📊 Parte 1 – Selección y descarga de datos

En esta sección se seleccionarán los activos financieros que formarán parte del análisis y se descargarán sus precios históricos diarios utilizando `yfinance`.

La elección de los activos busca lograr una **diversificación adecuada** para un perfil de riesgo **moderado**, incluyendo distintas clases de instrumentos (acciones, bonos, índices, etc.).

Se trabajará con datos comprendidos entre **enero 2022 y junio 2025**.

📦 Parte 1.1: Selección de Activos para la Cartera

Se seleccionan 12 instrumentos financieros de distintos sectores, países y características para construir una cartera diversificada, coherente con un perfil **moderado** de riesgo.

Ticker	Descripción	Tipo / Sector
SPY	ETF que replica el S&P 500	Índice amplio (USA)
BRK-B	Berkshire Hathaway (Clase B)	Conglomerado financiero
XLP	Consumer Staples Select Sector SPDR	Sector defensivo (consumo)
AAPL	Apple Inc.	Tecnología (crecimiento)
GOOG	Alphabet Inc. (Google)	Tecnología (crecimiento)
KO	Coca-Cola Co	Consumo defensivo
MELI	Mercado Libre	E-commerce LATAM
NVDA	NVIDIA Corp	Tecnología / AI
MA	Mastercard Inc	Servicios financieros

Ticker	Descripción	Tipo / Sector
YPF	YPF S.A.	Energía / Argentina
BTC-USD	Bitcoin	Criptoactivo / alternativo
PFE	Pfizer Inc.	Salud / farmacéutico

Justificación de los activos seleccionados

A continuación se presentan breves fundamentos para la inclusión de cada instrumento financiero en la cartera, considerando su aporte a la diversificación, su perfil de riesgo y su rol estratégico:

- **SPY (ETF del S&P 500):**
  Representa a las 500 compañías más grandes de EE.UU. Permite exposición amplia al mercado americano. Si bien está concentrado en grandes tecnológicas, continúa siendo el principal benchmark global. Aporta solidez, liquidez y referencia.
- **BRK-B (Berkshire Hathaway Clase B):**
  Holding diversificado liderado por Warren Buffett. No paga dividendos, reinvierte ganancias. Combina negocios tradicionales con posiciones en grandes tecnológicas. Actúa como activo de valor con gestión activa.
- **XLP (Consumer Staples ETF):**
  ETF del sector de consumo básico (empresas como P&G, Coca-Cola, Walmart). Se comporta defensivamente en momentos de volatilidad. Aporta estabilidad y bajo Beta.
- **AAPL (Apple Inc.):**
  Empresa tecnológica consolidada, líder en innovación y con sólida posición financiera. Participa de megatendencias y mantiene márgenes altos. Combina crecimiento y reputación.
- **GOOG (Alphabet – Google):**
  Dominante en publicidad digital, con inversiones en inteligencia artificial y servicios cloud. Exposición a tecnología de largo plazo con diversificación en negocios no tradicionales.
- **KO (Coca-Cola Co):**
  Empresa defensiva, con fuerte presencia global y marca consolidada. Estable y resiliente ante ciclos económicos. Ideal para moderar la volatilidad total del portafolio.
- **MELI (Mercado Libre):**
  Principal plataforma de e-commerce y fintech en Latinoamérica. Alta volatilidad pero gran potencial de crecimiento. Aporta exposición regional y a economías emergentes.

- **NVDA (NVIDIA Corp):**
  Líder mundial en chips gráficos, inteligencia artificial y data centers. Alto crecimiento y sensibilidad a expectativas del mercado. Expone al inversor a la revolución tecnológica actual.
- **MA (Mastercard Inc.):**
  Activo del sector financiero con modelo de negocio sólido y baja exposición crediticia. Se beneficia de la digitalización de pagos. Balance entre crecimiento e ingresos estables.
- **YPF (YPF S.A.):**
  Empresa energética argentina. Alta volatilidad y exposición al riesgo país, pero con proyección estratégica por el desarrollo de Vaca Muerta. Aporta diversificación geográfica y sectorial.
- **BTC-USD (Bitcoin):**
  Criptoactivo con alta volatilidad. No correlacionado con activos tradicionales. Se incluye en pequeña proporción como cobertura y para diversificación estructural.
- **PFE (Pfizer Inc.):**
  Farmacéutica global con fuerte presencia en vacunas y medicamentos. Ofrece flujo de ingresos defensivo. Su perfil conservador contrabalancea activos más riesgosos.

📥 Parte 1.2: Descarga de datos históricos

Se descargan los precios ajustados de cierre para los 12 activos seleccionados desde **Yahoo Finance**, cubriendo el período comprendido entre **enero de 2022** y **junio de 2025**.

La fuente de datos será el módulo `yfinance`, el cual permite acceder directamente a datos de mercado desde Python.

```
# Paso 1: Instalar y actualizar yfinance si es necesario
!pip install yfinance --upgrade --quiet

# Paso 2: Importar librerías necesarias
import yfinance as yf
import pandas as pd

# Paso 3: Definir los 12 tickers seleccionados
tickers = ['SPY', 'BRK-B', 'XLP', 'AAPL', 'GOOG', 'KO', 'MELI', 'NVDA

# Paso 4: Descargar los precios ajustados diarios entre enero 2022 y
data = yf.download(tickers, start='2022-01-01', end='2025-06-30')

# Paso 5: Seleccionar la columna de precios de cierre
```

```
data = data["Close"]

# Paso 6: Eliminar fechas con valores faltantes en algún activo
data = data.dropna()

# Paso 7: Mostrar las primeras filas del DataFrame resultante
data.head()
```

```
/tmp/ipython-input-1181831570.py:12: FutureWarning: YF.download() has
  data = yf.download(tickers, start='2022-01-01', end='2025-06-30')
[*********************100%***********************]  12 of 12 completed
```

Ticker	AAPL	BRK-B	BTC-USD	GOOG	KO	MA
Date						
2022-01-03	178.270309	300.790009	46458.117188	144.088470	52.996700	362.362640
2022-01-04	176.007767	308.529999	45897.574219	143.434952	53.881470	366.699921
2022-01-05	171.326019	309.920013	43569.003906	136.717896	54.328327	365.244385

🔍 Parte 2: Análisis exploratorio de los activos

Se realiza un análisis exploratorio de los 12 activos seleccionados, con el fin de comprender su comportamiento histórico en términos de precios, retornos y volatilidad.

Este paso permite evaluar las características de riesgo y rendimiento de cada instrumento, y es fundamental para fundamentar decisiones futuras en la construcción del portafolio.

Se incluyen:

- Evolución de precios históricos
- Cálculo de retornos diarios y acumulados
- Medidas de dispersión (volatilidad)
- Visualizaciones con gráficos

📈 Parte 2.1: Evolución de precios históricos

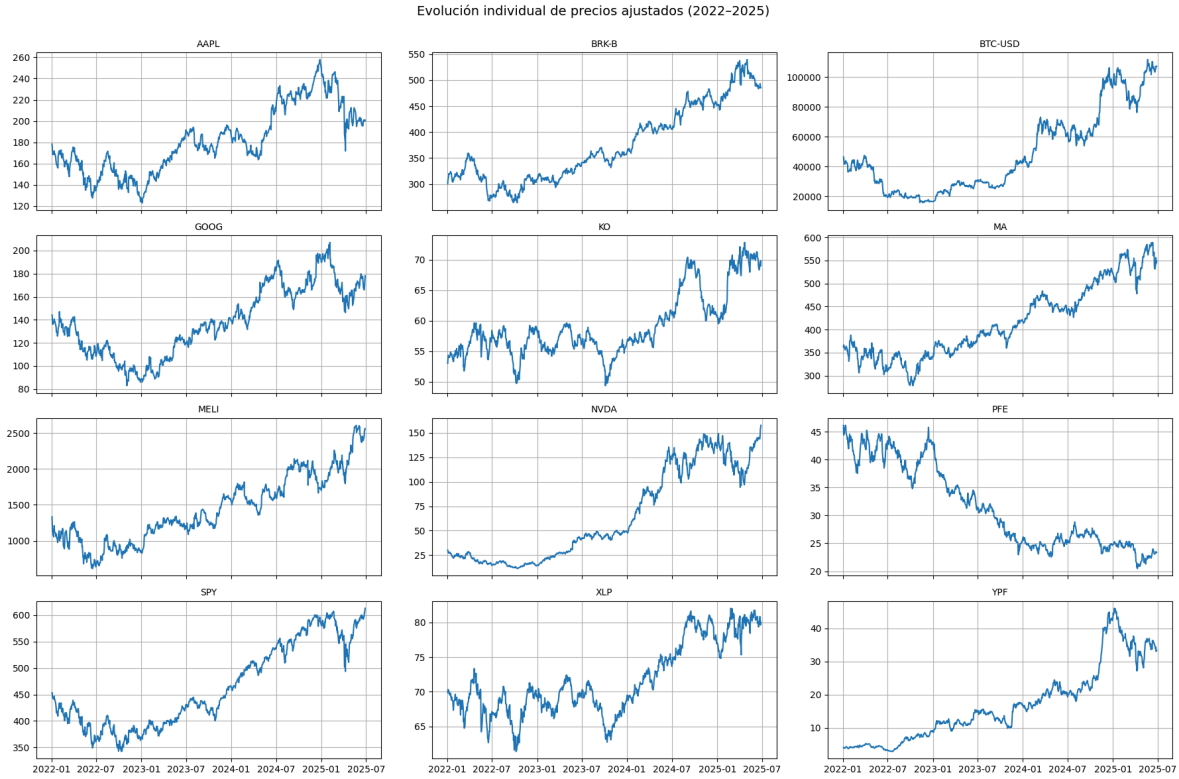
Se grafican los precios ajustados de los activos para observar su evolución entre enero de 2022 y junio de 2025.

La visualización se realiza en dos etapas complementarias:

1. **Gráficos individuales:** se representa cada activo por separado para identificar su comportamiento propio (tendencias, rupturas, niveles de volatilidad, etc.).
2. **Gráfico comparado (base 100):** se normalizan todos los precios a una misma base inicial (100) y se grafican juntos en un solo panel. Esto permite comparar el rendimiento relativo de cada instrumento a lo largo del tiempo, sin importar el valor nominal de sus precios.

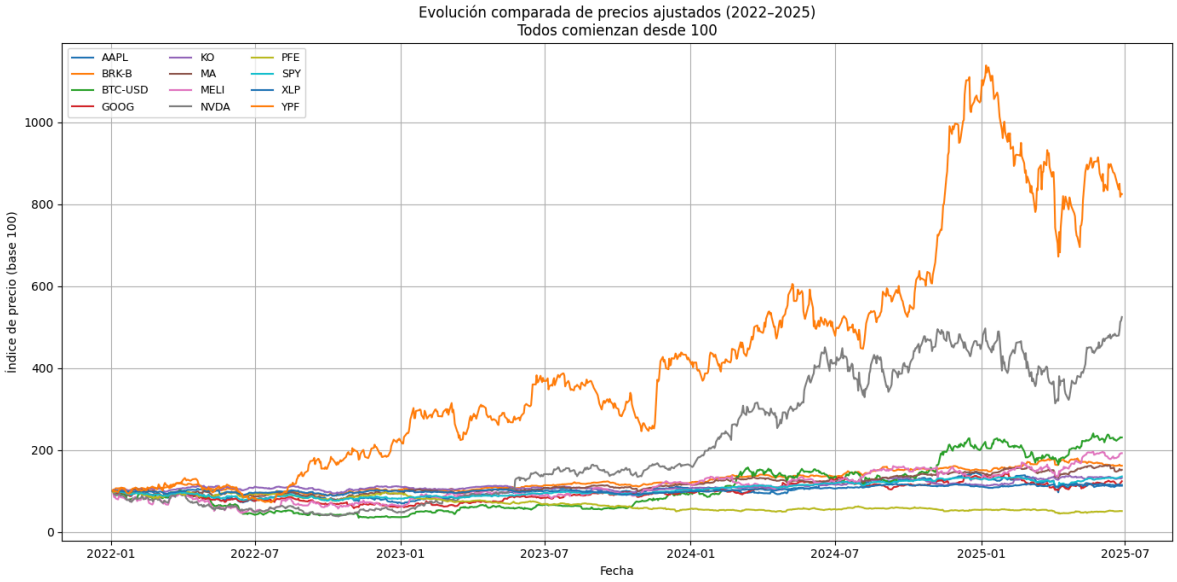
Estos gráficos permiten tener una visión clara del desempeño individual y colectivo de los activos seleccionados.

```
# Paso 1: Gráficos individuales
# Se genera un gráfico por cada activo para observar su comportamient

import matplotlib.pyplot as plt

# Configura el tamaño general y el diseño de subplots
n = len(data.columns)
fig, axes = plt.subplots(nrows=4, ncols=3, figsize=(18, 12), sharex=T
axes = axes.flatten()

# Genera un gráfico por cada activo
for i, ticker in enumerate(data.columns):
    axes[i].plot(data.index, data[ticker], lw=1.5)
    axes[i].set_title(f"{ticker}", fontsize=10)
    axes[i].grid(True)

# Ajusta el layout para que no se superpongan elementos
plt.suptitle("Evolución individual de precios ajustados (2022–2025)",
plt.tight_layout()
plt.subplots_adjust(top=0.92)
plt.show()
```




```
# Paso 2: Gráfico comparado (base 100)
# Se normalizan los precios para que todos comiencen desde 100 y se p

normalized_data = data / data.iloc[0] * 100

# Crear el gráfico
plt.figure(figsize=(14, 7))
for ticker in normalized_data.columns:
    plt.plot(normalized_data.index, normalized_data[ticker], label=ti

plt.title("Evolución comparada de precios ajustados (2022–2025)\nTodo
```

```
plt.xlabel("Fecha")
plt.ylabel("Índice de precio (base 100)")
plt.legend(loc="upper left", ncol=3, fontsize=9)
plt.grid(True)
plt.tight_layout()
plt.show()
```



