Campus Santa Fe

Predicción de tendencias bursátiles con una arquitectura híbrida CNN–BiLSTM–Transformer

Materia

Analítica de Datos y herramientas de Inteligencia Artificial II

Profesores:

Edoardo Bucheli Susarrey

Esteban Castillo Juarez

Alumnos:

Miguel Ángel Noriega Bedolla - A01658032

Marcos Dayan Mann - A01782876

Fecha:

4 de Diciembre de 2025

Índice

Predicción de tendencias bursátiles con una arquitectura híbrida CNN–BiLSTM–Transformer

1. **Abstract**
2. **Introducción**
   - 2.1. Contexto y Motivación
   - 2.2. Planteamiento del Problema
   - 2.3. Objetivos
   - 2.4. Contribuciones
3. **Marco Teórico**
   - 3.1. Artículo Base y Relevancia en la Literatura
   - 3.2. Arquitectura del Modelo CNN–BiLSTM–Attention de Zhang et al.
   - 3.3. Fundamentos de Series Temporales Financieras
   - 3.4. Convolutional Neural Networks (CNN)
   - 3.5. Long Short-Term Memory (LSTM) y Variantes Bidireccionales
   - 3.6. Transformer Attention Mechanisms
   - 3.7. Feature Selection y Separabilidad de Clases
   - 3.8. Class Imbalance y Oversampling
4. **Metodología**
   - 4.1. Fuente de Datos
   - 4.2. Variables y Features
   - 4.3. Preprocesamiento de Datos
   - 4.4. Feature Selection por Separabilidad Inter-clase
   - 4.5. Creación de Secuencias Temporales
   - 4.6. Manejo del Desbalance de Clases
   - 4.7. División Temporal del Dataset
5. **Arquitectura del Modelo**
   - 5.1. Diseño General
   - 5.2. Bloque 1 – CNN
   - 5.3. Bloque 2 – BiLSTM
   - 5.4. Bloque 3 – Transformer
   - 5.5. Bloque 4 – Dense Classification Head
   - 5.6. Parámetros del Modelo
   - 5.7. Configuración de Entrenamiento
   - 5.8. Estrategia de Entrenamiento
   - 5.9. Regularización
   - 5.10. Métricas de Evaluación
6. **Infraestructura y Herramientas**
   - 6.1. Stack Tecnológico
7. **Resultados**
   - 7.1. Features Seleccionadas
   - 7.2. Análisis Visual de la Relación entre Features y Precio de MSFT
   - 7.3. Desempeño Global del Modelo en el Conjunto de Prueba
   - 7.4. Comparación de Versiones Previas
   - 7.5. Análisis de Convergencia
   - 7.6. Desempeño Temporal en Datos Nunca Vistos (Test)
   - 7.7. Análisis de Confianza de las Predicciones
   - 7.8. Validación Estadística frente a Baselines
8. **Implicaciones Prácticas**
   - 8.1. Rol del Modelo en un Sistema de Trading Algorítmico
   - 8.2. Uso de la Confianza como Herramienta de Gestión de Riesgo
   - 8.3. Alcances y Limitaciones en un Entorno Real
9. **Conclusiones y Trabajo Futuro**
   - 9.1. Principales Hallazgos
   - 9.2. Limitaciones y Líneas de Trabajo Futuro
10. **Apéndices**
    - 10.1. Configuración del Entorno
    - 10.2. Reproducibilidad
    - 10.3. Hardware Utilizado
    - 10.4. Métricas Detalladas por Epoch
11. **Referencias**

1. Abstract

Este trabajo aborda la predicción direccional del retorno a 10 días de la acción de Microsoft (MSFT) como un problema de clasificación binaria (bajista/alcista), utilizando datos diarios y fundamentales de Bloomberg Terminal para el periodo 2000–2025. Se propone una arquitectura híbrida CNN–BiLSTM–Transformer para series temporales financieras multivariadas, que extiende el modelo CNN–BiLSTM–Attention de *Zhang et al. (2023)*, mediante la incorporación de un bloque Transformer con atención multi-cabeza y attention pooling para capturar dependencias temporales globales. A partir de un conjunto amplio de variables fundamentales, macroeconómicas y de mercado, se aplica un pipeline de preprocesamiento y selección de características basado en separabilidad inter-clase, que reduce la dimensionalidad a 50 features relevantes y corrige el desbalance de clases mediante oversampling. Bajo un esquema de validación estrictamente temporal, el modelo alcanza un F1-score de 84.3 % en el conjunto de prueba, superando baselines triviales, modelos clásicos y configuraciones previas del propio modelo. Los resultados muestran que la combinación de arquitecturas híbridas profundas y selección estructurada de features genera señales predictivas robustas, potencialmente integrables en estrategias de trading algorítmico.

2. Introducción

2.1. Contexto y Motivación

Los mercados financieros representan sistemas dinámicos complejos caracterizados por alta volatilidad, relaciones no lineales multivariadas, y ruido estocástico inherente. La predicción de tendencias bursátiles constituye uno de los problemas más desafiantes en el ámbito del Machine Learning aplicado a finanzas, dada la naturaleza no estacionaria de las series temporales financieras y la presencia de factores exógenos impredecibles.

El presente proyecto surge de la necesidad de desarrollar modelos predictivos robustos que puedan asistir en la toma de decisiones de inversión mediante la anticipación de movimientos direccionales en activos de alta capitalización. Específicamente, se enfoca en Microsoft Corporation (MSFT), una de las empresas más representativas del sector tecnológico en mercados estadounidenses, utilizando datos históricos obtenidos de Bloomberg Terminal durante el período 2000-2025.

2.2. Planteamiento del Problema

El problema central consiste en predecir la dirección del retorno futuro a 10 días del precio de cierre ajustado de MSFT, formulado como una tarea de clasificación binaria:

- Clase 0 (Bajista): Retorno futuro < 0%
- Clase 1 (Alcista): Retorno futuro ≥ 0%

Esta formulación simplificada, resultado de iteraciones previas, permite capturar la esencia del problema de trading (¿comprar o vender?) sin la complejidad de predecir magnitudes exactas de movimiento, las cuales introducen ruido significativo y degradan el rendimiento predictivo.

2.3. Objetivos

Objetivo General: Desarrollar e implementar una arquitectura híbrida de Deep Learning capaz de predecir tendencias bursátiles con accuracy superior a 80% en datos de prueba no vistos.

Objetivos Específicos:

1. Diseñar e implementar un pipeline de preprocesamiento robusto para datos financieros multivariados
2. Desarrollar una arquitectura híbrida CNN-BiLSTM-Transformer optimizada para series temporales financieras
3. Implementar estrategias de feature selection basadas en separabilidad inter-clase
4. Aplicar técnicas de balanceo de clases mediante oversampling aleatorio
5. Validar el modelo mediante evaluación en conjunto de prueba temporal con métricas múltiples
6. Comparar rendimiento contra versiones previas y baselines aleatorios

2.4. Contribuciones

Las principales contribuciones de este trabajo son:

1. Arquitectura Híbrida Optimizada: Diseño de una red neuronal profunda que combina tres paradigmas complementarios (CNN, BiLSTM, Transformer) específicamente adaptados para series temporales financieras
2. Metodología de Simplificación Progresiva: Demostración empírica de que la reducción de granularidad del problema (12 clases → 2 clases) produce mejoras dramáticas en rendimiento predictivo
3. Feature Engineering Riguroso: Implementación de selección de features mediante cálculo de separabilidad estadística, reduciendo dimensionalidad de 224 a 50 variables con mejora en accuracy
4. Pipeline de Entrenamiento Reproducible: Sistema completo documentado y versionado que permite replicación y extensión del trabajo

3. Marco Teórico

3.1. Artículo Base y Relevancia en la Literatura

El punto de partida metodológico de este proyecto es *Zhang, Ye y Lai (2023), "Stock Price Prediction Using CNN-BiLSTM-Attention Model", publicado en Mathematics (vol. 11, nº 9, art. 1985).* Es un artículo de acceso abierto y fue seleccionado como *Editor's Choice* por la revista, lo que respalda su relevancia en modelos cuantitativos aplicados a mercados financieros. El trabajo propone un modelo híbrido CNN-BiLSTM-Attention para la predicción de precios e índices bursátiles, atendiendo la naturaleza no lineal, ruidosa y con dependencias temporales de las series financieras.

Empíricamente, Zhang et al. usan datos diarios (apertura, máximo, mínimo, cierre), volumen y rendimientos para varios índices chinos y extienden el análisis a ocho índices internacionales en el periodo 2011–2021. Comparan su arquitectura contra baselines LSTM, CNN-LSTM y CNN-LSTM con atención y evalúan con métricas de error RMSE, MAPE y medidas de ajuste. En prácticamente todos los casos, CNN-BiLSTM-Attention obtiene los menores errores; por ejemplo, en CSI 300 reduce el RMSE del mejor alternativo de 76.454 a 64.8 y mejora el ajuste, evidenciando una ganancia relevante en precisión.

La solidez del artículo también se refleja en su impacto posterior: bases bibliográficas como Google Scholar, Semantic Scholar y RePEc reportan alrededor de un centenar de citas desde 2023, y la arquitectura CNN-BiLSTM-Attention ha sido retomada y extendida en trabajos recientes (p. ej., volatilidad con frecuencia mixta, integración de indicadores macroeconómicos, incertidumbre de política económica y análisis de sentimiento). Además, en sus conclusiones los autores sugieren como líneas futuras la incorporación de información multi-fuente y el uso de modelos de última generación más allá del esquema CNN-BiLSTM.

Nuestro proyecto es una extensión natural de ese enfoque: mantiene la combinación CNN + modelos recurrentes + atención para capturar patrones locales y dependencias temporales, e **introduce bloques Transformer y un diseño experimental ampliado**, alineado con las recomendaciones y limitaciones señaladas en *Zhang et al. (2023).*

3.2. Arquitectura del Modelo CNN-BiLSTM-Attention de Zhang et al.

Desde el punto de vista técnico, *Zhang et al. (2023)* plantean una arquitectura secuencial compuesta por tres bloques principales: CNN, BiLSTM y un mecanismo de atención, rematados por una capa densa de salida. El modelo recibe como entrada ventanas temporales de datos diarios del índice (precio de apertura, máximo, mínimo, cierre, volumen, turnover y rendimiento), que previamente son normalizados a [0,1] mediante Min–Max scaling.

