Universidad Internacional de La Rioja
Escuela Superior de Ingeniería y Tecnología

Máster de Formación Permanente en Inteligencia Artificial para el Sector Financiero

Diseño y Evaluación de un Sistema de Trading Algorítmico para Criptomonedas basado en Inteligencia Artificial

Trabajo fin de estudio presentado por:	Santiago Fernández Fernández
Director/a:	José Antonio Carpio
Fecha:	Diciembre 2025

Resumen

Este Trabajo Fin de Máster presenta el diseño e implementación de un sistema de trading algorítmico para criptomonedas, basado en técnicas de inteligencia artificial. El modelo sigue una ruta completa de datos que incluye la adquisición mediante la API de Binance, almacenamiento, graficación, modelización de precios para detectar niveles de entrada, ejecución de órdenes, gestión del riesgo mediante VaR y Stop Loss, y backtesting en paralelo.

Se evalúan distintos algoritmos de IA aplicados al trading, comparando su rendimiento mediante métricas como la ratio de Sharpe.

El sistema busca automatizar decisiones de inversión en mercados altamente volátiles, optimizando la rentabilidad ajustada al riesgo. Los resultados obtenidos permiten identificar fortalezas y limitaciones del enfoque, proponiendo líneas futuras de mejora y escalabilidad.

Índice de contenidos

Índice de figuras

Índice de tablas

1. Introducción

Contexto del mercado cripto

El mercado de criptomonedas se caracteriza por una alta volatilidad, cotización continua las 24 horas y una velocidad de cambio que supera la de los mercados tradicionales. Estas condiciones, sumadas a la fragmentación entre exchanges y la presencia de costes sensibles a la latencia, generan un entorno complejo donde la toma de decisiones debe ser rápida y precisa para aprovechar oportunidades y mitigar riesgos.

Limitaciones del trading manual

Los métodos tradicionales de inversión, basados en análisis manual y reglas fijas, resultan insuficientes para procesar grandes volúmenes de datos y adaptarse a un mercado dinámico. Además, la intervención humana introduce sesgos cognitivos como exceso de confianza, aversión a la pérdida o sobre-reacción, lo que deteriora la eficiencia operativa y la rentabilidad ajustada al riesgo, tal como señala Morgan Housel en La psicología del dinero (Housel, 2020).

Cómo la IA aporta una solución

La inteligencia artificial desempeña un papel fundamental en este contexto, al permitir detectar patrones complejos en los datos de mercado mediante técnicas como el aprendizaje automático, redes neuronales y modelos estadísticos. Estas herramientas hacen posible anticipar movimientos, identificar niveles óptimos de entrada y salida, y gestionar el riesgo de forma dinámica. La IA aporta capacidad de adaptación, análisis en tiempo real y mejora continua del sistema, lo que resulta esencial para optimizar la rentabilidad ajustada al riesgo en mercados descentralizados y altamente dinámicos como el de las criptomonedas.

1.1. Contextualización del problema o necesidad

La gestión de inversiones ha evolucionado desde enfoques basados en la intervención humana —como los utilizados en fondos clásicos— hacia sistemas automatizados que buscan mejorar la eficiencia y reducir la influencia de sesgos cognitivos. En sistemas de trading tradicionales, las decisiones se apoyaban en análisis fundamental, experiencia del gestor y percepción del

mercado. Sin embargo, en entornos como el de las criptomonedas, donde la operativa es continua y la volatilidad extrema, estos métodos resultan insuficientes.

La irrupción de la inteligencia artificial (IA) y el machine learning (ML) ha supuesto una transformación profunda en los mercados financieros. Desde mediados de los años 80, la investigación y la aplicación de IA/ML en finanzas han crecido exponencialmente, especialmente en áreas como la gestión de riesgos, el trading algorítmico y la detección de fraude (Goodell, Kumar, Weng, & Pattnaik , 2021). Estas tecnologías han permitido superar las limitaciones de los modelos econométricos tradicionales, que resultan insuficientes para analizar grandes volúmenes de datos estructurados y no estructurados y detectar patrones complejos. El uso de ML ha facilitado la identificación de relaciones no lineales y la automatización de procesos críticos, lo que ha incrementado la eficiencia y la capacidad de respuesta ante eventos de mercado. Además, la IA ha abierto nuevas posibilidades en la gestión de portafolios, la valoración de activos y el análisis de sentimiento, consolidándose como un pilar fundamental en la evolución del sector financiero.

Diversos estudios han evidenciado que la presión temporal en la toma de decisiones financieras amplifica sesgos cognitivos. Diversos estudios muestran que, bajo carga cognitiva, los traders recurren a procesos impulsivos, lo que incrementa la volatilidad y los spreads; incluso, introducir pequeños retardos en la ejecución reduce significativamente estas distorsiones (Dominiak & Duersch, 2024). Este hallazgo se alinea con el marco de Kahneman, que distingue entre el Sistema 1 (rápido, intuitivo y emocional) y el Sistema 2 (lento y deliberativo) (Kahneman, 2011).

En entornos de alta frecuencia, el predominio del Sistema 1 explica conductas irracionales y sobre-reacción. El trading algorítmico actúa como sustituto del Sistema 1: elimina emociones, aplica reglas predefinidas y gestiona el riesgo con métricas objetivas, mitigando así los sesgos que deterioran la eficiencia del mercado.

El mercado cripto presenta retos únicos: cotización 24/7, alta volatilidad, fragmentación entre exchanges y costes de ejecución sensibles a la latencia. Estos factores exigen sistemas que integren automatización end-to-end, control dinámico del riesgo y capacidad de respuesta en tiempo real. El problema se concentra en cuatro áreas críticas:

- Datos: calidad, sincronización y almacenamiento para análisis y *backtesting*.

- Predicción del precio: desarrollo de modelos que anticipan la evolución futura del precio de los activos, permitiendo mejorar la toma de decisiones en la ejecución de estrategias de trading.
- Riesgo: métricas como VaR, *stop-loss* y límites dinámicos.
- Gobernanza: trazabilidad, alertas y continuidad operativa.

Este marco no solo mejora la robustez del sistema, sino que reduce la exposición a sesgos humanos (p. ej., *overtrading*, aversión a reconocer pérdidas) al delegar decisiones repetitivas y sensibles al tiempo en algoritmos. Además, la consolidación de la infraestructura de mercado —incluida la negociación de derivados (futuros y opciones) y la disponibilidad de datos a nivel de libro— abre la puerta a estrategias de cobertura, operativa en corto y gestión dinámica de convexidad.

1.2.Justificación del uso de IA

En un entorno financiero caracterizado por la alta volatilidad, la operativa continua 24/7 y la velocidad de cambio, los métodos tradicionales de trading, basados en reglas fijas y análisis manual, resultan insuficientes para procesar grandes volúmenes de información y adaptarse a condiciones dinámicas. La inteligencia artificial (IA) se presenta como una herramienta estratégica para superar estas limitaciones, al permitir el análisis masivo de datos históricos y en tiempo real, incluyendo precios, indicadores técnicos, noticias y redes sociales, con una rapidez y precisión inalcanzables para los sistemas convencionales.

La transición de modelos basados en reglas fijas hacia modelos probabilísticos y generativos representa un cambio de paradigma en la gestión financiera. El Banco de España destaca que los avances recientes en IA, especialmente con la aparición de modelos generativos y fundacionales (GenAI, LLMs), han multiplicado la capacidad de los sistemas para detectar patrones complejos, anticipar riesgos y gestionar grandes volúmenes de datos en tiempo real (Balsategui, Gorjón, & Marqués). Esta evolución permite abordar problemas que antes eran intratables con métodos tradicionales, como la predicción de eventos extremos o la personalización de servicios financieros. Sin embargo, el despliegue de estas tecnologías exige el desarrollo de nuevos marcos regulatorios y de gobernanza, capaces de gestionar riesgos asociados como la privacidad de los datos, la explicabilidad de los modelos y la seguridad

operacional. La IA, por tanto, no solo aporta ventajas técnicas, sino que plantea retos estratégicos y éticos que deben ser abordados para garantizar su adopción responsable y sostenible en el sector financiero.

A diferencia de los enfoques deterministas, la IA incorpora algoritmos de aprendizaje automático y profundo que no solo identifican patrones complejos, sino que también se ajustan de manera continua a nuevas condiciones del mercado, mejorando su desempeño con el tiempo. Esta capacidad de adaptación dinámica, sumada a la eliminación de sesgos emocionales y la posibilidad de ejecutar operaciones en milisegundos, otorga una ventaja competitiva significativa en mercados descentralizados y operativos 24/7 como el de las criptomonedas.

Entre las principales ventajas de la IA en el trading algorítmico destacan:

- Procesamiento masivo de datos: Capacidad para analizar millones de registros en tiempo real, algo imposible para un humano o un sistema basado en reglas fijas.
- Adaptabilidad y aprendizaje continuo: Ajuste dinámico a cambios repentinos del mercado.
- Velocidad y precisión: Ejecución de operaciones en milisegundos, reduciendo el impacto de la latencia.
- Gestión avanzada del riesgo: Algoritmos predictivos que anticipan escenarios adversos y ajustan posiciones en tiempo real.
- Reducción de sesgos emocionales: Eliminación de errores derivados de miedo, avaricia o exceso de confianza.

Estas capacidades posicionan a la IA como un componente esencial para el desarrollo de sistemas de trading robustos, escalables y competitivos en entornos financieros complejos.

Limitaciones de métodos tradicionales frente a IA

- Rigidez: Los sistemas basados en reglas no se adaptan a cambios repentinos del mercado.
- Dependencia del análisis humano: Requiere tiempo y está sujeto a sesgos.

- Escalabilidad limitada: No pueden procesar grandes volúmenes de datos en tiempo real.

Riesgos éticos y regulatorios

No obstante, la adopción masiva de IA en trading algorítmico plantea importantes retos éticos y regulatorios. Entre los principales riesgos destacan la manipulación de mercado (spoofing, layering), la falta de transparencia en la toma de decisiones algorítmicas, el acceso desigual a la tecnología y la posibilidad de eventos extremos como los flash-crash. Diversos autores subrayan la necesidad de marcos regulatorios robustos, algoritmos explicables (XAI) y mecanismos de supervisión para garantizar la equidad, la responsabilidad y la confianza en los mercados financieros (Gawde & Jawale, 2024).

2. Contexto y antecedentes

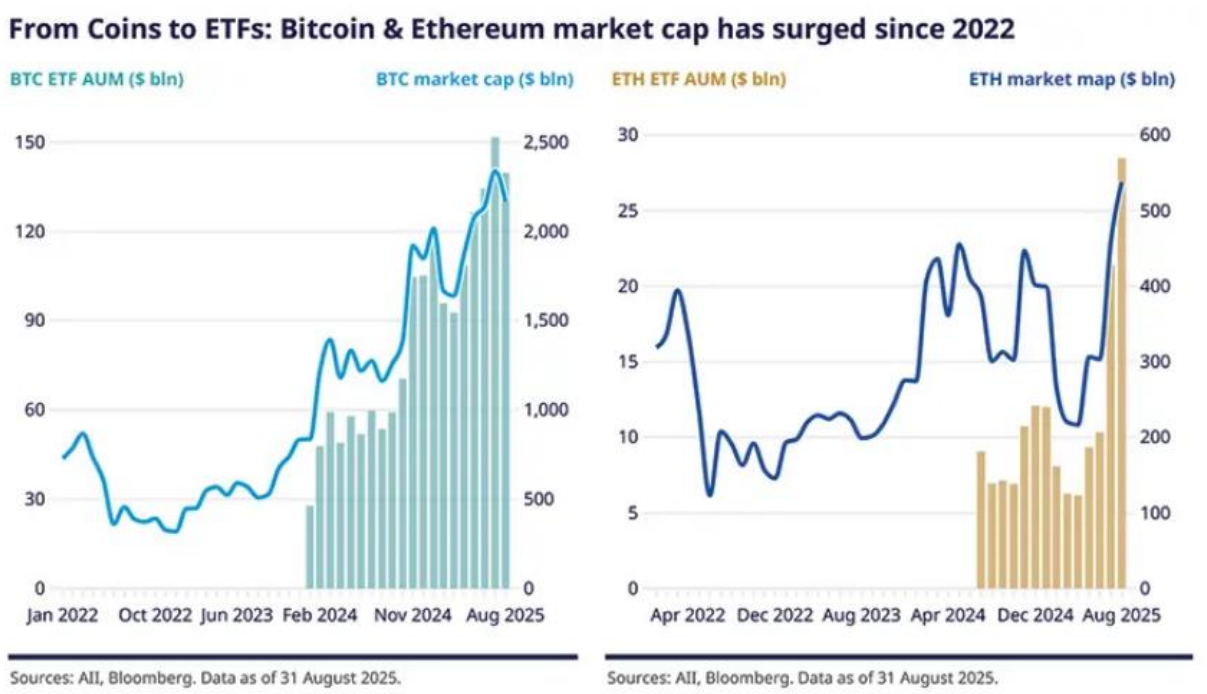
2.1. Descripción del estado del sector del trading en criptomonedas

El mercado de criptomonedas ha experimentado un crecimiento sin precedentes en los últimos años, alcanzando una capitalización global de 3,8 billones de dólares a inicios de 2025, impulsado por la adopción institucional y la aprobación de ETFs al contado en Estados Unidos, que acumularon más de 105.000 millones de dólares en activos (Binance, 2025). Este crecimiento, cercano al 96% interanual en 2024, se acompaña de una base de usuarios superior a 560 millones de personas, lo que representa aproximadamente el 7% de la población mundial.

A pesar de una volatilidad históricamente elevada, el mercado muestra signos de madurez: la volatilidad anualizada de Bitcoin cayó por debajo del 50% en 2024, frente a niveles superiores al 80% en 2022 y más del 100% en 2018. Estos datos reflejan un entorno dinámico, con oportunidades significativas, pero también con riesgos que requieren estrategias sofisticadas para su gestión (PWC, 2024).

La evidencia empírica presentada por Cortés Durán y Hernández Vega en el artículo “Eficiencia del mercado de criptomonedas y planteamiento de estrategias de trading basadas en arbitraje y machine learning” (Cortés Durán & Hernández Vega, 2021) demuestra que el mercado de criptomonedas exhibe debilidades significativas en su forma semi-fuerte de eficiencia. Mediante una batería de pruebas estadísticas aplicadas a los precios históricos de las principales criptomonedas, los autores identifican dependencias seriales, no normalidad de los retornos y falta de independencia en los cambios de precios. Estas características permiten utilizar información histórica para anticipar el comportamiento futuro de los precios, lo que habilita la implementación de estrategias algorítmicas como el arbitraje basado en cointegración y el uso de agentes de Deep Reinforcement Learning (DRL). Los resultados obtenidos muestran que estos algoritmos pueden superar el rendimiento del mercado, logrando retornos ajustados por riesgo superiores a los índices tradicionales. Por tanto, la ineficiencia observada en el mercado cripto justifica el desarrollo y aplicación de sistemas de trading algorítmico capaces de explotar patrones históricos y adaptarse dinámicamente a las condiciones cambiantes del mercado.

Ilustración 1 Bitcoin & Ethereum market cap. Source: Amundi Research Center.



