

# **Capstone Proyecto de IA**

**2025 - Semestre 2**

## **Reporte Técnico–Experimental**

**Profesor:**

Rodrigo López Aguilar

**Estudiantes:**

Consuelo Diaz

Rafael Guevara

Daniel Alcantar

José Klarian

Rodrigo Erazo

# Contexto y definición técnica del problema

## Organización y proceso.

El trabajo se enmarca en un *toolkit* experimental orientado al análisis cuantitativo y a la construcción de estrategias sobre portafolios multi-activo. El proceso implementa un pipeline de ciencia de datos para: adquisición y preparación de series temporales, EDA, modelamiento (forecasting), optimización de portafolio (MVO) y evaluación contra baselines.

## Definición del problema desde ciencia de datos.

El problema se plantea como una tarea compuesta:

- **Forecasting** de retornos (estimación de retorno esperado  $\mu$ ) a partir de información histórica.
- **Optimización** de portafolio para asignación de pesos (ej., maximización de Sharpe).
- **Aprendizaje por refuerzo (RL)** como enfoque alternativo de control secuencial (documentado; módulo deshabilitado en esta ejecución).

## Variables objetivo.

Las estrategias/modelos se evalúan mediante métricas riesgo/retorno:

- Retorno total, CAGR (aprox.), Volatilidad anual, Sharpe, Max Drawdown, Calmar.

## Variables predictoras.

El modelamiento utiliza retornos históricos e indicadores técnicos. El universo de activos corresponde a MAG7: AAPL, AMZN, GOOGL, META, MSFT, NVDA, TSLA.

## Supuestos y restricciones.

- Se utiliza split cronológico para evitar *data leakage*.
- Se trabaja con datos diarios (sin intradía).
- Se consideran costos de transacción/slippage simplificados (por defecto 0).
- Se utiliza una fuente gratuita ([yfinance](#)), por lo que pueden existir inconsistencias.

**Justificación (IA).** El problema es abordable con IA debido a: no estacionariedad y cambios de régimen, presencia de decisiones secuenciales (RL), y complejidad multi-activo que vuelve poco escalables soluciones manuales/heurísticas.

# Datos y preparación

## Fuentes de datos.

Se utiliza `yfinance` como fuente de series históricas de precios (frecuencia diaria). Se trabaja con precio *Close* (ajustado automáticamente).

## Caracterización del dataset.

El período cubre 2015-01-02 a 2024-12-30 para 7 activos.

Filas precios	Nº activos	Inicio	Fin	Filas features	Nº features
2515	7	2015-01-02	2024-12-30	2466	92

Cuadro 1: Resumen del dataset y features generadas.

## EDA: patrones, outliers, correlaciones y sesgos.

Se analiza correlación entre retornos, evolución de precios normalizados (base=100) para comparar crecimiento relativo, y distribución de retornos diarios promedio (*cross-assets*). Se reconoce riesgo de *survivorship bias* y dependencia de fuente (`yfinance`).

