

# S&P500 Analysis

February 25, 2024

## 1 ÁNALISIS BURSÁTIL POR MARCOS MEDINA

### 2 S&P 500

El S&P 500 es un índice bursátil que refleja el desempeño de las 500 mayores empresas públicas de Estados Unidos, seleccionadas por capitalización de mercado. Es uno de los indicadores más seguidos y utilizados para medir la salud y el rendimiento general del mercado de valores estadounidense.

#### 2.1 Analicemos los datos históricos del índice S&P 500 para ver si es un buen fondo indexado en el que invertir, qué estrategia de inversión puede ser más interesante y qué recomendaciones nos ofrecen los datos.

Un fondo indexado es un tipo de fondo cotizado en bolsa (ETF) que se gestiona de manera pasiva. Esto significa que su objetivo no es superar el rendimiento del mercado, sino seguir de cerca el rendimiento de un índice específico, como el S&P 500 o el IBEX 35.

Instalemos la biblioteca yfinance

```
[1]: pip install --upgrade yfinance
```

```
Requirement already satisfied: yfinance in ./anaconda3/lib/python3.11/site-  
packages (0.2.36)  
Requirement already satisfied: pandas>=1.3.0 in ./anaconda3/lib/python3.11/site-  
packages (from yfinance) (2.1.4)  
Requirement already satisfied: numpy>=1.16.5 in ./anaconda3/lib/python3.11/site-  
packages (from yfinance) (1.24.3)  
Requirement already satisfied: requests>=2.31 in  
./anaconda3/lib/python3.11/site-packages (from yfinance) (2.31.0)  
Requirement already satisfied: multitasking>=0.0.7 in  
./anaconda3/lib/python3.11/site-packages (from yfinance) (0.0.11)  
Requirement already satisfied: lxml>=4.9.1 in ./anaconda3/lib/python3.11/site-  
packages (from yfinance) (4.9.2)  
Requirement already satisfied: appdirs>=1.4.4 in  
./anaconda3/lib/python3.11/site-packages (from yfinance) (1.4.4)  
Requirement already satisfied: pytz>=2022.5 in ./anaconda3/lib/python3.11/site-  
packages (from yfinance) (2023.3.post1)  
Requirement already satisfied: frozendict>=2.3.4 in  
./anaconda3/lib/python3.11/site-packages (from yfinance) (2.4.0)
```

Requirement already satisfied: peewee>=3.16.2 in  
./anaconda3/lib/python3.11/site-packages (from yfinance) (3.17.1)  
Requirement already satisfied: beautifulsoup4>=4.11.1 in  
./anaconda3/lib/python3.11/site-packages (from yfinance) (4.12.2)  
Requirement already satisfied: html5lib>=1.1 in ./anaconda3/lib/python3.11/site-  
packages (from yfinance) (1.1)  
Requirement already satisfied: soupsieve>1.2 in ./anaconda3/lib/python3.11/site-  
packages (from beautifulsoup4>=4.11.1->yfinance) (2.4)  
Requirement already satisfied: six>=1.9 in ./anaconda3/lib/python3.11/site-  
packages (from html5lib>=1.1->yfinance) (1.16.0)  
Requirement already satisfied: webencodings in ./anaconda3/lib/python3.11/site-  
packages (from html5lib>=1.1->yfinance) (0.5.1)  
Requirement already satisfied: python-dateutil>=2.8.2 in  
./anaconda3/lib/python3.11/site-packages (from pandas>=1.3.0->yfinance) (2.8.2)  
Requirement already satisfied: tzdata>=2022.1 in  
./anaconda3/lib/python3.11/site-packages (from pandas>=1.3.0->yfinance) (2023.3)  
Requirement already satisfied: charset-normalizer<4,>=2 in  
./anaconda3/lib/python3.11/site-packages (from requests>=2.31->yfinance) (2.0.4)  
Requirement already satisfied: idna<4,>=2.5 in ./anaconda3/lib/python3.11/site-  
packages (from requests>=2.31->yfinance) (3.4)  
Requirement already satisfied: urllib3<3,>=1.21.1 in  
./anaconda3/lib/python3.11/site-packages (from requests>=2.31->yfinance)  
(1.26.16)  
Requirement already satisfied: certifi>=2017.4.17 in  
./anaconda3/lib/python3.11/site-packages (from requests>=2.31->yfinance)  
(2024.2.2)  
Note: you may need to restart the kernel to use updated packages.

