']))
print(df)
```

	Date	Economias	tasa
0	2020	Economias Avanzadas	0.7
1	2021	Economias Avanzadas	3.1

2	2020	Economías Emergentes	5.1
3	2021	Economías Emergentes	5.7

[411]: *# Funcion a necesitar*

```
import math
def add_value_labels2(ax, typ, spacing=5):
    space = spacing
    va = 'bottom'

    if typ == 'bar':
        for i in ax.patches:
            y_value = i.get_height()
            x_value = i.get_x() + i.get_width() / 2

            label = '{:.1%}'.format(y_value/100)
            ax.annotate(label, (x_value, y_value), xytext=(-3, 3),
                        textcoords="offset points", ha='center', va=va,
                        color = 'olive')
```

[412]: `sns.set_theme(style="whitegrid")`

```
plt.figure(figsize = (10,8))

ax = sns.barplot(
    data = df,
    x = 'Economías', y = 'tasa', hue = 'Date', alpha=1,
    palette = 'dark'
)

ax.yaxis.set_major_formatter(FuncFormatter(lambda y, _: '{:.0%}'.format(y/100)))

ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
ax.spines['left'].set_visible(False)
ax.spines['bottom'].set_color('#DDDDDD')
ax.tick_params(bottom=False, left=False)
ax.set_axisbelow(True)
ax.yaxis.grid(True, color='#EEEEEE')
ax.xaxis.grid(False)

add_value_labels2(ax, typ = 'bar')
fig.tight_layout()

ax.set_xlabel('')
ax.set_ylabel('')
ax.set_title('Inflación Total\n')
```

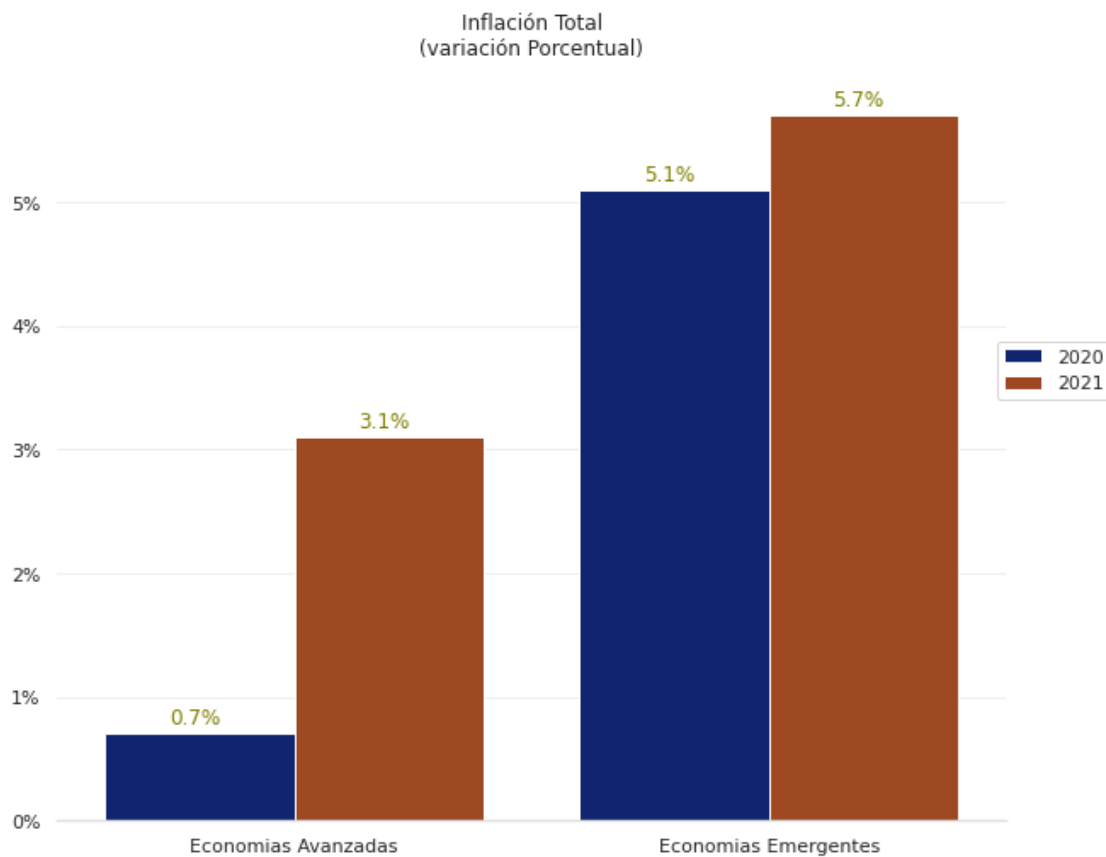
```

'(variación Porcentual)', pad=15)

ax.legend(bbox_to_anchor=(0.98, 0.66))

plt.savefig('Econo_Inflac.eps', format = 'eps', pdi = 1000, bbox_inches = "tight")
plt.show()
sns.set()

```



## 12 Plot - Proyeccion Inflacon - 2021 - 2023

```
[ ]: from google.colab import files
```

```
uploaded = files.upload()
```

Saving inflacion\_proyect\_21\_23.csv to inflacion\_proyect\_21\_23.csv

```
[ ]: df = pd.read_csv(io.BytesIO(uploaded['inflacion_proyect_21_23.csv']))
print(df)
```

	date	Economias	tasa
0	2021	Economias Avanzadas	3.1
1	2022	Economias Avanzadas	3.9
2	2023	Economias Avanzadas	2.1
3	2021	Economias Emergentes	5.7
4	2022	Economias Emergentes	5.9
5	2023	Economias Emergentes	4.7

```
[ ]: sns.set_theme(style="whitegrid")

plt.figure(figsize = (10,8))

ax = sns.barplot(
    data = df,
    x = 'date', y = 'tasa', hue = 'Economias', alpha=1,
    palette = 'dark'
)

ax.yaxis.set_major_formatter(FuncFormatter(lambda y, _: '{:.0%}'.format(y/100)))

ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
ax.spines['left'].set_visible(False)
ax.spines['bottom'].set_color('#DDDDDD')
ax.tick_params(bottom=False, left=False)
ax.set_axisbelow(True)
ax.yaxis.grid(True, color='#EEEEEE')
ax.xaxis.grid(False)

add_value_labels2(ax, typ = 'bar')
fig.tight_layout()

ax.set_xlabel('')
ax.set_ylabel('')
ax.set_title('Perspectiva: Inflación Economías Avanzadas y de Mercados_
→Emergentes\n'
'(variación Porcentual)', pad=15)

ax.legend(bbox_to_anchor=(1, 1))

plt.savefig('perspect_inf_econo.eps', format = 'eps', pdi = 1000, bbox_inches =_
→"tight")
plt.show()
sns.set()
```

Perspectiva: Inflación Economías Avanzadas y de Mercados Emergentes  
(variación Porcentual)