📉 Parte 2.2: Retornos diarios

En esta sección calculamos los retornos diarios de cada activo a partir de los precios ajustados. Esto permite analizar las variaciones porcentuales diarias y trabajar con medidas de rendimiento y riesgo.

```
# Cálculo de retornos diarios
# A partir del DataFrame 'data' que contiene precios ajustados, se ca

data_returns = data.pct_change().dropna()
```

```
# Se muestran las primeras filas para verificar
data_returns.head()
```

Ticker	AAPL	BRK-B	BTC-USD	GOOG	KO	MA	MEL
Date							
2022-01-04	-0.012692	0.025732	-0.012066	-0.004536	0.016695	0.011969	-0.06869
2022-01-05	-0.026600	0.004505	-0.050734	-0.046830	0.008293	-0.003969	-0.08997
2022-01-06	-0.016693	0.010648	-0.009366	-0.000745	-0.005264	-0.009132	0.02398

```
# Información general del DataFrame de retornos
# Se procede a verificar los tipo de datos, cantidad de observaciones

print("Shape:", data_returns.shape)
print("\nInformación:")
print(data_returns.info())

# También se puede ver un resumen estadístico general
data_returns.describe().T
```

```
Shape: (873, 12)

Información:
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 873 entries, 2022-01-04 to 2025-06-27
Data columns (total 12 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   AAPL     873 non-null    float64
 1   BRK-B    873 non-null    float64
 2   BTC-USD  873 non-null    float64
 3   GOOG     873 non-null    float64
 4   KO       873 non-null    float64
 5   MA       873 non-null    float64
 6   MELI     873 non-null    float64
 7   NVDA     873 non-null    float64
 8   PFE      873 non-null    float64
 9   SPY      873 non-null    float64
 10  XLP      873 non-null    float64
 11  YPF      873 non-null    float64
dtypes: float64(12)
memory usage: 88.7 KB
None
```

Ticker	count	mean	std	min	25%	50%	75%	
AAPL	873.0	0.000306	0.018529	-0.092456	-0.008826	0.000988	0.009720	0.
BRK-B	873.0	0.000615	0.011482	-0.069073	-0.005631	0.000628	0.007012	0.
BTC-USD	873.0	0.001554	0.034537	-0.226807	-0.015632	0.000615	0.017740	0.
GOOG	873.0	0.000461	0.020899	-0.096350	-0.011677	0.001172	0.011731	0.
KO	873.0	0.000368	0.010258	-0.069626	-0.005577	0.000698	0.006025	0.
MA	873.0	0.000586	0.014890	-0.076864	-0.007013	0.001054	0.008244	0.
MELI	873.0	0.001302	0.033260	-0.168789	-0.015319	0.001462	0.018677	0.
NVDA	873.0	0.002524	0.035520	-0.169682	-0.018416	0.003350	0.022457	0.
PFE	873.0	-0.000660	0.015439	-0.067180	-0.010670	-0.000875	0.008415	0.
SPY	873.0	0.000417	0.011875	-0.058543	-0.005486	0.000505	0.006617	0.
XLP	873.0	0.000193	0.008699	-0.064348	-0.004890	0.000527	0.005281	0.
YPF	873.0	0.003015	0.035120	-0.105812	-0.018233	0.001869	0.020562	0.

Interpretación de los retornos diarios

A partir de los precios históricos ajustados, calculamos los retornos diarios de cada activo. Esto nos permite observar las variaciones porcentuales diarias, que son clave para entender el comportamiento de cada inversión en el corto plazo.

- El conjunto de datos contiene retornos diarios desde la fecha inicial hasta la más reciente disponible, con más de **870 observaciones** por activo.
- Los retornos están expresados como porcentajes (por ejemplo, un valor de 0.01 representa un +1%).
- A través del resumen estadístico (media, desviación estándar, valores mínimos y máximos), podemos observar:
    - Qué tan volátil es cada activo (cuanto mayor la desviación estándar, mayor el riesgo).
    - Si los activos tienen tendencia a rendimientos positivos o negativos (según su media).
    - La presencia de valores extremos (outliers) que pueden indicar momentos de fuerte suba o baja.

📌 Análisis de resultados estadísticos

A partir del resumen estadístico de los retornos diarios, se destacan algunos aspectos relevantes:

- **Rentabilidad promedio**: Algunos activos como MELI, NVDA y BTC-USD presentan medias diarias más altas, lo que puede reflejar un mayor potencial de crecimiento, aunque también suelen ir acompañados de mayor volatilidad.
- **Volatilidad (riesgo)**: BTC-USD muestra una desviación estándar mucho más alta que el resto, indicando una mayor inestabilidad. MELI y NVDA también tienen riesgos elevados. En cambio, activos como KO, XLP o PFE presentan menor volatilidad, lo que los hace más estables.
- **Retornos extremos**: Varios activos tienen mínimos diarios cercanos al -10% o más, lo que indica que han tenido días de fuertes caídas. Esto es importante al evaluar el riesgo de pérdida en el corto plazo.

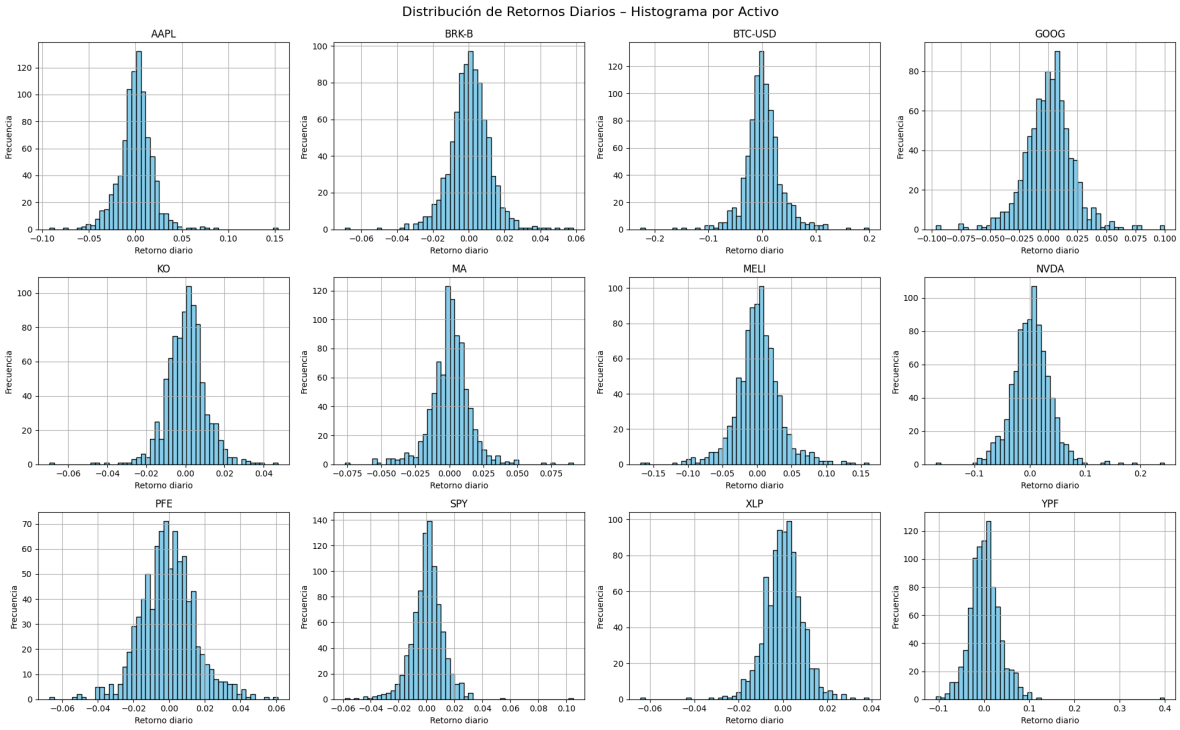
Este análisis inicial nos permite empezar a distinguir activos más riesgosos de los más estables, lo cual será clave al momento de construir una cartera diversificada para el cliente.

🧱 Parte 2.3: Histogramas de retornos diarios

Se procede a graficar la distribución de los retornos diarios de cada activo mediante histogramas. Esta visualización permite identificar la forma de la distribución (asimetría, curtosis, presencia de valores extremos) y comparar la dispersión entre distintos activos.

```
# Se define la grilla de subplots para visualizar todos los histogram
fig, axes = plt.subplots(nrows=3, ncols=4, figsize=(20, 12))
axes = axes.flatten()
```

```
# Se recorre cada activo y se grafica su histograma de retornos diari
for i, ticker in enumerate(data_returns.columns):
    ax = axes[i]
    ax.hist(data_returns[ticker], bins=50, color='skyblue', edgecolor
    ax.set_title(ticker)
    ax.set_xlabel('Retorno diario')
    ax.set_ylabel('Frecuencia')
    ax.grid(True)

# Ajuste de espaciado y título general
plt.tight_layout()
plt.suptitle("Distribución de Retornos Diarios – Histograma por Activ
plt.show()
```



📌 Análisis de los resultados

🔵 Activos estables y defensivos
(KO, XLP, BRK-B)

Presentan histogramas angostos, simétricos y sin colas pronunciadas. Reflejan baja volatilidad y comportamiento estable, ideales para reducir el riesgo total del portafolio. Son utilizados como activos defensivos, especialmente en contextos de incertidumbre.

🟡 Activos equilibrados y de volatilidad moderada
(AAPL, GOOG, MA, SPY, PFE)

Muestran distribuciones más dispersas pero relativamente simétricas, con algunas colas visibles. Representan activos con buen balance entre riesgo y retorno, adecuados para ocupar un rol central en la cartera de un inversor de perfil moderado. Pfizer, aunque pertenece al sector salud, presenta una mayor dispersión que otros activos defensivos, por lo que se clasifica en este grupo.

🔴 Activos con alta volatilidad o riesgo elevado
(MELI, NVDA, BTC-USD, YPF)

Exhiben histogramas amplios, colas largas o asimetría marcada. Indican mayor frecuencia de retornos extremos, tanto positivos como negativos. Estos activos implican un riesgo considerable, por lo que su inclusión debe ser cuidadosa y en proporciones limitadas dentro de una estrategia diversificada.

Esta clasificación visual permite orientar la selección de activos para lograr un equilibrio entre estabilidad, crecimiento y retorno esperado, respetando el perfil de riesgo del cliente.

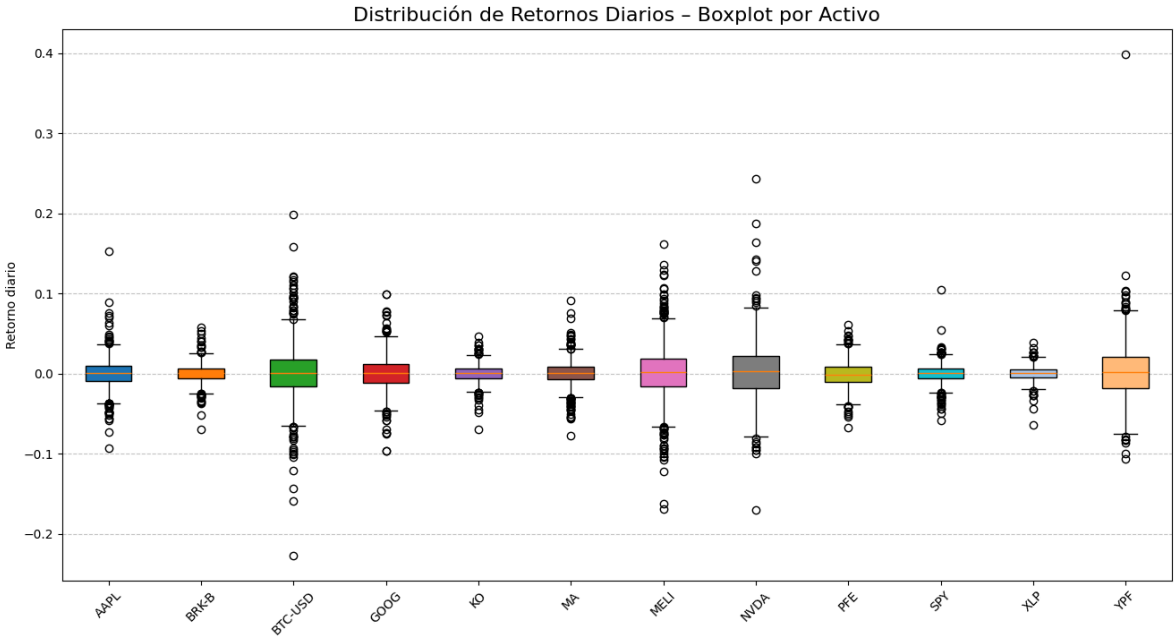
📦 Parte 2.4: Diagramas de caja (Boxplots) de retornos diarios

Se utilizan diagramas de caja para comparar visualmente la dispersión, la simetría y la presencia de valores atípicos en los retornos diarios de cada activo. Esta herramienta permite identificar de forma rápida cuáles activos presentan mayor riesgo (mayor amplitud de la caja y más outliers) y cuáles son más estables.

```
# Paleta de colores definida
colores = {
    'AAPL': '#1f77b4',
    'BRK-B': '#ff7f0e',
    'BTC-USD': '#2ca02c',
    'GOOG': '#d62728',
    'KO': '#9467bd',
    'MA': '#8c564b',
```

```
    'MELI': '#e377c2',
    'NVDA': '#7f7f7f',
    'PFE': '#bcbd22',
    'SPY': '#17becf',
    'XLP': '#aec7e8',
    'YPF': '#ffbb78'
}

# Creamos el boxplot con colores por activo
plt.figure(figsize=(16, 8))
bp = plt.boxplot([data_returns[col] for col in data_returns.columns],
                 labels=data_returns.columns,
                 patch_artist=True)

# Aplicamos los colores personalizados
for patch, ticker in zip(bp['boxes'], data_returns.columns):
    patch.set_facecolor(colores[ticker])

plt.title('Distribución de Retornos Diarios — Boxplot por Activo', fo
plt.ylabel('Retorno diario')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.xticks(rotation=45)
plt.show()
```

```
/tmp/ipython-input-1914491010.py:19: MatplotlibDeprecationWarning: The
  bp = plt.boxplot([data_returns[col] for col in data_returns.columns]
```



📌 Análisis de los resultados

A partir del gráfico de boxplots, los activos pueden agruparse según la dispersión de sus retornos y la presencia de valores atípicos:

🔵 Activos de baja volatilidad y comportamiento estable (KO, BRK-B, XLP, SPY)

Presentan cajas angostas, sin colas extendidas ni gran cantidad de outliers. Sus retornos diarios están concentrados cerca de la mediana, lo que sugiere un comportamiento predecible y menor exposición a riesgos extremos. Son ideales para aportar estabilidad a la cartera.

