

# Backtester Quant v5.0: Un Sistema Integral para el Análisis y la Gestión Avanzada de Carteras de Inversión

Informe Técnico y de Arquitectura

Autor: Álvaro Martínez Gamo

11 de junio de 2025

## Recursos en Línea

Repositorio en GitHub: [trentisiete/backtester\\_fondos](https://github.com/trentisiete/backtester_fondos)

Aplicación en Producción: [diosfamiliafe.streamlit.app](https://diosfamiliafe.streamlit.app)

*Una herramienta clave para la toma de decisiones estratégicas en los mercados financieros.*

# Índice

<b>1. Resumen Ejecutivo</b>	<b>2</b>
<b>2. Arquitectura del Sistema</b>	<b>3</b>
2.1. Estructura de Ficheros . . . . .	3
<b>3. Funcionalidades Clave y Análisis de Módulos</b>	<b>5</b>
3.1. Gestión de Datos y Pesos (Módulo <code>src/utils.py</code> ) . . . . .	5
3.2. Motor de Backtesting (Módulo <code>src/simulation.py</code> ) . . . . .	5
3.3. Modelos Avanzados y Simulaciones (Módulo <code>src/models.py</code> ) . . . . .	6
3.4. Optimización de Carteras (Módulo <code>src/optimization.py</code> ) . . . . .	6
<b>4. Hitos y Reconocimientos: Del Aula al Mercado Real</b>	<b>7</b>
4.1. Liderazgo en la Liga de Inversores de la UAM . . . . .	7
4.2. Entrevista en Radio Intereconomía . . . . .	7
<b>5. Futuras Líneas de Desarrollo</b>	<b>8</b>
5.1. Integración de Aprendizaje por Refuerzo (Reinforcement Learning) . . . . .	8
5.2. Modelos de Backtesting Avanzados . . . . .	8
<b>6. Conclusión</b>	<b>8</b>
<b>A. Apéndice A: Fórmulas Matemáticas Clave</b>	<b>10</b>
A.1. Rentabilidad Total . . . . .	10
A.2. Rentabilidad Anualizada (CAGR) . . . . .	10
A.3. Volatilidad Anualizada . . . . .	10
A.4. Máximo Drawdown . . . . .	10
A.5. Ratio de Sharpe . . . . .	10
A.6. Ratio de Sortino . . . . .	10
A.7. Ratio de Diversificación . . . . .	11
A.8. Beta ( $\beta$ ) . . . . .	11
A.9. Alpha ( $\alpha$ ) . . . . .	11
A.10. Contribución Porcentual al Riesgo . . . . .	11

# 1. Resumen Ejecutivo

El presente informe detalla la arquitectura, funcionalidades y logros del **Backtester Quant v5.0**, un sistema de software avanzado desarrollado en Python con la librería Streamlit. Esta herramienta ha sido diseñada para ofrecer a inversores y gestores de carteras un entorno robusto y completo para el análisis cuantitativo, la optimización y la gestión de riesgos. Es un proyecto vivo y **en constante desarrollo**, enfocado actualmente en la monitorización exhaustiva de carteras.

El sistema se destaca por su capacidad de realizar backtesting con pesos de cartera tanto fijos como dinámicos, permitiendo simular con alta fidelidad estrategias de inversión pasivas y activas. Además, integra un conjunto exhaustivo de métricas de rendimiento y riesgo, incluyendo análisis de correlación, contribución al riesgo, pruebas de estrés, escenarios históricos, optimización de carteras mediante la Frontera Eficiente y proyecciones estocásticas con simulaciones de Montecarlo.

Este proyecto no es solo un ejercicio académico; es una herramienta probada en batalla que ha sido fundamental para **alcanzar y mantener la primera posición en la prestigiosa Liga de Inversores de la Universidad Autónoma de Madrid (UAM)**. El éxito y la innovación del proyecto han sido reconocidos externamente, culminando en una **entrevista en el programa "Tu Dinero Nunca Duerme" de Radio Intereconomía**, donde se discutió su enfoque y su aplicación práctica en el mundo de la inversión.

