

Universidad Abierta Interamericana



Diseño e Implementación de una Plataforma de Análisis Técnico para Inversores Minoristas Argentinos

Profesora: Marcela Samela

Alumno: Sebastián Martín Schiuma

Trabajo Final de Carrera presentado para obtener el Título de la
Carrera de Licenciatura en Gestión de Tecnología Informática

(julio, 2026)

Índice General

Resumen.....	8
Abstract.....	9
Capítulo 1. Introducción	10
1.1 Naturaleza del proyecto.....	10
1.2 Justificación.....	11
1.3 Marco institucional.....	12
1.4 Objetivo general	12
1.4.1 Objetivos particulares	12
Capítulo 2. Estado del Arte	13
2.1 Delimitación del alcance de la revisión.....	13
2.2 Estudios sobre medias móviles clásicas	13
2.3 Hull Moving Average y la elección de HMA16	13
2.4 Antecedentes de dolarización de series para análisis técnico	14
2.5 Herramientas tecnológicas para análisis técnico y backtesting.....	15
2.6 Métodos de backtesting y evaluación de estrategias	15
2.7 Plataformas web de backtesting orientadas a usabilidad	16
2.8 Integración de Análisis Técnico con Procesamiento de Lenguaje Natural.....	16
2.9 Brechas identificadas.....	16
Capítulo 3. Marco Teórico.....	18
3.1 Conceptos fundamentales de trading y análisis técnico	18
3.1.1 Medias móviles.....	18
3.1.2 Media móvil simple (Simple Moving Average, SMA)	19
3.1.3 Media móvil exponencial (Exponential Moving Average, EMA)	19
3.1.4 Media móvil de Hull (Hull Moving Average, HMA16)	20
3.1.5 Generación de señales: cruces y puntos de giro	20
3.1.6 El volumen.....	21

3.1.7 Patrones de velas japonesas	21
3.2 Mercado de capitales argentino: acciones y ADRs	21
3.3 Paridad entre acciones locales y ADRs	22
3.3.1 El dólar MEP	22
3.3.2 Cálculo de la paridad teórica Local-ADR	23
3.4 Conceptos de backtesting y evaluación de estrategias	24
Capítulo 4. Abordaje Metodológico	25
4.1 Tipo de investigación	25
4.2 Diseño de la plataforma.....	25
4.2.1 Requisitos funcionales y no funcionales	25
4.2.2 Arquitectura del sistema	27
4.2.3 Estrategia de persistencia de datos	28
4.2.4 Diseño de la interfaz de usuario	30
4.3 Recolección y preprocesamiento de datos	30
4.3.1 Fuentes de datos.....	30
4.3.1.1 Criterios de calidad de los datos	31
4.3.1.2 Evolución y selección de la fuente de datos	31
4.3.2 Universo de análisis.....	32
4.3.3 Período de estudio y cantidad de ruedas.....	32
4.3.4 Normalización y ajuste de datos	33
4.4 Implementación de indicadores técnicos.....	33
4.5 Diseño del experimento de evaluación.....	33
4.5.1 Estrategias de trading.....	33
4.5.2 Parámetros de backtesting	34
4.5.3 Métricas de evaluación	34
4.5.4 Metodología de validación	35
4.6 Planificación y cronograma del desarrollo.....	35

Capítulo 5. Desarrollo e Implementación de la Plataforma	37
5.1 Tecnologías utilizadas	37
5.2 Módulos implementados	38
5.2.1 Ingesta de datos	38
5.2.2 Normalización y ajuste	38
5.2.3 Generación de series sintéticas MEP	39
5.2.4 Cálculo de indicadores.....	39
5.2.5 Generación de señales	39
5.2.6 Backtesting	40
5.2.7 Exposición y visualización web	40
5.3 Desafíos técnicos y soluciones.....	41
5.4 Reproducibilidad y trazabilidad	42
5.5 Marco de gobernanza	43
5.5.1 Gobierno de datos.....	43
5.5.2 Gobierno del código	44
5.5.3 Gobierno operativo	44
5.5.4 Gobierno de la curación.....	45
5.5.5 Gobierno del riesgo	46
5.6 Resiliencia operativa	46
Capítulo 6. Resultados y Análisis	47
6.1 Resultados del backtesting	47
6.1.1 HMA16.....	47
6.1.2 Medias móviles clásicas	48
6.2 Análisis comparativo de desempeño	48
6.3 Discusión de los resultados	51
6.3.1 La media como filtro de desastre.....	51
6.3.2 Corrección de look-ahead bias en los validadores.....	52