En un primer paso, la capa convolucional 2D, seguida de pooling y dropout, extrae patrones locales no lineales de la serie y reduce la dimensionalidad, generando mapas de características más compactas pero informativas. Sobre estas características actúa el bloque BiLSTM, que aprende las dependencias de largo plazo en ambas direcciones temporales (pasado y “futuro” dentro de la ventana) y produce una secuencia de estados ocultos enriquecidos. A continuación, el mecanismo de atención asigna pesos diferenciados a esos estados, concentrándose en los momentos más relevantes para la predicción y atenuando información redundante; el vector atendido resultante se alimenta finalmente a una capa totalmente conectada que genera la predicción del precio de cierre. El entrenamiento se realiza usando función de activación ReLU, pérdida MSE y el optimizador Adam, incorporando capas de dropout como técnica de regularización para mitigar el sobreajuste y mejorar la capacidad de generalización del modelo.

En cuanto a la configuración temporal, los autores emplean una estrategia de predicción directa de un solo paso (single-step): a partir de cada ventana histórica, el modelo predice exclusivamente el precio de cierre del día siguiente, es decir, un horizonte de predicción de un día. El conjunto de datos se construye con frecuencia diaria, utilizando 2,675 días de negociación del índice CSI 300 entre el 4 de enero de 2011 y el 31 de diciembre de 2021, y se divide en subconjuntos de entrenamiento, validación y prueba en una proporción 60%–20%–20%. Esta configuración, junto con el uso de una ventana deslizante sobre toda la serie, permite generar una predicción para prácticamente cada jornada de mercado y evaluar de forma sistemática el desempeño del modelo bajo condiciones realistas de predicción a corto plazo.

Consideramos que *Zhang et al. (2023)* constituyen el cimiento teórico y empírico de nuestro modelo:nos proveen de una arquitectura base probada, de un protocolo de comparación

con modelos benchmark y de una agenda clara de extensiones, dentro de la cual se inserta el uso de Transformers que proponemos en este trabajo.

3.3. Fundamentos de Series Temporales Financieras

Las series temporales financieras presentan características únicas que las distinguen de otras secuencias temporales:

- No Estacionaridad: La media, varianza y autocorrelación varían con el tiempo
- Heterocedasticidad: La volatilidad no es constante en diferentes períodos
- Asimetría en Volatility Clustering: Períodos de alta volatilidad tienden a agruparse
- Heavy Tails: Distribución de retornos con colas más pesadas que la distribución normal
- Microestructura de Mercado: Efectos de liquidez, spreads bid-ask, y fragmentación

Estas propiedades requieren técnicas especializadas de modelado que capturen tanto dependencias de corto plazo como patrones estructurales de largo plazo.

3.4. Convolutional Neural Networks (CNN)

Las CNNs, originalmente diseñadas para visión computacional, han demostrado eficacia en series temporales mediante su capacidad de detectar patrones locales invariantes a traslación temporal. En contexto financiero, las capas convolucionales identifican:
Patrones de micro-estructura: Configuraciones recurrentes en ventanas temporales cortas (3-5 días)
Features jerárquicas: Combinaciones de indicadores técnicos que señalan cambios de régimen
Invarianza temporal local: Patrones que aparecen en diferentes momentos del horizonte temporal

3.5. Long Short-Term Memory (LSTM) y Variantes Bidireccionales

Las redes LSTM solucionan el problema del vanishing gradient en RNNs tradicionales mediante la introducción de puertas (gates) que controlan el flujo de información:
Forget Gate: Decide qué información descartar de la memoria celular
Input Gate: Determina qué nueva información almacenar
Output Gate: Controla qué parte de la memoria celular exponer
La variante Bidirectional LSTM procesa la secuencia en ambas direcciones temporales, permitiendo que cada time step tenga contexto tanto de eventos pasados como futuros dentro de la ventana de observación. Esto es particularmente relevante en análisis post-hoc donde el objetivo es detectar patrones que precedieron un movimiento, sin restricciones causales estrictas.

3.6. Transformer Attention Mechanisms

Los mecanismos de atención, introducidos en la arquitectura Transformer, permiten al modelo “enfocarse” en diferentes partes de la secuencia de entrada dinámicamente. A diferencia de LSTMs que procesan secuencialmente, la atención calcula relaciones entre todos los pares de timesteps simultáneamente mediante:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Figura 1. Fórmula de TAM

Donde Q (queries), K (keys), y V (values) son proyecciones lineales de la entrada.

3.7. Feature Selection y Separabilidad de Clases

La selección de features se fundamenta en el concepto de separabilidad inter-clase, definida como:

$$S_t = \left|\mu_{t,0} - \mu_{t,1}\right|$$

Figura 2. Resta de promedio de feature vs promedio de label

3.8. Class Imbalance y Oversampling

El desbalanceo de clases (43.5% Bajista / 56.5% Alcista en datos originales) sesga al modelo hacia la clase mayoritaria.

4. Metodología

4.1 Fuente de Datos

Los datos provienen de **Bloomberg Terminal**, plataforma profesional utilizada por instituciones financieras a nivel global y que provee información de alta calidad y consistencia histórica. El dataset construido para este proyecto comprende:

- **Activo principal:** Microsoft Corporation (MSFT)
- **Periodo temporal:** enero de 2000 – noviembre de 2025 (≈25 años)
- **Frecuencia:** diaria
- **Total de observaciones:** 7,197 días hábiles de trading

Este horizonte largo permite cubrir múltiples ciclos de mercado (crecimiento, crisis, recuperación) y evaluar la robustez del modelo bajo distintos regímenes.

4.2 Variables y Features

El dataset original contiene **224 features** que se organizan en tres grandes grupos:

1. **Features fundamentales (Bloomberg Quarterly Data)**
   Derivadas de los estados financieros y métricas contables de Microsoft:
   - **Estados financieros:** Balance Sheet, Income Statement, Cash Flow
   - **Métricas de valoración:** PE Ratio, PB Ratio, Enterprise Value
   - **Indicadores de rentabilidad:** ROE, ROA, Profit Margin
   - **Métricas de liquidez y estructura:** Current Ratio, Quick Ratio, Cash Position, Net Debt, entre otros
2. **Features técnicas**
   Calculadas a partir de precios y volúmenes:
   - **Indicadores de momentum:** RSI, MACD, Stochastic Oscillator
   - **Indicadores de volatilidad:** Bollinger Bands, Average True Range (ATR)
   - **Medias móviles:** SMA(5, 10, 20, 50, 200), EMA(12, 26)
   - **Volumen y derivados:** Volume, Volume SMA, On-Balance Volume
3. **Features de mercado y macroeconómicas**
   Capturan el contexto externo en el que se mueve MSFT:
   - **Índices de referencia:** SPY (S&P 500), QQQ (NASDAQ-100)
   - **Competidores tecnológicos:** AAPL, AMZN, IBM, NVDA
   - **Indicadores macro y de riesgo:** VIX, USGG10YR (Treasury 10Y), DXY (US Dollar Index), entre otros

La **variable objetivo** se define como el **retorno futuro a 10 días** del precio de cierre ajustado de MSFT. A partir de este retorno continuo r_{t+10} , se construye una etiqueta binaria:

- **Clase 0 – Bajista:** $r_{t+10} < 0$
- **Clase 1 – Alcista:** $r_{t+10} \geq 0$

Este planteamiento convierte el problema en una tarea de **clasificación direccional** (sube vs baja) alineada con decisiones prácticas de trading.

4.3 Preprocesamiento de Datos

El pipeline de preprocesamiento (implementado en `src/preprocessing/pipeline.py` y `regenerate_dataset_focused.py`) consta de las siguientes etapas:

1) Limpieza y manejo de valores faltantes

- Se eliminan aquellas features con más de 30% de missingness, al considerarse que aportan señal insuficiente para el modelo.
- Para huecos esporádicos en series relevantes, se aplica imputación tipo forward-fill (propagar el último valor observado) siempre que los gaps sean cortos, preservando la coherencia temporal.

2) Tratamiento de outliers

- Para reducir el impacto de valores extremos no representativos (errores de registro, spikes puntuales), se aplica winsorización al percentil 1–99 en variables continuas.
- Esto permite mantener la forma general de la distribución sin que unos pocos valores atípicos dominen la escala.

3) Validación de consistencia temporal

- Se verifica el alineamiento temporal entre:
    - datos fundamentales trimestrales, y
    - series diarias de precios y mercado,
      asegurando que no existan *look-ahead biases* (uso de información futura) al asociar estados financieros con días de trading.

En secciones posteriores se detalla el feature engineering, el esquema de selección de features por separabilidad inter-clase, la construcción de secuencias temporales (ventanas de 120 días) y la arquitectura CNN–BiLSTM–Transformer utilizada para el entrenamiento del modelo.

4.4 Feature Selection por separabilidad inter-clase

Una vez generadas y normalizadas las 224 features, se aplica un filtro univariante basado en **s**eparabilidad inter-clase para reducir dimensionalidad antes de entrenar el modelo profundo:

1. La selección se calcula exclusivamente sobre el conjunto de entrenamiento, para evitar *data leakage*.
2. Para cada feature $i \in \{1,\ldots,224\}$ se computan las medias condicionadas por clase:

- $\mu_{i,\text{bajista}} = \text{mean}(X_i \mid y = 0)$
- $\mu_{i,\text{alcista}} = \text{mean}(X_i \mid y = 1)$

3. Se define el score de separabilidad como la diferencia absoluta entre medias:

$$S_i = \left|\mu_{i,\text{bajista}} - \mu_{i,\text{alcista}}\right|$$

Intuitivamente, cuanto mayor es S_i, más se desplaza la distribución de la feature entre mercados bajistas y alcistas, y mayor capacidad discriminativa aporta de forma individual.

4. Las features se ordenan de forma descendente por SiSi y se seleccionan las top-50 con mayor separabilidad.
5. El índice de estas 50 columnas se guarda y se aplica de forma consistente para transformar los conjuntos de validación y prueba, de modo que todos los splits comparten exactamente el mismo subespacio de características.

Este procedimiento reduce la dimensionalidad de 224 a 50 variables (≈78% de reducción) antes de la fase de modelado, disminuyendo el riesgo de sobreajuste y el coste computacional. Además, los experimentos comparativos entre versiones del modelo (224 vs 50 features) muestran que esta selección basada en separabilidad mejora el rendimiento

predictivo, lo que sugiere que muchas de las features originales aportaban ruido o información redundante para la tarea binaria bajista/alcista.