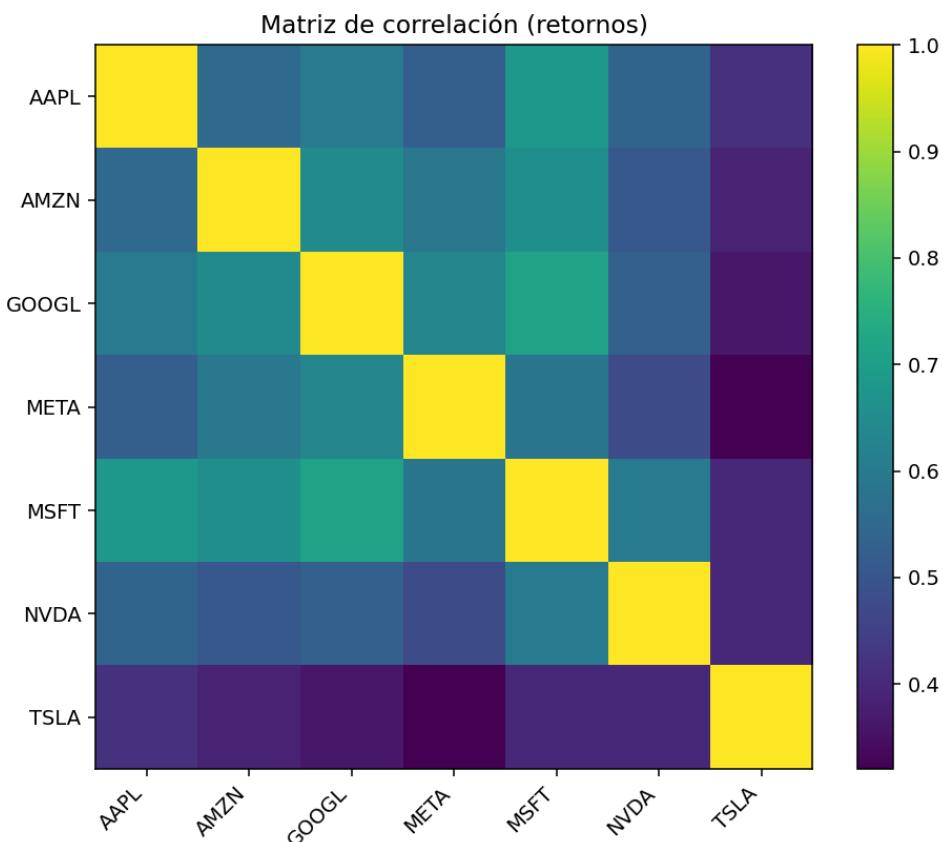


Figura 1: Matriz de correlación (retornos).

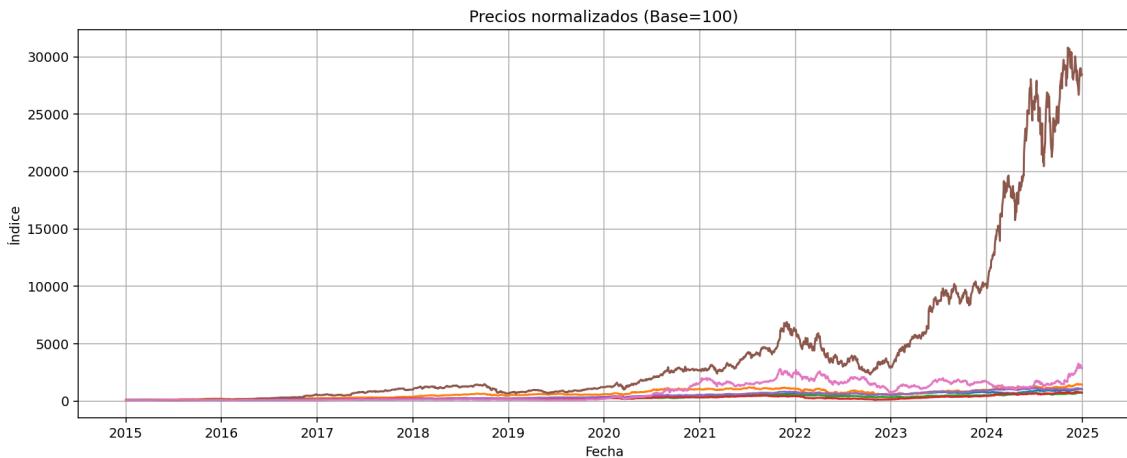


Figura 2: Precios normalizados (Base=100).