6.4 Validación visual de la serie sintética MEP	53
Capítulo 7. Conclusiones	59
7.1 Conclusiones principales.....	59
7.1.1 El caso de HMA16	60
7.2 Cumplimiento de objetivos	60
7.3 Contribuciones del trabajo	61
7.4 Limitaciones.....	62
7.5 Líneas de trabajo futuro.....	63
Anexos	65
Anexo I. Pseudocódigo y diagramas	65
I.1 Pseudocódigo de los módulos	65
I.2 Diagramas.....	69
Anexo II. Interfaz web: capturas y esquema JSON	71
II.1 Interfaz web	71
II.2 Esquema de los archivos JSON	73
Anexo III. Manual de operación y reproducción	76
III.1 Manual de operación	76
III.1.1 Ejecución del pipeline.....	76
III.1.2 Dependencias e instalación	76
III.1.3 Credenciales.....	77
III.1.4 Disparo de una corrida	77
III.1.5 Recuperación ante fallas	77
III.2 Procedimiento de curaduría del panel de eventos	78
III.2.1 Alcance y encuadre	78
III.2.2 Fuente y flujo de edición.....	79
III.2.3 Criterio de admisión y auditoría	80
III.2.4 Extensión futura	80

III.3 Protocolo de verificación del estado congelado	80
III.3.1 Qué garantiza y qué no	80
III.3.2 Procedimiento	81
III.3.3 Advertencia sobre fin de línea	82
Referencias.....	83

Índice de tablas

Tabla 1. Stack tecnológico de la plataforma	37
Tabla 2. Retorno anualizado de HMA16 por universo y corte	47
Tabla 3. Retorno anualizado de las medias clásicas por universo y corte	48
Tabla 4. Tickers que superan a Buy & Hold por método, universo y corte.....	49
Tabla 5. Drawdown máximo (mediana) por método, universo y corte	50
Tabla 6. Retorno anualizado por método, universo y corte según el costo	50
Tabla 7. Validación de la serie sintética MEP contra el ADR real, por acción	55
Tabla 8. Esquema de summary.json (resumen por serie)	75
Tabla 9. Esquema de ticker_{TICKER}.json (serie completa)	75
Tabla 10. Variables de entorno requeridas por el módulo de ingesta.....	77
Tabla 11. Campos de un evento en el archivo de origen	79

Índice de figuras

Figura 1. Arquitectura del pipeline	27
Figura 2. Costo computacional de la etapa de indicadores sobre series sintéticas	29
Figura 3. Cronograma de desarrollo de la plataforma	36
Figura 4. YPF, MEP sintético y ADR	56
Figura 5. YPF en base 100, las tres series.....	56
Figura 6. GGAL, MEP sintético y ADR.....	57
Figura 7. GGAL en base 100, las tres series.....	57
Figura 8. CRES, MEP sintético y ADR.....	58
Figura 9. CRES en base 100, las tres series	58
Figura 10. Flujo de datos entre módulos del pipeline	69
Figura 11. Secuencia de ejecución del pipeline	70
Figura 12. Máquina de estados de señales	70

Figura 13. Panel de señales del día	71
Figura 14. Detalle del activo	72
Figura 15. Eventos del mercado.....	72

Resumen

El trabajo desarrolla una plataforma que lee señales de mercado con medias móviles, apoyada en HMA16. Descarga las cotizaciones de las acciones locales y las deja normalizadas. Para cada *ticker* arma una serie sintética en dólares MEP, y sobre esa serie calcula los indicadores que después producen la señal de cada rueda. Esa señal se exporta en JSON. El archivo lo lee una web pensada para usuarios que no necesariamente manejan finanzas.