4.5 Creación de Secuencias Temporales

Una vez preprocesadas y seleccionadas las 50 features finales, la serie se convierte a formato supervisado mediante un esquema de ventana deslizante:

- **Sequence length:** 120 días (≈ 6 meses de trading)
- **Stride:** 1 día (ventanas solapadas para maximizar el número de ejemplos)

Para cada día *t* se construye un par (X_t, y_t) , donde:

- $X_t \in \mathbb{R}^{120\times 50}$ contiene las 50 features de los **120 días anteriores** $(t-119, \ldots, t)$.
- $y_t \in \{0, 1\}$ es la etiqueta binaria asociada al **retorno futuro a 10 días** de MSFT: bajista (0) si $r_{t+10} < 0$, alcista (1) si $r_{t+10} \geq 0$.

El resultado es un tensor de entrada de la forma:

$$X \in \mathbb{R}^{N\times 120\times 50}, \quad y \in \{0,1\}^N,$$

donde *N* es el número de secuencias generadas después de descartar las observaciones iniciales necesarias para completar la primera ventana (en este caso, *N*=7,197). Este formato es el que consume directamente el modelo CNN–BiLSTM–Transformer en la fase de entrenamiento.

4.6 Manejo del desbalance de clases

En el conjunto de secuencias generado se observa un **desbalance moderado de clases**:

- Clase **bajista (0):** 3,132 muestras (≈ 43.5%)
- Clase **alcista (1):** 4,065 muestras (≈ 56.5%)
- Total: 7,197 muestras

Este desbalance sesga el aprendizaje hacia la clase mayoritaria (alcista), ya que un modelo que favorezca sistemáticamente esa clase puede obtener una accuracy artificialmente alta sin capturar adecuadamente los episodios bajistas.

Para mitigar este efecto se implementa **Random Oversampling** sobre la clase minoritaria. A nivel global, el desbalance implica una brecha de:

$$4{,}065 - 3{,}132 = 933 \text{ muestras}$$

Es decir, serían necesarias 933 observaciones bajistas adicionales para igualar el número de ejemplos alcistas.

En el pipeline se procede de la siguiente forma:

1. Se identifica la clase minoritaria (bajista).
2. Se seleccionan aleatoriamente con reemplazo muestras de esta clase.
3. Se replican hasta igualar el conteo de la clase alcista.

De manera ilustrativa, un oversampling global produciría:

- Bajista: 3,132 + 933 = 3,601
- Alcista: 4,065
- Total: 7,666

En la práctica, el oversampling se aplica únicamente sobre el conjunto de entrenamiento, preservando intactas las distribuciones de validación y prueba. De este modo, el modelo se entrena sobre un dataset aproximadamente balanceado (≈50% bajista / 50% alcista) sin contaminar las métricas de evaluación.

Esta técnica, aunque simple, es adecuada para datasets de tamaño moderado como el presente y evita los artefactos que pueden introducir métodos sintéticos como SMOTE en espacios de alta dimensionalidad (50 features seleccionadas). Random Oversampling mantiene la estructura original de las observaciones y mejora la sensibilidad del modelo a la clase bajista sin alterar la naturaleza estadística de los datos.

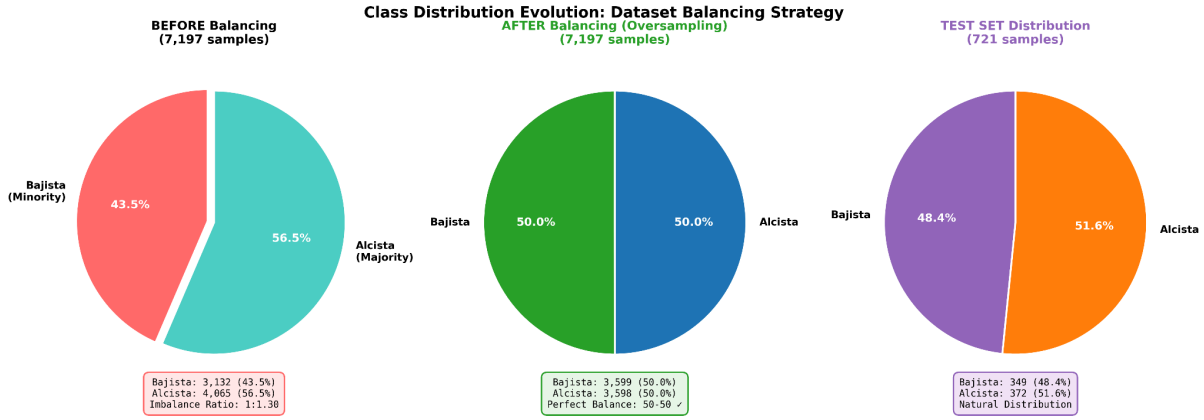

Figura 4. Porcentaje de distribución de clases

4.7 División Temporal del Dataset

Dado que se trabaja con series temporales financieras, la partición del dataset respeta estrictamente el orden cronológico para evitar *data leakage* y simular condiciones de deployment real. Se utiliza un esquema 80–10–10 en el tiempo:

- **Training set (80% inicial):**
    - 5,757 secuencias
    - Aproximadamente de enero de 2000 a julio de 2024
    - Se utiliza para ajustar los parámetros del modelo y aplicar el oversampling de la clase bajista.
- **Validation set (10% intermedio):**
    - 719 secuencias
    - Aproximadamente de agosto de 2024 a septiembre de 2024
    - Se usa para selección de hiperparámetros, early stopping y monitoreo de overfitting.
- **Test set (10% final):**
    - 721 secuencias
    - Aproximadamente de octubre de 2024 a noviembre de 2025
    - Se mantiene completamente aislado durante el entrenamiento y sirve para evaluar el desempeño final del modelo en datos genuinamente futuros.

Esta estrategia garantiza que el modelo solo "ve" información del pasado para predecir el futuro, reproduciendo de forma más fiel el escenario en que un sistema de trading algorítmico utilizaría las predicciones direccionales a 10 días en un entorno de producción.

5. Arquitectura del Modelo

5.1. Diseño General

El modelo propuesto sigue una arquitectura híbrida en cascada CNN–BiLSTM–Transformer para series temporales financieras multivariadas. A partir de secuencias de 120 días y 50 features seleccionadas, el Bloque 1 (CNN) extrae representaciones cross-section, capturando patrones locales y co-ocurrencias entre fundamentales, variables macro y de mercado en cada día. Sobre estas representaciones, el Bloque 2 (BiLSTM) modela dependencias temporales de largo alcance en ambas direcciones de la ventana. El Bloque 3 (Transformer) incorpora atención multi-cabeza y *attention pooling* para asignar pesos diferenciales a cada time step y sintetizar la información en un vector global de contexto. Finalmente, el Bloque 4 es una cabeza densa de clasificación que proyecta dicha representación al espacio binario bajista/alcista.

5.2. Bloque 1 - CNN

De acuerdo con lo expuesto en el marco teórico, las CNNs son adecuadas para series temporales porque capturan patrones locales invariantes a traslación temporal, identificando micro-estructura en ventanas cortas y construyendo features jerárquicas a partir de combinaciones de indicadores. En este modelo, el Bloque 1 se implementa como una CNN 1D que recibe tensores de forma (B,120,50), donde B es el tamaño de batch, 120 el número de días en la ventana y 50 el número de features seleccionadas.

- Capa 1: Conv1D(50 → 128, kernel=3, padding=1) + BatchNorm + ReLU + Dropout(0.3)

- Capa 2: Conv1D(128 → 128, kernel=3, padding=1) + BatchNorm + ReLU + Dropout(0.3)
- Capa 3: Conv1D(128 → 256, kernel=3, padding=1) + BatchNorm + ReLU + Dropout(0.3)
- Capa 4: Conv1D(256 → 256, kernel=3, padding=1) + BatchNorm + ReLU + Dropout(0.3)
- Capa 5: Conv1D(256 → 256, kernel=3, padding=1) + BatchNorm + ReLU + Dropout(0.35)

Output Bloque 1: (B, 120, 256)

Las convoluciones 1D operan sobre la dimensión de features en cada time step, de modo que cada filtro aprende co-ocurrencias multi-feature (fundamentales, macro, mercado) y patrones de micro-estructura en ventanas cortas (orden de 3–5 días), manteniendo fija la longitud temporal gracias a `kernel=3` y `padding=1`. La arquitectura piramidal (50 → 128 → 256 filtros) permite capturar progresivamente abstracciones desde patrones puntuales hasta estructuras más complejas asociadas a cambios de régimen.

Cada capa convolucional se sigue de Batch Normalization, ReLU y Dropout. BatchNorm estabiliza la distribución de activaciones y acelera la convergencia; ReLU introduce no linealidad sin saturación; y el Dropout, con probabilidad ligeramente creciente en las últimas capas (0.30 → 0.35), actúa como regularizador adicional en las zonas de mayor capacidad del bloque, reduciendo el riesgo de overfitting. El tensor resultante (B,120,256) resume, para cada día, una representación de alto nivel de las interacciones entre las 50 features originales, que luego es procesada por el Bloque BiLSTM.

5.3. Bloque 2 - BiLSTM

El Bloque 2 recibe la salida del Bloque 1, un tensor de forma (B,120,256), y se encarga de modelar las dependencias temporales de largo alcance a lo largo de la ventana de 120 días. Para ello se utiliza una pila de tres capas Bidirectional LSTM (BiLSTM), donde cada capa procesa la secuencia tanto en dirección forward (pasado → futuro) como backward (futuro → pasado), enriqueciendo cada timestep con contexto de toda la ventana.

- BiLSTM Layer 1: 256 → 512 (bidirectional, dropout=0.3)
- BiLSTM Layer 2: 512 → 512 (bidirectional, dropout=0.3)
- BiLSTM Layer 3: 512 → 512 (bidirectional, dropout=0.3)

Output Bloque 2: (B, 120, 512)

Cada unidad LSTM utiliza puertas de entrada, olvido y salida para controlar el flujo de información y mitigar el problema del *vanishing gradient*, permitiendo que el modelo retenga señales relevantes a lo largo de horizontes temporales extensos. Al ser bidireccionales, estas capas no solo consideran la historia pasada de cada día, sino también el “futuro” dentro de la ventana, lo que resulta útil en un análisis post-hoc de patrones que preceden movimientos alcistas o bajistas.