### 3 1/01/200 - 2/02/2024

```
[2]: import yfinance as yf
import pandas as pd
import matplotlib.pyplot as plt

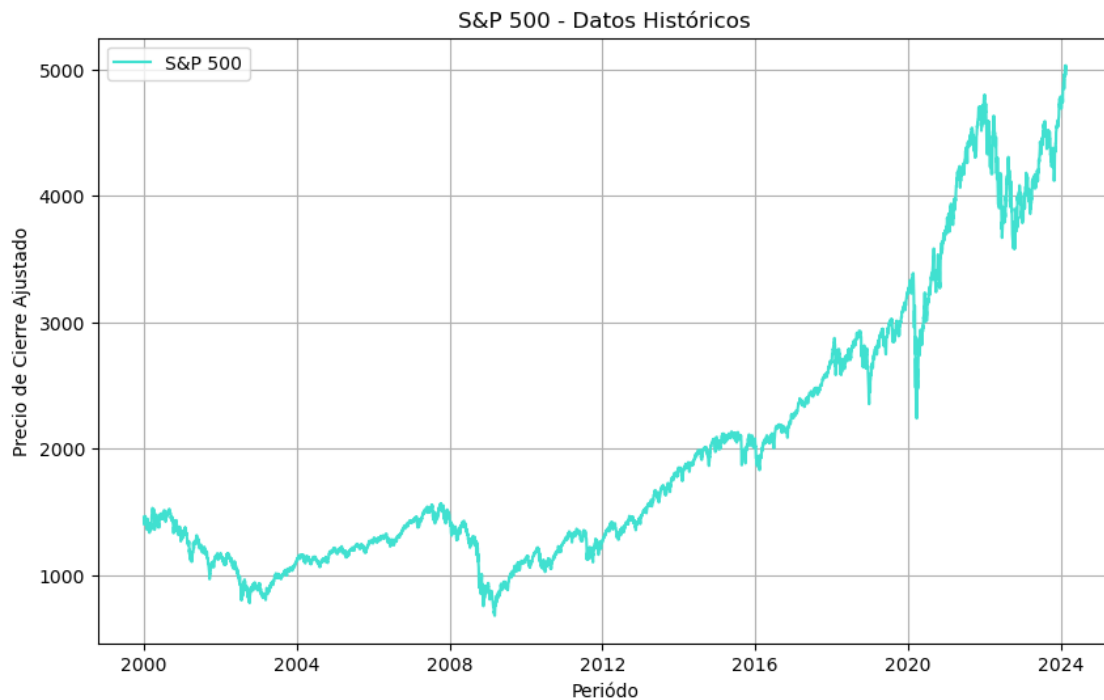
inicio = '2000-01-01'
fin = '2024-02-20'

# Datos del S&P 500 (GSPC)
data = yf.download('^GSPC', start=inicio, end=fin)

plt.figure(figsize=(10, 6))
plt.plot(data.index, data['Adj Close'], label='S&P 500', color='turquoise') #_
↳ Establecemos el índice
plt.title('S&P 500 - Datos Históricos')
plt.xlabel('Período')
plt.ylabel('Precio de Cierre Ajustado')
```

```
plt.legend()
plt.grid(True)
plt.show()
```

[\*\*\*\*\*100%\*\*\*\*\*] 1 of 1 completed



### 3.1 Rendimiento promedio anual del S&P 500

```
[3]: import numpy as np
data['Daily Return'] = data['Adj Close'].pct_change()

average_daily_return = data['Daily Return'].mean()
annualized_return = ((1 + average_daily_return)**252) - 1 # 252 días hábiles
↳ en un año

# Los mercados financieros suelen operar cinco días a la semana, de lunes a
↳ viernes
# Consideramos que hay aproximadamente 252 días hábiles en un año de
↳ operaciones típico

print("Rendimiento promedio anual del S&P 500: {:.2%}".
      ↳ format(annualized_return))
```