🟡 **Activos de volatilidad moderada y distribución controlada (AAPL, GOOG, MA, PFE)**

Muestran cajas más amplias y una presencia moderada de valores extremos. Combinan potencial de crecimiento con una volatilidad contenida, por lo que son adecuados para un portafolio balanceado. Pfizer, a pesar de pertenecer a un sector defensivo, muestra más dispersión de la esperada, alineándose mejor con este grupo.

🔴 **Activos con alta dispersión y riesgo elevado (BTC-USD, MELI, NVDA, YPF)**

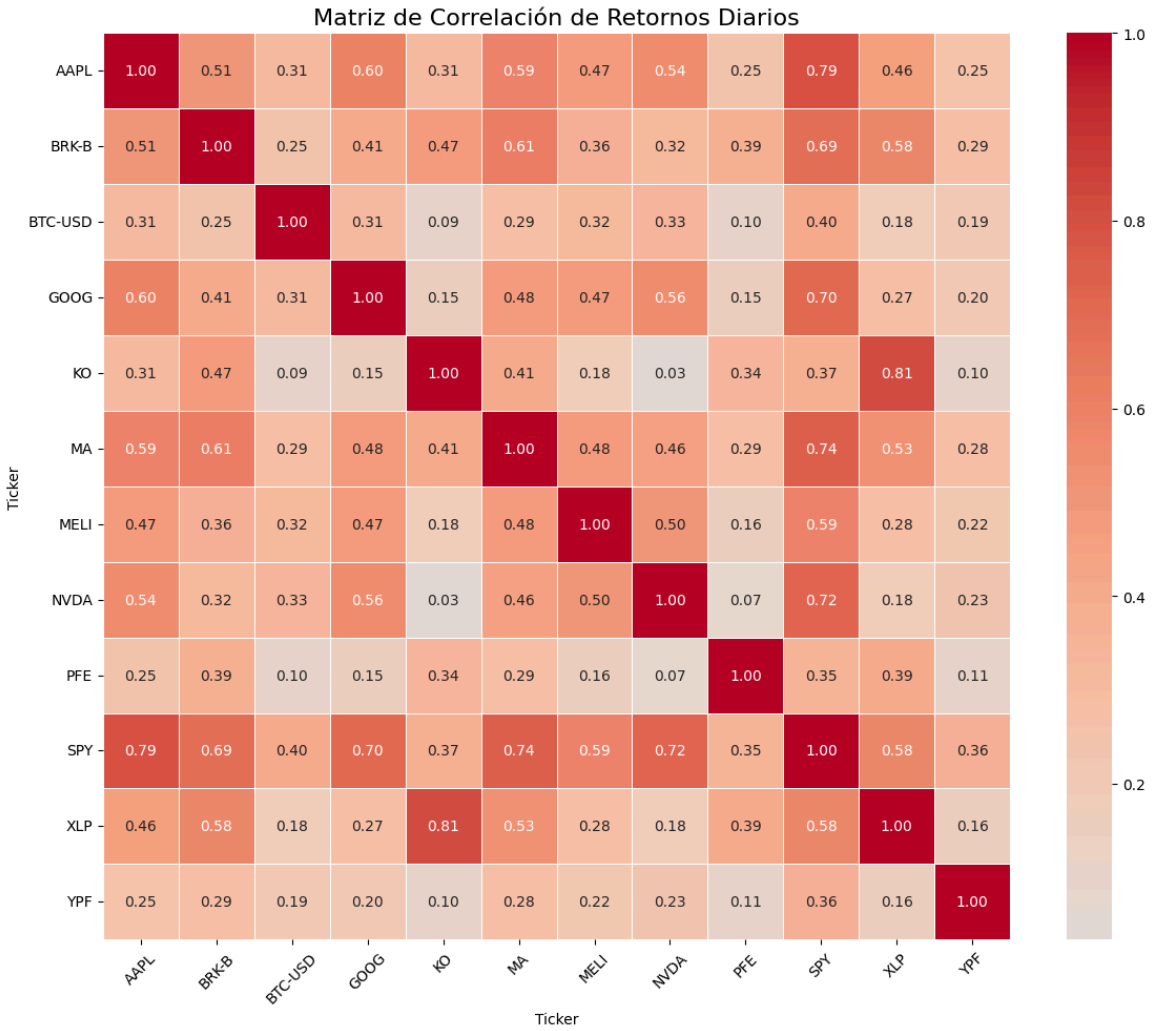
Se destacan por cajas más anchas y gran cantidad de outliers, lo que refleja alta volatilidad e inestabilidad. BTC-USD y NVDA presentan los casos más extremos, con valores alejados del rango intercuartílico tanto por subas como por bajas. Estos activos pueden potenciar el retorno, pero requieren una asignación limitada dentro de una estrategia moderada.

Esta clasificación permite profundizar el análisis de riesgo y complementar las observaciones realizadas previamente con histogramas.

🔥 Parte 2.5: Matriz de correlación y mapa de calor

Se calcula la matriz de correlación de los retornos diarios para identificar la relación entre los activos. Posteriormente, se visualiza con un mapa de calor (heatmap), lo que permite detectar qué instrumentos se mueven en conjunto y cuáles pueden aportar verdadera diversificación a la cartera.

```
import seaborn as sns

# Se calcula la matriz de correlación de los retornos
correlacion = data_returns.corr()

# Se grafica la matriz con un mapa de calor
plt.figure(figsize=(12, 10))
sns.heatmap(correlacion, annot=True, cmap='coolwarm', center=0, linew
plt.title('Matriz de Correlación de Retornos Diarios', fontsize=16)
plt.xticks(rotation=45)
plt.yticks(rotation=0)
plt.tight_layout()
plt.show()
```


Matriz de Correlación de Retornos Diarios
Ticker
AAPL 1.00 0.51 0.31 0.60 0.31 0.59 0.47 0.54 0.25 0.79 0.46 0.25
BRK-B 0.51 1.00 0.25 0.41 0.47 0.61 0.36 0.32 0.39 0.69 0.58 0.29
BTC-USD 0.31 0.25 1.00 0.31 0.09 0.29 0.32 0.33 0.10 0.40 0.18 0.19
GOOG 0.60 0.41 0.31 1.00 0.15 0.48 0.47 0.56 0.15 0.70 0.27 0.20
KO 0.31 0.47 0.09 0.15 1.00 0.41 0.18 0.03 0.34 0.37 0.81 0.10
MA 0.59 0.61 0.29 0.48 0.41 1.00 0.48 0.46 0.29 0.74 0.53 0.28
MELI 0.47 0.36 0.32 0.47 0.18 0.48 1.00 0.50 0.16 0.59 0.28 0.22
NVDA 0.54 0.32 0.33 0.56 0.03 0.46 0.50 1.00 0.07 0.72 0.18 0.23
PFE 0.25 0.39 0.10 0.15 0.34 0.29 0.16 0.07 1.00 0.35 0.39 0.11
SPY 0.79 0.69 0.40 0.70 0.37 0.74 0.59 0.72 0.35 1.00 0.58 0.36
XLP 0.46 0.58 0.18 0.27 0.81 0.53 0.28 0.18 0.39 0.58 1.00 0.16
YPF 0.25 0.29 0.19 0.20 0.10 0.28 0.22 0.23 0.11 0.36 0.16 1.00
AAPL BRK-B BTC-USD GOOG KO MA MELI NVDA PFE SPY XLP YPF
Ticker
1.0
0.8
0.6
0.4
0.2

📌 Análisis de los resultados – Matriz de correlación de retornos diarios

A partir del mapa de calor, se identifican patrones clave en la relación entre activos:

🔗 Activos con alta correlación con el mercado (SPY)

- **AAPL (0.79)**, **GOOG (0.70)**, **MA (0.74)** y **NVDA (0.72)** presentan una correlación elevada con SPY. Esto sugiere que se comportan de manera similar al mercado general estadounidense, siguiendo sus tendencias. Si bien aportan crecimiento al portafolio, no contribuyen significativamente a la diversificación.

📈 Activos con correlaciones medias

- **BRK-B (0.69 con SPY, 0.61 con MA)** mantiene una correlación moderada con distintos activos, lo que indica una exposición diversificada aunque no independiente.
- **MELI** muestra correlaciones medias con activos como NVDA (0.50) y SPY (0.59), lo cual sugiere que aporta cierta diferenciación sin estar completamente desvinculada del comportamiento del mercado.

🧊 Activos con baja correlación con el resto del portafolio

- **KO** y **XLP** tienen una correlación alta entre sí (**0.81**), pero baja respecto a activos más volátiles como AAPL, MELI o BTC-USD. Esto los posiciona como activos defensivos ideales para amortiguar movimientos bruscos del portafolio.
- **PFE** también presenta correlaciones bajas con la mayoría, aunque su comportamiento es más errático que KO y XLP.

🌑 BTC-USD: activo claramente descorrelacionado

- Exhibe correlaciones muy bajas con todos los activos tradicionales (ninguna supera 0.40). Desde un punto de vista técnico, esto lo convierte en un activo que podría aportar diversificación. Sin embargo, debido a su altísima volatilidad, su inclusión debe ser estratégica y limitada.

🇦🇷 YPF: descorrelación por factores no tradicionales

- YPF presenta correlaciones muy bajas con todos los activos analizados. Si bien esto podría parecer atractivo desde una perspectiva de diversificación, en este caso la falta de correlación responde a factores específicos de Argentina, como la intervención estatal, el riesgo país, el cepo cambiario o la volatilidad macroeconómica.
  En consecuencia, YPF no se comporta como una acción global tradicional, y su rendimiento es altamente impredecible. Su inclusión en una cartera moderada requiere un análisis cauteloso.

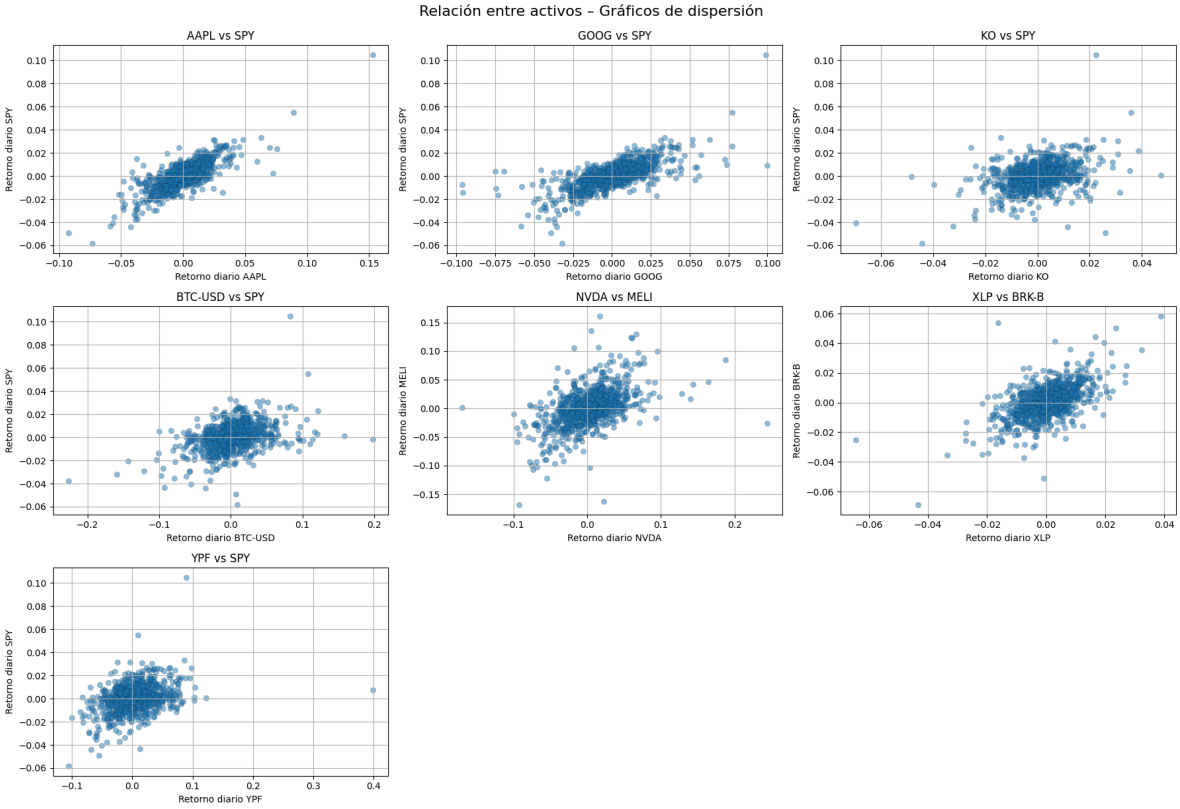
Conclusión

El portafolio propuesto incluye activos con distintos grados de correlación, lo que permite lograr un balance entre crecimiento y diversificación. La presencia de activos con baja correlación como KO, XLP, BTC-USD o incluso MELI ayuda a reducir el riesgo sistémico. Sin embargo, se recomienda limitar la exposición a aquellos activos cuya independencia no necesariamente implica estabilidad (como BTC-USD o YPF).