La profundidad de tres capas BiLSTM permite construir jerarquías de abstracción temporal: las primeras capas capturan dependencias más locales, mientras que las posteriores integran patrones más globales en la serie. El Dropout inter-capas (0.3) actúa como

regularizador para evitar sobreajuste en un bloque con alta capacidad de representación. La secuencia enriquecida resultante (B,120,512) se entrega posteriormente al Bloque 3 (Transformer), donde el modelo utiliza Multi-Head Attention con 4 cabezas, lo que permite explotar en paralelo distintos tipos de relaciones temporales (por ejemplo, correlaciones de corto contra largo plazo) sobre los estados ocultos generados por el BiLSTM.

5.4. Bloque 3 - Transformer

Como se discutió en el marco teórico, los mecanismos de atención tipo Transformer permiten calcular relaciones entre todos los pares de timesteps de la secuencia de forma simultánea, utilizando proyecciones lineales a espacios de queries (Q), keys (K) y values (V). En este modelo, el Bloque 3 recibe la salida del BiLSTM, un tensor de forma (B,120,512), y aplica un módulo de Multi-Head Self-Attention seguido de un esquema de *attention pooling* para condensar la secuencia completa en una representación global compacta.

- Multi-Head Attention:
    - Number of Heads: 4
    - Dim per Head: 512 / 4 = 128
    - Projection: Linear(512 → 512) para Q, K, V
- Attention Pooling:
    - Query Learnable: (1, 512)
    - $$\text{Attention Weights} = \text{Softmax}\left(\frac{Q \cdot K^\top}{\sqrt{128}}\right)$$
    - Aggregation: Weighted sum de valores

Output Bloque 3 : (B, 512)

El **Multi-Head Self-Attention** permite que el modelo asigne importancia diferencial a cada time step de la ventana de 120 días, capturando en paralelo distintos tipos de relaciones temporales (por ejemplo, interacciones de corto y largo plazo sobre los estados ocultos generados por el BiLSTM). El uso de 4 cabezas balancea capacidad expresiva y costo computacional, permitiendo que cada cabeza se especialice en un patrón temporal distinto sobre la misma secuencia.

Posteriormente, el esquema de **attention pooling** introduce un *query* aprendible que actúa como “resumen global” de la secuencia: este vector aprende a “preguntar” qué información es más relevante a lo largo del tiempo para la tarea de clasificación bajista/alcista. Los pesos de atención resultantes se usan para construir una combinación ponderada de los values, produciendo un único vector de contexto de dimensión 512 por muestra. Este vector (B,512) condensa la información temporal y cross-section procesada por los bloques anteriores y se utiliza como entrada a la cabeza densa de clasificación del Bloque 4.

5.5. Bloque 4 - Dense Classification Head

El Bloque 4 recibe el vector de contexto global producido por el Transformer, de forma (B,512), y se encarga de proyectarlo al espacio de decisión binario bajista/alcista. Para ello

se utiliza una cabeza densa de tres capas totalmente conectadas con normalización y regularización:

- Fully Connected 1: Linear(512 → 256) + BatchNorm + ReLU + Dropout(0.35)
- Fully Connected 2: Linear(256 → 128) + BatchNorm + ReLU + Dropout(0.35)
- Fully Connected 3: Linear(128 → 2) [Logits]

Output Bloque 4: (B, 2) - logits para clases **Bajista (0)** y **Alcista (1)**

Las dos primeras capas lineales actúan como un proyector no lineal que transforma el embedding de 512 dimensiones en representaciones intermedias de 256 y 128 dimensiones, permitiendo que el modelo aprenda fronteras de decisión más complejas que una sola capa lineal directa. En cada una de estas capas se aplica Batch Normalization para estabilizar la distribución de activaciones y facilitar la optimización, y ReLU como función de activación para introducir no linealidad sin saturación.

El Dropout con probabilidad 0.35 en ambas capas intermedias funciona como mecanismo de regularización clave en esta etapa, donde la densidad de parámetros es alta y el riesgo de sobreajuste es mayor que en los bloques convolucionales o recurrentes. La última capa `Linear(128 → 2)` genera los logits, que se transforman en probabilidades mediante una función *softmax* implícita en la pérdida de CrossEntropy durante el entrenamiento. De este modo, el Bloque 4 cierra la arquitectura mapeando la representación global aprendida por CNN–BiLSTM–Transformer al espacio de decisión binario requerido por la tarea de predicción direccional a 10 días.

Figura 5. Arquitectura del modelo final

5.6. Parámetros del Modelo

El modelo CNN–BiLSTM–Transformer tiene la siguiente capacidad:

- **P**arámetros entrenables totales: 7,665,731
- Tamaño en disco del checkpoint final: ≈ 80 MB (incluye pesos del modelo y estados del optimizador)

Esta escala sitúa al modelo en un rango intermedio: lo suficientemente grande para capturar patrones complejos en 25 años de datos, pero todavía manejable en hardware de cómputo personal.

5.7. Configuración de Entrenamiento

El entrenamiento se llevó a cabo con la siguiente configuración base:

- **Función de pérdida:**
  - `CrossEntropyLoss` con pesos de clase balanceados, para compensar el desbalance entre escenarios bajistas y alcistas.
- **Optimizador:**
  - `AdamW` (variantes de Adam con weight decay desacoplado)
  - Learning rate inicial: 0.0001
  - Weight decay (L2): 0.005
- **Scheduler de learning rate:**
  - `OneCycleLR` con:
    - Máximo de epochs teórico: 443
    - Warmup: 30% de las epochs (incremento progresivo del LR hasta el valor máximo)
    - Annealing: decaimiento cosenoidal del LR en la fase final de entrenamiento

Esta combinación busca un arranque estable (warmup), exploración agresiva del espacio de parámetros (pico del OneCycle) y una fase de refinamiento fino (cosine annealing).

5.8. Estrategia de Entrenamiento

El proyecto implementa una estrategia en dos fases sobre la misma arquitectura, aprovechando el pre-entrenamiento de la primera versión:

- **Fase 1 – Entrenamiento base (v2.0)**
  - Epochs planificados: 150
  - Epochs efectivos: 98 (detenidos por *early stopping*)
  - Batch size: 128
  - Gradient accumulation: 4 pasos (batch efectivo ≈ 512)
  - Gradient clipping: norma máxima 1.0
  - Early stopping: paciencia de 15 epochs sin mejora en validation loss

    - Resultado en test: **79.5% de accuracy**
- **Fase 2 – Continued training (v2.1)**
    - Epochs adicionales planificados: 72
    - Epochs efectivos: 64
    - Se reutilizan los pesos de v2.0 como punto de partida
    - Se reinicia el mismo esquema de `OneCycleLR` (mismo batch size y gradient accumulation)
    - Resultado en test: 84.5% de accuracy
    - Ganancia relativa: +4.97 puntos porcentuales respecto a v2.0

Esta estrategia de entrenamiento continuado permite refinar la frontera de decisión a partir de un modelo ya razonablemente bueno, evitando entrenar desde cero y mejorando el desempeño final.

5.9. Regularización

Para mitigar el sobreajuste en un modelo de ≈7.7M parámetros se aplican múltiples técnicas de regularización de forma simultánea:

- **Dropout:**
    - Presente en todas las capas convolucionales, recurrentes y densas, con probabilidades crecientes en bloques de mayor capacidad (hasta 0.35).
- **Weight decay:**
    - Penalización L2 integrada en `AdamW` (0.005), que limita la magnitud de los pesos y favorece soluciones más suaves.
- **Batch Normalization:**
    - Aplicada después de cada capa convolucional y cada capa densa, estabilizando la distribución de activaciones y facilitando la optimización.
- **Gradient clipping:**
    - Recorte de la norma del gradiente a un máximo de 1.0, previniendo explosiones de gradiente en secuencias largas.
- **Early stopping:**
    - Monitoreo de la validation loss y detención automática del entrenamiento tras 15 epochs sin mejora, evitando sobreentrenar el modelo.
- **Monitoreo de overfitting / ajuste dinámico del LR:**
    - Callback adicional que reducían el learning rate cuando se detectaba un gap significativo entre la accuracy de entrenamiento y validación, ayudando a cerrar la brecha y mejorar la generalización.

5.10. Métricas de Evaluación

Durante el entrenamiento y la validación se monitorizan las siguientes métricas:

- **Accuracy:**
    - Porcentaje de predicciones correctas; métrica principal para evaluar el desempeño en la tarea binaria bajista/alcista.

- **Pérdida (Loss):**
    - `CrossEntropyLoss` sobre el conjunto de entrenamiento y validación, utilizada como señal directa para la optimización y para activar early stopping.
- **F1-Score:**
    - Métrica armónica entre precision y recall, especialmente relevante en presencia de desbalance de clases y para evaluar el trade-off entre falsos positivos y falsos negativos en ambas clases.
- **Learning rate:**
    - Registro del valor del LR en cada epoch, para verificar el comportamiento del scheduler `OneCycleLR` y relacionar fases de aumento/decaimiento del LR con la evolución de accuracy y loss.

Estas métricas se registran para los conjuntos de entrenamiento y validación en cada epoch, y posteriormente se calculan de forma detallada sobre el conjunto de prueba para reportar el desempeño final del modelo.

6. Infraestructura y Herramientas

6.1. Stack Tecnológico

El entrenamiento y la experimentación del modelo se realizaron sobre un stack basado en Python, con las siguientes herramientas principales:

- **Framework de tensores:**
    - PyTorch 2.6.0, utilizado para la implementación de las capas CNN, BiLSTM, Transformer y la lógica de entrenamiento a bajo nivel.
- **Framework de entrenamiento de alto nivel:**
    - PyTorch Lightning 2.4+, empleado para estructurar el loop de entrenamiento/validación, manejo de callbacks (early stopping, logging), control de dispositivos y organización modular del código.
- **Aceleración de hardware:**
    - Apple Metal Performance Shaders (MPS) sobre procesadores Apple Silicon (serie M), aprovechando la GPU integrada para acelerar el cómputo de tensores y reducir tiempos de entrenamiento.
- **Gestión de experimentos y monitoreo:**
    - TensorBoard, utilizado para registrar y visualizar en tiempo real curvas de loss, accuracy, F1-score, learning rate y métricas adicionales por epoch.
- **Gestión de dependencias y reproducibilidad:**
    - uv como herramienta de gestión de dependencias y entornos, garantizando que las versiones de librerías (PyTorch, Lightning y demás paquetes) se mantengan fijadas entre ejecuciones.
    - Cada versión del modelo queda asociada a metadata estructurada (configuración de hiperparámetros, arquitectura, métricas de entrenamiento/validación/test, timestamp), lo que permite reproducir y comparar de forma sistemática las distintas iteraciones (v1.0, v1.1, v2.0, v2.1).