El tipo de cambio MEP necesita el precio del AL30D, y ese bono recién empezó a cotizar el 14 de septiembre de 2020. Por eso el *backtesting* no comienza antes de esa fecha. Sin esa cotización no hay MEP que calcular. Desde ahí en adelante, cada señal se contrasta con un *Buy & Hold* de referencia.

El objetivo es identificar temprano las señales de compra o venta. Las señales falsas son el riesgo. Para filtrarlas, el sistema suma la fuerza del volumen y confirma con patrones de velas. La evaluación se apoya en el retorno anualizado y en la máxima caída (*drawdown*), siempre contra ese *Buy & Hold* de base. Queda registrado además cuánto tiempo estuvo el capital invertido, con las reglas del sistema documentadas.

Se entrega una web pública. Muestra las salidas por instrumento y suma una sección aparte de análisis fundamental como complemento.

Palabras clave: análisis técnico, ADR, backtesting, HMA16, medias móviles, MEP

Abstract

This work develops a platform that reads market signals from moving averages, built around HMA16. It downloads local equity quotes and normalizes them. For each ticker it builds a synthetic series in MEP dollars, and from that series it computes the indicators that then produce the signal for each trading session. That signal is exported as JSON. The file is read by a web interface aimed at users who do not necessarily have a finance background.

The MEP exchange rate requires the price of the AL30D bond, which only began trading on September 14, 2020. For that reason, the backtesting does not begin before that date. Without that quote there is no MEP to compute. From then on, each signal is measured against a benchmark Buy & Hold.

The goal is to identify buy or sell signals early. False signals are the risk. To filter them, the system adds volume strength and confirms with candlestick patterns. Evaluation rests on the annualized return and the maximum drawdown, always against that baseline Buy & Hold. It also records how long capital stayed invested, with the system's rules documented.

A public web interface is delivered. It shows the outputs per instrument and adds a separate section of fundamental analysis as a complement.

Keywords: technical analysis, ADR, backtesting, HMA16, MEP, moving averages

Capítulo 1. Introducción

1.1 Naturaleza del proyecto

El mercado de capitales argentino es muy volátil. Esa volatilidad empuja a los inversores, profesionales y minoristas, a buscar métodos simples para identificar oportunidades y manejar el riesgo. Uno de los más usados es el análisis técnico, que se apoya en los datos de precios para anticipar movimientos y suele acompañarse de un análisis fundamental.

Este trabajo es una propuesta de intervención profesional, orientada a desarrollar una solución tecnológica para el análisis financiero, desde el diseño hasta la implementación. La solución es una herramienta informática que analiza acciones locales con American Depositary Receipt (ADR) internacional, junto con versiones sintéticas de esas acciones expresadas en dólares mediante el tipo de cambio MEP, para atenuar la distorsión que la inflación y la devaluación del peso introducen en la serie local. Sobre una serie en dólares, la tendencia queda a la vista, sin el efecto de la devaluación que la encubre en la serie en pesos. El cotejo con el ADR internacional confirma si la serie sintética replica de forma razonable el comportamiento del activo en el exterior.

El objetivo es darle al usuario un análisis preliminar que combine lo técnico y lo fundamental, con las señales de cada sistema de medias a la vista y su desempeño histórico medido contra comprar y mantener. El trabajo no recomienda operaciones. Le da al inversor la evidencia para juzgar cada señal por su historial y decidir con criterio propio.

El trabajo también expone una arquitectura de *software* simplificada y modular. Está pensada para que otros desarrolladores o investigadores la entiendan y prueben sobre ella sus propios métodos de análisis técnico, con indicadores nuevos o estrategias distintas.