Las futuras líneas de desarrollo se centrarán en la incorporación de modelos de *Reinforcement Learning* (Aprendizaje por Refuerzo) para la creación de estrategias de trading adaptativas y la expansión de los procesos de backtesting para incluir modelos más complejos, como los basados en eventos.

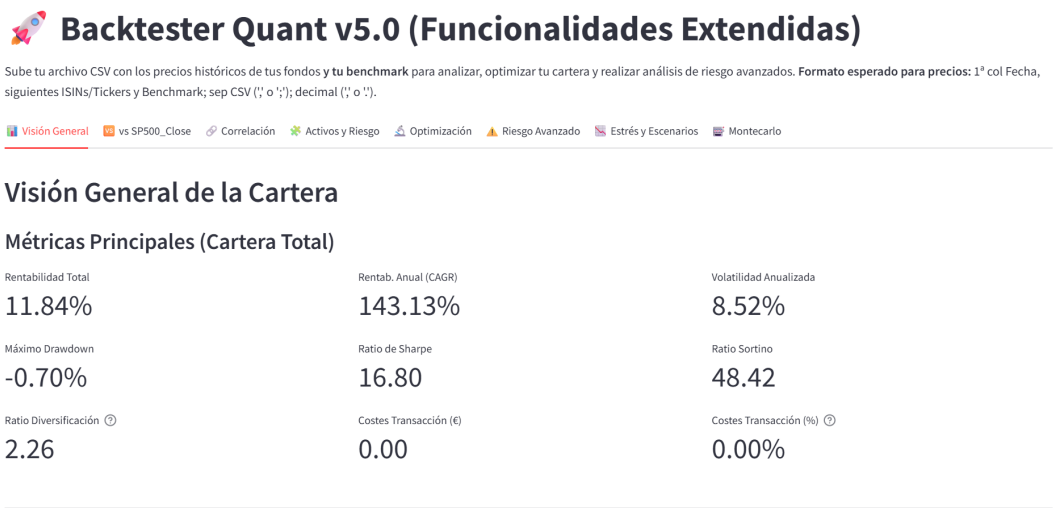


Figura 1: Interfaz de usuario principal del Backtester Quant v5.0.

## 2. Arquitectura del Sistema

El proyecto está diseñado siguiendo un enfoque modular, lo que facilita su mantenimiento, escalabilidad y legibilidad. La lógica de negocio está separada de la interfaz de usuario, permitiendo que cada componente se desarrolle y se pruebe de forma independiente.

### 2.1. Estructura de Ficheros

El código se organiza en un directorio principal que contiene la aplicación Streamlit y una subcarpeta `src/` que alberga los módulos de lógica.

- **app.py**: Es el punto de entrada y el orquestador principal de la aplicación. Gestiona la interfaz de usuario con Streamlit, el estado de la sesión, la interacción con el usuario (carga de archivos, configuración de parámetros) y la presentación de resultados.
- **/src**: Directorio que contiene los módulos especializados.
  - **constants.py**: Define constantes globales, como la tasa libre de riesgo por defecto y los diccionarios de escenarios históricos, centralizando la configuración.
  - **utils.py**: Contiene funciones de utilidad para la carga y el preprocesamiento de datos, tanto de los precios históricos como de los pesos dinámicos. Incluye una lógica robusta para manejar diferentes formatos de CSV.
  - **simulation.py**: Es el corazón del motor de backtesting. Implementa la simulación de la cartera a lo largo del tiempo, manejando el rebalanceo, los pesos dinámicos y los costes de transacción.
  - **analysis.py**: Proporciona todas las funciones para el cálculo de métricas de rendimiento, tanto para la cartera total como para los activos individuales (CAGR, Sharpe, Sortino, Drawdown, Beta, Alpha, etc.).
  - **models.py**: Implementa los modelos cuantitativos avanzados para el análisis de riesgo (VaR Histórico y Paramétrico, Expected Shortfall), las pruebas de estrés y la simulación de Montecarlo.
  - **optimization.py**: Contiene la lógica para la optimización de carteras utilizando la librería `PyPortfolioOpt`. Calcula la Frontera Eficiente, la cartera de Mínima Varianza (MVP) y la de Máximo Ratio de Sharpe (MSR), aplicando las restricciones definidas por el usuario.
  - **visualization.py**: Centraliza todas las funciones de generación de gráficos (usando Plotly, Matplotlib y Seaborn) para mantener el código de **app.py** más limpio.
- **requirements.txt**: Lista todas las dependencias de Python necesarias para ejecutar el proyecto, permitiendo una fácil instalación en cualquier entorno.
- **sp500.csv** y **sp500\_downloader.py**: Un archivo de datos de ejemplo que contiene los precios históricos del S&P 500 y un script para descargarlos y actualizarlos desde diversas fuentes.