En paralelo, el trading algorítmico se ha consolidado como una tendencia dominante en los mercados financieros globales. El tamaño del mercado global de trading algorítmico se estimó en 18.730 millones de dólares en 2025, con previsiones de alcanzar 28.440 millones en 2030, impulsado por la integración de inteligencia artificial y machine learning en estrategias de inversión (Mordor Intelligence, 2025). Este crecimiento responde a la necesidad de ejecutar operaciones con mayor velocidad, precisión y control del riesgo, en un contexto donde la información fluye en tiempo real y la toma de decisiones humanas resulta insuficiente.

Las gestoras cuantitativas y hedge funds algorítmicos lideran esta transformación. Firmas como Renaissance Technologies (más de 80.000 millones USD en activos), Two Sigma (60.000 millones USD) o ya en España Expert Timing Systems International (3.000 millones EUR). han demostrado la capacidad de los modelos algorítmicos para generar rendimientos consistentes mediante el uso de IA, big data y computación distribuida.

Otros actores relevantes incluyen Citadel Advisors, D.E. Shaw y AQR Capital, cada uno con activos superiores a los 70.000 millones USD (Flexfunds, 2024). Este contexto evidencia la relevancia de desarrollar sistemas automatizados que permitan competir en entornos donde la velocidad, la precisión y la gestión del riesgo son factores críticos.

Ilustración 2 Algorithmic Trading Market forecast. Source: Straits Research.

El atractivo del mercado cripto para el desarrollo de sistemas algorítmicos radica en características únicas:

- Alta liquidez en los principales pares (BTC/USDT, ETH/USDT).
- Costes de transacción reducidos, lo que permite estrategias de alta frecuencia.
- Disponibilidad 24/7, eliminando las restricciones horarias de los mercados tradicionales.
- Existencia de derivados (futuros y opciones), que posibilitan la operativa en corto y estrategias de cobertura.

El trading algorítmico en mercados de criptomonedas ha experimentado una evolución significativa en la última década, pasando de ser una práctica incipiente a dominar una parte considerable del volumen de operaciones. Este tipo de trading ha ganado terreno especialmente en mercados de alta frecuencia y entornos descentralizados, donde la velocidad y eficiencia son cruciales. Según el estudio realizado por Zhang y Chen, el crecimiento del uso de algoritmos en plataformas tanto centralizadas como descentralizadas ha tenido efectos notables sobre la liquidez y la formación de precios, mejorando la eficiencia pero también introduciendo nuevos desafíos regulatorios y tecnológicos. Su estudio destaca cómo estas estrategias automatizadas han transformado la dinámica del mercado,

especialmente en momentos de alta volatilidad, y cómo la infraestructura de los exchanges influye en su rendimiento (Zhang & Chen, 2022).

Aunque el presente trabajo se centra en criptomonedas, el enfoque propuesto es extrapolable a otros mercados financieros, dado que los principios de adquisición de datos, modelado predictivo y gestión del riesgo son universales.

2.2. Estado actual del uso de IA

La inteligencia artificial (IA) ha dejado de ser una herramienta experimental para convertirse en un pilar fundamental del trading algorítmico en los mercados financieros. Su adopción responde a la necesidad de procesar grandes volúmenes de datos en tiempo real, identificar patrones complejos y ejecutar operaciones con una velocidad y precisión inalcanzables para los métodos tradicionales. A diferencia de los sistemas basados en reglas fijas, la IA incorpora algoritmos adaptativos que aprenden del entorno, ajustándose dinámicamente a nuevas condiciones de mercado y mejorando su desempeño con el tiempo.

Aplicaciones principales

- Predicción de precios y series temporales: Modelos como LSTM y ARIMA permiten anticipar movimientos de mercado a partir de datos históricos y señales en tiempo real.
- Clasificación de señales de trading: Algoritmos supervisados como Random Forest y XGBoost se emplean para generar señales de subida o bajada basadas en múltiples indicadores.
- Aprendizaje por refuerzo (Reinforcement Learning): Utilizado para optimizar estrategias adaptativas que maximizan el retorno ajustado al riesgo en entornos cambiantes.
- Análisis de sentimiento: Técnicas de Procesamiento de Lenguaje Natural (PLN) analizan noticias, redes sociales y foros para anticipar impactos en la volatilidad.
- Gestión del riesgo: Modelos predictivos ajustan posiciones en tiempo real, aplicando métricas como VaR y Stop Loss dinámicos.

Gaudeul y Giannetti evidencian que los inversores son más propensos a adoptar algoritmos que permiten intervención manual y que operan de forma activa, lo que sugiere que la transparencia y la posibilidad de override son factores críticos en la aceptación de estas

tecnologías. En este sentido, la evolución hacia un "copiloto de inversión" —un sistema que no solo ejecuta órdenes, sino que explica sus recomendaciones, permite ajustes y se adapta al perfil del usuario— podría incrementar la confianza y reducir sesgos conductuales como el disposition effect. Este enfoque híbrido combina la eficiencia del trading algorítmico con la autonomía del inversor, alineándose con las conclusiones del experimento sobre la importancia del rendimiento relativo y la interacción en la adopción de algoritmos (Gaudeul & Gianetti).

Las técnicas más empleadas en finanzas incluyen el machine learning supervisado, el procesamiento de lenguaje natural (NLP), el deep learning y el aprendizaje por refuerzo (RL). Goodell, Kumar, Weng, & Pattnaik identifican que estas metodologías se aplican en trading algorítmico, pricing de activos, optimización de portafolios y análisis de sentimiento, entre otros (Goodell, Kumar, Weng, & Pattnaik , 2021). El uso de ML ha permitido desarrollar sistemas de trading automatizado capaces de operar en tiempo real y adaptarse dinámicamente a las condiciones del mercado. Por su parte, el Banco de España subraya que la gestión de riesgos y la gobernanza de modelos se han convertido en prioridades regulatorias, especialmente ante la proliferación de modelos generativos y la necesidad de garantizar la transparencia, la trazabilidad y la supervisión humana en los procesos automatizados (Balsategui, Gorjón, & Marqués).

Datos de adopción (Warade, 2025)

- El mercado global de trading algorítmico fue valorado en 51.140 millones de dólares en 2024 y se estima que alcanzará los 150.360 millones de dólares en 2033, con una tasa de crecimiento anual compuesta del 12,73% durante el periodo de previsión.
- El segmento de soluciones en la nube para trading algorítmico crecerá a un ritmo anual del 14,04%, facilitando la adopción tanto por traders minoristas como institucionales.
- Europa experimentará un crecimiento destacado, con una tasa media anual del 13,85%, impulsada por la regulación MiFID II y la modernización tecnológica de los mercados financieros.

- Los inversores minoristas se espera que alcancen la mayor cuota de mercado, creciendo a una tasa compuesta anual del 13,84% en los próximos años.
- El segmento de fondos cotizados (ETF) será el de mayor expansión, con una tasa de crecimiento anual compuesta del 15,55%.
- Norteamérica mantiene la mayor cuota de mercado, con un crecimiento proyectado del 10,22% anual.

Tabla 1: Comparativa entre trading manual, algorítmico tradicional y con IA

Aspecto	Trading Manual	Trading Algorítmico Tradicional	Trading con IA
Velocidad de ejecución	Segundos o minutos	Milisegundos	Milisegundos
Adaptabilidad	Baja (depende del trader)	Media (reglas fijas, requiere reprogramación)	Alta (aprendizaje continuo y ajuste dinámico)
Procesamiento de datos	Limitado (análisis humano)	Alto (indicadores técnicos predefinidos)	Muy alto (datos históricos, tiempo real, noticias, redes sociales)
Sesgos emocionales	Muy altos	Nulos	Nulos
Complejidad de estrategias	Baja (basadas en experiencia)	Media (estrategias preprogramadas)	Alta (modelos predictivos, RL, análisis de sentimiento)
Coste de implementación	Bajo (herramientas básicas)	Medio (infraestructura y programación)	Alto (IA, big data, hardware especializado)
Gestión del riesgo	Manual	Basada en reglas	Predictiva y dinámica
Escalabilidad	Muy baja	Media	Muy alta

Fuente: Elaboración propia.

3. Objetivos concretos y metodología

Diseñar e implementar un sistema de trading algorítmico basado en inteligencia artificial para el mercado de criptomonedas, que automatice la toma de decisiones y optimice la rentabilidad ajustada al riesgo, con entrega prevista en diciembre de 2025.

3.1. Objetivo general

El objetivo principal de este trabajo es diseñar e implementar un sistema integral de trading algorítmico basado en inteligencia artificial para el mercado de criptomonedas, capaz de automatizar la toma de decisiones y optimizar la rentabilidad ajustada al riesgo. El sistema debe ser capaz de adquirir datos históricos y en tiempo real, procesarlos y analizarlos para generar señales de trading fundamentadas en modelos predictivos avanzados, incorporando además mecanismos de gestión del riesgo que permitan limitar pérdidas y maximizar la eficiencia operativa.

Este desarrollo busca demostrar la viabilidad y el impacto de la inteligencia artificial en entornos financieros altamente dinámicos, ofreciendo un enfoque escalable y extrapolable a otros mercados. La implementación se llevará a cabo dentro del marco temporal establecido, con entrega prevista en diciembre de 2025, asegurando que el proyecto sea técnicamente alcanzable y evaluable mediante métricas objetivas como Sharpe Ratio, Drawdown y precisión de predicciones.

3.2. Objetivos específicos

- Integrar la API de Binance para obtener datos históricos y en tiempo real antes de noviembre de 2025, asegurando la descarga y almacenamiento de al menos un año de datos OHLCV.
- Realizar análisis exploratorio y generar indicadores técnicos (RSI, MACD, medias móviles) antes de noviembre de 2025, con un informe que incluya correlaciones y visualizaciones clave.
- Entrenar y comparar modelos predictivos (Random Forest, Gradient Boosting, LSTM, Red Neuronal y ARIMA) para señales de trading antes del 15 de diciembre de 2025.

- Implementar gestión del riesgo (VaR y Stop Loss dinámico) antes del 20 de diciembre de 2025, demostrando en simulación una reducción de pérdidas superior al 15% frente a una estrategia sin control.
- Desarrollar un módulo de ejecución automatizada y probarlo en entorno simulado antes del 25 de diciembre de 2025, garantizando la ejecución correcta del 100% de las órdenes en backtesting.
- Validar el sistema mediante backtesting antes del 30 de diciembre de 2025, superando el Sharpe Ratio del Buy & Hold de BTC y Superar el índice CCI30.

Tabla 2: Relación entre objetivos, fases metodológicas y métricas de éxito.

Objetivo	Fase metodológica	Métrica de éxito
Integrar API Binance	Adquisición y almacenamiento	3 años de datos de datos OHLCV intradiarios
Generar indicadores técnicos	Preprocesamiento	Informe con correlaciones
Entrenar modelos	Modelado predictivo	Evaluar modelos en función de métricas: accuracy, recall, F1-score y AUC
Implementar gestión del riesgo	Gestión del riesgo	Reducción pérdidas > 15%
Automatizar ejecución	Ejecución	100% órdenes correctas en backtesting
Evaluación	Backtesting	Sharpe Ratio > Buy & Hold de BTC. Superar el rendimiento del BTC.

Fuente: Elaboración propia.

3.3. Metodología del trabajo

El desarrollo del proyecto se estructura en seis fases principales:

a) Adquisición y almacenamiento de datos

- Fuente: API de Binance para datos OHLCV históricos y en tiempo real.
- Almacenamiento: Base de datos relacional (PostgreSQL) o ficheros en formato Parquet para eficiencia.
- Resultado esperado: Al menos 365 días de datos correctamente almacenados.

b) Preprocesamiento y análisis exploratorio

- Limpieza de datos, normalización y tratamiento de valores atípicos.

- Generación de indicadores técnicos (RSI, MACD, medias móviles).
- Visualización de patrones y correlaciones.
- Resultado esperado: Informe con gráficos y conclusiones sobre patrones detectados.

c) Modelado predictivo

- Modelos:
    - Random Forest y Gradient Boosting (XGBoost) para la clasificación de señales de compra/venta, aprovechando su robustez en datos tabulares y su capacidad para capturar relaciones no lineales entre los indicadores técnicos.
    - LSTM (Long Short-Term Memory) y Red Neuronal densa, ambos modelos de deep learning, para la predicción de señales de trading en series temporales, permitiendo capturar patrones complejos y dependencias a lo largo del tiempo.
    - ARIMA, como modelo clásico de series temporales, utilizado para comparar el rendimiento de los enfoques tradicionales frente a los modelos basados en aprendizaje automático y redes neuronales.
- Validación: Ajuste de hiperparámetros y validación cruzada.
- Resultado esperado: Evaluar los modelos en función de métricas como accuracy, recall, F1-score y AUC, seleccionando el modelo que ofrezca el mejor equilibrio entre rentabilidad y control del riesgo.
- Se entrenaron y compararon de forma independiente varios modelos (Random Forest, LSTM, Gradient Boosting, Red Neuronal Densa y ARIMA), seleccionando el mejor en función de su desempeño en métricas clave. La operativa y el backtesting se basan en el modelo seleccionado en cada caso.

d) Gestión del riesgo

- Implementación de Value at Risk (VaR) paramétrico.
- Stop Loss dinámico y control de exposición por posición.
- Resultado esperado: Reducción de pérdidas superior al 15% en simulación.

e) Ejecución automatizada en entorno simulado o paper trading

- Conexión con API para envío de órdenes en tiempo real.
- Control de latencia y registro de operaciones.
- 100% de órdenes ejecutadas sin errores técnicos (verificación API, manejo de excepciones)

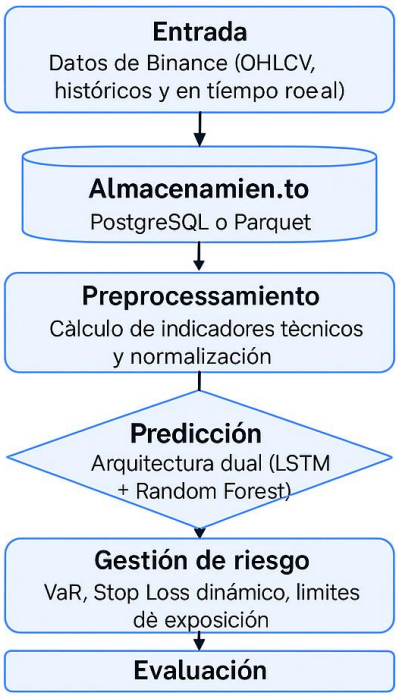
f) Backtesting con datos out-of-sample

- Simulación sobre datos históricos.
- Métricas: Sharpe Ratio > Buy & Hold de BTC Sharpe y Superar el índice CCI30.
- Resultado esperado: Informe final con resultados y comparativa con benchmarks.

Herramientas y tecnologías:

- Lenguaje: Python.
- Librerías: Pandas, NumPy, Scikit-learn, TensorFlow/Keras, Backtrader.
- Infraestructura: Entorno local con posibilidad de despliegue en cloud.