1.2 Justificación

El análisis financiero personal lleva tiempo y no es fácil de organizar. Ya existen herramientas para eso, y siempre está la opción de delegar en asesores o intermediarios. Este trabajo propone una plataforma que simplifica el proceso, con análisis técnico basado en medias móviles y énfasis en la HMA16 por su menor retraso y su mayor suavizado.

En 2020, la inflación y las restricciones cambiarias empujaron a los ahorristas argentinos a proteger su capital (Sánchez, 2024). De esa necesidad de herramientas accesibles, que no exijan formación técnica, parte este trabajo.

Sin procedimientos estandarizados o personalizables para evaluar señales técnicas, el inversor no puede comparar ni validar sus estrategias con criterio objetivo. Una plataforma modular y transparente, que integre análisis técnico y fundamental básico, responde a esa carencia.

La inflación local camufla un patrón lineal o incluso una tendencia bajista leve. Comparar la acción local contra su versión sintética en dólares MEP muestra lo que la serie en pesos oculta.

Con una arquitectura clara y extensible, un inversor no profesional puede replicar el trabajo sin gran dificultad técnica, al tiempo que ofrece a desarrolladores e investigadores una base para experimentar con otros indicadores y métodos.

1.3 Marco institucional

La herramienta apunta a inversores minoristas y a investigadores interesados en el análisis técnico reproducible en el contexto argentino. Sus datos vienen de fuentes públicas e incluyen precios de acciones locales y el tipo de cambio MEP, más algunos indicadores fundamentales básicos. El tratamiento de esa información respeta los términos de uso y las licencias que correspondan, dentro de las buenas prácticas del área.

Para acceder a los datos históricos hace falta la API del bróker *online* PPI. Las evaluaciones corren sobre series históricas desde el 14 de septiembre de 2020, sin ejecuciones reales. Cada resultado suma un costo de transacción fijo del 0,5% por punta (1% por operación completa), tomado como referencia para el mercado minorista argentino.

1.4 Objetivo general

Diseñar e implementar una plataforma modular para monitorear y evaluar señales técnicas basadas en distintos sistemas de medias móviles, con HMA16 como referencia principal. La plataforma valida esas señales con patrones de velas japonesas y compara el rendimiento de las medias móviles entre acciones locales y sus versiones sintéticas en dólares MEP.

1.4.1 Objetivos particulares

- Implementar el *pipeline* de ingesta de datos, cálculo de indicadores y exportación de resultados para su visualización y análisis.
- Registrar de forma reproducible cada ejecución del pipeline, con su configuración y sus resultados exportables.
- Desarrollar la interfaz web que muestre las señales a usuarios sin formación financiera.
- Comparar el rendimiento de HMA16 con otras medias móviles mediante backtests, con métricas que un inversor no especializado pueda leer.

Capítulo 2. Estado del Arte

2.1 Delimitación del alcance de la revisión

Este capítulo sintetiza la evidencia práctica y académica sobre medias móviles, HMA16, la relación Local–ADR y buenas prácticas de backtesting pertinentes.

La revisión cubre materiales aplicados, documentación de indicadores y metodologías reproducibles para evaluar señales basadas en medias móviles en mercados emergentes.

2.2 Estudios sobre medias móviles clásicas

Las medias móviles son herramientas de suavizado utilizadas para identificar tendencias y generar señales de *trading* (Ellareddy, 2025).

Lo et al. (2000) sostienen que el suavizado de precios como un problema estadístico donde el grado de promediado define un *trade-off*. Con poco suavizado la curva ajusta el ruido y con demasiado se pierde información local. Sus resultados muestran además que el contenido informativo de los indicadores técnicos varía según el mercado analizado, con diferencias marcadas entre los papeles de NYSE/AMEX y los de Nasdaq.

Revisiones de evidencia histórica indican que las estrategias basadas en medias móviles funcionan mejor en mercados tendenciales y no tanto en laterales (Hurst et al., 2017).