Rendimiento promedio anual del S&P 500: 7.31%

### 3.1.1 Calculemos el valor de una inversión mensual de 200 euros en el S&P 500 desde enero de 2000 hasta diciembre de 2024 y mostrar los resultados en una tabla.

```
[4]: cartera = pd.DataFrame(index=data.index)
cartera['Precio'] = data['Adj Close']
cartera['Inversion_Mensual'] = 200
cartera['Numero_Acciones'] = cartera['Inversion_Mensual'].cumsum() / \
    ↪cartera['Precio']
cartera['Valor_Cartera'] = cartera['Numero_Acciones'] * cartera['Precio']

valor_cartera_anual = cartera['Valor_Cartera'].resample('Y').last()

valor_cartera_anual = valor_cartera_anual.to_frame().reset_index()
valor_cartera_anual.columns = ['Año', 'Valor de la Cartera']

print(valor_cartera_anual)
```

	Año	Valor de la Cartera
0	2000-12-31	50400.0
1	2001-12-31	100000.0
2	2002-12-31	150400.0
3	2003-12-31	200800.0
4	2004-12-31	251200.0
5	2005-12-31	301600.0
6	2006-12-31	351800.0
7	2007-12-31	402000.0
8	2008-12-31	452600.0
9	2009-12-31	503000.0
10	2010-12-31	553400.0
11	2011-12-31	603800.0
12	2012-12-31	653800.0
13	2013-12-31	704200.0
14	2014-12-31	754600.0
15	2015-12-31	805000.0
16	2016-12-31	855400.0
17	2017-12-31	905600.0
18	2018-12-31	955800.0
19	2019-12-31	1006200.0
20	2020-12-31	1056800.0
21	2021-12-31	1107200.0
22	2022-12-31	1157400.0
23	2023-12-31	1207400.0
24	2024-12-31	1214000.0

**3.1.2** Para contrastar, ahora calculemos el ahorro acumulado de 200 euros al mes sin invertir durante el mismo período de tiempo, podemos usar un enfoque similar al anterior, pero sin multiplicar por el precio del S&P 500:

```
[5]: inicio = pd.Timestamp('2000-01-01')
    fin = '2024-02-20'

    fechas_mensuales = pd.date_range(start=inicio, end=fin, freq='MS')

    ahorro_acumulado = pd.DataFrame(index=fechas_mensuales, columns=['Ahorro_
    ↪Acumulado'])
    ahorro_acumulado['Ahorro Acumulado'] = 200 * (ahorro_acumulado.index - inicio).
    ↪days / 30

    ahorro_acumulado_anual = ahorro_acumulado.resample('Y').last()

    print(ahorro_acumulado_anual)
```

	Ahorro Acumulado
2000-12-31	2233.333333
2001-12-31	4666.666667
2002-12-31	7100.000000
2003-12-31	9533.333333
2004-12-31	11973.333333
2005-12-31	14406.666667
2006-12-31	16840.000000
2007-12-31	19273.333333
2008-12-31	21713.333333
2009-12-31	24146.666667
2010-12-31	26580.000000
2011-12-31	29013.333333
2012-12-31	31453.333333
2013-12-31	33886.666667
2014-12-31	36320.000000
2015-12-31	38753.333333
2016-12-31	41193.333333
2017-12-31	43626.666667
2018-12-31	46060.000000
2019-12-31	48493.333333
2020-12-31	50933.333333
2021-12-31	53366.666667
2022-12-31	55800.000000
2023-12-31	58233.333333
2024-12-31	58646.666667

```
[6]: import matplotlib.pyplot as plt

    años_cartera = valor_cartera_anual['Año'].dt.year
```