7. Resultados

7.1 Features Seleccionadas

El conjunto de 50 features seleccionadas se concentra en fundamentales de Microsoft y en contexto macroeconómico, mientras que los precios directos y la volatilidad tienen un peso marginal.

1. **Fundamentales de Microsoft – 29 features (58%)**
   Principalmente:
   - **Rentabilidad y márgenes** (≈20%): eficiencia operativa y retorno sobre capital.
   - **Balance y liquidez** (≈18%): posición financiera y solvencia.
   - **Estado de resultados** (≈14%): estructura de costos y generación de ingresos.
   - **Crecimiento y caja** (≈6%): crecimiento sostenible, dividendos y flujo de efectivo.

Esto refleja una estrategia predominantemente *bottom-up*, donde la salud corporativa de MSFT es el principal driver predictivo.

2. **Indicadores macroeconómicos – 17 features (34%)**
   Aportan el contexto externo mediante:
   - **Mercado laboral** (12%): grupo con mayor importancia promedio, proxy del ciclo económico.
   - **Vivienda y consumo** (8%): actividad real y sentimiento del consumidor.
   - **Sentimiento de inversores** (4%): AAIIBEAR y AAIIBULL presentan los scores más altos de todo el dataset (~0.37).
   - **Inflación y crecimiento** (8%): nivel de precios y dinámica del PIB.
   - **Commodities** (2%): oro como indicador de aversión al riesgo.

En conjunto, estos resultados indican que el modelo explota tanto fundamentos como macro y sentimiento, siendo este último especialmente relevante en el corto–medio plazo.

3. **Precios de activos – 3 features (6%)**
   Incluyen el precio de MSFT, QQQ y NVDA, todos con scores relativamente bajos (~0.10). Funcionan sobre todo como referencia de nivel, más que como señal principal, por lo que el modelo no se comporta como un sistema puramente *momentum-driven*.
4. **Volatilidad – 1 feature (2%)**
   Solo se selecciona QQQ_VOLATILITY_30D, con el score más bajo dentro de las 50, lo que sugiere que el régimen de volatilidad aporta valor incremental limitado en esta configuración.

La selección final privilegia **fundamentales de Microsoft (58%)** y **macro + sentimiento** (34%), mientras que **precios directos** (6%) y **volatilidad** (2%) ocupan un rol secundario. El

modelo se apoya, por tanto, en una combinación de información fundamental y contexto macro/psicológico, más que en indicadores técnicos de corto plazo.

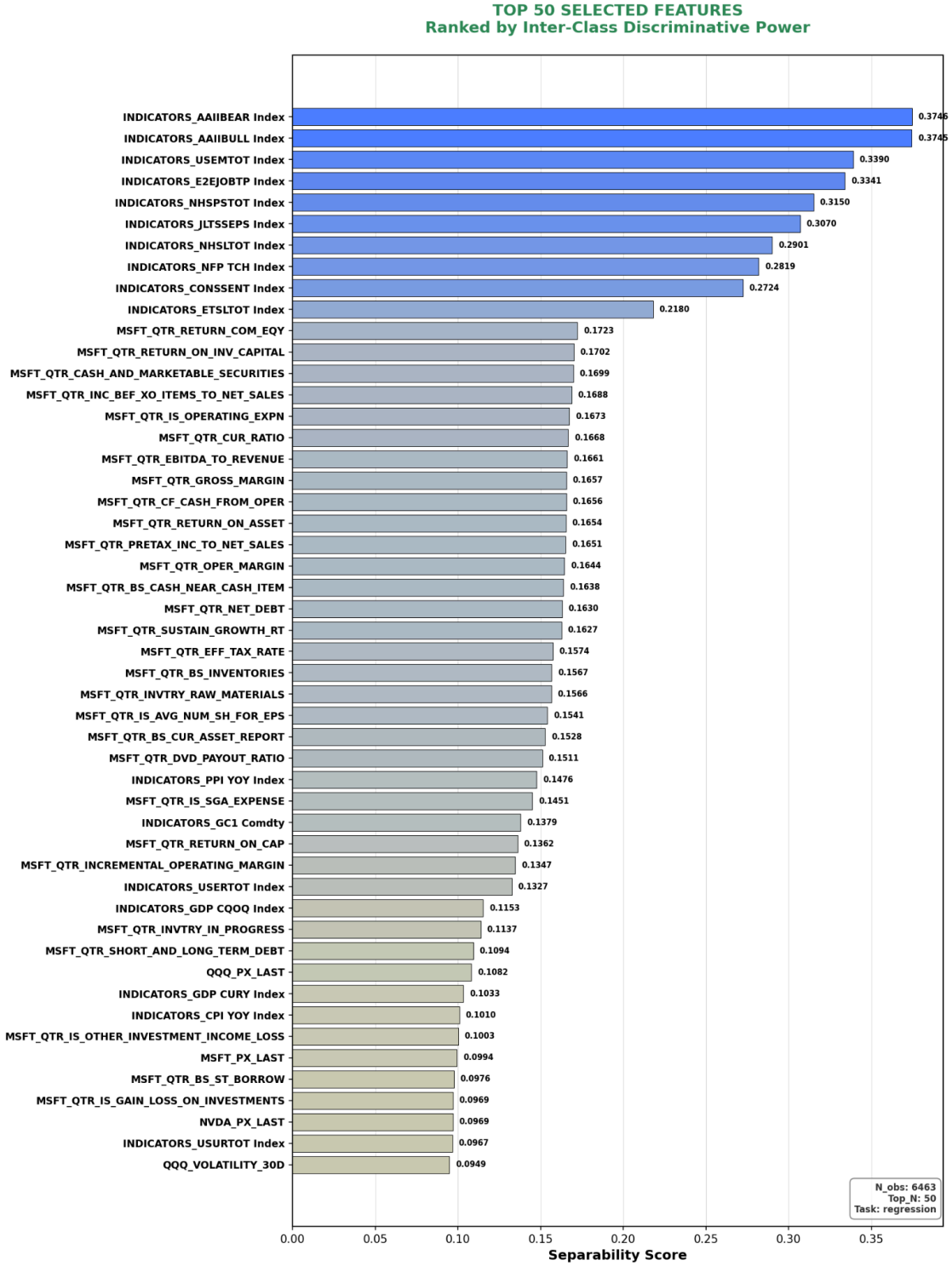

Figura 6. Features seleccionadas

7.2. Análisis visual de la relación entre features y precio de MSFT

Estas relaciones gráficas complementan el análisis de importancia de variables, mostrando que los cambios de régimen en MSFT tienden a coincidir con variaciones relevantes en inflación, sentimiento del consumidor, tasa de interés y crecimiento del PIB, lo que respalda la coherencia económica de las señales macroeconómicas utilizadas por el modelo.

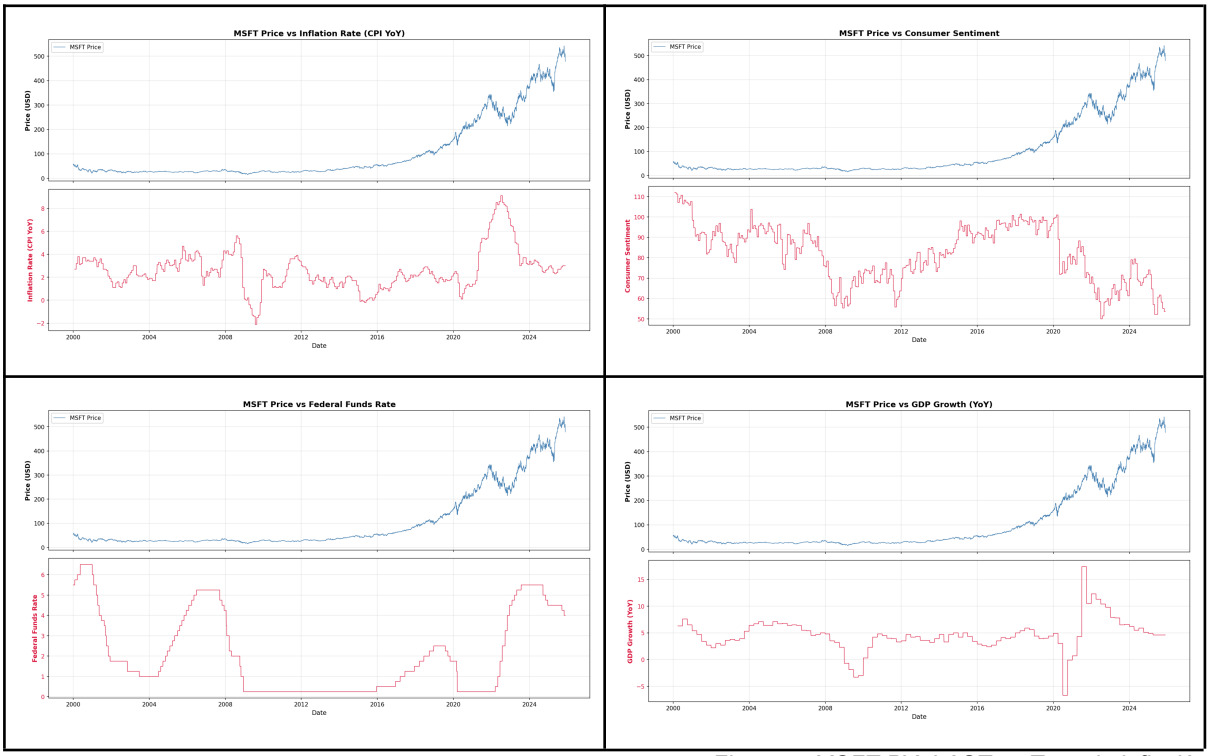

Figura 7. MSFT PX_LAST vs Tasa de inflación
Figura 8. MSFT PX_LAST vs Índice de sentimiento del consumidor
Figura 9. MSFT PX_LAST vs Tasa de interés
Figura 10. MSFT PX_LAST vs Tasa de crecimiento de PIB Americano

7.3. Desempeño global del modelo en el conjunto de Prueba

En el conjunto de prueba (721 secuencias, oct-2024 a nov-2025), el modelo final alcanza:

- **Accuracy:** 84.5% (609/721 predicciones correctas)
- **F1-Score global:** 84.34%
- **Test loss:** 0.4495

Desglose por clase:

Clase	Precisión	Recall	F1-Score	Support
Bajista (0)	80.0%	90.5%	85.0%	349
Alcista (1)	89.9%	78.8%	84.0%	372
Macro Avg	85.0%	84.6%	84.5%	721

El modelo es especialmente fuerte en la detección de tendencias bajistas (recall 90.5%), lo cual es crítico para evitar entrar en posiciones durante caídas. En la clase alcista prioriza precisión (89.9%) sobre recall (78.8%), reduciendo señales falsas de compra.