🔁 Parte 2.6: Gráficos de dispersión entre activos

Se grafican diagramas de dispersión entre pares seleccionados de activos para observar visualmente la relación entre sus retornos diarios. Esta herramienta complementa el análisis de correlación, permitiendo detectar relaciones lineales o patrones conjuntos de comportamiento.

```
# Lista de pares de activos a comparar
pares = [
    ("AAPL", "SPY"),       # alta correlación esperada
    ("GOOG", "SPY"),       # tecnológica vs mercado
    ("KO", "SPY"),         # defensiva vs mercado
    ("BTC-USD", "SPY"),    # cripto vs mercado
    ("NVDA", "MELI"),      # dos activos de alto crecimiento
    ("XLP", "BRK-B"),      # consumo básico vs value
    ("YPF", "SPY")         # activo atípico vs mercado
]

# Tamaño del gráfico general
fig, axes = plt.subplots(nrows=3, ncols=3, figsize=(18, 12))
axes = axes.flatten()

# Generación de gráficos
for i, (x, y) in enumerate(pares):
    ax = axes[i]
    ax.scatter(data_returns[x], data_returns[y], alpha=0.5, edgecolor
    ax.set_xlabel(f'Retorno diario {x}')
    ax.set_ylabel(f'Retorno diario {y}')
    ax.set_title(f'{x} vs {y}')
    ax.grid(True)

# Eliminar paneles vacíos si sobran
for j in range(len(pares), len(axes)):
    fig.delaxes(axes[j])

plt.tight_layout()
plt.suptitle("Relación entre activos — Gráficos de dispersión", fonts
plt.show()
```



📌 Análisis de los resultados

- **AAPL vs SPY** y **GOOG vs SPY** muestran una relación lineal clara. Sus retornos diarios se alinean de forma ascendente, confirmando que ambas acciones se mueven en sincronía con el mercado general.
- **XLP vs BRK-B** presenta una correlación positiva visible, aunque algo más dispersa. Muestra que ambos activos tienden a moverse en la misma dirección,

pero con menor intensidad.

- **NVDA vs MELI** confirma que ambos activos, de alto crecimiento, comparten una dinámica de comportamiento similar, aunque con mayor volatilidad y presencia de valores extremos.
- **KO vs SPY** muestra una nube sin dirección clara, reflejando la independencia de KO respecto al mercado. Esto confirma su rol como activo defensivo, útil para reducir la sensibilidad del portafolio ante caídas generales.
- **BTC-USD vs SPY** presenta una distribución sin patrón definido, lo que confirma la descorrelación de Bitcoin respecto a los activos tradicionales.
- **YPF vs SPY** revela una ligera tendencia ascendente, no tan evidente en la matriz de correlación. Si bien existe cierta respuesta al contexto internacional, la alta dispersión se explica por factores locales que afectan a YPF de forma particular, como regulaciones, riesgo país y shocks macroeconómicos argentinos.

⚖️ Parte 3: Métricas de riesgo y retorno

En esta sección se analizan distintas métricas financieras que permiten evaluar el desempeño de los activos desde una perspectiva ajustada por riesgo. Estas herramientas permiten comparar activos no solo por su rentabilidad, sino también por su nivel de riesgo y eficiencia relativa.

Se calcularán los principales ratios utilizados en finanzas modernas, tales como:

- Ratio de Sharpe
- Ratio de Sortino
- Ratio de Treynor
- CAPM y Betas

Cada uno de estos indicadores ofrece una visión complementaria del equilibrio entre retorno esperado y volatilidad, lo cual resulta fundamental para diseñar carteras alineadas con el perfil de riesgo del cliente.

📐 Parte 3.1: CAPM y Betas

Se estima la beta de cada activo con respecto al mercado utilizando el modelo CAPM (Capital Asset Pricing Model). La beta mide la sensibilidad del activo frente a los movimientos del mercado. Valores de beta:

- mayores a 1: implican mayor riesgo sistemático que el mercado
- iguales a 1: se mueven en línea con el mercado
- menores a 1: son menos volátiles que el mercado

Fórmula estimada por regresión:

$r_i - rf = \alpha + \beta \times (r_m - rf) + \varepsilon$

Donde:

- r_i = retorno del activo
- r_m = retorno del mercado (usamos SPY)
- rf = tasa libre de riesgo
- β = coeficiente estimado → mide la sensibilidad al mercado

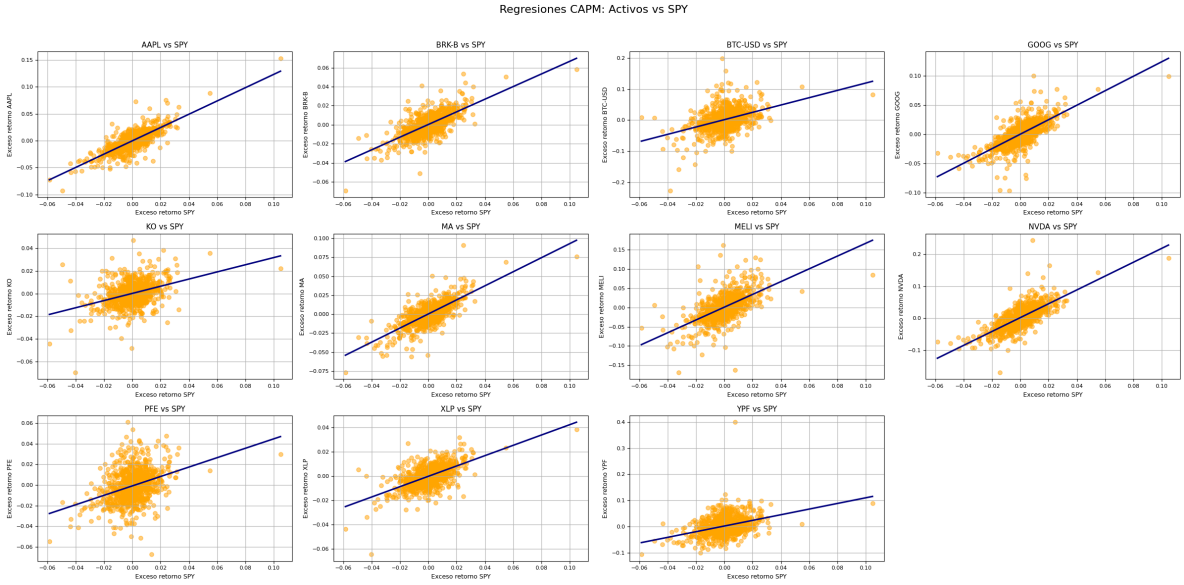
```
import statsmodels.api as sm

# Tasa libre de riesgo diaria (usamos la misma que antes)
rf_daily = 0.04 / 252

# Calculamos retornos excedentes (activo - rf)
excess_returns = data_returns.sub(rf_daily)

# Retornos del mercado (SPY)
market_excess = data_returns["SPY"] - rf_daily

# Estimamos beta para cada activo (excepto SPY)
betas = {}

for ticker in data_returns.columns:
    if ticker != "SPY":
        y = excess_returns[ticker]
        x = market_excess
        x = sm.add_constant(x)
        modelo = sm.OLS(y, x).fit()
        betas[ticker] = modelo.params[1]  # Coeficiente beta (pendien

# Convertimos a DataFrame ordenado
df_betas = pd.Series(betas).sort_values().to_frame(name="Beta").round
df_betas
```

```
/tmp/ipython-input-2035161724.py:21: FutureWarning: Series.__getitem__
  betas[ticker] = modelo.params[1]  # Coeficiente beta (pendiente)
/tmp/ipython-input-2035161724.py:21: FutureWarning: Series.__getitem__
  betas[ticker] = modelo.params[1]  # Coeficiente beta (pendiente)
/tmp/ipython-input-2035161724.py:21: FutureWarning: Series.__getitem__
  betas[ticker] = modelo.params[1]  # Coeficiente beta (pendiente)
/tmp/ipython-input-2035161724.py:21: FutureWarning: Series.__getitem__
  betas[ticker] = modelo.params[1]  # Coeficiente beta (pendiente)
/tmp/ipython-input-2035161724.py:21: FutureWarning: Series.__getitem__
  betas[ticker] = modelo.params[1]  # Coeficiente beta (pendiente)
/tmp/ipython-input-2035161724.py:21: FutureWarning: Series.__getitem__
  betas[ticker] = modelo.params[1]  # Coeficiente beta (pendiente)
/tmp/ipython-input-2035161724.py:21: FutureWarning: Series.__getitem__
  betas[ticker] = modelo.params[1]  # Coeficiente beta (pendiente)
/tmp/ipython-input-2035161724.py:21: FutureWarning: Series.__getitem__
  betas[ticker] = modelo.params[1]  # Coeficiente beta (pendiente)
/tmp/ipython-input-2035161724.py:21: FutureWarning: Series.__getitem__
  betas[ticker] = modelo.params[1]  # Coeficiente beta (pendiente)
/tmp/ipython-input-2035161724.py:21: FutureWarning: Series.__getitem__
  betas[ticker] = modelo.params[1]  # Coeficiente beta (pendiente)
/tmp/ipython-input-2035161724.py:21: FutureWarning: Series.__getitem__
  betas[ticker] = modelo.params[1]  # Coeficiente beta (pendiente)
```

	Beta
KO	0.3163
XLP	0.4251
PFE	0.4553
BRK-B	0.6640
MA	0.9268
YPF	1.0767
BTC-USD	1.1767
AAPL	1.2335
GOOG	1.2380
MELI	1.6563
NVDA	2.1621

```
tickers = [ticker for ticker in data_returns.columns if ticker != "SP

fig, axes = plt.subplots(3, 4, figsize=(24, 12))
axes = axes.flatten()

for i, ticker in enumerate(tickers):
    x = data_returns["SPY"] - rf_daily
    y = data_returns[ticker] - rf_daily

    sns.regplot(x=x, y=y, ci=None,
                scatter_kws={"alpha": 0.5, "color": "orange"},
```

```
                    line_kws={"color": "navy"},
                    ax=axes[i])

        axes[i].set_title(f"{ticker} vs SPY", fontsize=11)
        axes[i].set_xlabel("Exceso retorno SPY", fontsize=9)
        axes[i].set_ylabel(f"Exceso retorno {ticker}", fontsize=9)
        axes[i].tick_params(axis='both', labelsize=8)
        axes[i].grid(True)

    # Eliminar ejes sobrantes si hay
    for j in range(i + 1, len(axes)):
        fig.delaxes(axes[j])

    fig.suptitle("Regresiones CAPM: Activos vs SPY", fontsize=16)
    plt.tight_layout(rect=[0, 0, 1, 0.96])
    plt.show()
```



📉 Interpretación de las regresiones CAPM (Activos vs SPY)

A continuación se presentan conclusiones breves sobre la relación entre los excesos de retorno de cada activo y el mercado (SPY), observada a partir de los gráficos de

regresión CAPM:

- **AAPL**: Alta correlación y fuerte pendiente positiva. AAPL muestra una sensibilidad elevada al mercado, típica de un activo tecnológico de crecimiento.
- **BRK-B**: Relación moderada con SPY, con mayor dispersión. El holding de Berkshire actúa como activo más defensivo y diversificado, con beta inferior a 1.
- **BTC-USD**: Relación débil y dispersa. El exceso de retorno de Bitcoin es poco explicable por el mercado tradicional; actúa más como activo alternativo, con alto riesgo no sistemático.
- **GOOG**: Alta correlación con SPY y bajo nivel de dispersión. Comportamiento fuertemente alineado al mercado, con beta cercana o superior a 1.
- **KO**: Pendiente baja y mayor dispersión. Refleja su perfil defensivo y baja sensibilidad a los movimientos del mercado.
- **MA**: Buena alineación con SPY, con pendiente significativa. Activo pro-mercado, con beta moderada y comportamiento sistemático.
- **MELI**: Relación positiva clara pero con más dispersión. Activo volátil, sigue en promedio la dirección del mercado aunque con mayor riesgo.
- **NVDA**: Relación muy fuerte y pendiente elevada. Beta claramente superior a 1, confirma su carácter agresivo frente a los movimientos del SPY.
- **PFE**: Pendiente suave y nube dispersa. Comportamiento más independiente del mercado, típico del sector salud con menor exposición sistemática.
- **XLP**: Pendiente baja y moderada dispersión. ETF defensivo del sector consumo básico, con beta menor a 1 como era esperable.
- **YPF**: Relación débil y muy dispersa. Comportamiento desvinculado del mercado estadounidense, posiblemente influido por factores locales como el riesgo país y la política argentina.

Estas observaciones permiten comprender mejor el grado de exposición sistemática de cada activo, aportando información clave para la construcción de carteras diversificadas en línea con el perfil del inversor.

🔢 Parte 3.2: Cálculo de ratios de performance: Sharpe, Sortino y Treynor

Para evaluar el desempeño ajustado por riesgo de cada activo, se calculan tres ratios financieros fundamentales:

- **Ratio de Sharpe**: mide el exceso de retorno sobre la tasa libre de riesgo por unidad de volatilidad total. Es útil para comparar activos con distinto nivel de

riesgo.

- **Ratio de Sortino**: similar al de Sharpe, pero solo penaliza la volatilidad negativa (caídas), lo cual lo vuelve más preciso para evaluar inversiones de perfil moderado.
- **Ratio de Treynor**: relaciona el exceso de retorno con el riesgo sistemático (beta). Es adecuado para carteras bien diversificadas, donde solo importa el riesgo de mercado.