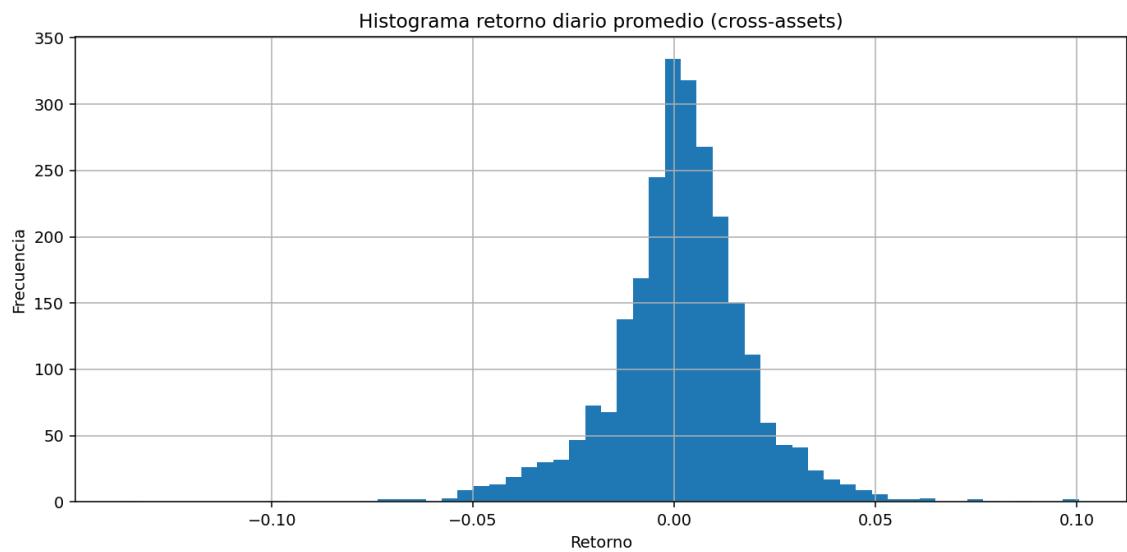


Figura 3: Histograma del retorno diario promedio (*cross-assets*).

Ticker	mean	std	min	max	skew	kurt
AAPL	0.001091	0.017928	-0.128647	0.119808	-0.003489	5.314588
AMZN	0.001272	0.020649	-0.140494	0.141311	0.399637	6.172538
GOOGL	0.000948	0.017933	-0.116341	0.162585	0.240716	6.699563
META	0.001087	0.023616	-0.263901	0.232824	-0.338717	21.890718
MSFT	0.001084	0.017095	-0.147390	0.142169	0.074760	7.811970
NVDA	0.002193	0.026776	-0.187860	0.251199	0.521710	10.434095
TSLA	0.001746	0.034017	-0.215154	0.243578	0.338988	7.404651

Cuadro 2: Estadísticos descriptivos de retornos diarios por activo.

Ticker	Tasa outliers ( $ z >3$ )
AAPL	0.0143
AMZN	0.0159
GOOGL	0.0147
META	0.0127
MSFT	0.0155
NVDA	0.0119
TSLA	0.0163

Cuadro 3: Tasa de outliers por activo (criterio z-score).

### Limpieza y preprocesamiento.

- Se eliminan columnas completamente nulas.
- Se eliminan filas con valores faltantes para asegurar consistencia multi-activo.
- Se aplica normalización/estandarización en el set de *features* para alimentar módulos de modelamiento/RL.

### Feature engineering.

Se construyen indicadores técnicos típicos: SMA (10, 20, 50), EMA (12, 26), RSI (14), Bandas de Bollinger (20,  $k = 2$ ) y Volatilidad (20). Además, se generan features normalizadas (92 variables) como insumo general para los modelos.

# Formulación experimental

## Enfoque experimental y partición de datos.

Dado el carácter temporal de las series, se utiliza un split cronológico:

- Train: 2015-01-01 → 2021-12-31
- Test: 2022-01-01 → 2024-12-31

Este diseño reduce el riesgo de fuga de información y aproxima el escenario real de despliegue.

## Métricas de evaluación y justificación.

Se emplean métricas enfocadas en riesgo/retorno para comparar estrategias: retorno total, CAGR (aprox.), volatilidad anual, Sharpe, Max Drawdown y Calmar. En particular, Sharpe y Calmar permiten comparar rendimiento ajustado por riesgo y por caídas máximas, respectivamente.

## Baselines.

Se utiliza Buy&Hold por activo como baseline principal: estrategia simple, interpretable y estándar para referencia en horizonte largo.

# Modelos y métodos de IA

**Modelos implementados.** Se consideran tres familias de métodos:

- **Forecasting (SARIMA)** sobre retornos mensuales para estimar  $\mu$  por activo.
- **Optimización media-varianza (MVO)** usando  $\mu$  y matriz de covarianza para obtener portafolio *Max Sharpe*.
- **Reinforcement Learning (PPO)** multi-activo: documentado como alternativa, pero **deshabilitado** en la ejecución actual.

## Justificación técnica.

- SARIMA permite modelar componentes autorregresivos y estacionales y entrega una estimación directa de retorno esperado.
- MVO es un estándar para asignación bajo supuestos de varianza/covarianza y permite construir un punto eficiente (maximización de Sharpe).
- PPO permite decisiones secuenciales y aprendizaje end-to-end de política; resulta útil cuando la dinámica no es bien capturada por modelos lineales.

## Hiperparámetros relevantes (SARIMA).

Se utiliza un SARIMA con orden y orden estacional (estacionalidad anual  $s = 12$  en frecuencia mensual). En esta ejecución se emplea configuración  $(1,1,1) \times (1,1,1,12)$  para todos los activos.