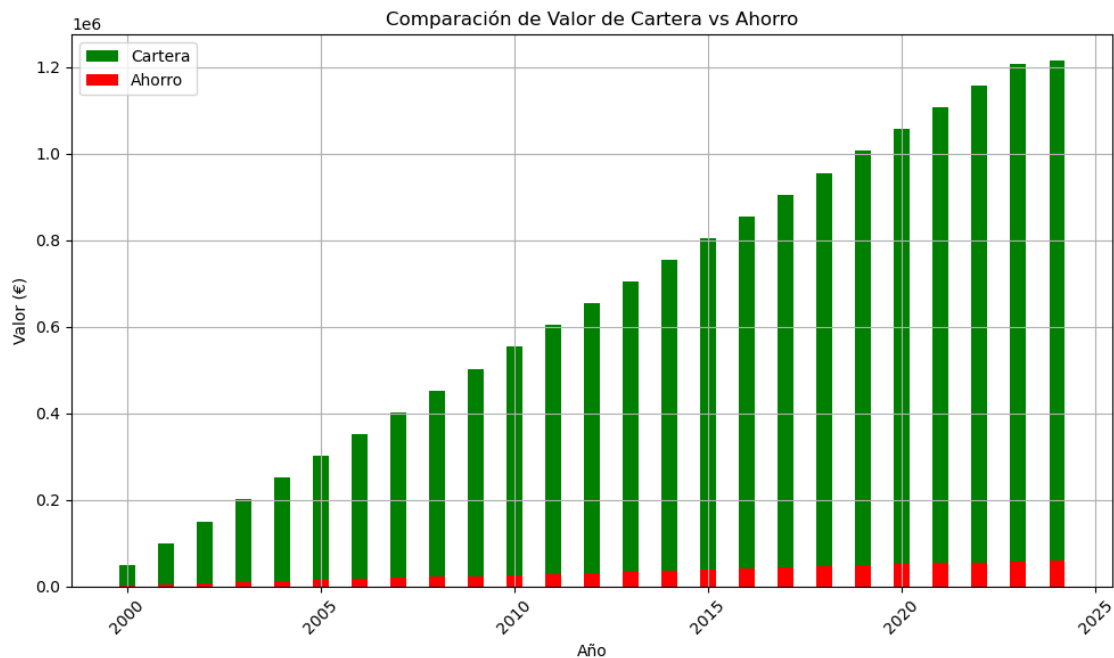
```

valor_cartera = valor_cartera_anual['Valor de la Cartera']

años_ahorro = ahorro_acumulado_anual.index.year
valor_ahorro = ahorro_acumulado_anual['Ahorro Acumulado']

plt.figure(figsize=(10, 6))
plt.bar(años_cartera, valor_cartera, width=0.4, label='Cartera', color='green')
plt.bar(años_ahorro, valor_ahorro, width=0.4, label='Ahorro', color='red')
plt.title('Comparación de Valor de Cartera vs Ahorro')
plt.xlabel('Año')
plt.ylabel('Valor (€)')
plt.legend()
plt.grid(True)
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

```



### 3.1.3 Conclusiones:

La cartera invertida en el S&P 500 muestra un crecimiento significativo a lo largo de los años, superando el ahorro acumulado sin inversión. Esto sugiere que, históricamente, invertir en el S&P 500 ha sido una estrategia rentable a largo plazo.

La diferencia entre el valor de la cartera y el ahorro acumulado se amplía con el tiempo. Esto demuestra el poder del interés compuesto y cómo puede acelerar el crecimiento del capital invertido.

Se observa que el crecimiento de la cartera y el ahorro acumulado son gradualmente crecientes a

lo largo de los años. Esto resalta la importancia de comenzar a invertir o ahorrar temprano para aprovechar al máximo el tiempo y permitir que el interés compuesto tenga un mayor impacto.

A pesar de que la cartera muestra un crecimiento general, se observan fluctuaciones en el valor de la cartera debido a la volatilidad del mercado. Esto es típico en las inversiones en acciones y resalta la importancia de tener una estrategia de inversión a largo plazo y una tolerancia al riesgo adecuada.

En resumen, los datos muestran la importancia de invertir y ahorrar de manera consistente a lo largo del tiempo para alcanzar objetivos financieros a largo plazo. Además, ilustran cómo la inversión en el mercado de valores, como el S&P 500, puede generar retornos significativos a largo plazo.