Los valores se anualizan y permiten identificar cuáles activos ofrecen mejor retorno ajustado por el tipo de riesgo que representan.

```
import numpy as np

# Diccionario para guardar los ratios
ratios = {}

# Calcular ratios para todos los activos excepto SPY
for ticker in data_returns.columns:
    if ticker != "SPY":
        r = data_returns[ticker]
        ex_r = r - rf_daily  # Exceso de retorno diario

        # Sharpe Ratio
        sharpe = (ex_r.mean() / ex_r.std()) * np.sqrt(252)

        # Sortino Ratio
        downside_std = r[r < rf_daily].std()
        sortino = (ex_r.mean() / downside_std) * np.sqrt(252)

        # Treynor Ratio
        treynor = (ex_r.mean() * 252) / betas[ticker]

        # Guardar resultados
        ratios[ticker] = {
            "Sharpe": sharpe,
            "Sortino": sortino,
            "Treynor": treynor
        }

# Crear tabla ordenada y redondeada
ratios_df = pd.DataFrame(ratios).T.round(2)
ratios_df.sort_values(by="Sharpe", ascending=False)
```

	Sharpe	Sortino	Treynor
YPF	1.29	2.39	0.67
NVDA	1.06	1.72	0.28
BTC-USD	0.64	0.92	0.30
BRK-B	0.63	0.92	0.17
MELI	0.55	0.78	0.17
MA	0.46	0.62	0.12
KO	0.32	0.45	0.17
GOOG	0.23	0.33	0.06
AAPL	0.13	0.18	0.03
XLP	0.06	0.09	0.02
PFE	-0.84	-1.32	-0.45

📊 Interpretación de los ratios de performance

A partir de los cálculos realizados, se destacan los siguientes aspectos:

- **YPF** obtiene sorpresivamente los valores más altos en los ratios de **Sharpe** y **Sortino**, lo que indica un alto retorno ajustado por riesgo. Sin embargo, este resultado debe analizarse con cautela, ya que YPF también mostró alta volatilidad y poca correlación con el mercado (lo que puede generar inestabilidad en el portafolio).
- **NVDA** es el segundo activo más eficiente según estos ratios, con muy buen retorno ajustado tanto por volatilidad total como por riesgo a la baja. Confirma su atractivo como activo de crecimiento, aunque con sensibilidad alta al mercado.
- **BTC-USD** y **MELI** presentan ratios intermedios. Si bien implican riesgo, ofrecen un retorno relativamente compensado, sobre todo si se los mantiene en proporciones controladas.
- **BRK-B** mantiene un perfil balanceado, con ratios consistentes que reflejan estabilidad y eficiencia, especialmente en el ratio de Treynor, lo que lo hace interesante para portafolios diversificados.
- **MA, KO y GOOG** muestran retornos ajustados por riesgo más bajos, aunque aceptables dentro del contexto de activos moderados. Son adecuados para complementar una cartera con otros activos más dinámicos.
- **AAPL y XLP** tienen valores muy bajos en los tres ratios, lo que sugiere que su riesgo no fue bien compensado por retornos en este período particular.

- **PFE** es el único activo con valores negativos en todos los ratios, lo que indica que no compensó su riesgo ni su volatilidad. Su inclusión en el portafolio debería ser reevaluada.

Estos indicadores permiten priorizar activos según su rendimiento ajustado por riesgo, algo clave para diseñar carteras eficientes adaptadas al perfil del cliente.

📘 Parte 3.3: Regresión con el Modelo Fama-French de 3 Factores

Para mejorar la explicación del retorno de los activos, se aplica el **modelo Fama-French de tres factores**, una extensión del CAPM que considera no solo el riesgo de mercado, sino también el efecto del tamaño de la empresa y su valoración.

Fórmula:
$$R - Rf = \alpha + \beta_1 \times (Rm - Rf) + \beta_2 \times SMB + \beta_3 \times HML + error$$

📌 ¿Qué significan los factores?

- **Rm - Rf**: Exceso de retorno del mercado (igual que en el CAPM).
- **SMB (Small Minus Big)**: mide la diferencia de rendimiento entre acciones de empresas pequeñas y grandes. Captura el efecto de tamaño.
- **HML (High Minus Low)**: mide la diferencia entre acciones con alto valor contable/mercado y aquellas con bajo. Captura el efecto de "value vs. growth".
- **α**: el rendimiento adicional que no puede explicarse por los tres factores (alpha).
- **β_1, β_2, β_3**: sensibilidad de la acción a cada factor.

Este modelo permite evaluar si un activo tiene rendimientos anormales ($\alpha \neq 0$) o si sus retornos pueden explicarse completamente por estos factores sistemáticos.

```
# Descarga y preparación de los factores (diarios) desde la Kenneth F
url = "https://mba.tuck.dartmouth.edu/pages/faculty/ken.french/ftp/F-
factors = pd.read_csv(url, skiprows=3)
factors = factors.rename(columns={"Unnamed: 0": "Date"})
factors = factors[factors["Date"].str.isnumeric()]
factors["Date"] = pd.to_datetime(factors["Date"], format="%Y%m%d")
factors = factors.set_index("Date").astype(float) / 100  # de % a dec
factors = factors[["Mkt-RF", "SMB", "HML", "RF"]]

# Visualizamos las primeras filas del DataFrame ya limpio
factors
```

Date	Mkt-RF	SMB	HML	RF
1926-07-01	0.0009	-0.0025	-0.0027	0.0001
1926-07-02	0.0045	-0.0033	-0.0006	0.0001
1926-07-06	0.0017	0.0030	-0.0039	0.0001
1926-07-07	0.0009	-0.0058	0.0002	0.0001
1926-07-08	0.0022	-0.0038	0.0019	0.0001
...	...	...	...	...
2025-09-24	-0.0034	-0.0036	0.0099	0.0002
2025-09-25	-0.0058	-0.0077	0.0055	0.0002
2025-09-26	0.0060	0.0028	0.0045	0.0002
2025-09-29	0.0025	-0.0006	-0.0075	0.0002
2025-09-30	0.0029	-0.0013	-0.0057	0.0002

26087 rows × 4 columns

```
# Unión del DataFrame de factores con los retornos diarios

# Se combinan los factores Fama-French con los retornos diarios de lo
# Solo se conservan las fechas en común (intersección de ambos índice
data_ff = data_returns.join(factors, how="inner")

import statsmodels.api as sm

# Diccionario vacío para guardar los resultados de cada activo
ff_results = {}

# Se calcula el modelo Fama-French para cada activo del portafolio
for ticker in data_returns.columns:
    # Exceso de retorno del activo: Ri - Rf
    Ri = data_ff[ticker] - data_ff["RF"]

    # Variables independientes: factores del modelo
    X = data_ff[["Mkt-RF", "SMB", "HML"]]
    X = sm.add_constant(X)  # agrega constante para estimar alpha

    # Regresión lineal OLS
    modelo = sm.OLS(Ri, X).fit()

    # Se guardan los coeficientes y el R² ajustado
    ff_results[ticker] = {
        "Alpha": round(modelo.params["const"], 4),
        "Beta_Mkt": round(modelo.params["Mkt-RF"], 4),
        "Beta_SMB": round(modelo.params["SMB"], 4),
        "Beta_HML": round(modelo.params["HML"], 4),
        "R2_ajustado": round(modelo.rsquared_adj, 4)
```

```
    }

# Mostrar la tabla resumen con los resultados
pd.DataFrame(ff_results).T
```

	Alpha	Beta_Mkt	Beta_SMB	Beta_HML	R2_ajustado
AAPL	-0.0002	1.1660	-0.3066	-0.2563	0.6315
BRK-B	0.0001	0.8381	-0.2521	0.5115	0.6227
BTC-USD	0.0014	0.9777	0.6503	-0.2810	0.1954
GOOG	0.0000	1.1078	-0.2486	-0.4782	0.5348
KO	-0.0000	0.4087	-0.3218	0.1943	0.1855
MA	0.0001	0.9641	-0.2679	0.0622	0.5511
MELI	0.0010	1.3593	0.3517	-0.8462	0.4319
NVDA	0.0019	1.8718	-0.4568	-1.1034	0.6010
PFE	-0.0010	0.5178	-0.1370	0.2305	0.1284
SPY	-0.0000	0.9929	-0.1136	0.0132	0.9941
XLP	-0.0002	0.4872	-0.2205	0.1579	0.3618
YPF	0.0025	1.2946	0.0405	0.8269	0.1714

📊 Interpretación de los ratios de performance

Los coeficientes estimados permiten analizar la sensibilidad de cada activo a los tres factores explicativos: mercado (Mkt-RF), tamaño (SMB) y estilo (HML). Además, el alpha indica el rendimiento anómalo no explicado por el modelo, y el R^2 ajustado muestra qué tan bien se ajusta la regresión.

🟩 Alta sensibilidad al mercado ($\beta_m > 1$)

- **AAPL (1.17), GOOG (1.11), MELI (1.36), YPF (1.30), NVDA (1.87):**
  Muestran una alta exposición al mercado. NVDA y MELI son los más volátiles. YPF, a pesar de ser una acción argentina, también tiene un comportamiento muy dependiente del mercado internacional.

🟨 Sensibilidad moderada al mercado ($\beta_m \approx 1$)

- **SPY (0.99), BTC-USD (0.98), MA (0.96):**
  Tienen una beta cercana a 1, por lo que siguen de forma similar el comportamiento del mercado.

🟦 Baja sensibilidad al mercado ($\beta_m < 1$)

- **BRK-B (0.84), KO (0.41), PFE (0.52), XLP (0.49):**
  Se comportan de forma más defensiva. Especialmente KO, XLP y PFE, que

muestran baja exposición al riesgo sistémico.

Tamaño (SMB)

- **BTC-USD (0.66), MELI (0.36):**
  Se comportan como activos de empresas pequeñas, capturando el efecto tamaño.
- **NVDA, AAPL, GOOG, KO, BRK-B, etc.**
  Tienen coeficientes SMB negativos, lo cual indica un comportamiento más parecido a empresas grandes consolidadas.

Estilo (HML)

- **BRK-B (0.51), YPF (0.82):**
  Tienen comportamiento asociado a empresas de **valor**.
- **NVDA (-1.10), MELI (-0.85), GOOG (-0.48), AAPL (-0.26):**
  Se comportan como empresas de **crecimiento**, típicas del sector tecnológico.

Alpha (rendimiento no explicado)

- **Positivo** en MELI, NVDA, YPF, MA, BRK-B ⟶ sugieren rendimientos superiores a lo que predice el modelo.
- **Negativo** en AAPL y PFE ⟶ podrían estar rindiendo por debajo del modelo.

R^2 ajustado

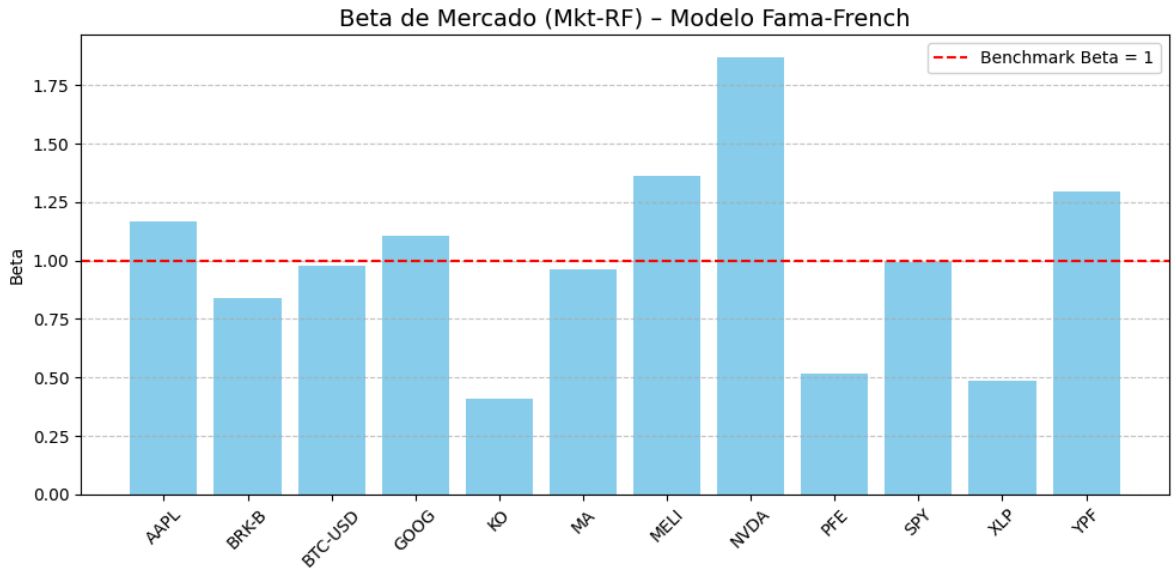
- **SPY (0.99):** como ETF del mercado, el modelo lo explica casi a la perfección.
- **AAPL, BRK-B, GOOG, MA, NVDA:** el modelo también se ajusta muy bien.
- **PFE, KO, BTC-USD, YPF:** el modelo explica poco (bajo R^2), por lo que otros factores podrían estar influyendo.

📊 Visualización de resultados del modelo Fama-French

A continuación se muestran de forma gráfica los resultados obtenidos en la regresión del modelo de tres factores para cada activo. Estas visualizaciones permiten interpretar fácilmente las sensibilidades (betas), el alpha y el nivel de ajuste del modelo (R^2) para cada uno de los activos analizados.