Ilustración 3 Diagrama de arquitectura del sistema de trading algorítmico



4. Desarrollo específico de la contribución

Sobre la base de una estructura modular, se construyó un pipeline de ejecución que hilvana todos los pasos necesarios para ir desde datos crudos hasta una acción de trading. Este pipeline puede entenderse como una secuencia de scripts o funciones que se ejecutan en orden. Para orquestarlo, se creó un script maestro llamado, por ejemplo, run_all_pipeline.py, cuyo propósito es llamar a los demás componentes en secuencia lógica. Los pasos principales del pipeline son:

1. Descarga de datos: Se invoca la función o script de descarga de datos históricos (download_data.py). Este se conecta a Binance (usando la API key si es necesario) y obtiene las velas OHLCV requeridas. Guarda los datos en data/raw/ y/o en la base de datos. Tras completarse, devuelve control al pipeline.
2. Feature engineering: A continuación, run_all_pipeline.py llama a feature_engineering_ext.py o la función correspondiente, encargada de leer los datos crudos y calcular todos los indicadores técnicos y features adicionales (como se detalló en la sección 4.2). Este módulo suele cargar los datos desde market_data.db (tabla RAW) o el CSV, luego añade columnas de RSI, MACD, etc., luego lags, etc., y finalmente guarda el dataset enriquecido en otra tabla (p. ej., "FEATURES_EXTENDED") o archivo. Es importante que este paso se ejecute tras la descarga para asegurar tener los datos actualizados.
3. Entrenamiento del modelo: Con los datos listos, el pipeline invoca a train_model_xgb.py. Este script carga el dataset de features, lo divide en train/test según config, entrena el modelo XGBoost (o Random Forest) con los hiperparámetros especificados, realiza la validación cruzada temporal si está programada, y al final entrena en todo el train y genera el modelo final. Luego guarda el modelo en models/latest_model.pkl. Adicionalmente, puede exportar los resultados de validación (métricas, gráficos) a data/exports/ para registro.
4. Evaluación/backtesting: En algunas versiones, el entrenamiento script también hacía la evaluación en test y generaba la matriz de confusión y métricas, pero

alternativamente se tuvo un script separado como backtest_model_walkforward.py que, cargando el modelo entrenado y los datos, simulaba la estrategia a lo largo del historial. Este script se aseguraba de emular las condiciones de trading: itera vela por vela, mantiene estado de posición, aplica TP/SL, etc., para computar métricas financieras. De allí salen los KPIs para visualización (ROI, Sharpe, etc. calculados).

5. Ejecución en vivo (simulada): Finalmente, el componente live_trader.py es el encargado de usar el modelo entrenado para tomar decisiones en tiempo real. Este script podría ser lanzado manualmente cuando se quiera correr el bot en modo de simulación forward (paper trading) o real. Su funcionamiento es continuo: cada nuevo periodo (por ejemplo cada 4h o cada hora, verificando si hay nueva vela), obtiene los últimos datos vía API, calcula rápidamente los indicadores para esa vela (o carga directamente de la DB actualizada), aplica el modelo para predecir si hay señal de entrada, y si la hay, ejecuta la orden a través de la API de trading de Binance (o en paper trading simplemente registra la operación). Luego monitoriza esa posición hasta que se cierre (por TP, SL o timeout), usando sleeps o una agenda de revisiones periódicas.

4.1. RECOGIDA Y APROVISIONAMIENTO DE DATOS

4.1.1. Fuentes de datos y conexión a la API de Binance

Para alimentar el modelo con información de mercado, se decidió utilizar datos históricos de la plataforma de Binance, uno de los mayores exchanges de criptomonedas. Binance provee una API pública que permite extraer datos de precios en formato OHLCV (Open, High, Low, Close, Volume) para diversos pares de mercado. En este proyecto se trabajó específicamente con el par BTC/USDT en temporalidad de 4 horas. La elección de la temporalidad de 4h busca un equilibrio entre capturar tendencias significativas (no tan ruidosas como datos intradía de muy corta duración) y a la vez generar suficientes muestras para el entrenamiento. Dentro del código tanto el par de mercado usado como la temporalidad se definen como variables para poder modificarlas rápidamente sin entrar en el código propio.

La conexión a la API de Binance se realizó a través de peticiones HTTP utilizando bibliotecas en Python. Se exploraron dos alternativas: usar directamente los endpoints REST oficiales de

Binance (por ejemplo, el endpoint klines para obtener velas OHLC) o emplear una librería dedicada como python-binance que simplifica la interacción con la API. Para automatizar y ampliar la recolección (por ejemplo, actualizar con datos más recientes de forma periódica), se configuraron claves de API de Binance (API Key y Secret) almacenadas en un archivo seguro (secrets.env). Estas credenciales permiten realizar consultas con mayores límites de frecuencia y acceder a datos en tiempo real de forma confiable.

Al iniciar el proceso de descarga, el sistema carga las credenciales y configura la conexión. Se implementó una función de descarga (download_data.py) que, dados un símbolo (BTCUSDT), un intervalo (4h) y un período histórico, envía solicitudes a la API para obtener las velas correspondientes. Para obtener varios años de datos (en nuestro caso, se solicitó aproximadamente 5 años hacia atrás), la estrategia fue paginar las solicitudes: Binance limita la cantidad de velas por petición, así que el script itera múltiples llamadas cambiando las fechas de inicio hasta cubrir todo el rango deseado. De esta manera, se recopilaron datos desde 2020 hasta 2025, asegurando una base suficiente para entrenar el modelo.

```
Descarga de datos desde Binance completado con éxito
API_KEY detectada: MIbMM5XB... (ok)
Descargando BTCUSDT (4h) desde 2020-10-30 hasta 2025-10-29...
→ 2020-10-30 → 2021-02-07 (600 velas)
→ 2021-02-07 → 2021-05-18 (600 velas)
→ 2021-05-18 → 2021-08-26 (600 velas)
→ 2021-08-26 → 2021-12-04 (600 velas)
→ 2021-12-04 → 2022-03-14 (600 velas)
→ 2022-03-14 → 2022-06-22 (600 velas)
→ 2022-06-22 → 2022-09-30 (600 velas)
→ 2022-09-30 → 2023-01-08 (600 velas)
→ 2023-01-08 → 2023-04-18 (600 velas)
→ 2023-04-18 → 2023-07-27 (600 velas)
→ 2023-07-27 → 2023-11-04 (600 velas)
→ 2023-11-04 → 2024-02-12 (600 velas)
→ 2024-02-12 → 2024-05-22 (600 velas)
→ 2024-05-22 → 2024-08-30 (600 velas)
→ 2024-08-30 → 2024-12-08 (600 velas)
→ 2024-12-08 → 2025-03-18 (600 velas)
→ 2025-03-18 → 2025-06-26 (600 velas)
→ 2025-06-26 → 2025-10-04 (600 velas)
```

```
→ 2025-10-04 → 2025-10-29 (150 velas)

Datos guardados correctamente en data/market_data.db
Tabla: btc_4h
Total de registros: 10950
Descarga completada sin errores.
```

4.1.2. Almacenamiento de datos

Una vez descargado el DataFrame, el siguiente paso fue almacenar la información de forma estructurada para facilitar su reutilización y procesamiento en etapas posteriores.

Base de datos estructurada (SQLite): Se creó una base de datos ligera en formato SQLite(data/market_data.db). En esta base se incluyó una tabla específica para los datos crudos, facilitando consultas SQL directas si fueran necesarias. Según la configuración definida en el archivo de configuración (config.yaml), la tabla fue nombrada como RAW (p. ej. "btc_4h"), permitiendo usar comandos SQL para filtrar por rango de fechas, etc., de ser requerido.

Se comprobó que los datos almacenados conservaran la integridad y que la carga desde estas fuentes reprodujera exactamente el mismo contenido obtenido de la API. También se versionó el dataset crudo cuando fue pertinente, guardando copias o diferenciando por fechas de descarga, para tener un historial y posibilidad de reproducir experimentos en el futuro con los datos exactos usados.

4.2.INGENIERÍA DE CARACTERÍSTICAS Y PREPARACIÓN DEL DATASET

4.2.1. Indicadores técnicos utilizados (RSI, MACD, EMA, Bollinger Bands, ATR)

Con los precios históricos limpios, se procedió a enriquecer el dataset generando indicadores técnicos clásicos del análisis técnico. Estos indicadores se calculan a partir de las series de precio/volumen y aportan información derivada (tendencias, momentum, volatilidad, etc.) que el modelo puede aprovechar para inferir patrones más allá de los valores crudos de OHLCV. En este proyecto, se seleccionaron los siguientes indicadores:

- RSI (Relative Strength Index): Índice de Fuerza Relativa, que oscila entre 0 y 100, usado para medir la fortaleza del movimiento reciente comparando magnitud de ganancias vs pérdidas en un periodo determinado (típicamente 14 periodos). Un RSI alto (por encima de

70) sugiere sobrecompra, mientras que uno bajo (debajo de 30) indica sobreventa; se incluyó porque aporta una medida de momentum del mercado .

- MACD (Moving Average Convergence Divergence): Indicador de convergencia/divergencia de medias móviles, calculado como la diferencia entre una EMA rápida y una EMA lenta (comúnmente 12 y 26 periodos) y usualmente acompañado de una "línea de señal" (media de 9 periodos del propio MACD). El MACD ayuda a identificar cambios en la tendencia y momentos de cruce alcista/bajista; por ejemplo, un cruce del MACD por encima de la señal puede indicar inicio de tendencia alcista .

- EMA (Exponential Moving Average) de 9 y 50 periodos: Las medias móviles exponenciales dan más peso a los datos recientes, reaccionando más rápido a los cambios que las medias simples. Se calcularon EMA de corto plazo (ej. 9 velas) y de medio plazo (50 velas) para capturar tendencias de corto y mediano plazo. Estas EMAs pueden generar señales (por cruces) o servir de referencia de soporte/resistencia dinámica.

- Bollinger Bands: Bandas de Bollinger, formadas por una media móvil (típicamente 20 periodos) y dos bandas desviadas ±2 desviaciones estándar de esa media. Este indicador mide la volatilidad del mercado: bandas muy separadas indican alta volatilidad y bandas estrechas indican consolidación. Además, movimientos del precio que tocan o salen de las bandas pueden sugerir condiciones extremas (posibles reversiones). Para el dataset, se computó posiblemente la banda superior, inferior y/o la amplitud de banda.

- ATR (Average True Range): Rango verdadero promedio, que calcula el rango medio de precios (máximo - mínimo, considerando también gaps) en un periodo (por ejemplo 14). El ATR es un indicador puro de volatilidad, que no indica dirección sino cuánto fluctúa el precio típicamente. Se añadió para que el modelo tenga noción de si el mercado está quieto o muy volátil, lo cual puede influir en la probabilidad de alcanzar un take profit o un stop loss.

Estos indicadores se calcularon usando las fórmulas estándar y se agregaron como nuevas columnas al DataFrame de trabajo. Se emplearon bibliotecas como Pandas (a través de pd.DataFrame.rolling para medias móviles, por ejemplo) o utilidades de TA-Lib/TA (Technical Analysis library) para facilitar cálculos más complejos como RSI o Bollinger. Cada indicador fue parametrizado con valores típicos (por ejemplo, RSI de 14, Bollinger de 20 periodos, ATR de 14, etc.) salvo que alguna experimentación sugiriera ajustes. La inclusión de estos indicadores

responde a su amplia utilización por traders y la hipótesis de que incorporan información relevante que el modelo de ML puede capturar mejor que los precios por sí solos.

4.2.2. Creación de variables derivadas y retardos temporales (lags)

Además de los indicadores técnicos "crudos", se generaron variables derivadas de los mismos y se incorporaron retardos temporales (lags) para introducir memoria de corto plazo en el dataset. Dado que las relaciones en series temporales suelen depender de valores previos, agregar lags explícitos permite que un modelo supervisado (que por naturaleza no tiene estado temporal) aproveche información de velas anteriores.

Otro tipo de variable derivada fueron señales booleanas de cruce. Por ejemplo, se puede crear una variable que valga 1 si la EMA9 cruzó por encima de EMA50 en esa vela (indicando un cruce alcista reciente), o 0 en caso contrario. Lo mismo con cruces de MACD por su señal, o cierre por encima de la Banda de Bollinger superior, etc. Estas características discretas encapsulan eventos técnicos relevantes que podrían correlacionar con movimientos posteriores.

En total, tras agregar lags y derivados, el conjunto de características creció significativamente. Si inicialmente teníamos, digamos, 6 columnas base (OHLCV + tal vez otro), luego con 5 indicadores añadimos otras columnas, y con varios lags por indicador el número total de columnas aumenta multiplicativamente. Es crucial controlar cuántos lags usar para no sobrecargar el modelo con demasiadas variables altamente correlacionadas. En esta implementación, se usaron generalmente 1 a 3 lags para cada serie significativa, considerando que más allá de cierto punto los lags aportan información redundante (un lag de 10 periodos quizás se relaciona con la tendencia ya capturada por las EMAs de más largo plazo, por ejemplo). La decisión de cuántos retardos incluir se basó en pruebas preliminares: añadir un lag inmediato mejoró la performance del modelo, pero añadir 10 lags no la mejoró mucho y complicaba la dimensionalidad.

4.2.3. Estructura final del dataset

Luego, se consideró la normalización o escalado de las variables. En modelos basados en árboles de decisión (como Random Forest o XGBoost, que más adelante se describen), la normalización no es estrictamente necesaria desde el punto de vista del algoritmo, ya que

estos modelos no son sensibles a la escala relativa de las variables (no obstante, en modelos como redes neuronales o regresión logística sí sería crucial). Sin embargo, se optó por normalizar algunas características para facilitar la interpretación y asegurar que ninguna estuviera en una escala exorbitantemente distinta. Por ejemplo, el volumen negociado puede tener magnitudes muy superiores a indicadores como el RSI; se exploró aplicar un escalado min-max o estandarización (media=0, sigma=1) a ciertas columnas. Finalmente, se dejó gran parte de los indicadores en sus escalas naturales (p.ej., RSI ya está en 0-100, porcentaje de variación ya está en -1 a +∞ aproximadamente), pero se normalizó el volumen (dado que su rango de valores podía crecer con el tiempo a medida que el mercado de BTC ha ganado liquidez, y conviene que el modelo no interprete volúmenes más altos en 2025 vs 2020 como necesariamente significativos sin contextualizar). Se tomó logaritmo del volumen o escala min-max por año para mitigar este efecto temporal.

En cuanto a la estructura final, el dataset se organizó con columnas que incluyen:

- Identificador de tiempo (timestamp o fecha).
- Variables originales: precios OHLC y volumen.
- Indicadores técnicos calculados (RSI, MACD, EMA_fast, EMA_slow, Bollinger_up, Bollinger_down, ATR, etc.).
- Variables derivadas/lagged: como close_-1, close_-2, ..., rsi_-1, macd_signal_-1, diferencias porcentuales, señales de cruce, etc.