- `README.md`: Documentación exhaustiva sobre el proyecto, sus funcionalidades y cómo utilizarlo.

Esta arquitectura modular asegura que cada parte del sistema tenga una responsabilidad única, adhiriéndose al principio de Responsabilidad Única (SRP) del diseño de software.

### 3. Funcionalidades Clave y Análisis de Módulos

El sistema ofrece un abanico completo de herramientas para el inversor cuantitativo. A continuación, se detallan las funcionalidades más importantes y los módulos que las implementan.

#### 3.1. Gestión de Datos y Pesos (Módulo `src/utils.py`)

La base de cualquier análisis cuantitativo reside en la calidad de los datos. El módulo de utilidades está diseñado para ser extremadamente flexible.

- **Carga de Datos de Precios:** La función `load_data` es capaz de procesar archivos CSV con diferentes delimitadores (',' o ';') y separadores decimales ('.' o ','). Realiza una limpieza automática, rellenando valores nulos mediante propagación hacia adelante y hacia atrás (`ffill`, `bfill`) para asegurar la continuidad de las series temporales.
- **Gestión de Pesos Dinámicos (¡Funcionalidad Estrella!):** La función `load_dynamic_weights` es una de las características más potentes de la v5.0. Permite al usuario subir un archivo que especifica cómo deben cambiar los pesos de la cartera en fechas concretas. Esto es crucial para simular estrategias de gestión activa, como el *market timing* o los ajustes tácticos de activos, en lugar de una simple estrategia de comprar y mantener con rebalanceo. El sistema normaliza automáticamente los pesos para que sumen 1.

#### 3.2. Motor de Backtesting (Módulo `src/simulation.py`)

El motor de simulación es el núcleo del sistema. La función `run_backtest` reconstruye la evolución del valor de la cartera día a día.

- **Inicialización:** Comienza con una inversión inicial y calcula el número de participaciones de cada activo según los pesos iniciales.
- **Iteración Diaria:** Para cada día del período de backtest, calcula el valor de la cartera en función de la variación de los precios de los activos.
- **Lógica de Rebalanceo:**
  - **Basado en Frecuencia:** Si se establece una frecuencia (mensual, trimestral, anual), el sistema reajusta las participaciones para volver a los pesos objetivo al inicio de cada período.
  - **Basado en Pesos Dinámicos:** El sistema detecta si para la fecha actual existe una nueva asignación de pesos en el archivo del usuario. Si es así, ejecuta un rebalanceo a estos nuevos pesos.
- **Costes de Transacción:** Se aplica un coste configurable (en puntos básicos) sobre el valor de las operaciones en cada rebalanceo, lo que añade un mayor realismo a la simulación.

### 3.3. Modelos Avanzados y Simulaciones (Módulo `src/models.py`)

Para una gestión de riesgos proactiva, el sistema incluye modelos predictivos y de estrés.

- **Valor en Riesgo (VaR) y Expected Shortfall (ES):** Cuantifican la pérdida máxima esperada en un horizonte temporal (diario) con un nivel de confianza determinado (e.g., 99 %).
- **Pruebas de Estrés:**
  - **Shock Hipotético:** Permite al usuario simular el impacto inmediato de una caída de mercado (e.g., -20 %) en el valor de la cartera.
  - **Escenarios Históricos:** Simula el comportamiento de la cartera actual durante crisis pasadas reales.
- **Simulación de Montecarlo:** Proyecta miles de posibles trayectorias futuras para el valor de la cartera, proporcionando una distribución de resultados probables.