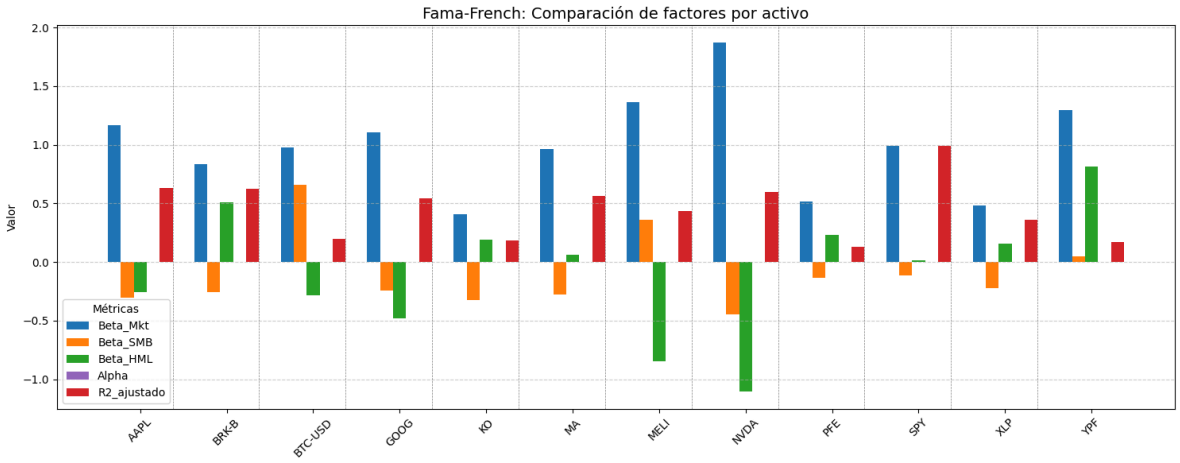
```
# Diccionario con los Betas de mercado (Mkt-RF)
beta_mercado = {
    "AAPL": 1.1660,
    "BRK-B": 0.8382,
    "BTC-USD": 0.9776,
    "GOOG": 1.1080,
    "KO": 0.4067,
    "MA": 0.9631,
    "MELI": 1.3631,
```

```
    "NVDA": 1.8711,
    "PFE": 0.5152,
    "SPY": 0.9925,
    "XLP": 0.4857,
    "YPF": 1.2956
}

# Gráfico de barras
plt.figure(figsize=(10, 5))
plt.bar(beta_mercado.keys(), beta_mercado.values(), color='skyblue')
plt.axhline(1, color='red', linestyle='--', label='Benchmark Beta = 1
plt.title("Beta de Mercado (Mkt-RF) – Modelo Fama-French", fontsize=1
plt.ylabel("Beta")
plt.xticks(rotation=45)
plt.legend()
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```

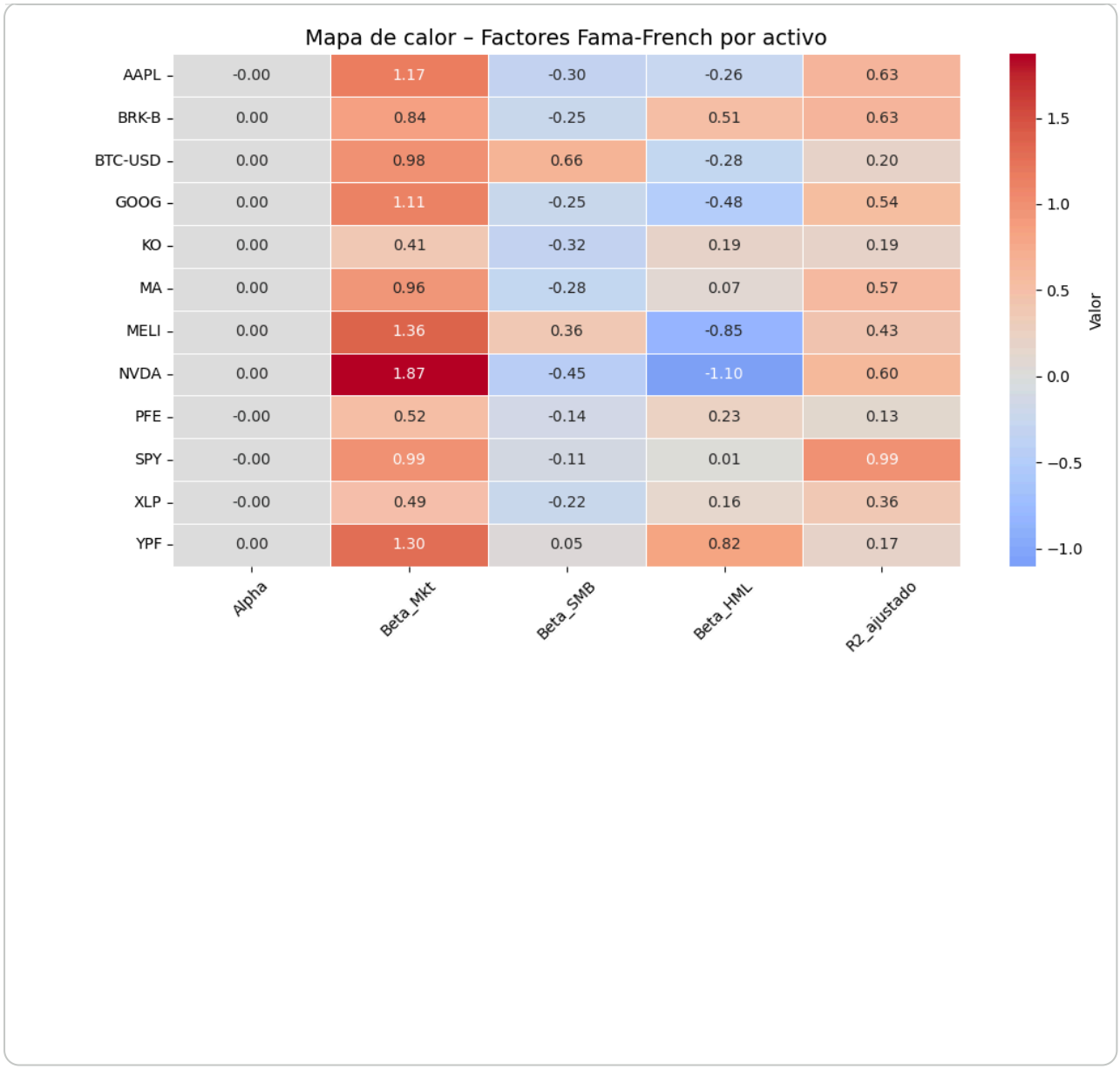



```
# DataFrame con los resultados del modelo Fama-French
ff_results = pd.DataFrame({
    "AAPL": {"Alpha": -0.0001, "Beta_Mkt": 1.1660, "Beta_SMB": -0.302
    "BRK-B": {"Alpha": 0.0002, "Beta_Mkt": 0.8382, "Beta_SMB": -0.253
    "BTC-USD": {"Alpha": 0.0014, "Beta_Mkt": 0.9776, "Beta_SMB": 0.65
    "GOOG": {"Alpha": 0.0000, "Beta_Mkt": 1.1080, "Beta_SMB": -0.2460
    "KO": {"Alpha": 0.0000, "Beta_Mkt": 0.4067, "Beta_SMB": -0.3248,
    "MA": {"Alpha": 0.0002, "Beta_Mkt": 0.9631, "Beta_SMB": -0.2782,
```

```
    "MELI": {"Alpha": 0.0011, "Beta_Mkt": 1.3631, "Beta_SMB": 0.3638,
    "NVDA": {"Alpha": 0.0019, "Beta_Mkt": 1.8711, "Beta_SMB": -0.4486
    "PFE": {"Alpha": -0.0010, "Beta_Mkt": 0.5152, "Beta_SMB": -0.1373
    "SPY": {"Alpha": -0.0000, "Beta_Mkt": 0.9925, "Beta_SMB": -0.1142
    "XLP": {"Alpha": -0.0001, "Beta_Mkt": 0.4857, "Beta_SMB": -0.2235
    "YPF": {"Alpha": 0.0026, "Beta_Mkt": 1.2956, "Beta_SMB": 0.0456,
}).T

# Orden de métricas y colores
ff_metrics = ["Beta_Mkt", "Beta_SMB", "Beta_HML", "Alpha", "R2_ajusta
colors = ['#1f77b4', '#ff7f0e', '#2ca02c', '#9467bd', '#d62728']

# Gráfico
x = np.arange(len(ff_results.index))
width = 0.15

plt.figure(figsize=(15, 6))

# Dibujar las barras
for i, metric in enumerate(ff_metrics):
    plt.bar(x + i * width, ff_results[metric], width, label=metric, c

# Agregar divisores verticales entre activos
for i in range(len(ff_results.index) - 1):
    xpos = x[i] + width * len(ff_metrics)  # fin del grupo de barras
    plt.axvline(x=xpos - width / 2, color='gray', linestyle='--', lin

# Ejes y etiquetas
plt.xticks(x + width * 2, ff_results.index, rotation=45)
plt.title("Fama-French: Comparación de factores por activo", fontsize
plt.ylabel("Valor")
plt.legend(title="Métricas")
plt.grid(axis='y', linestyle='--', alpha=0.6)
plt.tight_layout()
plt.show()
```




```
# Diccionario con los resultados del modelo Fama-French
ff_results = pd.DataFrame({
    "AAPL": {"Alpha": -0.0001, "Beta_Mkt": 1.1660, "Beta_SMB": -0.302
    "BRK-B": {"Alpha": 0.0002, "Beta_Mkt": 0.8382, "Beta_SMB": -0.253
    "BTC-USD": {"Alpha": 0.0014, "Beta_Mkt": 0.9776, "Beta_SMB": 0.65
    "GOOG": {"Alpha": 0.0000, "Beta_Mkt": 1.1080, "Beta_SMB": -0.2460
    "KO": {"Alpha": 0.0000, "Beta_Mkt": 0.4067, "Beta_SMB": -0.3248,
    "MA": {"Alpha": 0.0002, "Beta_Mkt": 0.9631, "Beta_SMB": -0.2782,
    "MELI": {"Alpha": 0.0011, "Beta_Mkt": 1.3631, "Beta_SMB": 0.3638,
    "NVDA": {"Alpha": 0.0019, "Beta_Mkt": 1.8711, "Beta_SMB": -0.4486
    "PFE": {"Alpha": -0.0010, "Beta_Mkt": 0.5152, "Beta_SMB": -0.1373
    "SPY": {"Alpha": -0.0000, "Beta_Mkt": 0.9925, "Beta_SMB": -0.1142
    "XLP": {"Alpha": -0.0001, "Beta_Mkt": 0.4857, "Beta_SMB": -0.2235
    "YPF": {"Alpha": 0.0026, "Beta_Mkt": 1.2956, "Beta_SMB": 0.0456,
}).T

# Mapa de calor
plt.figure(figsize=(10, 6))
sns.heatmap(ff_results, annot=True, fmt=".2f", cmap="coolwarm", cente
plt.title("Mapa de calor — Factores Fama-French por activo", fontsize
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



💼 Parte 4: Simulación de estrategias de inversión

A continuación se simularán múltiples combinaciones de portafolios diversificados construidos a partir de los 12 activos seleccionados.
El objetivo es identificar estrategias de inversión eficientes para un inversor de perfil moderado, considerando el equilibrio entre riesgo y retorno.

Se utilizará una simulación tipo Monte Carlo para generar cientos de combinaciones aleatorias de portafolios, y se evaluará el rendimiento esperado, la volatilidad y el ratio de Sharpe de cada uno. Posteriormente, se analizará la **frontera eficiente** y se compararán al menos **dos estrategias óptimas**.

Esta etapa permitirá fundamentar con mayor precisión qué combinaciones de activos se adaptan mejor al perfil del cliente.

🚀 Parte 4.1: Identificación del Portafolio de Máxima Sharpe

En esta sección se analizará una estrategia que prioriza la **eficiencia en el uso del riesgo**, buscando maximizar el retorno esperado por cada unidad de volatilidad asumida.

El objetivo es identificar el portafolio que presenta el **mayor ratio de Sharpe** entre las 10.000 combinaciones simuladas. Esta métrica es ampliamente utilizada en finanzas para evaluar la calidad de una estrategia de inversión, ya que ajusta el retorno en función del riesgo total.

Una vez identificado, se analizará su rendimiento, composición y posición dentro de la frontera eficiente, permitiendo evaluar su conveniencia para un inversor con tolerancia media al riesgo y objetivos de crecimiento sostenido.

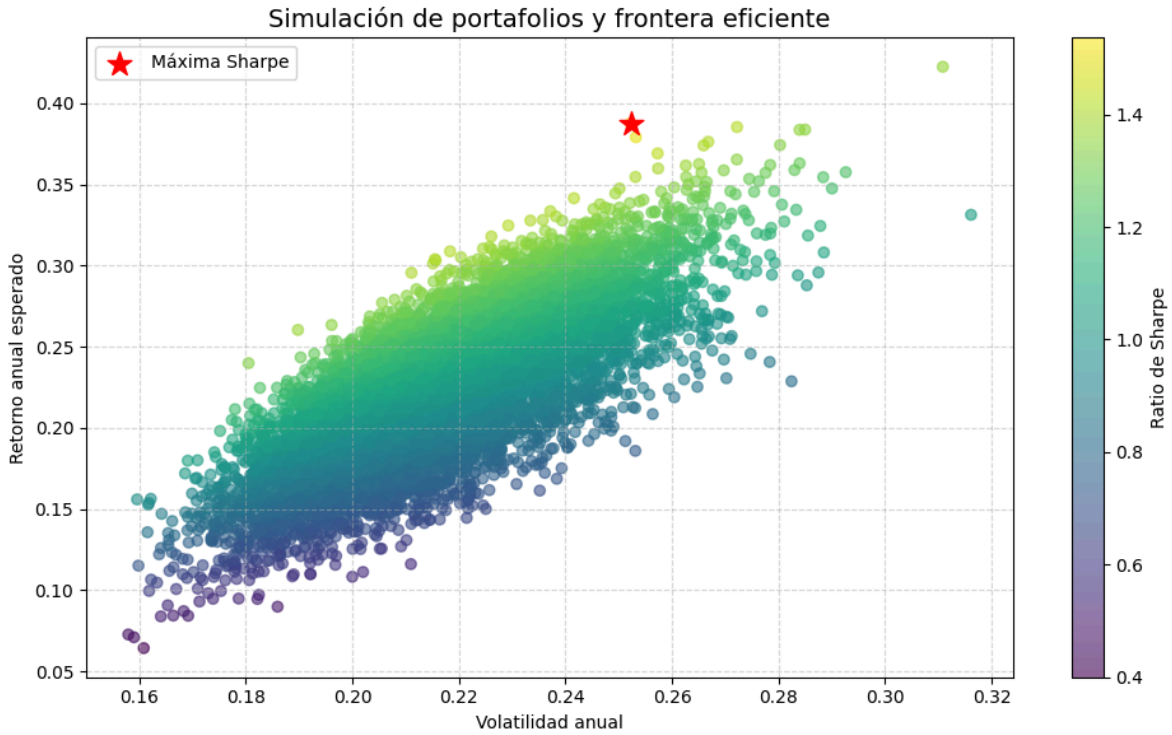
Este enfoque representa una estrategia balanceada pero orientada al rendimiento, ideal para perfiles moderados que buscan **optimizar su rentabilidad sin desprotegerse del riesgo**.

```
# Fijar semilla para reproducibilidad
np.random.seed(42)

# Establecemos número de simulaciones
num_portafolios = 10000

# Calculamos medias y matriz de covarianza
media_retornos = data_returns.mean() * 252        # retorno anualizad
matriz_cov = data_returns.cov() * 252             # covarianza anuali
tickers = data_returns.columns

# Inicializamos listas para guardar resultados
resultados = {
    "Retorno": [],
    "Volatilidad": [],
    "Sharpe": [],
    "Pesos": []
}

# Simulación de portafolios aleatorios
for _ in range(num_portafolios):
    pesos = np.random.random(len(tickers))
    pesos /= np.sum(pesos)  # normalizamos

    retorno = np.dot(pesos, media_retornos)
    volatilidad = np.sqrt(np.dot(pesos.T, np.dot(matriz_cov, pesos)))
    sharpe_ratio = retorno / volatilidad

    resultados["Retorno"].append(retorno)
    resultados["Volatilidad"].append(volatilidad)
    resultados["Sharpe"].append(sharpe_ratio)
```

```
    resultados["Pesos"].append(pesos)

# Convertimos resultados a DataFrame
portafolios = pd.DataFrame(resultados)
portafolios["Pesos"] = portafolios["Pesos"].apply(lambda x: np.round(

# Mostramos primeras filas
portafolios.head()
```