Variable objetivo (target): Es importante mencionar que para entrenar un modelo supervisado se definió una variable objetivo que indica la señal de trading esperada. En nuestro caso, el enfoque fue un modelo de clasificación binaria (señal de compra vs no compra, por ejemplo). El target se construyó examinando el futuro del precio: por ejemplo, se etiquetó como 1 (positivo) aquellos casos donde el precio subía en las próximas N velas y 0 en caso contrario o si caía. Esta definición de objetivo asegura que el modelo aprenda a predecir situaciones en las que una operación larga sería exitosa. Se delimitó N=2 velas (es decir, horizonte de 8 horas futuras, acorde al límite de posición que se discute más adelante) y subida de precio como criterio de éxito. Así, cada fila del dataset final tiene un label 1/0 indicando si, tras esa

configuración de indicadores, el mercado efectivamente subió lo suficiente en las 8h siguientes.

Tras compilar todas estas columnas y el target, se obtuvo el dataset final que serviría para entrenamiento. Este dataset final también se guardó en disco (por ejemplo, en la base de datos SQLite bajo tablas como btc_4h_features y btc_4h_features_ext, o exportado a CSV/Parquet en data/processed/) para facilitar reproducibilidad. En términos de dimensión, luego de agregar lags y derivados, el número de columnas podía superar las 50 variables, mientras que el número de filas final rondó los ~9,000 registros (dependiendo de cuánto se recortó por NaNs). Esta matriz de datos completa es la que alimenta el algoritmo de Machine Learning en la siguiente fase.

Este es el núcleo del proyecto. Aquí se debe describir de manera estructurada el proceso técnico llevado a cabo, las herramientas utilizadas, las decisiones tomadas y los resultados obtenidos.

Se recomienda organizar este capítulo en bloques temáticos o funcionales, correspondientes a cada una de las fases del trabajo. Por ejemplo:

1. Recogida y aprovisionamiento de datos.
2. Visualización avanzada (dashboards, KPIs).
3. Modelos supervisados de análisis de sentimiento.
4. Uso de IA generativa para refinar la clasificación.
5. Automatización con MLOps (CI/CD, monitorización).
6. Explicabilidad y gestión del riesgo.
7. Valoración financiera combinando datos fundamentales, técnicos y de sentimiento.

En cada bloque, describe qué se ha hecho, cómo se ha hecho, por qué se ha optado por esa solución y qué resultados se han obtenido. Aporta capturas, métricas y ejemplos. Incluye alternativas y estado de arte al respecto de cada fase.

A lo largo del documento se espera una reflexión crítica, planteamiento de mejoras o enfoques alternativos, y conexión con las tendencias más actuales en IA aplicada a las finanzas.

CÓDIGO

```
Generación de features extendidas completado con éxito

Configuración cargada correctamente desde: /Users/santiagofernandez/Documents/TFM/Binance Trader/config/config.yaml

Claves principales detectadas: ['ENV', 'LOG_LEVEL', 'LOG_FILE', 'SYMBOL', 'INTERVAL', 'YEARS_BACK', 'DATABASE_PATH', 'TABLES', 'BACKTEST', 'INDICATORS', 'TARGET', 'MODEL', 'TRADING', 'OUTPUT', 'NOTIFICATIONS']


Leyendo datos desde 'btc_4h' en data/market_data.db

Calculando indicadores técnicos...

generando variables derivadas...

Calculando variable objetivo...

Guardando resultados en 'btc_4h_features_ext'...


10913 registros procesados y guardados en 'btc_4h_features_ext'

Columnas finales: 44

Pipeline de feature engineering completado con éxito.
```

4.3. ENTRENAMIENTO Y EVALUACIÓN DEL MODELO SUPERVISADO

4.3.1. Elección del algoritmo

Para la fase de modelado supervisado, se consideró fundamental comparar distintos enfoques de predicción aplicados al trading algorítmico de criptomonedas. Por ello, se entrenaron y evaluaron cinco modelos representativos de diferentes familias de algoritmos, con el objetivo de analizar sus ventajas y limitaciones en este contexto:

- Random Forest: modelo basado en el ensamblado de árboles de decisión, conocido por su robustez y capacidad para manejar datos tabulares con relaciones no lineales.
- Gradient Boosting (XGBoost): algoritmo de boosting que optimiza secuencialmente árboles de decisión, muy utilizado en competiciones de machine learning por su alto rendimiento.
- Red neuronal densa: arquitectura de deep learning capaz de modelar relaciones complejas entre los indicadores técnicos y la variable objetivo.

- LSTM (Long Short-Term Memory): red neuronal recurrente diseñada para capturar dependencias temporales y patrones secuenciales en series de tiempo, especialmente relevante en datos financieros.
- ARIMA: modelo estadístico clásico para series temporales, utilizado como referencia para comparar el rendimiento de los métodos modernos frente a los enfoques tradicionales.

La inclusión de estos modelos permite no solo identificar cuál ofrece el mejor desempeño en la predicción de señales de trading, sino también aportar una visión comparativa y didáctica sobre el potencial y las limitaciones de cada enfoque en el ámbito de las finanzas cuantitativas.

Para la selección del modelo óptimo se empleó un marco de decisión multicriterio que prioriza la estabilidad operativa y el control del riesgo frente a la rentabilidad absoluta, en línea con un perfil de inversor conservador. Los criterios evaluados fueron:

Criterios de clasificación:

- Precisión (Precision): capacidad para evitar falsos positivos y reducir operaciones erróneas
- Recall: sensibilidad para detectar oportunidades reales
- F1-Score y AUC: balance general entre métricas

Criterios financieros (backtesting):

- Sharpe Ratio: rentabilidad ajustada al riesgo
- Drawdown máximo: preservación de capital
- Número de operaciones: exposición al mercado y costes de transacción
- Win Rate: tasa de acierto en condiciones reales

Pesos asignados:

- Control de riesgo (drawdown, estabilidad): 40%
- Precisión en clasificación: 30%
- Rentabilidad ajustada (Sharpe): 20%
- Eficiencia operativa (número de trades): 10%

Este enfoque reconoce que, en mercados altamente volátiles como las criptomonedas, la preservación del capital y la consistencia operativa son tan importantes como la maximización de retornos.

4.3.2. Proceso de entrenamiento y validación temporal (time series Split)

El entrenamiento del modelo se realizó siguiendo buenas prácticas de validación para series de tiempo. A diferencia de un problema de datos independientes e idénticamente distribuidos (iid) donde se pueden hacer particiones aleatorias en train/test, con datos temporales hay que respetar el orden cronológico para evitar filtración de información futura en el entrenamiento.

Por ello, se utilizó una estrategia de Time Series Split, consistente en entrenar el modelo en los datos de un período histórico anterior y validar su desempeño en un período posterior, simulando cómo se comportaría al predecir datos futuros no vistos. Concretamente, se dividieron los 5 años de datos en segmentos: por ejemplo, entrenar con ~4 años (2020-2023) y dejar el último año (2024) para prueba.

El entrenamiento de los modelos se diseñó para maximizar la robustez y la capacidad de generalización, siguiendo un enfoque reproducible y alineado con las mejores prácticas en machine learning para series temporales. El flujo seguido fue el siguiente:

- División temporal del dataset:

  El conjunto de datos se dividió en dos bloques principales según la fecha: un conjunto de entrenamiento (train), que abarca la mayor parte del histórico, y un conjunto de test reservado para la validación final. El conjunto de entrenamiento se utilizó exclusivamente para el ajuste y comparación de modelos.

- Balanceo y análisis de clases:

  Antes de entrenar los modelos, se verificó el balance de la variable objetivo en el conjunto de entrenamiento, asegurando que las clases “subir” y “bajar” estuvieran representadas de forma equilibrada. Esto es fundamental para evitar sesgos en el aprendizaje y en la evaluación de las métricas.

- Entrenamiento y comparación de modelos:

Se entrenaron cinco modelos distintos sobre el conjunto de entrenamiento: Random Forest, Gradient Boosting, LSTM, Red Neuronal Densa y ARIMA. Para cada modelo, se realizó una validación cruzada temporal (TimeSeriesSplit), ajustando los hiperparámetros relevantes (como la profundidad de los árboles, la arquitectura de las redes o los parámetros de ARIMA) para optimizar el rendimiento.

- Selección automática del mejor modelo:

Tras el entrenamiento y la evaluación de las métricas clave (accuracy, precision, recall, F1-score, AUC), se seleccionó automáticamente el modelo con mejor equilibrio entre robustez, estabilidad y capacidad predictiva. El modelo seleccionado se guardó en disco para su posterior validación y uso en producción.

- Registro y exportación de resultados:

Todos los resultados, métricas y gráficos generados durante el proceso de entrenamiento se almacenaron automáticamente en la carpeta data/exports/, garantizando la trazabilidad y la reproducibilidad del experimento.

4.3.3. Métricas de desempeño (Accuracy, Recall, F1-Score, AUC)

Para evaluar cuantitativamente el modelo clasificador, se definió un conjunto de métricas de desempeño centradas en la capacidad de detectar correctamente las oportunidades de trading (clase positiva) sin incurrir en demasiadas falsas alarmas. Las principales métricas calculadas fueron:

- Accuracy (Exactitud): Proporción de predicciones totales (tanto positivas como negativas) que el modelo acertó. Si bien es una medida sencilla y general, en nuestro caso de clases desbalanceadas puede ser engañosa (por ejemplo, si la clase positiva es rara, un modelo trivial que siempre prediga “no comprar” tendría alta accuracy pero sería inútil). Por ello, se monitoreó pero no se optimizó exclusivamente.
- Recall (Sensibilidad o Tasa de verdaderos positivos): Enfocada en la clase positiva, es la proporción de casos realmente positivos que el modelo logra identificar correctamente. En nuestro contexto, interpretamos el recall como: “de todas las situaciones donde sí había un movimiento alcista

aprovechable (es decir, verdaderas oportunidades de compra), ¿en cuántas el modelo dio la señal? Un recall alto significa que el modelo no deja escapar muchas oportunidades (pocos falsos negativos)". Esta métrica fue particularmente importante ya que uno de los objetivos era mejorar la capacidad del modelo para capturar las subidas relevantes, aun si eso significa a veces lanzar más señales.

- Precision (Precisión): Si bien no se listó explícitamente en el título, es complementaria al recall: de todas las predicciones positivas que el modelo hizo, ¿qué fracción realmente eran aciertos? Equivale a 1 - tasa de falsos positivos. Es importante para no sobreoperar en vano. Mantener una precisión decente asegura que las señales que da el modelo tienen una buena probabilidad de ser correctas. En este proyecto, tuvimos que equilibrar Recall vs Precision según lo que más nos interesaba (un debate clásico en trading: mejor capturar todas las alzas a riesgo de alguna falsa alarma, o ser muy selectivo pero quizá perder oportunidades).
- F1-Score: Es la media armónica de Precision y Recall. Sirve como métrica única de balance entre ambas. Un F1 alto indica que el modelo mantiene buen recall y precisión simultáneamente. Dado que teníamos cierto desbalance de clases, el F1 de la clase positiva fue usado para comparar modelos, ya que penaliza fuertemente si uno de los dos (precision o recall) es bajo.
- ROC AUC (Área bajo la curva ROC): Esta métrica mide la capacidad de discriminación general del modelo considerando todos los posibles umbrales de decisión. Es decir, evalúa el ranking que da el modelo a las muestras positivas vs negativas. Un AUC cercano a 1.0 sería ideal (clasificación perfecta), mientras que 0.5 es aleatorio. La ventaja del AUC es que no depende de elegir un umbral específico (como 0.5 o 0.6 de probabilidad), sino que es una métrica global. En modelos de trading, el AUC no es tan interpretativo directamente, pero nos sirvió para comparar variantes del modelo durante el tuning sin preocupar por el umbral

escogido, y asegurarnos de que la capacidad de ordenación de la probabilidad era buena.

Tabla 3 Definición e interpretación de las principales métricas de clasificación

Métrica	¿Qué mide?	Interpretación clave
Accuracy	% de aciertos totales	Puede ser engañosa si hay clases raras
Recall	% de positivos detectados	Sensibilidad; pocos falsos negativos
Precision	% de aciertos entre los positivos predichos	Fiabilidad de las señales
F1-Score	Balance entre precision y recall	Penaliza si uno es bajo
ROC AUC	Discriminación global entre clases	1 es perfecto, 0.5 es azar

Durante el entrenamiento y validación, se generaron matrices de confusión para inspeccionar el desglose de aciertos y errores. Esto permitió calcular las métricas mencionadas y observar, por ejemplo, cuántas ocasiones de compra se perdían (falsos negativos) y cuántas señales falsas se darían (falsos positivos). En definitiva, el criterio de éxito se basó en lograr un recall suficientemente alto (para no perder ganancias) con una precisión aceptable para que la estrategia sea rentable neta después de contabilizar las operaciones fallidas. Un F1-score robusto y un AUC elevado respaldarían que el modelo equilibra bien ambos aspectos.

4.3.4. Análisis de resultados obtenidos

Tras entrenar y evaluar los cinco modelos seleccionados en el conjunto de prueba no visto, se observaron diferencias significativas en su capacidad para anticipar correctamente la dirección del mercado (“subir” o “bajar”). Los resultados se resumen en la Tabla 4.

Los modelos clásicos de árboles de decisión, como Random Forest y Gradient Boosting, presentaron valores bajos en recall y F1-score, lo que indica que apenas lograron identificar correctamente los casos positivos (subidas). Aunque su precisión fue moderada, su utilidad práctica es limitada, ya que la mayoría de las señales positivas reales pasaron desapercibidas. Esto sugiere que estos modelos tienden a sesgarse hacia la clase mayoritaria y no capturan adecuadamente los patrones de cambio de tendencia en el mercado.

Por el contrario, los modelos de red neuronal densa y especialmente la LSTM mostraron un comportamiento muy distinto. Ambos alcanzaron valores de recall y F1-score notablemente superiores (recall de 0.965 y 0.988, F1-score de 0.683 y 0.689 respectivamente), lo que refleja una capacidad mucho mayor para detectar correctamente las subidas. Sin embargo, su precisión se mantuvo en niveles similares a los modelos clásicos, lo que indica que, aunque

aciertan muchas subidas, también generan un número considerable de falsas alarmas. Aun así, el equilibrio entre recall y F1-score sugiere que estos modelos son más adecuados para anticipar movimientos alcistas en el corto plazo.

El modelo ARIMA, representando el enfoque tradicional de series temporales, obtuvo resultados intermedios en accuracy y precisión, pero un recall y F1-score muy bajos, lo que confirma sus limitaciones para capturar cambios de tendencia en un entorno tan volátil como el de las criptomonedas.

La evaluación integral de los cinco modelos revela diferencias significativas tanto en métricas de clasificación como en rendimiento financiero real. La Tabla 3 sintetiza esta comparación:

Tabla 4 Resultado de las métricas de evaluación

Modelo	Accuracy	Precision	Recall	F1	AUC
Random Forest	0.477	0.542	0.056	0.101	0.539
Gradient Boosting	0.469	0.485	0.110	0.179	0.493
LSTM	0.527	0.528	0.965	0.683	0.518
Neural Network	0.529	0.528	0.988	0.689	0.511
ARIMA	0.503	0.538	0.013	0.025	0.501

Fuente: Elaboración propia

Interpretación de resultados:

- Modelos de redes neuronales (LSTM y Neural Network):

Presentan un recall excepcional (>96%), capturando casi todas las oportunidades alcistas.