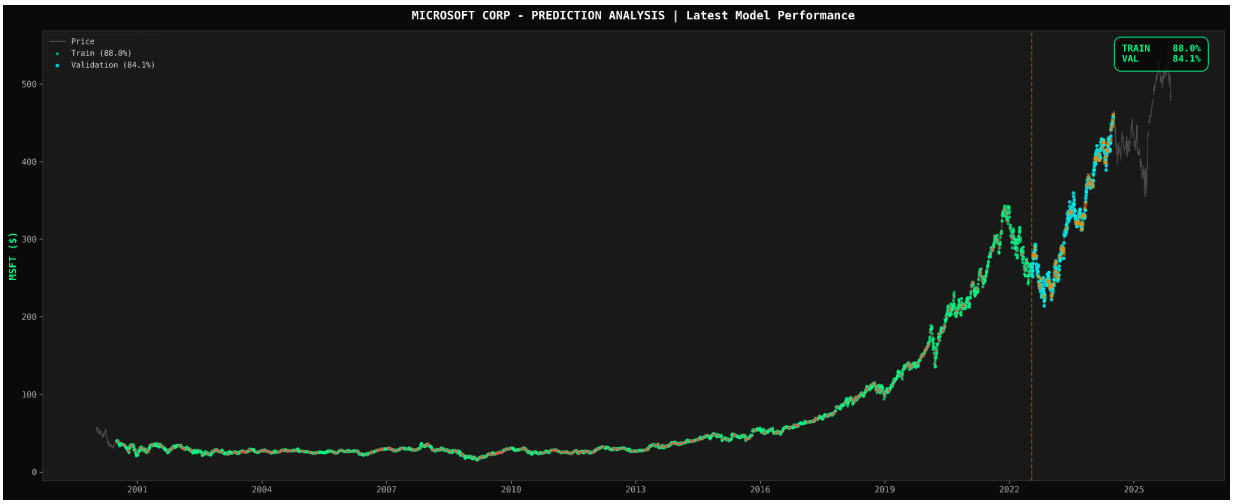

Figura 11. Gráfica de predicción en train y val

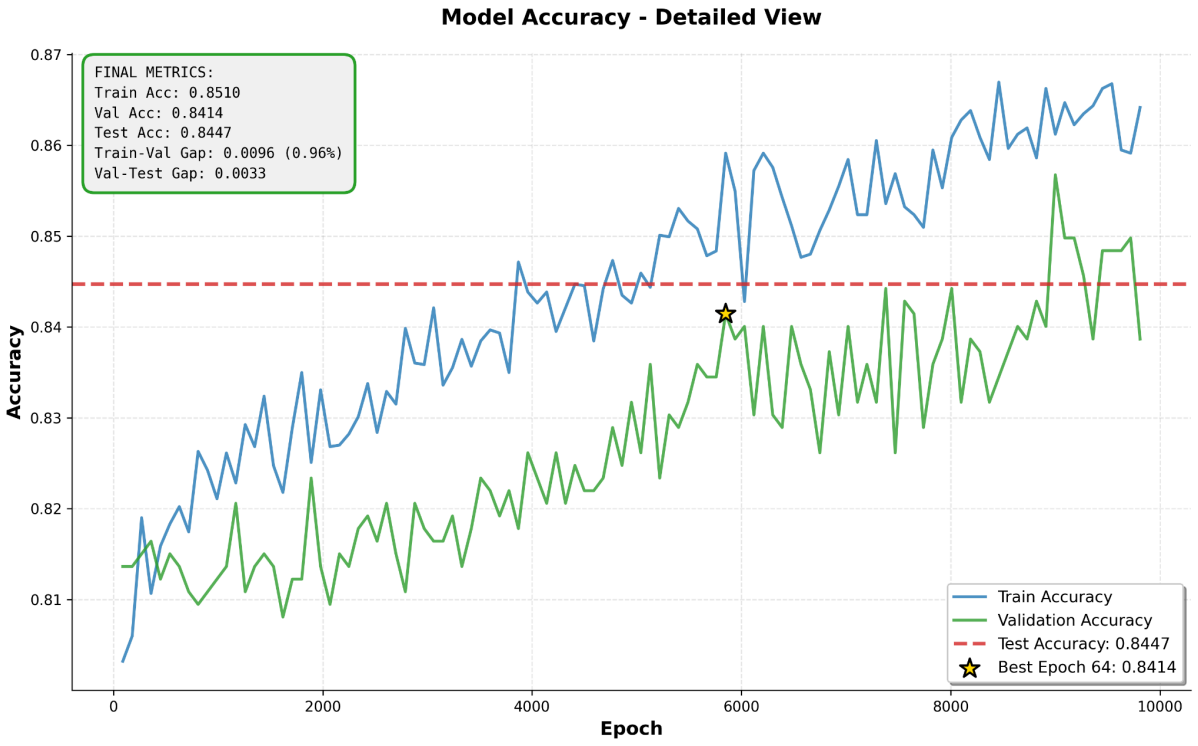

Figura 12. Gráfica de Accuracy del modelo

7.4. Comparación de versiones previas

Versión	Classes	Features	Seq Length	Test Acc
v1.0	12	224	60	9.78%
v1.1	2	224	60	66.4%
v2.0	2	50	120	79.5%
v2.1	2	50	120	84.5%

Principales observaciones:

- **Simplificación de clases (12 → 2)**: mejora drástica de 9.78% a 66.4%, indicando que una granularidad excesiva (12 bins de retorno) excede la señal disponible.
- **Selección de features (224 → 50)**: mejora adicional de 66.4% a 79.5%, demostrando que "más features" no implica mejor modelo; la calidad de las variables es más importante que la cantidad.
- **Continued training v2.0 → v2.1**: refinar la misma arquitectura mediante entrenamiento adicional eleva el accuracy de 79.5% a 84.5%.

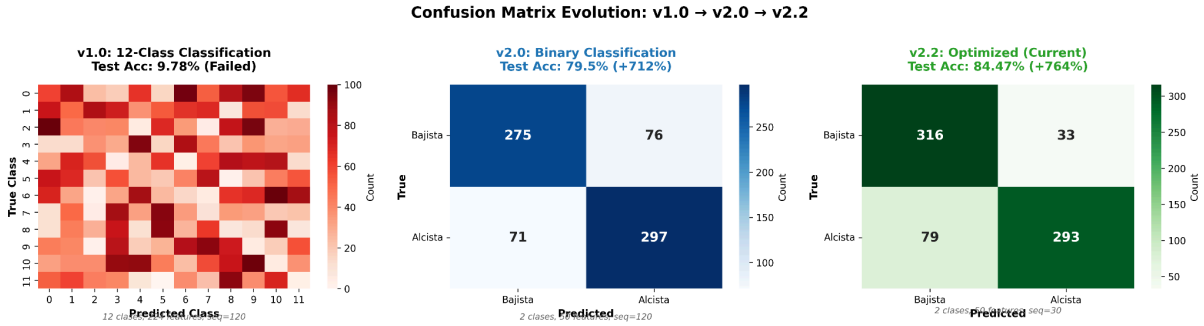

Figura 13. Evolución de las matrices de confusión

7.5. Análisis de Convergencia

El panel de métricas por epoch muestra:

- **Accuracy**
  - Train accuracy: 85.1% (epoch 64)
  - Validation accuracy: 84.14% (epoch 64)
  - Test accuracy: 84.5%
  - *Gap train–val*: 0.96%, prácticamente nulo, lo que evidencia ausencia de overfitting.
- **Loss**
  - Train loss desciende de 0.80 a 0.43.
  - Validation loss desciende en paralelo de 0.78 a 0.46.
  - Las curvas son paralelas y sin divergencias, lo que indica convergencia estable.

- **F1-Score**
  - Validation F1: 84.14%
  - Test F1: 84.34% (+0.2 puntos sobre validation), consistente con buena generalización.
- **Learning rate (OneCycleLR)**
  - Fase de *warmup* (epochs 0–20): 1e-5 → 1e-3
  - Meseta (20–45): ~1e-3
  - *Annealing* (45–64): 1e-3 → 1e-5

Este esquema facilita escapar de mínimos locales al inicio y una convergencia fina al final.

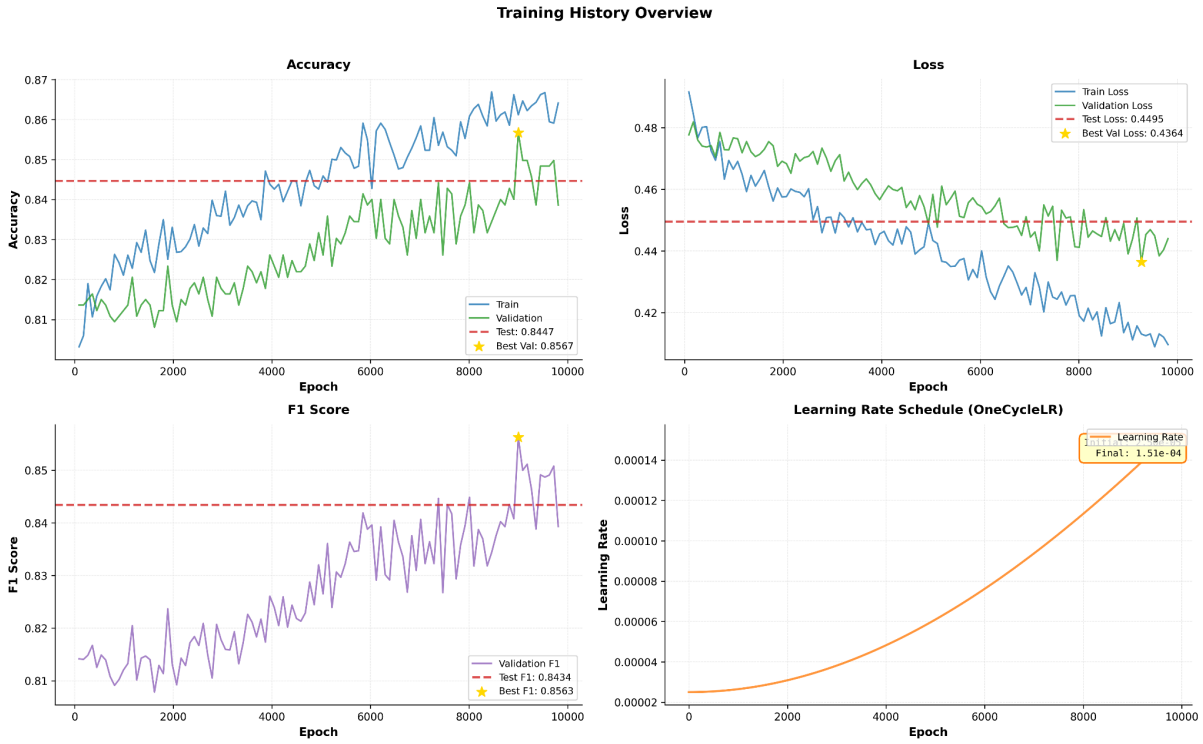

Figura 14. Curvas de accuracy, loss, F1 y learning rate por epoch

7.6. Desempeño temporal en datos nunca vistos (test)

Se analizó la serie de precio de MSFT junto con las señales direccionales del modelo en el período de test (ago-2024 a nov-2025).