### 3.1.4 Datos Extra: Vamos a analizar el comportamiento del índice durante la crisis del COVID-19 en un gráfico:

```
[7]: inicio_pre_covid = '2019-01-01'

inicio_covid = '2020-02-20'
fin_covid = '2020-04-01'

inicio_post_covid = '2020-04-01'
fin_post_covid = '2021-04-01'

data_pre_covid = yf.download('^GSPC', start=inicio_pre_covid, end=inicio_covid)
data_covid = yf.download('^GSPC', start=inicio_covid, end=fin_covid)
data_post_covid = yf.download('^GSPC', start=inicio_post_covid,
    ↪end='2022-01-01')

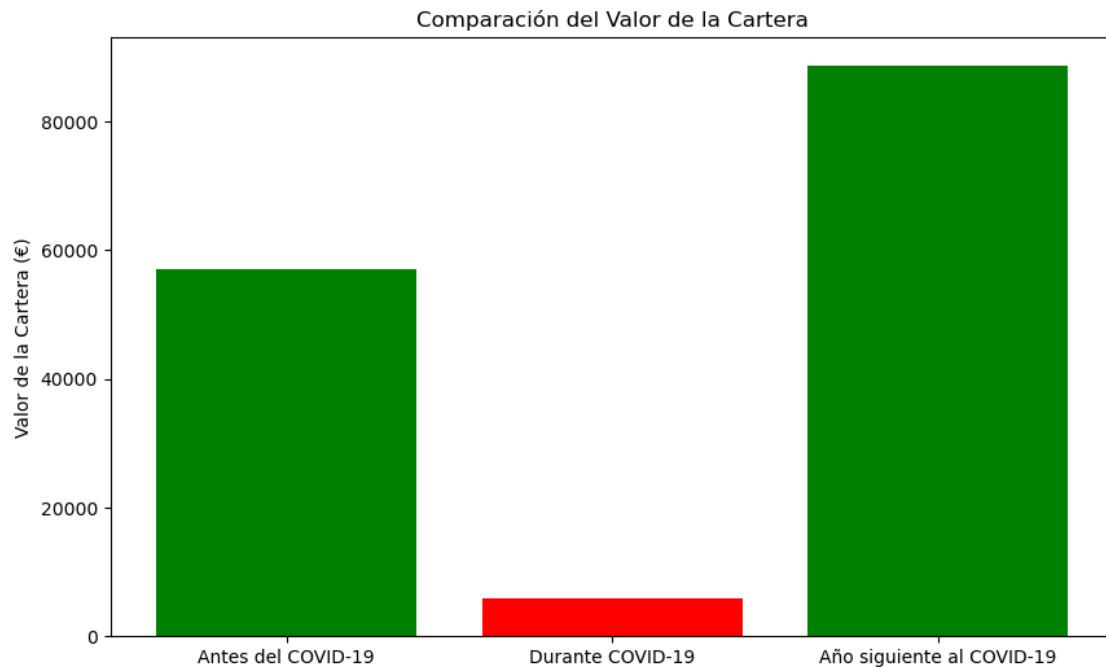
def calcular_valor_cartera(data):
    carter = pd.DataFrame(index=data.index)
    carter['Precio'] = data['Adj Close']
    carter['Inversion_Mensual'] = 200
    carter['Numero_Acciones'] = carter['Inversion_Mensual'].cumsum() /
    ↪carter['Precio']
    carter['Valor_Cartera'] = carter['Numero_Acciones'] * carter['Precio']
    return carter['Valor_Cartera']

valor_cartera_pre_covid = calcular_valor_cartera(data_pre_covid)
valor_cartera_covid = calcular_valor_cartera(data_covid)
valor_cartera_post_covid = calcular_valor_cartera(data_post_covid)

plt.figure(figsize=(10, 6))
plt.bar(['Antes del COVID-19', 'Durante COVID-19', 'Año siguiente al COVID-19'],
    [valor_cartera_pre_covid.iloc[-1], valor_cartera_covid.iloc[-1],
    ↪valor_cartera_post_covid.iloc[-1]],
    color=['green', 'red', 'green'])
plt.title('Comparación del Valor de la Cartera')
```

```
plt.ylabel('Valor de la Cartera (€)')
plt.show()
```

```
[*****100%*****] 1 of 1 completed
[*****100%*****] 1 of 1 completed
[*****100%*****] 1 of 1 completed
```



## 4 Conclusión:

Durante la crisis del COVID-19, el valor de la cartera experimentó una caída significativa debido a la volatilidad del mercado y la incertidumbre económica. Esto se refleja en el valor más bajo de la cartera durante este período.