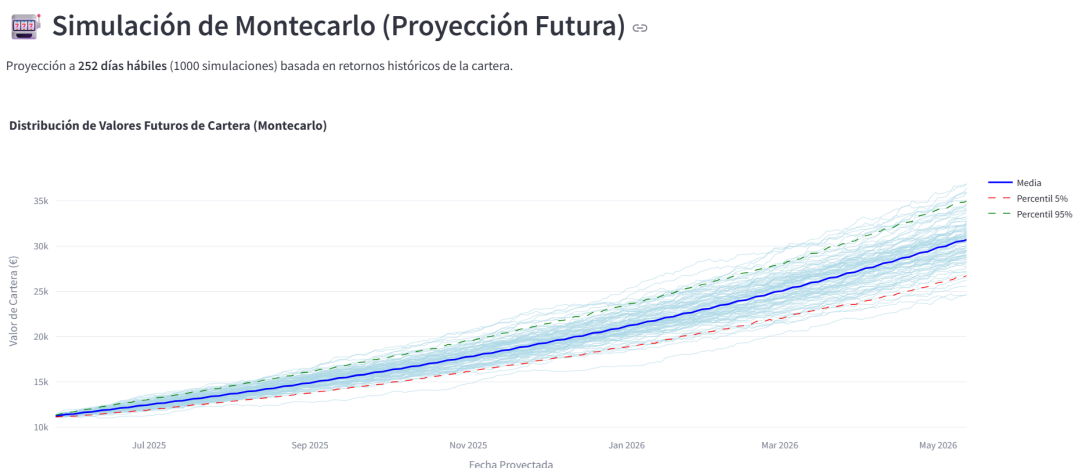


Figura 2: Ejemplo de una proyección de Montecarlo generada por la aplicación.

### 3.4. Optimización de Carteras (Módulo `src/optimization.py`)

Utilizando la Teoría Moderna de Carteras (MPT), este módulo ayuda a encontrar la asignación de activos óptima.

- **Frontera Eficiente:** Se calcula y grafica el conjunto de carteras que ofrecen el máximo retorno para un nivel de riesgo dado.
- **Carteras Óptimas:** Se identifican la **Cartera de Mínima Varianza (MVP)** y la **Cartera de Máximo Ratio de Sharpe (MSR)**.
- **Restricciones:** Se pueden aplicar restricciones de inversión, como limitar la exposición total a Renta Fija o a activos Monetarios.

## 4. Hitos y Reconocimientos: Del Aula al Mercado Real

El desarrollo del Backtester Quant v5.0 ha estado intrínsecamente ligado a su aplicación en un entorno competitivo real, lo que ha validado su eficacia y ha guiado su evolución.

### 4.1. Liderazgo en la Liga de Inversores de la UAM

La principal prueba de fuego para esta herramienta ha sido su uso como sistema central para la toma de decisiones en la **Liga de Inversores de la Universidad Autónoma de Madrid**. Gracias a la capacidad del sistema para:

1. **Monitorizar el Riesgo:** Evaluar en tiempo real la contribución al riesgo de cada activo y el drawdown potencial de la cartera.
2. **Optimizar la Asignación:** Identificar asignaciones más eficientes que las propuestas inicialmente, comparando nuestra cartera actual con las carteras de Mínima Varianza y Máximo Sharpe.
3. **Simular Estrategias Activas:** La introducción de los pesos dinámicos nos permitió testear y ejecutar cambios tácticos en la cartera, adaptándonos a las condiciones cambiantes del mercado.

Estos factores nos proporcionaron una ventaja competitiva decisiva, permitiéndonos alcanzar y consolidar el **primer puesto** en la competición. La herramienta transformó la toma de decisiones, pasando de un enfoque puramente discrecional a uno sistemático y basado en datos.