Ticker	Orden (p,d,q)	Estacional (P,D,Q,s)	AIC	$\mu$ forecast	$\mu$ hist	$\sigma$ hist
AAPL	(1, 1, 1)	(1, 1, 1, 12)	-159.9702	0.0167	0.0224	0.0796
AMZN	(1, 1, 1)	(1, 1, 1, 12)	-156.1824	0.1118	0.0252	0.0882
GOOGL	(1, 1, 1)	(1, 1, 1, 12)	-199.4001	0.0357	0.0190	0.0691
META	(1, 1, 1)	(1, 1, 1, 12)	-118.9601	0.1249	0.0222	0.0973
MSFT	(1, 1, 1)	(1, 1, 1, 12)	-224.3052	0.0076	0.0231	0.0616
NVDA	(1, 1, 1)	(1, 1, 1, 12)	-66.2788	0.1449	0.0578	0.1343
TSLA	(1, 1, 1)	(1, 1, 1, 12)	-3.6631	0.2359	0.0438	0.1799

Cuadro 4: Diagnóstico del modelo SARIMA sobre retornos mensuales.

## Proceso de entrenamiento (resumen).

- Se construyen retornos mensuales a partir de precios.
- Se ajusta SARIMA por activo con estacionalidad anual.
- Se obtiene  $\mu$  pronosticado por activo como insumo para optimización.

# Implementación y arquitectura técnica

## Arquitectura de la solución (flujo de datos).

- Ingesta de precios diarios (`yfinance`) → limpieza → cálculo de retornos.
- Feature engineering (indicadores técnicos) → normalización → dataset final de features.
- Forecasting (SARIMA) → estimación de  $\mu$ .
- Optimización MVO → portafolio *Max Sharpe*.
- Evaluación comparativa y generación de reportes (tablas/figuras).

## Componentes e integración.

La solución se implementa como scripts/notebooks que producen:

- Archivos de salida (`prices`, `returns`, `features`).
- Tablas de EDA (`eda_returns_stats`, `eda_returns_corr`, `eda_outliers`).
- Tabla de comparación de modelos (`model_comparison`) y figura de curvas de valor.

**Tecnologías y librerías utilizadas.** Se utiliza Python y librerías estándar del stack cuantitativo (ej., `pandas`, `numpy`, `matplotlib`), además de `yfinance` para adquisición de datos y `statsmodels` para el ajuste SARIMA.

## Requerimientos computacionales.

El ajuste SARIMA se realiza por activo (7 modelos), lo que mantiene tiempos de entrenamiento acotados en CPU. El módulo RL, al estar deshabilitado, no impacta tiempos de cómputo en esta ejecución.

# Evaluación y comparación de modelos

## Resultados cuantitativos.

Se reportan métricas en el conjunto de test (2022–2024) para cada baseline *Buy&Hold* y para el portafolio MVO (Max Sharpe).

Estrategia	Retorno total	CAGR	Volatilidad anual	Sharpe	Max Drawdown	Calmar	Valor final
BuyHold_NVDA	283.5738	0.9792	0.4859	2.0153	-0.6634	1.4762	284.5738
MVO_MaxSharpe	17.8658	0.4062	0.3051	1.3311	-0.5870	0.6920	18.8658
BuyHold_MSFT	9.5804	0.3141	0.2713	1.1575	-0.3715	0.8454	10.5804
BuyHold_AMZN	13.3459	0.3775	0.3277	1.1518	-0.5615	0.6723	14.3459
BuyHold_TSLA	27.5493	0.6456	0.5714	1.1299	-0.7363	0.8769	28.5493
BuyHold_AAPL	9.3585	0.3161	0.2845	1.1110	-0.3852	0.8208	10.3585
BuyHold_GOOGL	6.2489	0.2698	0.2846	0.9479	-0.4432	0.6087	7.2489
BuyHold_META	6.5653	0.3149	0.3748	0.8401	-0.7674	0.4103	7.5653

Cuadro 5: Comparación de estrategias/modelos en test (2022–2024).

## Comparación sistemática y trade-offs.

En términos generales, se observa un trade-off entre:

- **Retorno** (alto en activos con fuerte crecimiento) vs. riesgo (volatilidad y drawdown).
- **Sharpe** alto en estrategias con buen ajuste riesgo/retorno, pero con sensibilidad a cambios de régimen.