El año siguiente a la crisis del COVID-19 muestra una recuperación gradual en el valor de la cartera. A medida que la economía se estabilizó y los mercados se recuperaron, el valor de la cartera comenzó a aumentar nuevamente.

Esto resalta la importancia de mantener una perspectiva a largo plazo y no dejar que las fluctuaciones a corto plazo influyan en las decisiones de inversión.

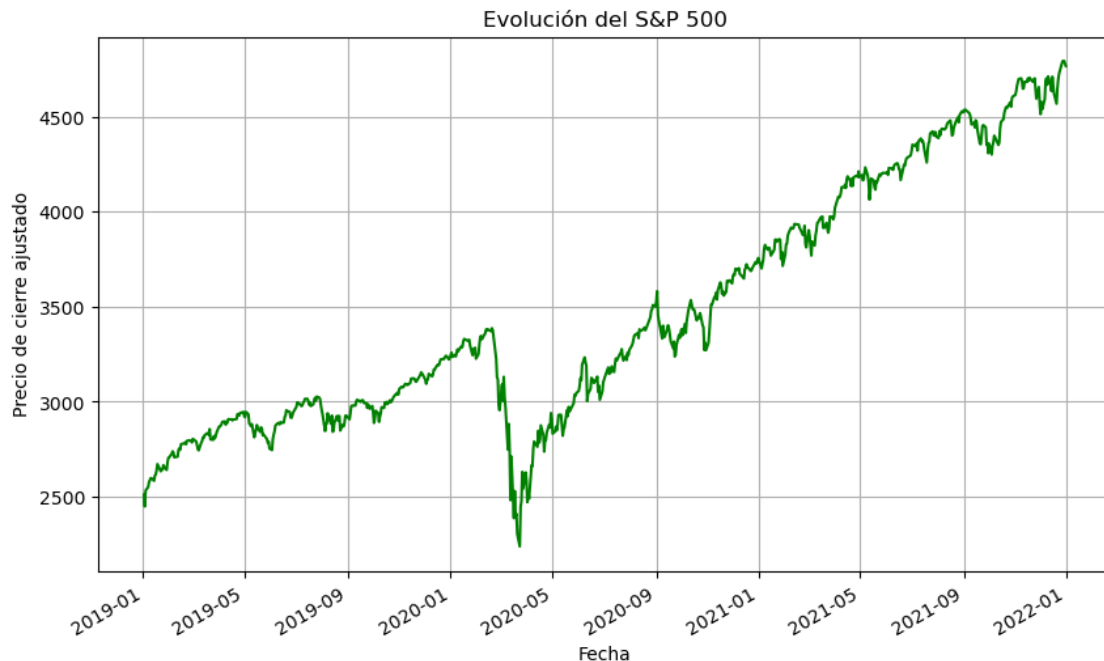
Los resultados también resaltan la importancia del momento de la inversión. Aquellos que pudieron invertir durante la crisis del COVID-19 y aprovechar los precios más bajos podrían haber experimentado un mayor crecimiento en el valor de su cartera en el año siguiente.

```
[8]: sp500 = yf.download('^GSPC', start='2019-01-01', end='2022-01-01')
```



```
plt.figure(figsize=(10, 6))
sp500['Adj Close'].plot(color='green')
plt.title('Evolución del S&P 500')
plt.xlabel('Fecha')
plt.ylabel('Precio de cierre ajustado')
plt.grid(True)
plt.show()
```