	Retorno	Volatilidad	Sharpe	Pesos
0	0.299130	0.239439	1.249299	[0.0605, 0.1535, 0.1182, 0.0967, 0.0252, 0.025...
1	0.189335	0.210440	0.899708	[0.1904, 0.0486, 0.0416, 0.0419, 0.0696, 0.12,...
2	0.229586	0.199324	1.151824	[0.074, 0.1275, 0.0324, 0.0835, 0.0962, 0.0075...
				[0.0629, 0.0202, 0.1414, 0.0909, 0.0252,

```
# Identificamos el portafolio con mayor Sharpe
idx_max_sharpe = portafolios["Sharpe"].idxmax()
mejor_portafolio = portafolios.loc[idx_max_sharpe]

# Gráfico
plt.figure(figsize=(10, 6))
scatter = plt.scatter(portafolios["Volatilidad"], portafolios["Retorn
                      c=portafolios["Sharpe"], cmap="viridis", alpha=
plt.colorbar(scatter, label="Ratio de Sharpe")

# Marcamos el portafolio óptimo
plt.scatter(mejor_portafolio["Volatilidad"], mejor_portafolio["Retorn
            color="red", marker="*", s=200, label="Máxima Sharpe")

# Estética
plt.title("Simulación de portafolios y frontera eficiente", fontsize=
plt.xlabel("Volatilidad anual")
plt.ylabel("Retorno anual esperado")
plt.legend()
plt.grid(True, linestyle="--", alpha=0.5)
plt.tight_layout()
plt.show()
```




```
# Mostrar métricas clave del portafolio de máxima Sharpe
mejor_portafolio[["Retorno", "Volatilidad", "Sharpe"]]
```

	9477
Retorno	0.387634
Volatilidad	0.252241
Sharpe	1.536762

dtype: object

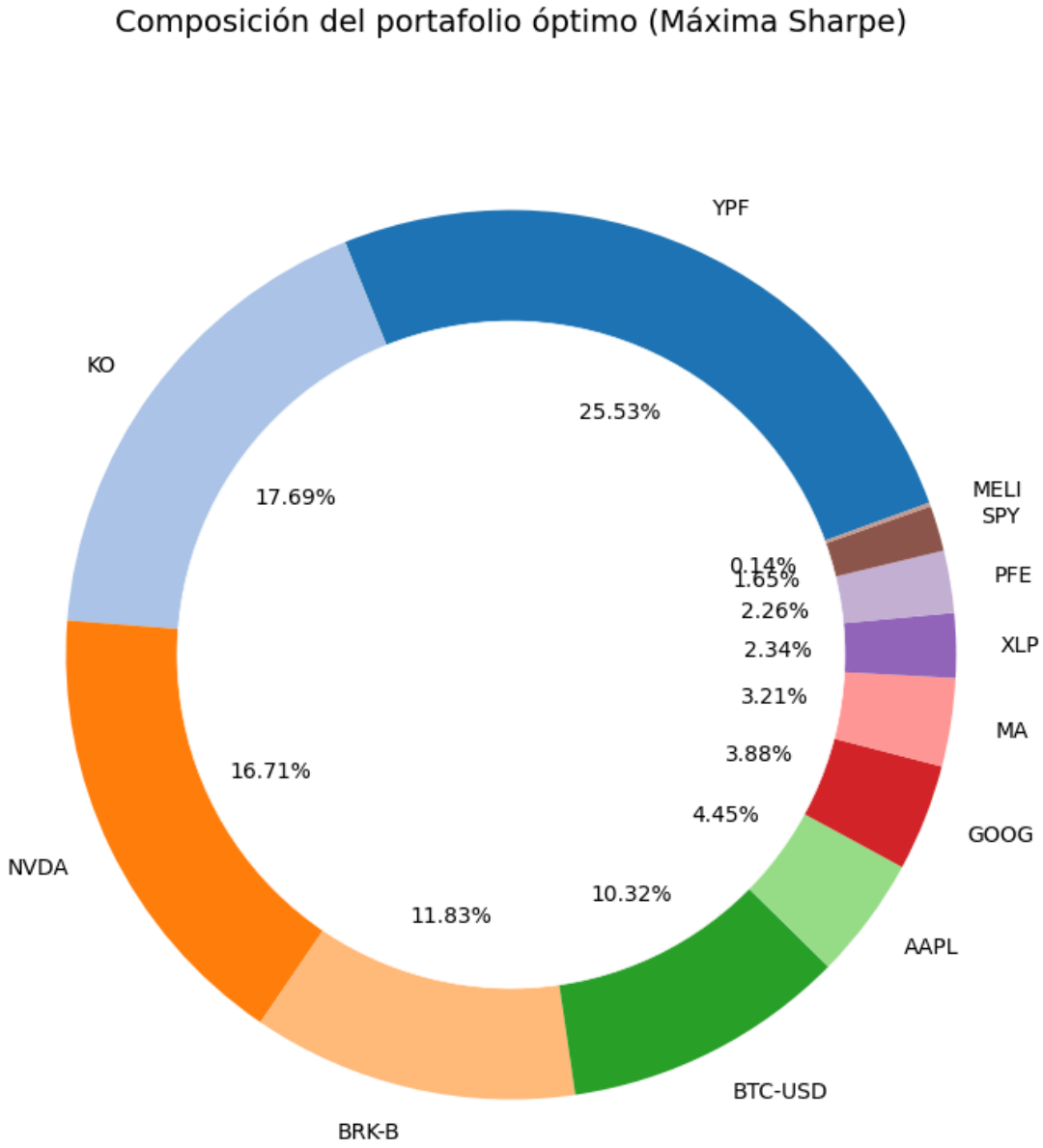
```
# Mostrar los pesos del portafolio óptimo como porcentaje
pesos_optimos = pd.Series(mejor_portafolio["Pesos"], index=tickers)
pesos_optimos = pesos_optimos.sort_values(ascending=False)
pesos_optimos = (pesos_optimos * 100).round(2).astype(str) + '%'

# Mostrar la tabla
pesos_optimos
```

Ticker	0
YPF	25.53%
KO	17.69%
NVDA	16.71%
BRK-B	11.83%
BTC-USD	10.32%
AAPL	4.45%
GOOG	3.88%
MA	3.21%
XLP	2.34%
PFE	2.26%
SPY	1.65%
MELI	0.14%

dtype: object

```
# Datos
pesos_optimos = [0.2553, 0.1769, 0.1671, 0.1183, 0.1032, 0.0445,
                 0.0388, 0.0321, 0.0234, 0.0226, 0.0165, 0.0014]
tickers = ["YPF", "KO", "NVDA", "BRK-B", "BTC-USD", "AAPL",
           "GOOG", "MA", "XLP", "PFE", "SPY", "MELI"]

# Usar paleta tab20 (20 colores únicos)
colors = plt.cm.tab20.colors[:len(tickers)]

# Crear figura
plt.figure(figsize=[10, 8])
plt.style.use("ggplot")
plt.title("Composición del portafolio óptimo (Máxima Sharpe)", fontsi

# Gráfico de torta
wedges, texts, autotexts = plt.pie(
    x=pesos_optimos,
    labels=tickers,
    autopct="%.2f%%",
    startangle=20,
    colors=colors
)

# Donut central blanco
circle = plt.Circle((0, 0), 0.75, color='white')
plt.gca().add_artist(circle)

# Asegura forma circular
```

```
plt.axis=("equal")
plt.tight_layout()
plt.show()
```



El portafolio identificado como el de **máxima eficiencia** dentro de las 10.000 simulaciones presenta un **retorno anual esperado de 38.76%** y una **volatilidad del 25.22%**, con un **ratio de Sharpe de 1.54**, lo que lo posiciona como la mejor combinación retorno-riesgo entre las estrategias evaluadas.

Durante el período completo analizado (**enero 2022 – junio 2025**), este portafolio habría generado un **retorno acumulado total de 242.94%**, lo que implica que una inversión de $1 al inicio se habría convertido en aproximadamente **$3.43** al final del período.

La composición resultante muestra una estrategia diversificada con sesgo al crecimiento, pero manteniendo elementos estabilizadores:

- **Alta ponderación en YPF (25.53%)**, una acción de alto potencial pero también elevada volatilidad, lo que evidencia una apuesta decidida por rendimiento.
- **KO (17.69%) y BRK-B (11.83%)** aportan solidez y estabilidad al portafolio, compensando el riesgo de activos más agresivos.
- **Participación importante de NVDA (16.71%) y BTC-USD (10.32%)**, que refuerzan la expectativa de crecimiento, aunque con riesgo elevado.
- **Tecnológicas como AAPL (4.45%), GOOG (3.88%) y MA (3.21%)** están presentes con pesos intermedios, aportando dinamismo y diversificación.
- **Activos conservadores como XLP (2.34%), PFE (2.26%), SPY (1.65%) y MELI (0.14%)** figuran con menor participación, funcionando como estabilizadores de la cartera.

En conjunto, este portafolio representa una estrategia con **orientación al crecimiento pero diversificada**, adecuada para un inversor de perfil **moderado con tolerancia al riesgo controlado**. Su implementación real debería contemplar un seguimiento constante, análisis de contexto y políticas de rebalanceo periódico.

🧘 Parte 4.2: Identificación del Portafolio de Mínima Volatilidad

En esta sección se analizará una estrategia alternativa al enfoque de máxima eficiencia (Sharpe), priorizando la **estabilidad** del portafolio por sobre la rentabilidad.

El objetivo es identificar el portafolio que presenta la **menor volatilidad anualizada** entre las 10.000 combinaciones simuladas. Esta estrategia puede ser especialmente relevante para inversores con una alta aversión al riesgo o con un horizonte de inversión conservador.

Una vez identificado, se analizará su rendimiento esperado, su composición y se comparará visualmente con el resto de las combinaciones simuladas.

Este enfoque permite evaluar una estrategia basada en la **minimización del riesgo absoluto**, lo cual también puede ser compatible con un perfil de inversor moderado en ciertos contextos de mercado.

```
# Fijar semilla para reproducibilidad
np.random.seed(42)

# Establecer número de simulaciones
num_portafolios = 10000

# Calcular medias y matriz de covarianza
media_retornos = data_returns.mean() * 252          # retorno anualiza
matriz_cov = data_returns.cov() * 252               # covarianza anual
tickers = data_returns.columns                      # nombres de los a

# Inicializar listas para guardar resultados
resultados = {
    "Retorno": [],
    "Volatilidad": [],
    "Sharpe": [],
    "Pesos": []
}

# Simulación de portafolios aleatorios
for _ in range(num_portafolios):
    pesos = np.random.random(len(tickers))
    pesos /= np.sum(pesos)  # normalizar los pesos para que sumen 1

    retorno = np.dot(pesos, media_retornos)
    volatilidad = np.sqrt(np.dot(pesos.T, np.dot(matriz_cov, pesos)))
    sharpe_ratio = retorno / volatilidad

    resultados["Retorno"].append(retorno)
    resultados["Volatilidad"].append(volatilidad)
    resultados["Sharpe"].append(sharpe_ratio)
    resultados["Pesos"].append(pesos)

# Convertir a DataFrame y redondear los pesos
portafolios = pd.DataFrame(resultados)
portafolios["Pesos"] = portafolios["Pesos"].apply(lambda x: np.round(

# Mostrar las primeras filas
portafolios.head()
```

	Retorno	Volatilidad	Sharpe	Pesos
0	0.299130	0.239439	1.249299	[0.0605, 0.1535, 0.1182, 0.0967, 0.0252, 0.025...
1	0.189335	0.210440	0.899708	[0.1904, 0.0486, 0.0416, 0.0419, 0.0696, 0.12,...
2	0.229586	0.199324	1.151824	[0.074, 0.1275, 0.0324, 0.0835, 0.0962, 0.0075...
[illegible]	[illegible]	[illegible]	[illegible]	[0.0629, 0.0202, 0.1414, 0.0909, 0.0252,