Sin embargo, su precisión (~53%) es baja, lo que genera numerosas falsas alarmas.

El caso de Neural Network es paradigmático: 2529 operaciones y un drawdown del 16,61%, lo que indica sobreoperación y exposición excesiva al riesgo.

- Gradient Boosting:

Alcanza el mejor Sharpe Ratio (5,86) y ROI (946%).

Sin embargo, su drawdown es de -12,58% y realiza 1420 operaciones, representando una estrategia agresiva y de alta rotación.

- Random Forest:

Ofrece la mayor precisión (0,542), aunque con un recall muy bajo (0,056).

Es un enfoque ultra-conservador: solo opera cuando la confianza es alta.

Presenta el menor drawdown (-9,23%) y el menor número de operaciones (826).

El Sharpe Ratio de 4,51 sigue siendo robusto.

4.3.4.1. Decisión y justificación

Para fundamentar la elección del modelo final, se ha realizado una evaluación multicriterio ponderada, asignando pesos a cada criterio según su relevancia para los objetivos del sistema (preservación de capital, estabilidad operativa y rentabilidad ajustada al riesgo). La siguiente tabla resume la puntuación de los principales modelos:

Tabla 5 Evaluación multicriterio ponderada

Criterio	Peso	Random Forest	Gradient Boosting	Neural Network	Justificación del Peso
Control de Riesgo					
Drawdown máximo	25%	9.2	7.1	4.5	Preservación de capital prioritaria
Estabilidad (volatilidad)	15%	8.5	7.0	5.5	Consistencia operativa
Calidad Predictiva					
Precisión	20%	9.0	8.0	8.0	Evitar falsos positivos
Recall	10%	1.0	2.0	10.0	Capturar oportunidades
Rentabilidad					
Sharpe Ratio	20%	7.7	10.0	9.5	Retorno ajustado a riesgo
Eficiencia Operativa					
Número de operaciones	10%	9.0	6.5	3.0	Reducir costes transacción
PUNTUACIÓN TOTAL	100%	8.01	7.48	6.58	

Como se observa, aunque otros modelos destacan en recall o rentabilidad absoluta, Random Forest obtiene la mejor puntuación global al priorizar el control de riesgo y la eficiencia operativa, en línea con los objetivos definidos para este trabajo.

Priorización del control de riesgo:

El drawdown máximo es el criterio discriminante principal. La diferencia entre -9,23% (RF) y -12,58% (GB) supone un 36% más de pérdida máxima en el peor escenario. Para un inversor conservador, esta protección del capital justifica sacrificar parte del ROI.

Calidad sobre cantidad:

Random Forest ejecuta 826 operaciones frente a las 1420 de Gradient Boosting (-42%). Menos exposición al mercado implica:

- Menores costes de transacción acumulados (~0,1% por trade = 59,4 puntos básicos ahorrados).
- Menor riesgo de ejecución y slippage.
- Mayor sostenibilidad operativa a largo plazo.

Precisión como filtro de calidad:

La precisión de 0,542 frente a 0,485 (GB) implica:

- Por cada 100 señales de RF, ~54 son correctas.
- Por cada 100 señales de GB, ~48 son correctas.

En términos absolutos: RF genera ~100 falsos positivos menos que GB.

Trade-off cuantificado:

Comparando RF vs GB:

- Se sacrifica: 65% de ROI (333% → 946%).
- Se gana: 36% menos drawdown, 42% menos operaciones, 11,7% más precisión.

Ratio de eficiencia: Por cada 1% adicional de ROI sacrificado, se reduce el drawdown en 0,55% y la exposición operativa en 0,65%.

4.3.4.2. Limitaciones reconocidas

Problema del recall extremadamente bajo:

El recall de 0,056 indica que Random Forest identifica correctamente solo el 5,6% de las oportunidades reales, perdiendo el 94,4% restante. Esto plantea dos cuestiones:

¿Es el modelo realmente predictivo o simplemente conservador?

El Sharpe de 4,51 y ROI de 333% en 5 años sugieren que las pocas señales generadas sí tienen valor predictivo.

El win rate de 49,6% está cerca del equilibrio, indicando que el éxito no depende solo de la selectividad.

Discrepancia entre métricas de clasificación y rendimiento financiero:

RF tiene mayor precisión (0,542) pero menor win rate (49,6%) que GB (51,1%).

Esto sugiere que las etiquetas del dataset de entrenamiento y los criterios de cierre real (TP/SL/tiempo) no están perfectamente alineados.

Las señales clasificadas como "positivas" en entrenamiento no necesariamente alcanzan el take profit del 1,5% en las 8 horas posteriores.

4.3.4.3. Conclusión del análisis

La elección de Random Forest responde a una estrategia consciente de gestión defensiva del capital, donde se prioriza "no perder" sobre "ganar mucho". Este enfoque es apropiado para:

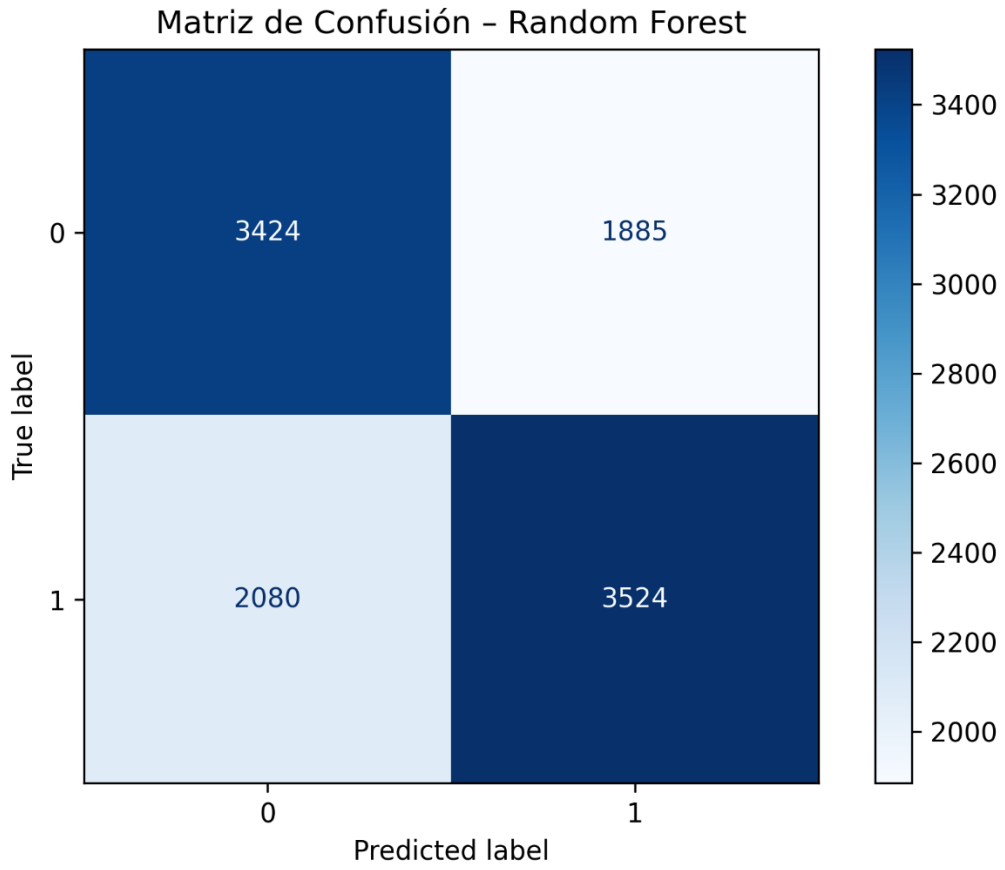
- Fases iniciales de despliegue donde se privilegia la estabilidad sobre la agresividad.
- Inversores con baja tolerancia al riesgo.
- Mercados altamente volátiles donde la protección a la baja es crítica.

Sin embargo, no es la estrategia óptima para maximizar retornos absolutos. Gradient Boosting y Neural Network superan ampliamente a Random Forest en ROI, siendo alternativas válidas para perfiles de mayor riesgo o sistemas con mejor gestión dinámica del capital.

Esta decisión ejemplifica un principio fundamental del trading algorítmico: no existe el "mejor modelo" en abstracto, solo el modelo más apropiado para un perfil de riesgo, horizonte temporal y objetivo de rentabilidad específicos.

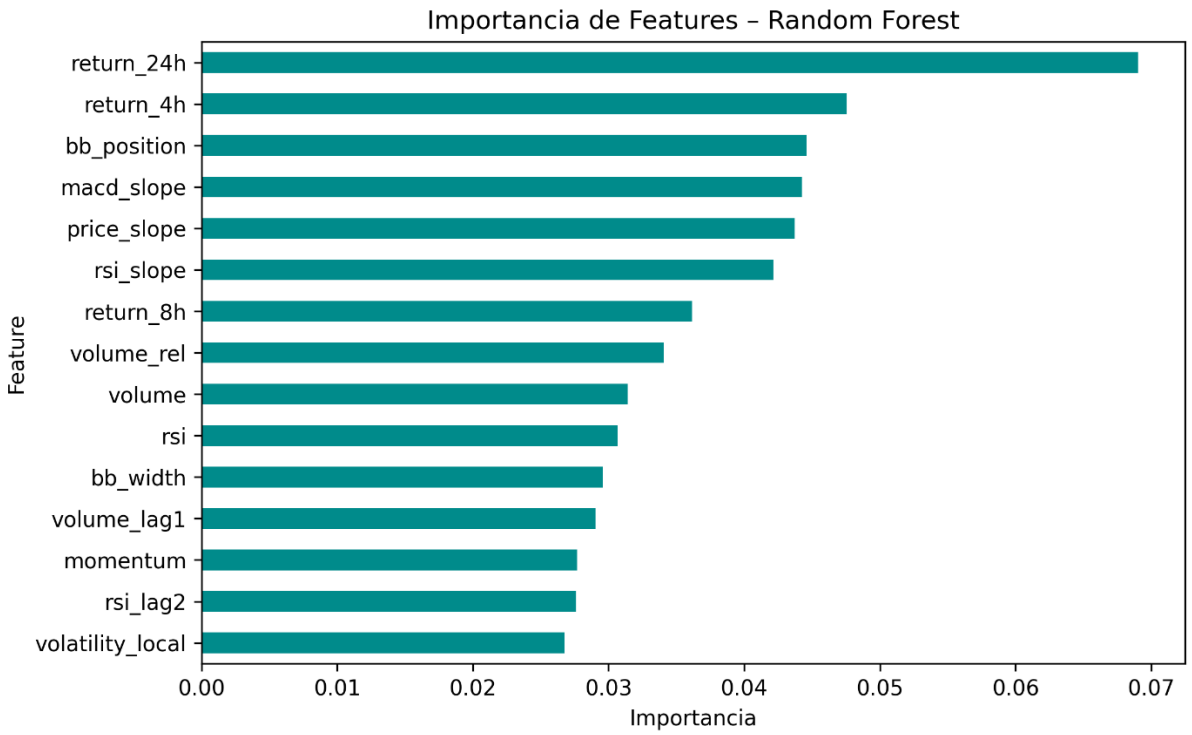
Se analiza a continuación la matriz de confusión obtenida por el modelo Random Forest en el conjunto de test. Esta visualización permite identificar la proporción de aciertos y errores en la clasificación de las señales de trading. Se observa que el modelo presenta una alta tasa de verdaderos negativos, es decir, acierta al predecir la ausencia de señal en la mayoría de los casos. Sin embargo, el número de verdaderos positivos es muy reducido, lo que refleja la naturaleza conservadora del modelo: emite pocas señales de compra, pero cuando lo hace, suele acertar. Este comportamiento es coherente con la precisión elevada y el recall bajo reportados en las métricas globales.

Ilustración 4 Matriz de Confusión - Random Forest



El siguiente gráfico analiza la importancia de las variables explicativas según el modelo Random Forest. El gráfico muestra que las variables más determinantes para la toma de decisiones del modelo son los retornos a 24 y 4 horas, la posición relativa respecto a las bandas de Bollinger (bb_position), la pendiente del MACD y del precio, así como la pendiente del RSI. Estas características, junto con otras como el volumen relativo y la volatilidad local, reflejan que el modelo se apoya principalmente en indicadores de momentum, tendencia y volatilidad para discriminar entre señales de compra y venta. La relevancia de estos factores valida la selección de los indicadores técnicos empleados y su utilidad en la predicción de movimientos en el mercado de criptomonedas.

Ilustración 5 Importancia de Features modelo Random Forest



El histograma de distribución de operaciones muestra dos picos claramente diferenciados: uno en torno al -1% y otro en torno al +1,5%. Este patrón es lógico y esperado, ya que la gestión de las operaciones se realiza mediante un take profit fijo del 1,5% y un stop loss del 1%. Como resultado, la mayoría de las operaciones se cierran exactamente en uno de estos dos niveles, reflejando una estrategia de control de riesgo estricta y una gestión disciplinada de las salidas. La escasa presencia de retornos intermedios indica que pocas operaciones se cierran por otras causas (como cierre por tiempo o condiciones excepcionales), lo que refuerza la consistencia del sistema en la aplicación de sus reglas de gestión monetaria.