## Modelo final seleccionado.

El portafolio MVO\_MaxSharpe se considera una alternativa robusta por diversificación y mejor equilibrio riesgo/retorno frente a baselines individuales, aun cuando un activo específico (ej., NVDA) pueda dominar en retorno total. La selección final depende del objetivo del negocio (maximizar retorno vs. controlar drawdown).



Figura 4: Curvas de valor normalizadas: baselines Buy&Hold por activo vs. portafolio MVO MaxSharpe.

# Análisis de resultados

**Interpretación.** La evidencia sugiere que el desempeño está altamente influenciado por el régimen de mercado y por la concentración en activos de alto crecimiento. En el período analizado, algunos activos presentan retornos extraordinarios, pero también drawdowns profundos, por lo que la comparación se realiza incorporando métricas de riesgo.

## Casos de éxito.

- Estrategias con mejor Sharpe tienden a suavizar volatilidad agregada, especialmente al diversificar.
- Rebalanceo/optimización puede reducir exposición a caídas extremas relativas.

## Casos donde el enfoque falla o se degrada.

- **Forecasting:** es sensible a shocks exógenos y cambios de régimen; puede sesgar  $\mu$ .
- **MVO:** depende de covarianza histórica y supuestos aproximados; puede subestimar riesgo de cola.
- **Evaluación sin fricciones:** no modelar slippage/turnover real puede sobreestimar performance.

## Análisis de errores e impacto.

- **Outliers** alteran estimaciones de riesgo y covarianza, cambiando asignaciones.
- **Correlación alta** reduce diversificación efectiva; aumenta vulnerabilidad ante caídas sistémicas.

# Validación, generalización y limitaciones

## Generalización (datos no vistos).

Se utiliza evaluación *out-of-sample* mediante split temporal (Train 2015–2021, Test 2022–2024), aportando evidencia de desempeño en escenarios posteriores con condiciones potencialmente distintas.

## Limitaciones técnicas.

- Datos diarios: no capturan dinámica intradía ni microestructura.
- Costos y *market impact* simplificados (slippage 0): posible sobreestimación.
- Dependencia de `yfinance`: calidad/ajustes variables.
- RL multi-activo (si se habilita): requiere mayor estabilidad de entrenamiento, diseño de recompensas y restricciones operativas.

## Recomendaciones de validación adicional.

- Backtesting con fricciones (costos, slippage, límites de posición).
- Evaluación por regímenes (bull/bear/high-vol) y análisis de sensibilidad de hiperparámetros.
- Robustez estadística (bootstrap de retornos, intervalos de confianza de Sharpe).

# Referencias

- Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). *Time Series Analysis: Forecasting and Control* (5th ed.). Wiley.
- Hyndman, R. J., & Athanasopoulos, G. (2021). *Forecasting: Principles and Practice* (3rd ed.). OTexts. <https://otexts.com/fpp3/>
- López Aguilar, R. (s/f). *Proyecto de Capstone – Diploma en Ciencia de Datos e Inteligencia Artificial (Unidad 1.2)*. UEjecutivos, Facultad de Economía y Negocios, Universidad de Chile.
- López Aguilar, R. (s/f). *Proyecto de Capstone – Diploma en Ciencia de Datos e Inteligencia Artificial (Unidad 2.2)*. UEjecutivos, Facultad de Economía y Negocios, Universidad de Chile.
- Markowitz, H. (1952). Portfolio Selection. *The Journal of Finance*, 7(1), 77–91.
- Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal Policy Optimization Algorithms. *arXiv preprint arXiv:1707.06347*.
- Sharpe, W. F. (1966). Mutual Fund Performance. *The Journal of Business*, 39(1), 119–138.
- Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning: An Introduction* (2nd ed.). MIT Press.
- Young, T. W. (1991). Calmar Ratio: A Smoother Tool. *Futures*, 20(1), 40.

## Librerías utilizadas en el proyecto

- Gymnasium: <https://gymnasium.farama.org/>
- Matplotlib: <https://matplotlib.org/>
- NumPy: <https://numpy.org/>
- pandas: <https://pandas.pydata.org/>
- PyPortfolioOpt: <https://pyportfolioopt.readthedocs.io/en/latest/>
- PyYAML: <https://pyyaml.org/>
- scikit-learn: <https://scikit-learn.org/>
- SciPy: <https://scipy.org/>
- Stable-Baselines3: <https://stable-baselines3.readthedocs.io/>
- statsmodels (SARIMA/SARIMAX): <https://www.statsmodels.org/>
- yfinance: <https://pypi.org/project/yfinance/>