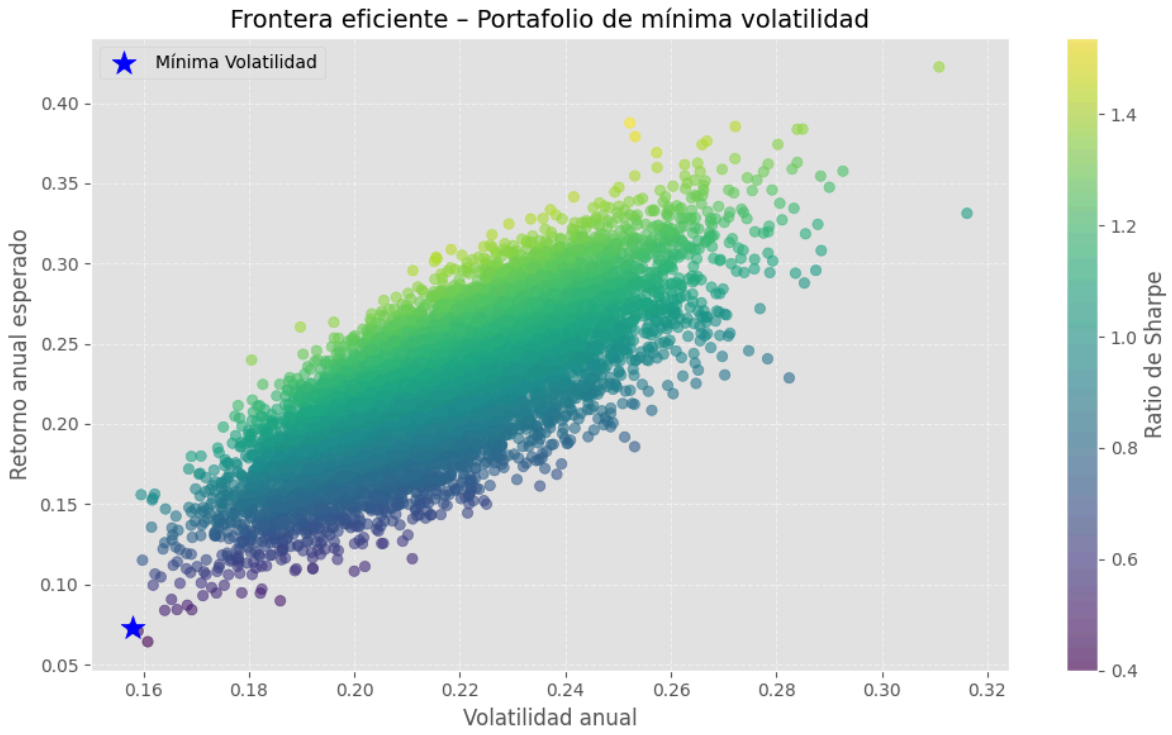
```
# Identificamos el portafolio con menor volatilidad
idx_min_vol = portafolios["Volatilidad"].idxmin()
min_vol_portafolio = portafolios.loc[idx_min_vol]
```

```
# Mostramos sus métricas principales
min_vol_portafolio[["Retorno", "Volatilidad", "Sharpe"]]
```

	6185
Retorno	0.072667
Volatilidad	0.157928
Sharpe	0.460128

dtype: object

```
# Gráfico de portafolios simulados
plt.figure(figsize=(10, 6))
scatter = plt.scatter(portafolios["Volatilidad"], portafolios["Retorn
                      c=portafolios["Sharpe"], cmap="viridis", alpha=
plt.colorbar(scatter, label="Ratio de Sharpe")

# Marcamos el portafolio de mínima volatilidad
plt.scatter(min_vol_portafolio["Volatilidad"], min_vol_portafolio["Re
            color="blue", marker="*", s=200, label="Mínima Volatilida

# Estética
plt.title("Frontera eficiente — Portafolio de mínima volatilidad", fo
plt.xlabel("Volatilidad anual")
plt.ylabel("Retorno anual esperado")
plt.legend()
plt.grid(True, linestyle="--", alpha=0.5)
plt.tight_layout()
plt.show()
```



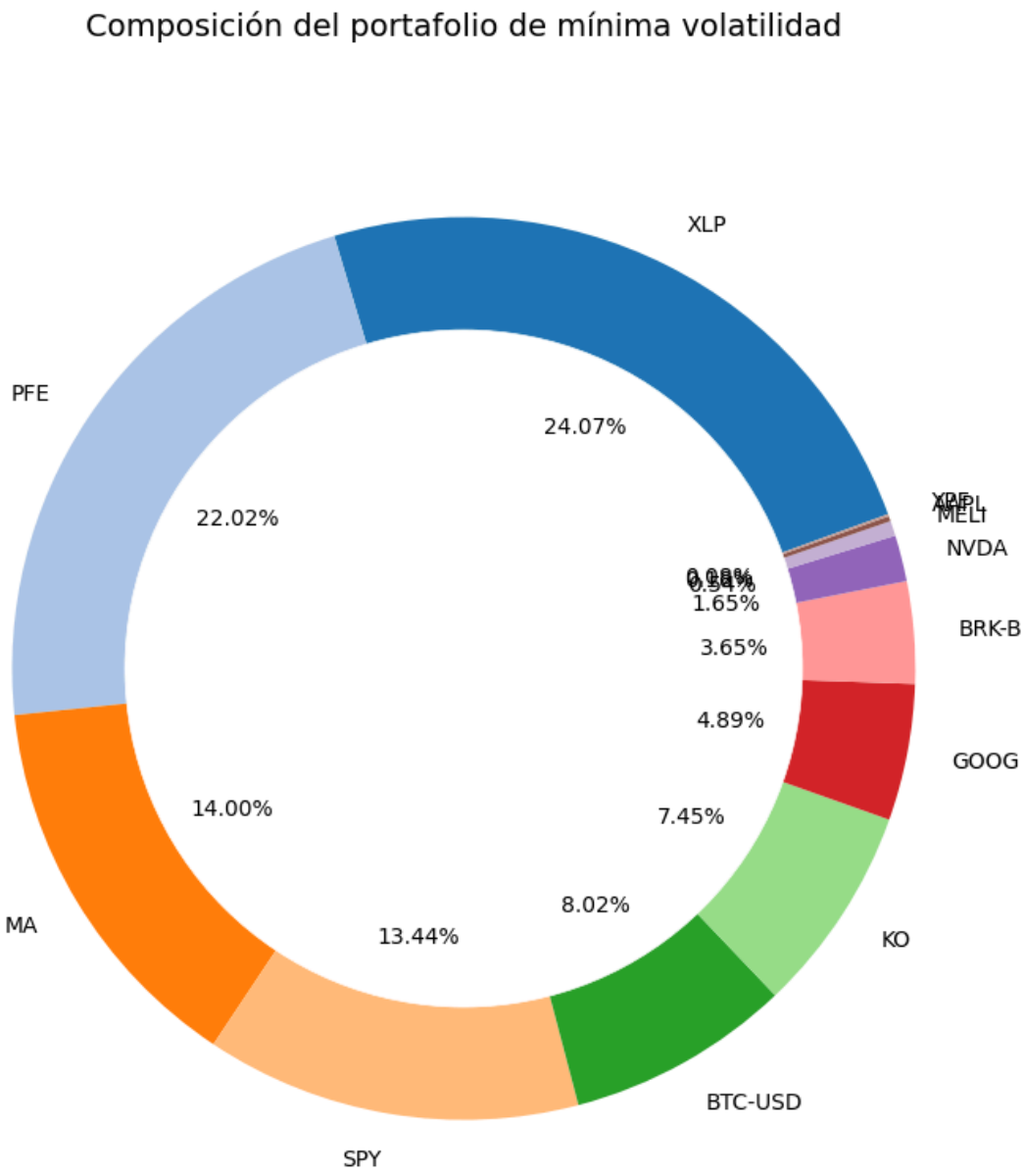

```
# Extraer y ordenar los pesos del portafolio de mínima volatilidad
pesos_min_vol = pd.Series(min_vol_portafolio["Pesos"], index=tickers)
pesos_min_vol = pesos_min_vol.sort_values(ascending=False)
pesos_min_vol = (pesos_min_vol * 100).round(2).astype(str) + '%'

# Mostrar la tabla de pesos
pesos_min_vol
```

Ticker	0
XLP	24.07%
PFE	22.02%
MA	14.0%
SPY	13.44%
BTC-USD	8.02%
KO	7.45%
GOOG	4.89%
BRK-B	3.65%
NVDA	1.65%
MELI	0.54%
AAPL	0.18%
YPF	0.08%

dtype: object

```
# Datos reales del portafolio de mínima volatilidad
pesos = [0.2407, 0.2202, 0.14, 0.1344, 0.0802, 0.0745,
         0.0489, 0.0365, 0.0165, 0.0054, 0.0018, 0.0008]

tickers_min_vol = ["XLP", "PFE", "MA", "SPY", "BTC-USD", "KO",
                   "GOOG", "BRK-B", "NVDA", "MELI", "AAPL", "YPF"]

# Paleta de colores (tab20)
colors = plt.cm.tab20.colors[:len(tickers_min_vol)]

# Crear figura
plt.figure(figsize=[10, 8])
plt.style.use("ggplot")
plt.title("Composición del portafolio de mínima volatilidad", fontsiz

# Gráfico de torta
wedges, texts, autotexts = plt.pie(
    x=pesos,
    labels=tickers_min_vol,
    autopct="%.2f%%",
    startangle=20,
    colors=colors
)

# Donut blanco en el centro
circle = plt.Circle((0, 0), 0.75, color='white')
plt.gca().add_artist(circle)
```

```
# Asegura forma circular
plt.axis=("equal")
plt.tight_layout()
plt.show()
```



El portafolio identificado como el de **mínima volatilidad** dentro de las 10.000 simulaciones presenta una **volatilidad anualizada de 15.79%**, lo que lo convierte en la estrategia más estable desde el punto de vista del riesgo absoluto.

A pesar de priorizar la estabilidad, el portafolio mantiene un **retorno anual esperado de 7.27%** y un **ratio de Sharpe de 0.46**, lo que sugiere una relación riesgo-retorno moderada, acorde a las necesidades de un inversor con perfil conservador o moderado.

Durante el período completo analizado (**enero 2022 – junio 2025**), este portafolio habría generado un **retorno acumulado total de 23.15%**, lo que implica que una inversión de $1 al inicio se habría convertido en aproximadamente **$1.23** al final del período.

La composición de este portafolio muestra una clara inclinación hacia activos de comportamiento más estable y correlaciones favorables:

- **Alta ponderación en XLP (24.07%) y PFE (22.02%)**, ambos activos considerados defensivos, que contribuyen directamente a reducir la volatilidad total.
- **MA (14.00%) y SPY (13.44%)** aportan equilibrio entre estabilidad y diversificación de mercado.
- **Presencia moderada de BTC-USD (8.02%)**, que, si bien es volátil, puede mejorar el perfil de riesgo-retorno por su baja correlación con otros activos tradicionales.
- **KO (7.45%) y GOOG (4.89%)** complementan la estructura, ofreciendo calidad y diversificación.
- **Exposición marginal a BRK-B, NVDA, MELI, AAPL y YPF**, todos con pesos inferiores al 4%, lo que evidencia un ajuste fino en la asignación para contener el riesgo total del portafolio.

En conjunto, este portafolio representa una **estrategia conservadora y balanceada**, coherente con un perfil moderado que prioriza la estabilidad sin resignar completamente las oportunidades de rendimiento. Su implementación real debería contemplar seguimiento activo y políticas de rebalanceo periódico según las condiciones del mercado.

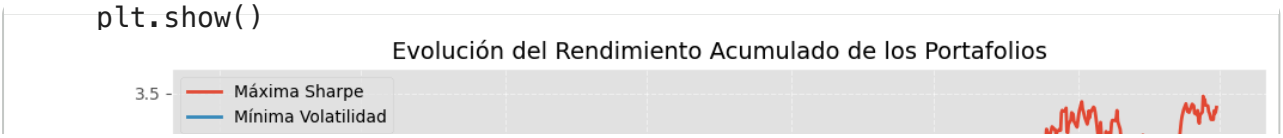
📈 Parte 4.3: Comparación de la Evolución del Rendimiento Acumulado

A continuación se presenta un gráfico comparativo entre los portafolios de **Máxima Sharpe** y **Mínima Volatilidad**, que permite observar visualmente cómo habría evolucionado una inversión en cada estrategia a lo largo del tiempo (2022–2025).

El análisis se basa en la simulación de rendimiento diario acumulado, partiendo de una inversión inicial equivalente a $1.

```
# Convertir pesos definitivos a arrays de numpy
pesos_sharpe_array = np.array([
    0.2553,  # YPF
    0.1671,  # NVDA
    0.1769,  # KO
    0.1032,  # BTC-USD
    0.0234,  # XLP
    0.0165,  # SPY
    0.0226,  # PFE
    0.0014,  # MELI
    0.0445,  # AAPL
    0.1183,  # BRK-B
    0.0321,  # MA
    0.0388   # GOOG
])

pesos_vol_array = np.array([
    0.0008,  # YPF
    0.0165,  # NVDA
    0.0745,  # KO
    0.0802,  # BTC-USD
    0.2407,  # XLP
    0.1344,  # SPY
    0.2202,  # PFE
    0.0054,  # MELI
    0.0018,  # AAPL
    0.0365,  # BRK-B
    0.14,    # MA
    0.0489   # GOOG
])

# Reordenar columnas de retornos para que coincidan con el orden de
orden_tickers = ["YPF", "NVDA", "KO", "BTC-USD", "XLP", "SPY",
                 "PFE", "MELI", "AAPL", "BRK-B", "MA", "GOOG"]
data_returns_ordenado = data_returns[orden_tickers]

# Retornos diarios de cada portafolio
retorno_diario_sharpe = data_returns_ordenado.dot(pesos_sharpe_array
retorno_diario_vol = data_returns_ordenado.dot(pesos_vol_array)

# Retorno acumulado (multiplicativo)
retorno_acum_sharpe = (1 + retorno_diario_sharpe).cumprod()
retorno_acum_vol = (1 + retorno_diario_vol).cumprod()

# Gráfico final
plt.figure(figsize=(10, 6))
plt.plot(retorno_acum_sharpe, label="Máxima Sharpe", linewidth=2)
plt.plot(retorno_acum_vol, label="Mínima Volatilidad", linewidth=2)
plt.title("Evolución del Rendimiento Acumulado de los Portafolios",
plt.xlabel("Fecha")
plt.ylabel("Crecimiento del capital (Base = 1)")
plt.legend()
plt.grid(True, linestyle="--", alpha=0.6)
plt.tight_layout()
```

```
plt.show()
```


Evolución del Rendimiento Acumulado de los Portafolios
3.5
Máxima Sharpe
Mínima Volatilidad