[\*\*\*\*\*100%\*\*\*\*\*] 1 of 1 completed



## 5 Si no invertimos y preferimos ahorrar, perdemos dinero por culpa de la inflación

El siguiente código calculará cómo se vería un ahorro de 10000 euros desde el año 2000 al año 2024, siguiendo la variación en la tasa de inflación en España. Después, el gráfico mostrará la tendencia general de cómo habría variado el valor real de tu ahorro debido a la inflación.

```
[9]: import pandas as pd

# Datos de inflación
data = [
    {"Año": 2000, "Tasa de Inflación (%)": 2.50},
    {"Año": 2001, "Tasa de Inflación (%)": 1.86},
    {"Año": 2002, "Tasa de Inflación (%)": 2.65},
    {"Año": 2003, "Tasa de Inflación (%)": 3.52},
```

```

{"Año": 2004, "Tasa de Inflación (%)": 1.77},
{"Año": 2005, "Tasa de Inflación (%)": 1.77},
{"Año": 2006, "Tasa de Inflación (%)": 3.58},
{"Año": 2007, "Tasa de Inflación (%)": 2.77},
{"Año": 2008, "Tasa de Inflación (%)": 1.53},
{"Año": 2009, "Tasa de Inflación (%)": 2.54},
{"Año": 2010, "Tasa de Inflación (%)": 1.54},
{"Año": 2011, "Tasa de Inflación (%)": 1.53},
{"Año": 2012, "Tasa de Inflación (%)": 2.24},
{"Año": 2013, "Tasa de Inflación (%)": 0.09},
{"Año": 2014, "Tasa de Inflación (%)": 0.28},
{"Año": 2015, "Tasa de Inflación (%)": 1.44},
{"Año": 2016, "Tasa de Inflación (%)": 0.99},
{"Año": 2017, "Tasa de Inflación (%)": 2.31},
{"Año": 2018, "Tasa de Inflación (%)": 1.09},
{"Año": 2019, "Tasa de Inflación (%)": 0.59},
{"Año": 2020, "Tasa de Inflación (%)": 3.47},
{"Año": 2021, "Tasa de Inflación (%)": 1.77},
{"Año": 2022, "Tasa de Inflación (%)": 2.07},
{"Año": 2023, "Tasa de Inflación (%)": 0.58},
{"Año": 2024, "Tasa de Inflación (%)": 1.46}
]

df = pd.DataFrame(data)

valor_inicial = 10000
valores_ajustados = [valor_inicial]

for i in range(1, len(df)):
    valor_ajustado = valores_ajustados[-1] * (1 - df.loc[i, "Tasa de Inflación_↵(%)"] / 100)
    valores_ajustados.append(valor_ajustado)

df["Valor Ajustado del Ahorro (€)"] = valores_ajustados

print(df)

```

	Año	Tasa de Inflación (%)	Valor Ajustado del Ahorro (€)
0	2000	2.50	10000.000000
1	2001	1.86	9814.000000
2	2002	2.65	9553.929000
3	2003	3.52	9217.630699
4	2004	1.77	9054.478636
5	2005	1.77	8894.214364
6	2006	3.58	8575.801490
7	2007	2.77	8338.251788
8	2008	1.53	8210.676536
9	2009	2.54	8002.125352

10	2010	1.54	7878.892622
11	2011	1.53	7758.345565
12	2012	2.24	7584.558624
13	2013	0.09	7577.732521
14	2014	0.28	7556.514870
15	2015	1.44	7447.701056
16	2016	0.99	7373.968816
17	2017	2.31	7203.630136
18	2018	1.09	7125.110567
19	2019	0.59	7083.072415
20	2020	3.47	6837.289802
21	2021	1.77	6716.269773
22	2022	2.07	6577.242988
23	2023	0.58	6539.094979
24	2024	1.46	6443.624192

```
[10]: plt.figure(figsize=(10, 6))
plt.plot(df["Año"], df["Valor Ajustado del Ahorro (€)"], marker='o',
        linestyle='-', color='red')
plt.title("Evolución del Valor Ajustado del Ahorro (2000-2024)")
plt.xlabel("Año")
plt.ylabel("Valor Ajustado del Ahorro (€)")
plt.grid(True)
plt.xticks(df["Año"], rotation=45)
plt.tight_layout()
plt.show()
```