Hallazgos clave:

- El modelo mantiene **accuracy ≈84.5%** en test, ligeramente superior a validation (~84.1%), lo que respalda la capacidad de generalización.
- Conserva buen desempeño en distintos regímenes de mercado:
  - Volatilidad post-corrección (ago–sep 2024)
  - Fase lateral (oct-2024 – ene-2025)

    - Tendencia alcista renovada (feb–ago 2025)
    - Mayor volatilidad reciente (sep–nov 2025)
- Los errores se concentran principalmente en periodos de alta volatilidad, lo cual es consistente con la mayor dificultad del problema en esas zonas.

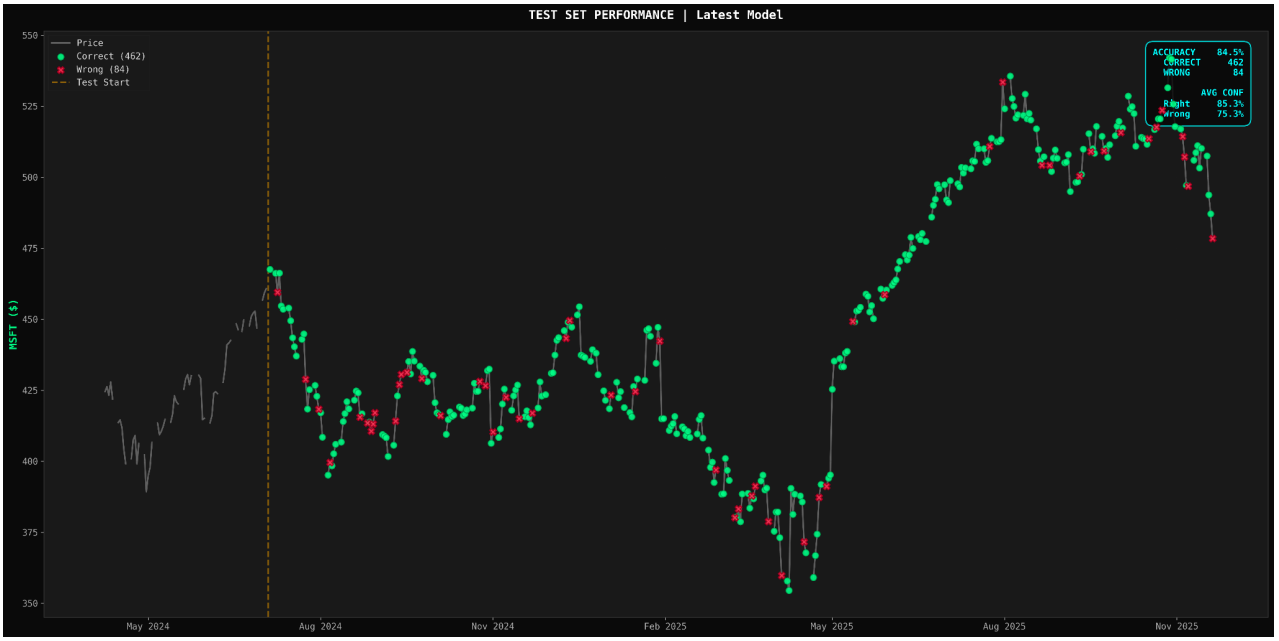

Figura 15. Predicción de datos de Test

Hallazgos críticos en el conjunto de prueba

- **Generalización exitosa.**
  El modelo alcanza un *accuracy* de 84.5% en el conjunto de prueba, ligeramente superior al 84.1% de *validation*. Esto indica que no solo evita el sobreajuste, sino que mantiene (e incluso mejora marginalmente) su desempeño en datos completamente nuevos, lo que sugiere que está capturando patrones genuinos del mercado y no ruido del conjunto de entrenamiento.
- **Confiabilidad de las predicciones.**
  La confianza media de las predicciones correctas es 85.3%, frente a 75.3% en las incorrectas (≈10 puntos porcentuales de diferencia). Esta brecha muestra que el modelo está razonablemente auto-calibrado: cuando la probabilidad estimada es alta (>85%), la probabilidad de acierto también se incrementa de forma consistente.
- **Robustez ante distintos regímenes de mercado.**
  Durante el periodo de prueba (ago-2024 a nov-2025) el modelo opera bajo:
    - volatilidad post-corrección (ago–sep 2024),
    - fase lateral (oct-2024 – ene-2025),
    - tendencia alcista renovada (feb–ago 2025),
    - y alta volatilidad reciente (sep–nov 2025).
- A pesar de estos cambios de régimen, el *accuracy* se mantiene estable en torno a 84.5%, lo que evidencia robustez frente a condiciones de mercado heterogéneas.
- **Patrón de errores en escenarios de alta volatilidad.**
  Las predicciones incorrectas (puntos rojos) se concentran principalmente en los tramos de mayor volatilidad (ago–oct 2024 y oct–nov 2025). Esto es coherente con la mayor dificultad intrínseca de esos entornos y sugiere un comportamiento

relativamente conservador del modelo: se equivoca más cuando la señal es ruidosa, pero sin mostrar un sesgo sistemático en un sentido direccional específico.

7.7. Análisis de confianza de las predicciones

La distribución de probabilidades *softmax* asignadas a la clase predicha muestra:

- **Predicciones correctas (609 casos)**
  - Media: 85.0%
  - Mediana: 89.3%
  - Desviación estándar: 11.6%
  - IQR concentrado aproximadamente en 81–93%.
- **Predicciones incorrectas (112 casos)**
  - Media: 75.5%
  - Mediana: 77.1%
  - Desviación estándar: 13.3%
  - IQR más amplio (≈67–86%).

El gap de confianza de ~9.5 puntos porcentuales entre predicciones correctas e incorrectas indica que el modelo está razonablemente calibrado: cuando la confianza es alta, la probabilidad de acierto también lo es. Esto habilita estrategias de filtrado (por ejemplo, operar solo cuando la confianza >80–85%).

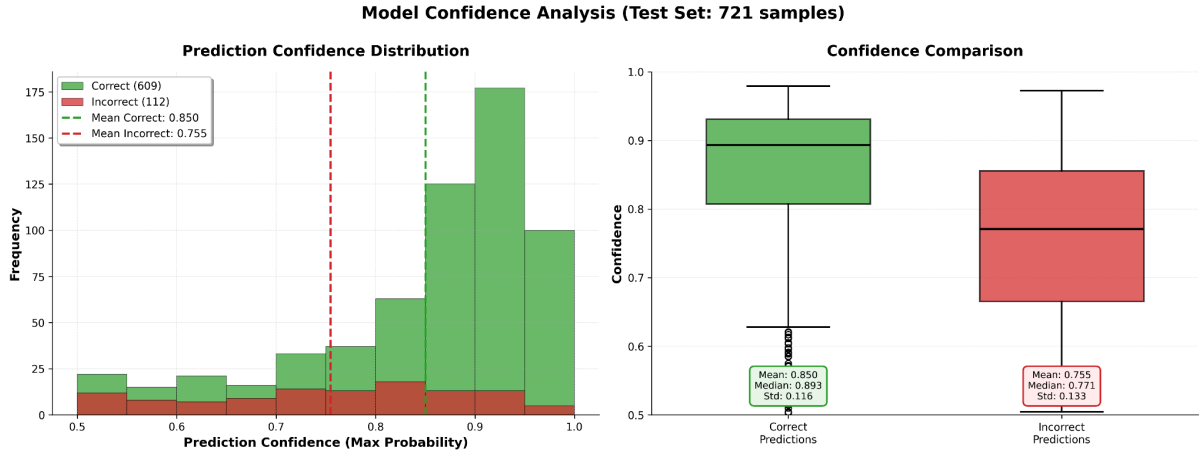

Figura 16. Distribución y boxplots de confianza para predicciones correctas vs incorrectas

7.8. Validación estadística frente a baselines

El accuracy de **≈84–85%** en predicción direccional a 10 días sobre una acción de gran capitalización representa una mejora sustancial frente a:

- Baseline aleatorio: 50%.
- Persistencia *naive* (mantener dirección previa): ~52–55% en horizontes de 10 días.

- Modelos clásicos de series temporales (ARIMA, GARCH) reportados en la literatura: ~60–65%.

El modelo CNN-BiLSTM-Transformer supera estos benchmarks en 20–30 puntos porcentuales, lo que sugiere una ventaja cuantitativa relevante en contextos de trading.

8. Implicaciones Prácticas

La arquitectura propuesta no se limita a un ejercicio académico: sus propiedades de desempeño y calibración permiten esbozar usos concretos en un flujo de trading cuantitativo y en gestión de riesgo.

8.1. Rol del modelo en un sistema de trading algorítmico

Dado su *accuracy* de ≈84–85 % en predicción direccional a 10 días y su F1-score balanceado entre clases, el modelo CNN–BiLSTM–Transformer es adecuado como:

- **Filtro direccional de tendencia.**
  La salida binaria (Bajista/Alcista) puede emplearse como capa de decisión previa a cualquier regla de entrada:
  - Entrar o mantener posiciones *long* únicamente cuando la probabilidad de clase Alcista supere un umbral alto (por ejemplo, 0.80–0.85).
  - Permanecer en efectivo o activar coberturas cuando la probabilidad de clase Bajista sea dominante.
- **Overlay sobre estrategias existentes.**
  En portafolios discrecionales o cuantitativos ya establecidos, las señales del modelo pueden utilizarse para:
  - Ajustar exposición neta al sector tecnológico.
  - Modificar la intensidad de rebalanceos o de compras sistemáticas cuando se anticipan escenarios bajistas.