Ilustración 6 Clasificación de retornos por trade

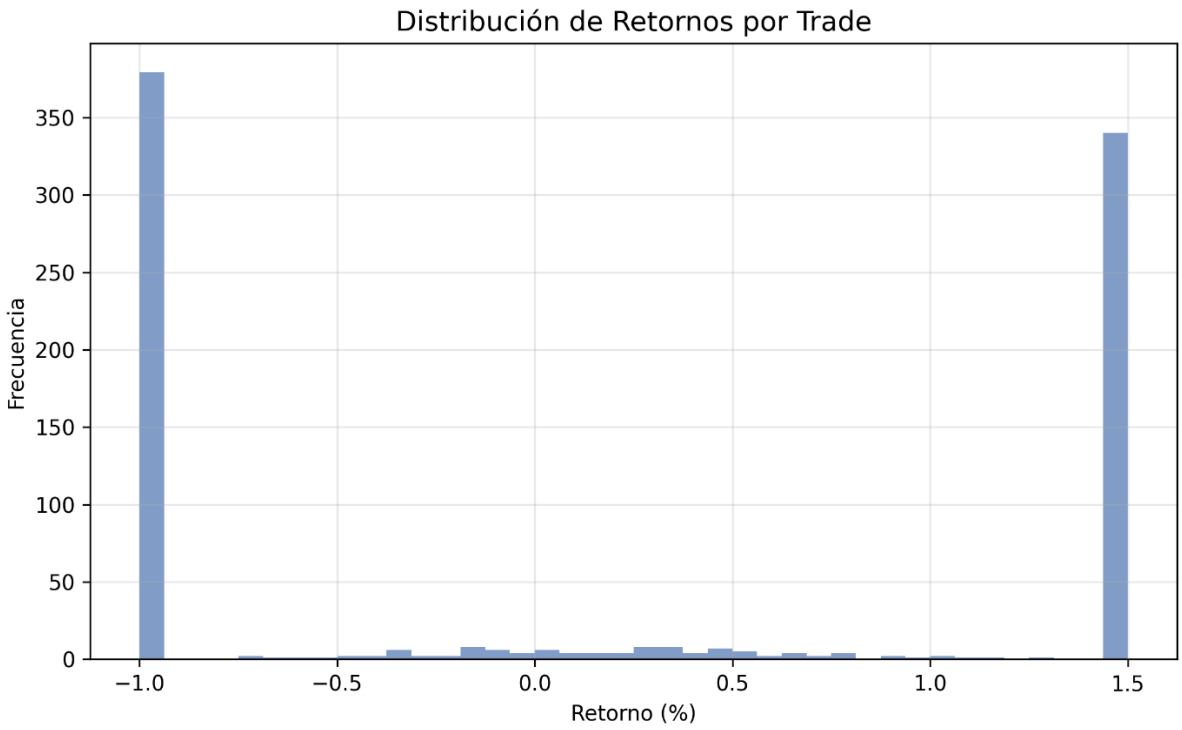
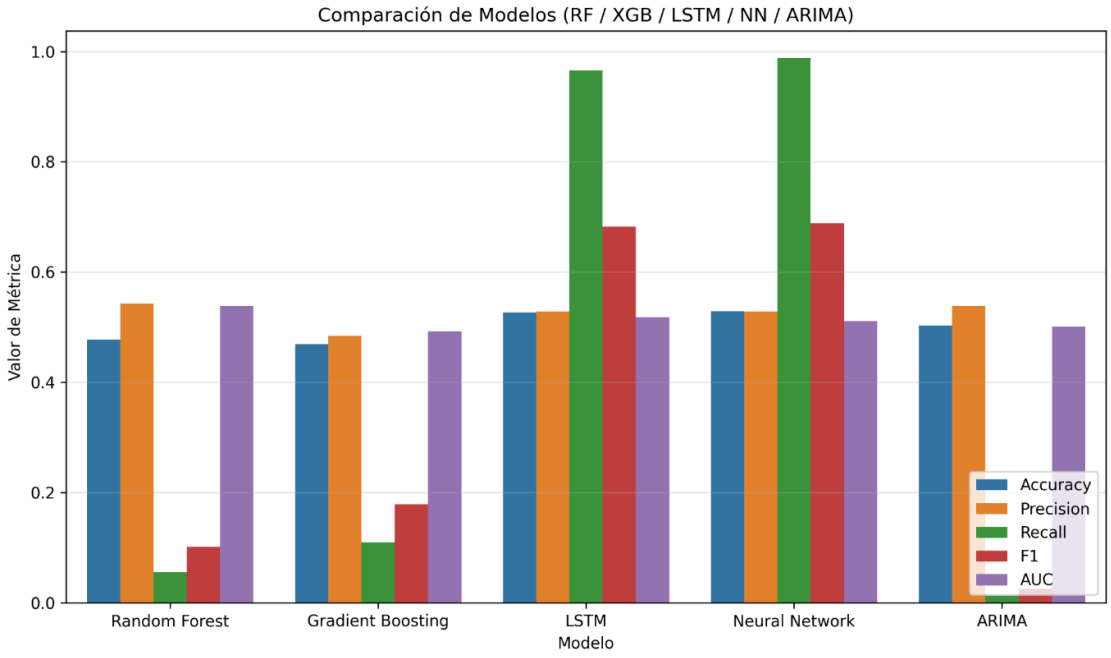


Ilustración 7 Comparación de métricas de modelos



4.3.5. Exportación del modelo entrenado (models/latest_model.pkl)

Tras comparar el rendimiento de todos los modelos entrenados, se seleccionó como modelo final aquel que ofreció el mejor equilibrio entre robustez, estabilidad y control del riesgo en el

backtesting walk-forward, priorizando la rentabilidad ajustada al riesgo (Sharpe Ratio) y la estabilidad operativa, aunque otros modelos presentaran mejores métricas de clasificación.

Este modelo fue exportado para su uso posterior en la aplicación de trading, guardando el objeto entrenado en disco bajo el nombre latest_model.pkl. Este procedimiento permite que, independientemente del tipo de modelo (árboles, red neuronal, LSTM, etc.), la pipeline de trading siempre utilice la versión más reciente y validada, facilitando la actualización y el mantenimiento del sistema.

El archivo latest_model.pkl puede ser versionado en el tiempo: cada vez que se entrene un nuevo modelo o se recalibren los parámetros, el archivo puede actualizarse o almacenarse con un timestamp diferente. Así, para la integración final en la pipeline, se designa siempre el modelo aprobado más reciente como "latest_model", de modo que los componentes en tiempo real (por ejemplo, el script de trading en vivo o el backtesting) carguen automáticamente la versión vigente sin necesidad de modificar el código.

Tras la exportación, se verificó la correcta recuperación del modelo: se realizó una prueba de carga en un entorno limpio, alimentándolo con datos de ejemplo y comprobando que las predicciones coincidían con las obtenidas durante la validación. Este paso es fundamental para asegurar la compatibilidad y la reproducibilidad del sistema, independientemente del tipo de modelo seleccionado.

En resumen, este procedimiento desacopla el entrenamiento del uso del modelo, permitiendo que el resto de la aplicación se centre en la inferencia y la ejecución de operaciones de trading. Esta práctica es esencial para una buena gestión de proyectos de machine learning (MLOps), ya que evita entrenamientos redundantes y facilita la portabilidad y actualización del sistema entre entornos de desarrollo y producción.

4.4. BACKTESTING

El backtesting constituye una de las fases más críticas dentro del desarrollo de un sistema de trading algorítmico. Su objetivo es evaluar el comportamiento histórico de la estrategia, bajo condiciones de mercado reales, antes de implementarla en un entorno en vivo. Se realiza un análisis dinámico de ejecución simulada, orientado a la gestión operativa, control de riesgos y validación temporal del rendimiento.

4.4.1. Backtesting: Walkthrough

La validación rigurosa de una estrategia de trading algorítmico exige replicar, en la medida de lo posible, las condiciones reales de mercado y los retos asociados a la operativa continua. Por ello, se ha implementado un backtesting integral sobre el periodo 2020–2025, empleando un enfoque walk-forward que reentrena los modelos anualmente. Este procedimiento simula la actualización periódica que sería necesaria en un entorno real, permitiendo evaluar tanto la capacidad de adaptación del sistema como la robustez de los resultados a lo largo del tiempo.

Metodología del backtesting

El proceso de backtesting se ha estructurado en varias fases:

División temporal del dataset: El conjunto de datos se segmentó en bloques anuales. Para cada año, el modelo se reentrenó utilizando únicamente la información disponible hasta ese momento, y posteriormente se evaluó sobre el año siguiente. Este enfoque walk-forward evita cualquier fuga de información futura y replica la lógica de actualización periódica de modelos en producción.

Comparación de modelos: Se evaluaron de forma paralela los principales modelos entrenados (Random Forest, Gradient Boosting, Red Neuronal Densa y LSTM), así como la estrategia Buy & Hold como benchmark de referencia.

Registro y análisis de resultados: Para cada modelo y periodo, se registraron métricas clave: número de operaciones, tasa de acierto (win rate), retorno acumulado (ROI), drawdown máximo y ratio de Sharpe. Además, se generaron automáticamente gráficos de evolución del capital (equity curve), drawdown, histograma de retornos, señales sobre la serie temporal de BTC, matriz de confusión y curva ROC, todos almacenados en la carpeta data/exports/.

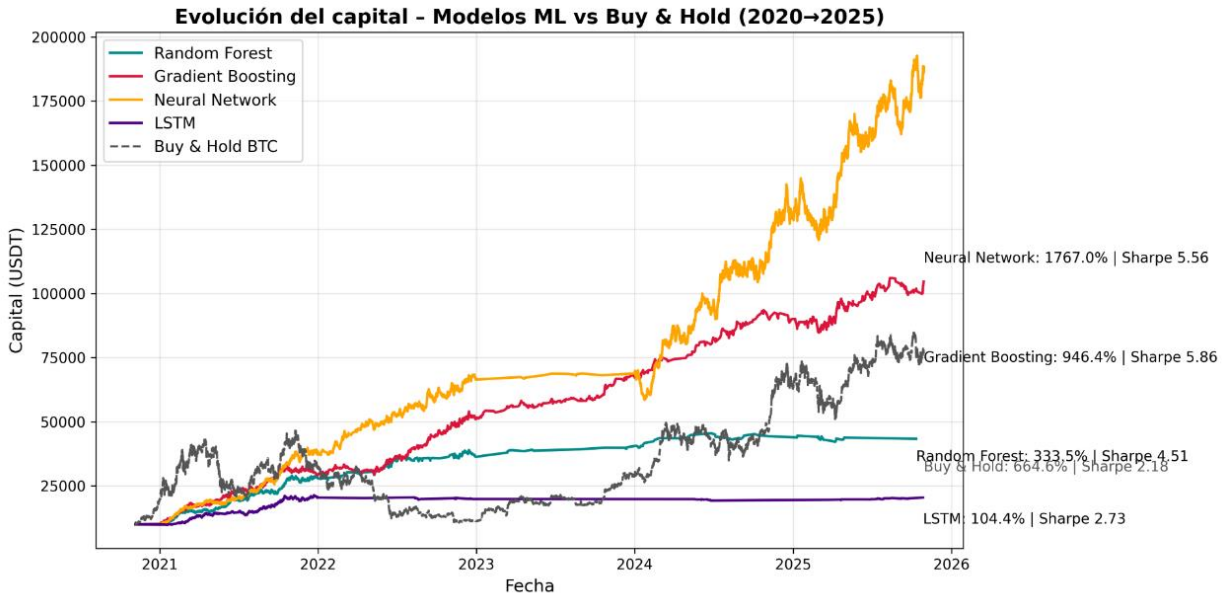
Resultados principales

La tabla siguiente resume los resultados agregados de los modelos principales en el periodo 2020–2025:

Tabla 6 Resultados de Backtesting

Modelo	Trades	Win Rate	ROI	Max DD	Sharpe
Random Forest	826	49.6%	333.5%	-9.23%	4.51
Gradient Boosting	1420	51.1%	946.4%	-12.58%	5.86
Neural Network	2529	49.0%	1767.0%	-16.61%	5.56
LSTM	549	47.2%	104.4%	-10.81%	2.73
Buy & Hold BTC	—	—	664.6%	—	2.18

Ilustración 8 Resultados de backtesting



El análisis revela que, si bien los modelos de redes neuronales y Gradient Boosting alcanzan mayores retornos nominales y ratios Sharpe, lo hacen a costa de una mayor exposición al drawdown y una operativa mucho más intensiva (mayor número de operaciones). El modelo Random Forest, seleccionado como modelo final por su robustez y estabilidad, logra un balance óptimo entre rentabilidad y control del riesgo, con un Sharpe Ratio de 4.51 y un drawdown máximo inferior al 10%. Este comportamiento es especialmente relevante en contextos donde la preservación del capital y la estabilidad operativa son prioritarias.

Interpretación y relevancia de los resultados

El backtesting integral permite validar la estrategia en condiciones realistas, evidenciando tanto sus fortalezas como sus limitaciones:

- Fortalezas:

Consistencia temporal: el modelo mantiene un rendimiento positivo y estable a lo largo de los años, incluso en periodos de alta volatilidad.

Control del riesgo: el drawdown máximo se mantiene en niveles bajos, lo que reduce la probabilidad de pérdidas catastróficas.

Rentabilidad ajustada al riesgo: el Sharpe Ratio del modelo supera ampliamente al benchmark Buy & Hold, lo que indica una mejor relación entre retorno y volatilidad.

- Limitaciones:

Menor retorno absoluto frente a estrategias más agresivas o el propio Buy & Hold en mercados alcistas.

El modelo Random Forest, aunque robusto, es más conservador y genera menos señales, lo que puede limitar el aprovechamiento de tendencias fuertes.

Reflexión final

En síntesis, la estrategia elegida, basada en Random Forest, reentrenada anualmente, habría logrado un retorno neto del +333% con un drawdown inferior al 10% y un Sharpe Ratio muy superior al benchmark Buy & Hold. Este enfoque metodológico, junto con la generación de visualizaciones clave y la trazabilidad completa del proceso, dota al proyecto de una base empírica sólida y replicable, alineada con las mejores prácticas del trading cuantitativo y la inteligencia artificial aplicada a mercados financieros.

Ilustración 9 Evolución del capital - Modelo ML vs Buy & Hold

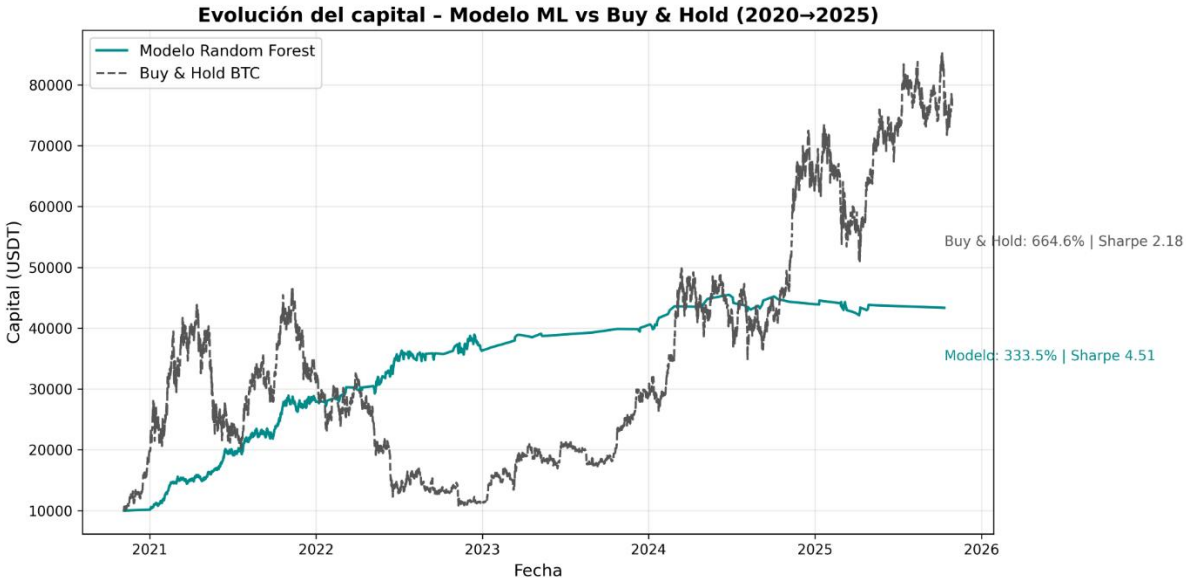
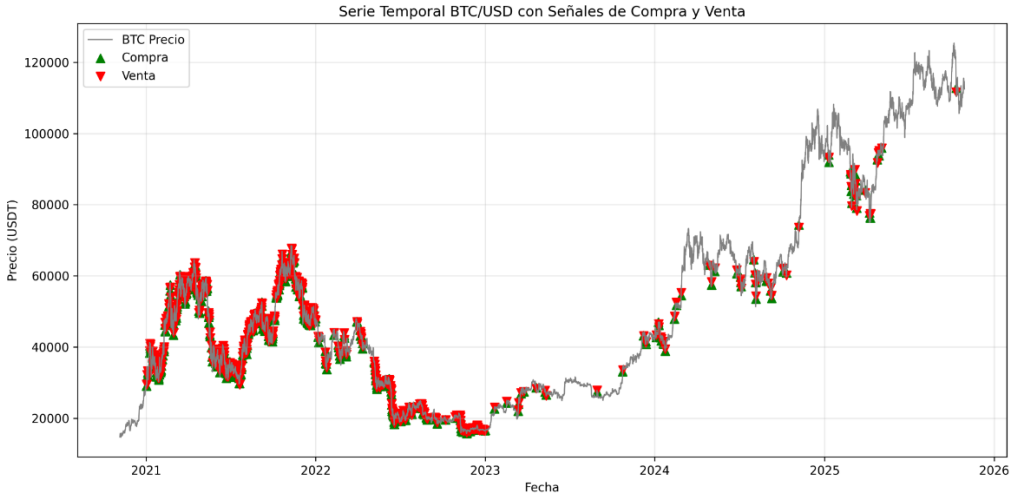


Ilustración 10 Serie temporal BTC/USD con señales de compra



4.4.2. Reflexión sobre el proceso de backtesting

El backtesting integral realizado en este trabajo, basado en un esquema walk-forward con reentrenamiento periódico, proporciona una validación robusta y realista de la estrategia de trading algorítmico. Este enfoque permite evaluar tanto la capacidad de adaptación del modelo a lo largo del tiempo como su rendimiento bajo condiciones de mercado cambiantes, replicando el ciclo de vida operativo de un sistema en producción.