### 4.2. Entrevista en Radio Intereconomía

El éxito y la innovación del proyecto captaron la atención de medios especializados. Fuimos invitados a participar en el programa "**Tu Dinero Nunca Duerme**" de **Radio Intereconomía**, una de las emisoras líderes en información económica y financiera en España.

Durante la entrevista, tuvimos la oportunidad de exponer la filosofía detrás del Backtester Quant: democratizar el acceso a herramientas de análisis cuantitativo sofisticadas, tradicionalmente reservadas para grandes instituciones. Se destacó cómo un enfoque disciplinado y apoyado por tecnología puede mejorar significativamente los resultados de inversión, un mensaje que resonó con la audiencia del programa y que validó públicamente la relevancia y el potencial de nuestro trabajo.



## 5. Futuras Líneas de Desarrollo

Es importante destacar que el Backtester Quant v5.0 es un proyecto en **proceso de mejora continua**. Las siguientes funcionalidades representan las próximas líneas estratégicas de desarrollo para transformar la herramienta de un sistema de monitorización a una plataforma de generación de estrategias.

### 5.1. Integración de Aprendizaje por Refuerzo (Reinforcement Learning)

El siguiente gran salto cualitativo será la integración de modelos de **Aprendizaje por Refuerzo (RL)**. El objetivo es desarrollar agentes inteligentes capaces de aprender estrategias de gestión de cartera de forma autónoma.

- **Agente de Trading:** El agente de RL aprenderá a tomar decisiones de compra, venta o mantenimiento de activos basándose en el estado del mercado.
- **Función de Recompensa:** La función de recompensa del agente se diseñará para optimizar métricas como el Ratio de Sharpe o para minimizar el drawdown, yendo más allá de la simple maximización del retorno.
- **Entorno de Simulación:** El backtester actual servirá como el entorno de entrenamiento perfecto para que el agente interactúe y aprenda de miles de simulaciones de mercado.

### 5.2. Modelos de Backtesting Avanzados

Para aumentar aún más el realismo, se planea implementar un **motor de backtesting basado en eventos**. A diferencia del enfoque actual (vectorial), un sistema basado en eventos procesa los datos secuencialmente, lo que permite:

- **Simulación de Órdenes más Realista:** Modelar con mayor precisión el deslizamiento (*slippage*) y las comisiones por tipo de orden.
- **Estrategias Intradía:** Abrir la puerta al análisis de estrategias de alta frecuencia.
- **Mayor Flexibilidad:** Facilitar la implementación de estrategias complejas que dependen de la llegada de nueva información.

## 6. Conclusión

El Backtester Quant v5.0 representa una solución integral y potente para la gestión de carteras de inversión. Su diseño modular, su amplio conjunto de funcionalidades y su probada eficacia en un entorno competitivo real lo posicionan como una herramienta de gran valor. Desde el análisis fundamental de métricas de riesgo y retorno hasta la optimización avanzada y la simulación de escenarios, el sistema proporciona a los usuarios los insights necesarios para tomar decisiones de inversión informadas y estratégicas. Los éxitos obtenidos y el reconocimiento mediático son un testimonio de su robustez, y las futuras líneas de desarrollo prometen llevar sus capacidades al siguiente nivel mediante la integración de inteligencia artificial.

Puede explorar el código fuente y la aplicación en producción en los siguientes enlaces:

- **Repositorio en GitHub:** [https://github.com/trentisiete/backtester\\_fondos](https://github.com/trentisiete/backtester_fondos)
- **Aplicación Demo:** <https://diosfamiliafe.streamlit.app/>

## A. Apéndice A: Fórmulas Matemáticas Clave

A continuación se presentan las explicaciones y fórmulas para las métricas clave calculadas por el sistema, extraídas de la documentación del proyecto.