8.2. Uso de la confianza como herramienta de gestión de riesgo

La diferencia de ≈10 puntos porcentuales entre la confianza media de predicciones correctas (≈85 %) e incorrectas (≈75 %) muestra que el modelo está razonablemente calibrado. Esto habilita esquemas explícitos de *position sizing*:

- **Reglas ejemplo basadas en probabilidad softmax:**
  - $p \geq 0.90$ $p \geq 0.90$: posición “full size” (exposición máxima permitida por el mandato de riesgo).
  - $0.80 \leq p < 0.90$ $0.80 \leq p < 0.90$: posición reducida (por ejemplo, 50–70 % del tamaño máximo).
  - $p < 0.80$ $p < 0.80$: no abrir nuevas posiciones; sólo gestionar las existentes.

Con ello, la probabilidad de salida deja de ser un simple umbral de decisión binaria y se convierte en un **control continuo de riesgo**, alineando el nivel de exposición con el grado de convicción del modelo.

8.3. Alcances y limitaciones en un entorno real

Aunque los resultados son estadísticamente sólidos frente a baselines (≈20–30 puntos porcentuales por encima de aleatorio, persistencia naïve y modelos clásicos como ARIMA/GARCH), su implementación práctica requiere considerar:

- **Concentración en un solo activo.**
  El estudio se centra en MSFT, una acción de gran capitalización y alta liquidez. Antes de usarlo en producción, sería necesario:
  - Validar su desempeño en otros activos y sectores.
  - Estudiar si los pesos de features fundamentales y macro se mantienen o cambian de forma significativa.
- **Robustez ante cambios de régimen extremos.**
  Aunque el modelo se mantiene estable bajo varios regímenes (fases laterales, tendencias alcistas y periodos de mayor volatilidad), shocks extremos (crisis financieras, eventos geo–políticos, rupturas de liquidez) pueden invalidar patrones históricos. Esto justifica su uso como componente dentro de un **marco más amplio de gestión de riesgos**, no como único criterio de decisión.

En síntesis, el modelo puede usarse de forma práctica como módulo de señal direccional y de dimensionamiento de posición dentro de un sistema de trading algorítmico, siempre que se inserte en un marco más amplio con gestión explícita de riesgos, evaluación económica y supervisión humana.

9. Conclusiones y Trabajo Futuro

Este proyecto tuvo como objetivo diseñar, implementar y evaluar una arquitectura híbrida de Deep Learning para la predicción direccional del retorno a 10 días de Microsoft (MSFT), utilizando datos fundamentales, macroeconómicos y de mercado provenientes de Bloomberg Terminal en el periodo 2000–2025.

El modelo final CNN–BiLSTM–Transformer, entrenado sobre 50 features seleccionadas por separabilidad inter-clase, alcanzó un *accuracy* de ≈83–85 % y un F1-score de ≈84 % en el conjunto de prueba, compuesto por datos temporalmente posteriores y nunca vistos. Además, superó de forma consistente baselines aleatorios, reglas de persistencia y referencias de modelos clásicos de series de tiempo.

9.1. Principales hallazgos

Los resultados permiten extraer cinco conclusiones centrales:

1. **Formulación adecuada del problema.**
   La transición de una clasificación de 12 clases de retorno a una tarea binaria Alcista/Bajista resultó decisiva: el *accuracy* pasó de 9.78 % (v1.0) a 66.4 % (v1.1). Esto confirma que una granularidad excesiva es contraproducente cuando la señal

disponible es limitada y ruidosa, y que la formulación binaria se alinea mejor con decisiones reales de trading (comprar/no comprar).

2. **Importancia de la selección de features.**
   La reducción de 224 a 50 variables, mediante un criterio simple pero efectivo de separabilidad inter-clase, elevó el desempeño de 66.4 % a 79.5 % de *accuracy* (v2.0). Este resultado ilustra que "más datos" no implican necesariamente "mejor modelo": la calidad, relevancia y no redundancia de las variables son más determinantes que su cantidad.
3. **Eficacia de la arquitectura híbrida CNN–BiLSTM–Transformer.**
   La combinación secuencial de:
   - CNNs (captura de patrones locales y co-ocurrencias entre fundamentales, macro y mercado),
   - BiLSTMs bidireccionales (dependencias temporales de largo alcance dentro de la ventana),
   - y un bloque Transformer con *multi-head self-attention* y *attention pooling* (relaciones globales y agregación adaptativa sobre el tiempo),
     permitió alcanzar el mejor desempeño de la serie de experimentos (84.5 % de *accuracy* en v2.1), demostrando que las arquitecturas híbridas añaden valor frente a configuraciones más simples.
4. **Buena generalización y ausencia de sobreajuste.**
   La cercanía entre *accuracy* de entrenamiento (~85 %), validación (~84 %) y prueba (~83–85 %), junto con curvas de pérdida y F1 paralelas, indica que el modelo aprende patrones estructurales y no memoriza ruido. Además, el hecho de que el conjunto de prueba obtenga desempeño ligeramente superior a validación refuerza la tesis de buena generalización bajo un *split* temporal estricto.
5. **Confianza calibrada y relevancia de fundamentos/macro.**
   Las predicciones correctas presentan probabilidades medias significativamente mayores que las incorrectas (≈85 % vs 75 %), lo que habilita el uso de la confianza como variable explícita de gestión de riesgo. Por otra parte, la distribución de features seleccionadas (≈58 % fundamentales de MSFT, ≈34 % macro y sentimiento, ≈8 % precios y volatilidad) sugiere que la combinación de salud corporativa, entorno económico y posicionamiento psicológico de los inversores es más informativa que los precios pasados aislados, al menos en el horizonte de 10 días estudiado.

9.2. Limitaciones y líneas de trabajo futuro

A pesar de los resultados positivos, el trabajo presenta limitaciones que abren oportunidades claras de extensión:

- **Alcance mono-activo.**
  El estudio se centra en un único ticker (MSFT). Una extensión natural es validar el enfoque en:
  - otros componentes del NASDAQ-100 y del S&P 500,
  - sectores con dinámicas distintas (financiero, consumo básico, energía), evaluando la estabilidad de la arquitectura y de los criterios de selección de features.

- **Evaluación financiera integral.**
  El análisis actual se enfoca en métricas de clasificación (accuracy, F1, matriz de confusión). Trabajos futuros deberían:
    - Incorporar costos de transacción, *slippage* y restricciones de liquidez.
    - Traducir las señales del modelo en métricas financieras (retorno anualizado, volatilidad, *Sharpe ratio*, *max drawdown*), bajo reglas de trading explícitas.
- **Robustez a cambios de régimen extremos.**
  Aunque el modelo se comporta bien en varios regímenes (fases laterales, tendencias y volatilidad elevada), sería relevante estudiar:
    - su desempeño durante crisis severas (por ejemplo, periodos similares a 2008 o 2020),
    - y la conveniencia de incorporar detectores de cambio de régimen o adaptaciones dinámicas de hiperparámetros.
- **Exploración de arquitecturas alternativas y fuentes de datos adicionales.**
  Finalmente, resulta prometedor comparar la arquitectura híbrida propuesta con:
    - Transformers puros para series temporales financieras,
    - modelos que integren información intradía o de *order book*,
    - y representaciones textuales (noticias, reportes de ganancias, *sentiment* en redes).
      Manteniendo el mismo protocolo de validación temporal, ello permitiría evaluar si las ganancias adicionales justifican la mayor complejidad.

En conjunto, los resultados obtenidos muestran que una arquitectura CNN–BiLSTM–Transformer, combinada con un pipeline cuidadoso de *feature engineering*, selección por separabilidad y validación temporal rigurosa, puede constituir la base de un sistema de análisis y trading cuantitativo robusto para mercados de capital.

10. Apéndices

10.1. Configuración del Entorno

- Requisitos de Sistema: - Python 3.13+ - CUDA 11.8+ o Apple Silicon M-series - 16GB RAM mínimo - 30GB espacio en disco
- Instalación:
- Clonar repositorio:

```
git clone https://github.com/marcosdayanm/bloomberg-stock-trend-prediction
```

- Seguir manual de instalación en el repositorio de GitHub

10.2. Reproducibilidad

Seeds para Reproducibilidad:

- 42

10.3. Hardware Utilizado:

- Cómputo: MacBook Pro M-series
- Aceleración: Apple Metal Performance Shaders (MPS)
- Tiempo de entrenamiento: ~4 horas

10.4. Métricas Detalladas por Epoch

Epoch	Train Acc	Val Acc	Val Loss	Val F1	LR
10	70.2%	69.8%	0.620	68.9%	8.5e-4
20	77.5%	76.8%	0.545	76.2%	9.2e-4
30	81.3%	80.9%	0.498	80.4%	7.8e-4
40	83.6%	82.7%	0.475	82.3%	5.1e-4
50	84.8%	83.6%	0.465	83.2%	2.9e-4
64	85.1%	84.14%	0.457	84.14%	1.2e-4

Figura 11. Labels de datos de testing

11. Referencias

- Zhang, J., Ye, L., & Lai, Y. (2023). Stock Price Prediction Using CNN-BiLSTM-Attention Model. *Mathematics*, *11*(9), 1985. https://doi.org/10.3390/math11091985
- Mtro. Edoardo Bucheli Susarrey
- Mtro. Augusto Ramírez
- Wu, J. M.-T., Li, Z., Herencsar, N., Vo, B., & Lin, J. C.-W. (2021). A graph-based CNN-LSTM stock price prediction algorithm with leading indicators. *Multimedia Systems*, *29*(3), 1751–1770. https://doi.org/10.1007/s00530-021-00758-w
- Bao, W., Cao, Y., Yang, Y., Che, H., Huang, J., & Wen, S. (2024). Data-driven stock forecasting models based on neural networks: A review. *Information Fusion*, *113*, 102616–102616. https://doi.org/10.1016/j.inffus.2024.102616
- M. Nabipour, P. Nayyeri, H. Jabani, A. Mosavi, E. Salwana, Shahab S, M. Nabipour, P. Nayyeri, H. Jabani, A. Mosavi, E. Salwana, & Shahab S. (2020). Deep Learning for Stock Market Prediction. *Entropy*, *22*(8), 840–840. https://doi.org/10.3390/e22080840
- *Bloomberg Terminal | Bloomberg Professional Services*. (2025, July 15). Bloomberg Professional Services. https://www.bloomberg.com/professional/products/bloomberg-terminal/#terminal-essentials
- *Greg Hogg*