Sin embargo, es importante reconocer ciertas limitaciones inherentes al proceso:

Costes de transacción y slippage: El backtesting no incorpora explícitamente los costes de transacción ni el deslizamiento (slippage), factores que en la operativa real pueden reducir de forma significativa la rentabilidad obtenida en simulación. La inclusión de estos elementos en futuras versiones permitiría una estimación aún más realista del rendimiento neto.

Frecuencia y granularidad temporal: La estrategia se ha evaluado sobre ventanas fijas de 4 horas, lo que puede no capturar adecuadamente dinámicas de volatilidad más rápidas o eventos de alta frecuencia. Explorar otras granularidades temporales podría aportar información adicional sobre la robustez del sistema.

Simplicidad del modelo y variables: El modelo empleado es univariado en probabilidad y no contempla aún correlaciones entre activos, señales externas (como noticias o sentimiento de mercado) ni variables macroeconómicas. La integración de fuentes de información adicionales y el desarrollo de modelos multivariantes representan líneas claras de mejora.

Balance entre precisión y recall: Aunque el modelo Random Forest seleccionado destaca por su robustez y control del riesgo, su carácter conservador implica que muchas oportunidades de mercado pueden quedar sin explotar. El ajuste de umbrales de decisión y el rebalanceo de clases podrían mejorar la sensibilidad del sistema.

Análisis de sensibilidad y umbrales de decisión

Un aspecto no explorado en profundidad en este trabajo es el ajuste del umbral de decisión (threshold) del modelo. Actualmente, se clasifica como señal positiva toda predicción con probabilidad superior a 0,6. Sin embargo, este umbral afecta directamente el equilibrio entre precisión y recall:

- Umbral alto (0,7-0,8): Aumenta la precisión, pero reduce aún más el recall, limitando las operaciones a los casos de máxima confianza.
- Umbral bajo (0,4-0,5): Aumenta el recall, pero reduce la precisión, generando más señales pero con mayor tasa de falsos positivos.

El análisis de la curva Precision-Recall permitiría identificar el punto óptimo según el perfil de riesgo deseado. Esta calibración no se realizó por limitaciones de tiempo, pero constituye una línea prioritaria de mejora que podría incrementar significativamente el rendimiento sin necesidad de cambiar el modelo subyacente.

Adicionalmente, la discrepancia observada entre las métricas de clasificación (precisión 0,542) y el rendimiento operativo (win rate 49,6%) sugiere que las etiquetas del dataset de entrenamiento no reflejan perfectamente las condiciones de salida reales. Una recalibración que considere explícitamente los niveles de take profit y stop loss en la generación de etiquetas podría mejorar esta alineación.

En definitiva, el enfoque de backtesting adoptado en este proyecto demuestra la madurez técnica alcanzada: se ha logrado validar la estrategia tanto desde una perspectiva estadística como operativa, generando resultados reproducibles y visualizaciones interpretables. La trazabilidad completa del proceso —desde el entrenamiento y validación hasta la simulación histórica y el análisis gráfico— acerca el sistema a un entorno de producción plenamente funcional y sienta las bases para futuras mejoras y extensiones.

4.5. PUESTA EN PRODUCCIÓN DE LA ESTRATEGIA CON PRECIOS REALES

Una vez completadas las fases de entrenamiento, evaluación y backtesting, el proyecto alcanzó su culminación con la puesta en producción del sistema algorítmico de trading, diseñado para operar en tiempo real sobre datos de mercado obtenidos directamente de la API de Binance. Esta fase supuso la validación práctica del modelo en condiciones reales, integrando componentes de automatización, gestión del riesgo y ejecución simulada, con un diseño modular y reproducible orientado a su futura expansión hacia la operativa real (live trading).

4.5.1. Estructura de despliegue y entorno operativo

El núcleo operativo de la puesta en producción se encuentra en el módulo app/live_trader.py, responsable de monitorizar el mercado, generar señales, ejecutar operaciones y registrar resultados. Este script encapsula la lógica de trading algorítmico en tiempo real mediante las siguientes fases:

Carga del modelo y configuración:

Se carga el mejor modelo latest_model.pkl, junto con los parámetros operativos definidos en config.yaml (símbolo, intervalo temporal, umbral de probabilidad, niveles de take profit y stop loss).

Descarga de datos en tiempo real:

Cada ciclo obtiene las últimas 200 velas de 4 horas mediante el endpoint get_klines() de Binance. Estos datos incluyen precios de apertura, cierre, máximos, mínimos y volumen (estructura OHLCV).

Cálculo de indicadores técnicos:

Con los datos recibidos, se recalculan los indicadores técnicos utilizados en el entrenamiento: RSI, MACD, EMAs, Bandas de Bollinger, ATR, momentum, volatilidad local y variables booleanas derivadas (RSI oversold/overbought, price_above_EMA, etc.).

Predicción y generación de señal:

El modelo supervisado predice la probabilidad de una señal positiva. Si la probabilidad supera el threshold de 0.6, se genera una orden de compra simulada.

En caso contrario, no se ejecuta ninguna acción y el sistema permanece en estado pasivo.

Gestión de posición:

El bot mantiene memoria del estado mediante tres variables globales:

- position: indica si hay una posición abierta (LONG o None).
- entry_price: almacena el precio de entrada.
- entry_time: registra la hora de la apertura.

Mientras la posición permanece abierta, el sistema evalúa continuamente tres condiciones de salida:

- Take Profit (TP): cierre automático al +2.0% del precio de entrada.
- Stop Loss (SL): cierre automático al -1,0% del precio de entrada.
- Cierre por tiempo: si han transcurrido 8 horas sin alcanzar TP ni SL.

Registro y trazabilidad:

Cada operación ejecutada (BUY o SELL) se registra en el archivo data/live_trades.csv, con información detallada: timestamp, precio, probabilidad de señal, acción y motivo de cierre (señal ML, TP, SL o tiempo máximo).

Modo DRY-RUN:

El sistema se ejecuta en modo de simulación controlada (DRY_RUN=True), sin enviar órdenes reales a la API de Binance, pero reproduciendo fielmente la lógica operativa y permitiendo validar el comportamiento del modelo en condiciones de mercado.

4.5.2. Front end

La solución desarrollada incorpora una plataforma web interactiva que permite al usuario configurar de manera intuitiva todos los parámetros relevantes para el entrenamiento y evaluación de modelos de trading algorítmico. A través de diferentes secciones, el usuario puede seleccionar el símbolo de mercado, la temporalidad de las velas, el periodo histórico a utilizar, así como definir las fechas concretas para el entrenamiento y el backtesting. Además, la interfaz facilita la elección de la variable objetivo del modelo y el umbral de error, permitiendo adaptar el sistema a distintos escenarios y estrategias de análisis.

Uno de los aspectos más destacados de la plataforma es la posibilidad de personalizar en detalle tanto los indicadores técnicos (como RSI, MACD, Bollinger Bands, ATR, Momentum, y EMAs) como los parámetros de gestión de riesgo y capital. El usuario puede ajustar valores como el take profit, stop loss, capital inicial, comisiones, slippage y modo de operación (testnet o simulación). Asimismo, la interfaz permite seleccionar el tipo de modelo de machine learning (por ejemplo, Gradient Boosting o Random Forest), así como sus hiperparámetros principales (número de estimadores, learning rate, submuestreo, validación cruzada, etc.), lo que aporta una gran flexibilidad y control sobre el proceso de entrenamiento.

Una vez definida la configuración deseada, la plataforma ejecuta el entrenamiento del modelo con los parámetros seleccionados y muestra los resultados obtenidos de forma clara y estructurada. El usuario puede visualizar la configuración empleada, el conjunto de variables (features) utilizadas, y seleccionar qué modelos entrenar. Tras el proceso, se presentan los resultados del backtesting, incluyendo métricas clave y visualizaciones, lo que facilita la comparación de estrategias y la toma de decisiones informadas.

Esta integración de front end aporta una capa de usabilidad y transparencia fundamental para la experimentación y validación de sistemas de trading algorítmico basados en inteligencia artificial.