### A.1. Rentabilidad Total

El porcentaje de crecimiento acumulado de la cartera desde el inicio hasta el fin del backtest.

$$\text{Rentabilidad Total} = \left( \frac{\text{Valor Final} - \text{Valor Inicial}}{\text{Valor Inicial}} \right)$$

### A.2. Rentabilidad Anualizada (CAGR)

La tasa de crecimiento promedio anual compuesta. Es esencial para comparar el rendimiento de inversiones de diferente duración, ya que normaliza el retorno a una base anual.

$$\text{CAGR} = \left[ \left( \frac{\text{Valor Final}}{\text{Valor Inicial}} \right)^{\frac{1}{\text{Nº de Años}}} - 1 \right]$$

### A.3. Volatilidad Anualizada

Mide la dispersión de los retornos de la cartera, indicando cuánto fluctúa su valor. Es una medida del riesgo, calculada como la desviación estándar de los retornos diarios y anualizada.

$$\text{Volatilidad Anualizada} = \text{Desv. Estándar de Retornos Diarios} \times \sqrt{252}$$

### A.4. Máximo Drawdown

La mayor caída porcentual desde un pico hasta un valle. Mide la peor pérdida potencial que se habría experimentado.

$$\text{Drawdown}_t = \frac{\text{Valor Cartera}_t - \text{Pico Anterior}_t}{\text{Pico Anterior}_t}, \quad \text{Máximo Drawdown} = \min(\text{Drawdown}_t)$$

### A.5. Ratio de Sharpe

Mide el exceso de retorno de la cartera por unidad de riesgo (volatilidad). Cuanto mayor, mejor es el retorno ajustado al riesgo.

$$\text{Ratio de Sharpe} = \frac{\text{CAGR} - \text{Tasa Libre de Riesgo}}{\text{Volatilidad Anualizada}}$$

### A.6. Ratio de Sortino

Similar al Sharpe, pero solo penaliza la volatilidad a la baja (retornos por debajo de un umbral), lo que lo hace útil para inversores que se preocupan principalmente por el riesgo de pérdidas.

$$\text{Ratio Sortino} = \frac{\text{CAGR} - \text{Tasa Mínima Aceptable}}{\text{Desviación a la Baja Anualizada}}$$

## A.7. Ratio de Diversificación

Mide cuánto la diversificación ha reducido la volatilidad de la cartera. Un valor  $\geq 1$  indica una buena diversificación.

$$\text{Ratio de Diversificación} = \frac{\sum_{i=1}^N (w_i \times \sigma_i)}{\sigma_p}$$

Donde  $w_i$  es el peso del activo  $i$ ,  $\sigma_i$  es su volatilidad anualizada, y  $\sigma_p$  es la volatilidad anualizada de la cartera.

## A.8. Beta ( $\beta$ )

Mide la sensibilidad de la cartera a los movimientos del benchmark.  $\beta > 1$  implica mayor volatilidad que el mercado;  $\beta < 1$  implica menor volatilidad.

$$\beta = \frac{\text{Cov}(\text{Retornos Cartera}, \text{Retornos Benchmark})}{\text{Var}(\text{Retornos Benchmark})}$$

## A.9. Alpha ( $\alpha$ )

El retorno "excedente" de la cartera en comparación con lo que se esperaría dado su Beta (según el modelo CAPM). Un Alpha positivo indica que se ha superado al mercado ajustado por riesgo.

$$\alpha = \text{CAGR}_{\text{Cartera}} - (\text{Tasa Libre de Riesgo} + \beta \times (\text{CAGR}_{\text{Benchmark}} - \text{Tasa Libre de Riesgo}))$$

## A.10. Contribución Porcentual al Riesgo

Descompone la volatilidad total para atribuir un porcentaje a cada activo. La contribución marginal al riesgo (MCTR) de un activo  $i$  a la volatilidad de la cartera  $\sigma_p$  es:

$$\text{MCTR}_i = \frac{\partial \sigma_p}{\partial w_i} = \frac{(\Sigma w)_i}{\sigma_p}$$

La contribución porcentual al riesgo de cada activo es entonces:

$$\text{Contribución Porcentual}_i = \frac{w_i \times \text{MCTR}_i}{\sigma_p}$$