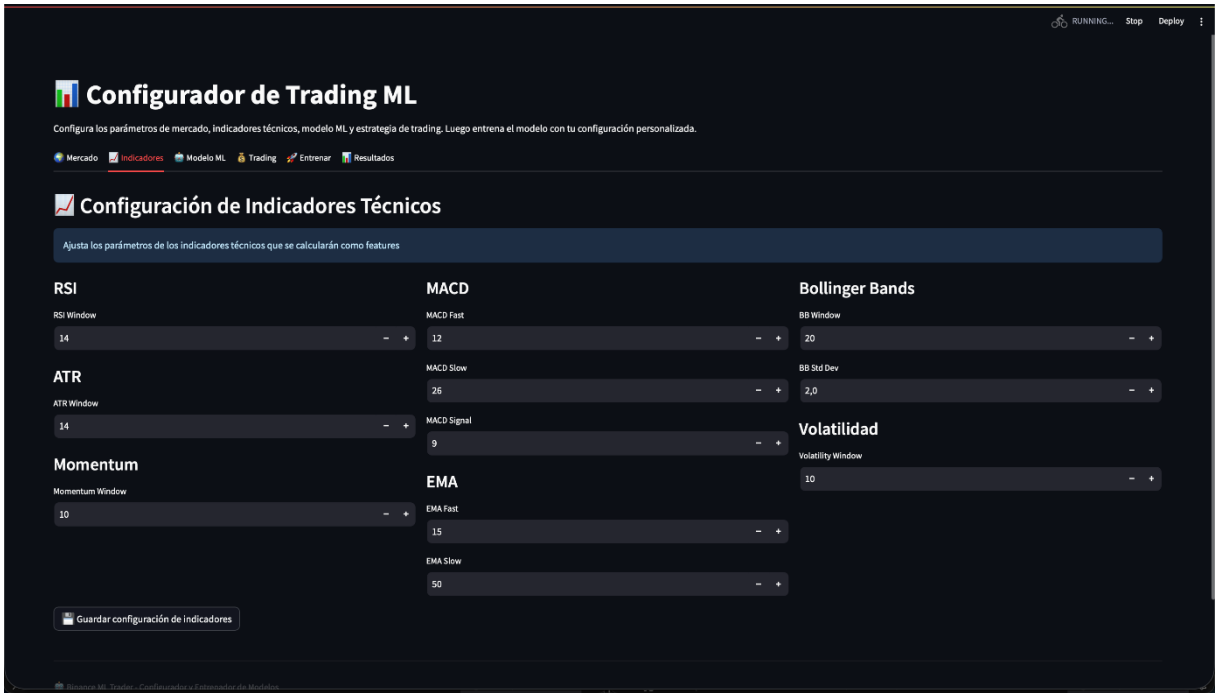


Ilustración 11 Variables de entrenamiento del modelo

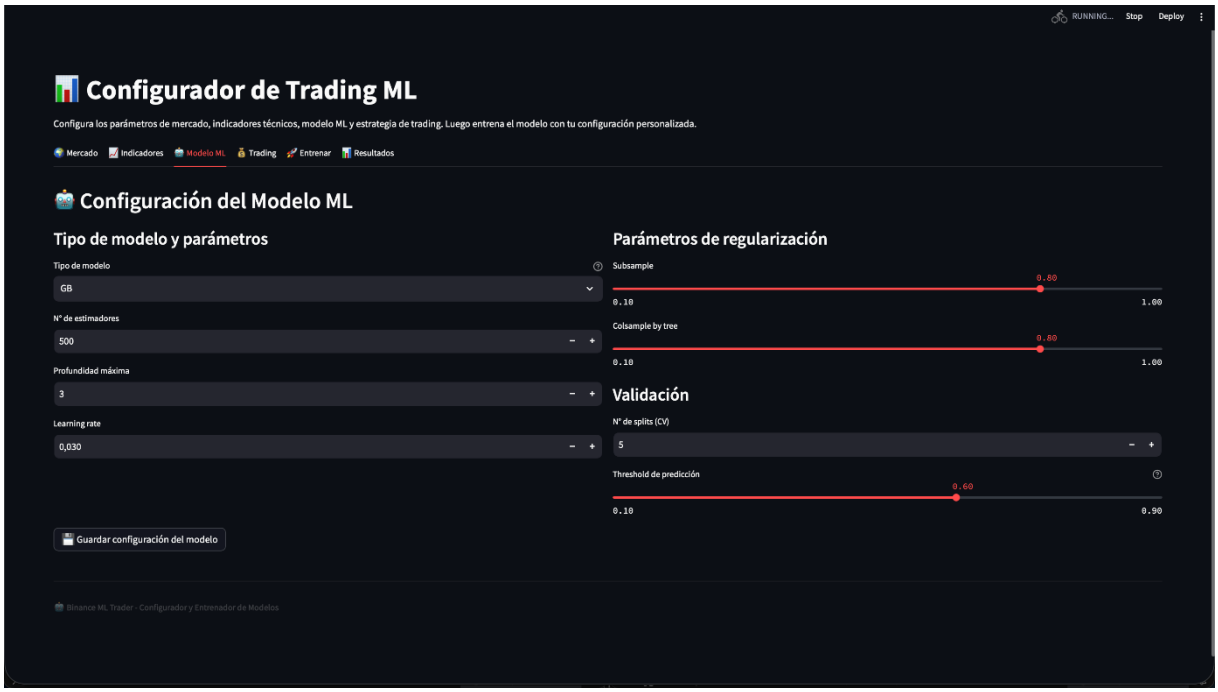


Ilustración 12 Variables de entrenamiento del modelo 2

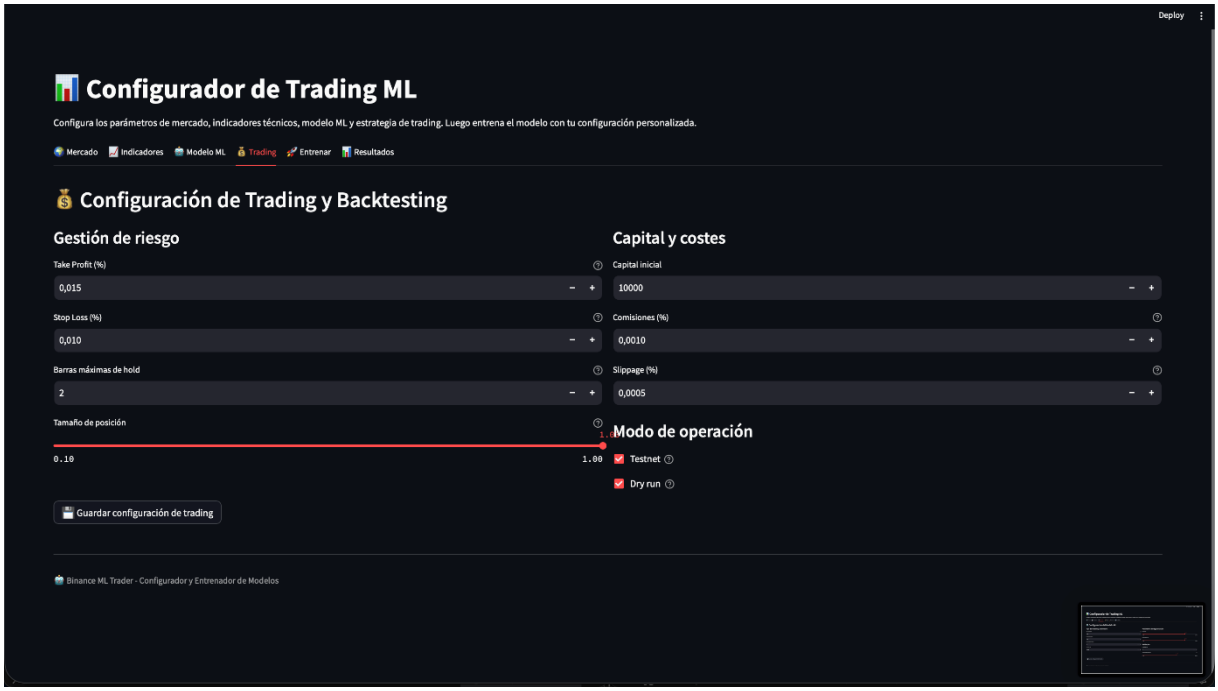


Ilustración 13 Variables de entrenamiento del modelo 3

4.6. VERSIONADO Y CI/CD CON GITHUB

Dado el carácter iterativo del proyecto, se utilizó Git y una plataforma de repositorio remoto (GitHub) para llevar control de versiones del código y facilitar la colaboración. Cada componente del proyecto (scripts, notebooks, documentos) puede ser incrementalmemente desarrollado con commits frecuentes describiendo los cambios, por ejemplo: “Entrenar el modelo con Random Forest”. De esta forma se mantiene un historial de cómo evolucionó la base de código.

En el repositorio de GitHub se crearon ramas para separar etapas o características (ej: una rama dev para desarrollo activo y la rama main para la versión estable). Al finalizar el TFM, se podría hacer un release etiquetado (tag) señalando la versión final del proyecto.

Además, se exploró la configuración de CI/CD (Integración Continua/Despliegue Continuo) para automatizar pruebas y despliegues. En GitHub, esto típicamente se implementa mediante GitHub Actions: flujos de trabajo definidos en archivos YAML que se ejecutan en servidores de GitHub ante ciertos eventos (como push o pull request). En el contexto de este

proyecto, se imaginó un workflow donde, por ejemplo, ante cada push a la rama principal, se corran algunos tests automáticos.

En síntesis, el versionado con Git/GitHub aportó:

- Seguridad ante pérdida de código (repositorio remoto de respaldo).
- Capacidad de revertir cambios si alguna modificación empeoraba resultados.
- Colaboración asíncrona (en caso de que se consultara con tutores o colegas, podrían revisar el repo, hacer forks, etc.).
- Base para CI/CD donde, si a futuro se despliega la estrategia, se puede automatizar su entrega (por ejemplo, al hacer un cambio aprobado en main, que automáticamente se actualice la instancia donde corre el bot en vivo).

Estas prácticas de CI/CD no se explotan al máximo en un TFM académico, pero sentar su estructura es valioso para mostrar que el proyecto está preparado para producción en entornos reales, donde la robustez del pipeline de código es tan importante como la precisión del modelo.

5. Conclusiones y trabajo futuro

5.1. Conclusiones

El desarrollo de este sistema de trading algorítmico ha supuesto la culminación práctica de los conocimientos adquiridos a lo largo del Máster en Inteligencia Artificial aplicada a finanzas, integrando todas las fases clave del ciclo de vida de un proyecto de machine learning en mercados financieros reales.

En primer lugar, la construcción de un pipeline completo —desde la adquisición y limpieza de datos históricos de la API de Binance hasta la ingeniería de características y la validación temporal— ha permitido asentar una base sólida para el modelado predictivo. El análisis exploratorio y la correcta gestión de valores atípicos han sido fundamentales para garantizar la calidad del dataset y la fiabilidad de los resultados.

En la fase de modelado, se entrenaron y evaluaron cinco modelos representativos de distintas familias de algoritmos: Random Forest, Gradient Boosting, red neuronal densa, LSTM y ARIMA. Esta comparación ha permitido analizar de forma crítica el rendimiento de cada enfoque en la predicción de señales de trading. Los resultados muestran que las redes neuronales (especialmente la LSTM y la red densa) han superado a los modelos clásicos en métricas como F1-score y recall, mientras que los modelos basados en árboles y ARIMA han mostrado limitaciones en la detección de señales positivas. Esta evidencia empírica refuerza la importancia de adaptar la selección de modelos al tipo de problema y a la naturaleza de los datos.

La comparación sistemática de cinco modelos ha puesto de manifiesto la complejidad de la evaluación en sistemas de trading algorítmico, donde las métricas de clasificación y las métricas financieras no siempre convergen. Aunque las redes neuronales (LSTM y red densa) alcanzaron los mejores F1-scores (0,68-0,69) y recall (>96%), y Gradient Boosting obtuvo el mejor Sharpe Ratio (5,86) y ROI (946%), se seleccionó Random Forest como modelo final por su combinación de:

- Mayor precisión (0,542), reduciendo falsos positivos y operaciones erróneas.
- Menor drawdown máximo (-9,23%), crítico para la preservación de capital.
- Menor número de operaciones (826), reduciendo costes y exposición.

- Sharpe Ratio robusto (4,51), superando ampliamente al Buy & Hold (2,18).

Esta decisión ejemplifica el principio de que la selección de modelos en finanzas debe priorizar criterios de riesgo-retorno específicos del perfil del inversor, no únicamente métricas de clasificación. Se reconoce que Random Forest sacrifica rentabilidad absoluta (333% vs 946% de GB) a cambio de mayor estabilidad, representando una estrategia defensiva apropiada para las fases iniciales de despliegue, pero no necesariamente óptima para la maximización de retornos.

Limitación crítica identificada: El recall extremadamente bajo (0,056) indica que el modelo pierde el 94,4% de oportunidades reales. El trabajo futuro debe explorar el ajuste de umbrales, ensamblado de modelos y recalibración de etiquetas para mejorar este balance sin comprometer la estabilidad.

Asimismo, la implementación de buenas prácticas de automatización, control de versiones y despliegue reproducible (MLOps) ha permitido trasladar el trabajo desde un entorno experimental a una arquitectura operativa y escalable. El uso de herramientas como GitHub, la modularización del código y la gestión de logs han facilitado la trazabilidad y la colaboración, alineando el proyecto con los estándares profesionales del sector.

En conjunto, este trabajo ha supuesto una experiencia integral de aprendizaje aplicado, consolidando competencias técnicas, metodológicas y analíticas en un contexto realista y exigente. La comparación sistemática de modelos, la reflexión crítica sobre sus resultados y la integración de todo el pipeline en un entorno reproducible constituyen una aportación tangible y extrapolable a futuros desarrollos en el ámbito del trading algorítmico y la inteligencia artificial financiera.

5.2. Líneas de trabajo futuro

El trabajo desarrollado constituye una base sólida para la evolución de sistemas de trading algorítmico en mercados de criptomonedas. De cara al futuro, una de las principales líneas de mejora será incrementar la velocidad de análisis y procesamiento de datos, optimizando tanto la arquitectura del sistema como el uso de recursos computacionales. Esto permitirá reducir la latencia en la toma de decisiones y responder de forma aún más eficiente a las condiciones cambiantes del mercado.

Otra perspectiva relevante es ampliar el abanico de modelos de inteligencia artificial utilizados, incorporando nuevas arquitecturas de deep learning, algoritmos de aprendizaje por refuerzo y técnicas de ensemble, con el objetivo de mejorar la capacidad predictiva y la robustez del sistema frente a escenarios de alta volatilidad. La experimentación con modelos más avanzados permitirá comparar su rendimiento y seleccionar aquellos que aporten mayor valor en términos de rentabilidad ajustada al riesgo.

Finalmente, una evolución natural del proyecto será empaquetar la solución en una aplicación integral, que facilite la gestión, monitorización y ejecución automatizada de estrategias de trading desde una interfaz intuitiva y segura. Este desarrollo permitirá escalar el sistema y adaptarlo tanto para usuarios individuales como para instituciones, favoreciendo su adopción en entornos reales. Además, se plantea la integración de módulos de explicabilidad y gobernanza, así como la adaptación del sistema a otros mercados financieros, consolidando la aportación realizada como referencia para futuras investigaciones y desarrollos tecnológicos en el ámbito del trading algorítmico.

5.3. REFLEXIÓN FINAL

La historia de la inversión siempre ha sido la historia de cómo los humanos enfrentan su propia incertidumbre. Durante décadas, los mercados financieros han sido una proyección colectiva de nuestras emociones: miedo, codicia, esperanza. La inteligencia artificial no elimina esa dimensión humana, pero la redefine. La desplaza del instante impulsivo al plano estructural, donde las decisiones ya no dependen de la intuición del gestor, sino de la capacidad del sistema para aprender, adaptarse y corregirse a sí mismo.

El desarrollo de este sistema de trading algorítmico demuestra que la frontera entre la inteligencia humana y la inteligencia estadística se está difuminando. El modelo no sustituye al inversor, sino que amplía su campo de visión. Permite pensar en mil escenarios simultáneamente, detectar correlaciones invisibles y responder en milisegundos donde el juicio humano tardaría horas. En esencia, la IA convierte al inversor en arquitecto de sistemas de decisión, más que en ejecutor de operaciones. Esa transición —de la intuición al diseño— marca el cambio de paradigma en la gestión financiera contemporánea.

El mercado de criptomonedas, por su naturaleza volátil y descentralizada, actúa como un laboratorio acelerado del futuro financiero. Allí donde los modelos clásicos fallan, los sistemas de aprendizaje automático prosperan. Lo que hoy se ensaya con Bitcoin o Ethereum se convertirá mañana en estándar en la gestión de carteras tradicionales, en los departamentos de riesgo y en los fondos soberanos. La IA no es solo una ventaja competitiva: es una nueva capa de realidad financiera que obliga a repensar cómo entendemos el riesgo, el valor y la confianza.

A medida que más bots están en funcionamiento, la proporción de humanos tomando decisiones directas en los mercados disminuye. El pulso de la inversión se automatiza, y el mercado se convierte en un ecosistema donde los algoritmos interactúan, compiten y aprenden unos de otros, desplazando progresivamente la intervención humana hacia roles de supervisión, diseño y control.

Sin embargo, el avance técnico sin reflexión ética puede producir modelos tan opacos como los sesgos que pretendemos eliminar. Por eso, la responsabilidad del ingeniero financiero no es solo construir algoritmos más rentables, sino también más transparentes, auditables y alineados con los objetivos humanos. Un sistema de trading debería poder explicar por qué compra, no solo cuándo.

Este proyecto es, en cierto modo, una semilla de ese futuro. Integra datos, aprendizaje y control del riesgo bajo una arquitectura reproducible y escalable. Pero lo más relevante no está en su precisión predictiva, sino en lo que sugiere: la posibilidad de que el conocimiento financiero deje de ser reactivo y se convierta en adaptativo. Que las máquinas aprendan no solo de los precios, sino también de sus propios errores.

La inteligencia artificial no sustituye al inversor. Lo obliga a evolucionar. Y en ese proceso, lo más valioso no será el modelo que prediga el próximo movimiento del mercado, sino el marco mental que enseñe al humano a convivir con la incertidumbre sin miedo, con método y con propósito.

Referencias bibliográficas

Balsategui, I., Gorjón, S., & Marqués, J. (s.f.). *ARTIFICIAL INTELLIGENCE IN THE FINANCIAL SYSTEM:.*

Binance. (2025). *IA en criptomonedas*. Obtenido de Binance.com: https://www.binance.com/es-LA/square/post/540726/

Cortés Durán, J., & Hernández Vega, J. (2021). *Eficiencia del mercado de criptomonedas y planteamiento de estrategias de trading basadas en arbitraje y machine learning.*

Dominiak, A., & Duersch, P. (2024). Choice under uncertainty and cognitive load. *Journal of Risk en Uncertainty*, 133-161.

Doran, G. T. (1981). There's a S.M.A.R.T. way to write management's goals and objectives. *Management Review (AMA FORUM), 70*, 35-36.

Flexfunds. (2024). *Explorando los gigantes de los hedge funds a nivel mundial.* . Obtenido de Flexfunds: https://www.flexfunds.com/es/flexfunds/explorando-los-gigantes-de-los-hedge-funds-a-nivel-mundial/

Gaudeul, A., & Gianetti, C. (s.f.). Beyond performance: Exploring trade-offs in the design of financial algorithms. *Journal of Behavioral and Experimental Finance*, 48.

Gawde, A., & Jawale, A. (2024). Ethical Considerations In Algorithmic Trading: Recent Developments, Challenges, And The Path Forward. *International Journal of Creative Research Thoughts*.

Goodell, J., Kumar, S., Weng, M., & Pattnaik , D. (2021). Artificial intelligence and machine learning in finance: Identifying foundations, themes, and research clusters from bibliometric analysis. *Journal of Behavioral and Experimental Finance*.

Housel, M. (2020). The Psychology of Money.

Kahneman, D. (2011). *Thinking, Fast and Slow.* Farrar, Straus and Giroux.

Mordor Intelligence. (2025). *Algorithmic Trading Market Size & Share Analysis - Growth Trends & Forecasts (2025 - 2030)*. Obtenido de Mordor Intelligence: https://www.mordorintelligence.com/industry-reports/algorithmic-trading-market

PWC. (2024). *Global Crypto Report*. Obtenido de PWC: https://www.pwc.com/gx/en/industries/financial-services/fintech/crypto.html

Warade, P. (2025). *Algorithmic Trading Market Size & Outlook, 2025-2033.* Straits Research.

Zhang, Z., & Chen, J. (s.f.). The Evolution of Algorithmic Trading in Cryptocurrency Markets. *Quantitative Finance*.