El material comparativo disponible es mayormente de proveedores, blogs y repositorios de scripts, con pocos artículos revisados por pares. Conviene tratarlo como evidencia secundaria y verificarlo con backtests reproducibles (Bailey et al., 2015).

2.3 Hull Moving Average y la elección de HMA16

La mayor parte del material disponible sobre HMA16 y su comparación con medias móviles clásicas es material aplicado, como guías técnicas, artículos de divulgación, repositorios de scripts y otras fuentes no académicas. Estas fuentes explican el cálculo, muestran comparaciones visuales y ofrecen ejemplos de implementación, pero no reemplazan estudios académicos con metodología reproducible y análisis estadístico.

Las implementaciones públicas muestran variantes y filtros usados en la práctica, pero no validan rendimiento. En scripts y repositorios públicos son frecuentes las advertencias sobre *repainting* y señales anticipadas en contextos ruidosos (Bailey et al., 2015).

Jiménez-Méndez y Álvarez-Lamus (2024) concluyen que el análisis técnico basado en indicadores comunes, como MACD, RSI y medias móviles, obtiene mejores resultados al combinar varios indicadores.

En el trabajo de Ospina-Holguín y Padilla-Ospina (2025) se propone un algoritmo basado en aprendizaje por refuerzo¹ para combinar señales de reglas de cruce de medias móviles con el objetivo de maximizar el alfa (retorno excedente ajustado por riesgo). Como conclusión se evidencia que la utilización combinada de medias móviles entrega mejores resultados que centrarse en una única, aunque vale aclarar que se cubren medias móviles tradicionales y no HMA en ninguna de sus variantes.

2.4 Antecedentes de dolarización de series para análisis técnico

La práctica de expresar las series de acciones argentinas en dólares para su análisis técnico no es nueva ni exclusiva del ámbito académico. En el plano de las herramientas, AnalisisTecnico.com.ar (s. f.) es una plataforma comercial con más de una década de operación, orientada al inversor minorista local, que ofrece gráficos de acciones, ADRs y CEDEARs con la posibilidad de ajustar cualquier símbolo por distintos tipos de cambio, entre ellos el CCL y el MEP, sumando un sistema de alarmas sobre precio, volumen y medias móviles. Su propósito es la visualización y el monitoreo. Presenta las series ajustadas y habilita el análisis gráfico, pero no incorpora un motor de backtesting que evalúe el desempeño de una estrategia, ni reporta métricas de rendimiento o riesgo, ni documenta controles sobre los sesgos que afectan la validez de una evaluación histórica. Tampoco valida la construcción de sus series dolarizadas contra una referencia externa como el ADR, y opera como servicio de datos cerrado, sin mostrar cómo calcula lo que publica.

En el ámbito académico local, Vintimilla (2021) construye el índice Merval en dólares mediante el CCL para un sistema de trading que busca superar a índices de referencia, y advierte sobre la sensibilidad metodológica de toda conversión de monedas en variables financieras argentinas. Su trabajo opera sobre el índice agregado y mediante el CCL, no sobre acciones individuales dolarizadas por MEP, y no contrasta las series construidas contra el ADR como control de consistencia.

¹ El aprendizaje por refuerzo (*reinforcement learning*) es un paradigma de ML en donde un agente se entrena utilizando un sistema de recompensas o penalizaciones como consecuencia de sus acciones.

2.5 Herramientas tecnológicas para análisis técnico y backtesting

Entre las herramientas de código abierto más utilizadas para backtesting se pueden mencionar Backtrader y Zipline, que permiten simular estrategias sobre datos históricos. Sarasa-Cabezuelo (2023) presenta una aplicación web académica de backtesting que permite la simulación de estrategias y gestión de carteras con foco en usabilidad para inversores no especializados.

Estas plataformas están diseñadas para mercados con una única moneda. Argentina formalmente lo es, pero en la práctica convive con una dualidad monetaria en varios segmentos, incluido el financiero, y estas herramientas no soportan de forma nativa la normalización a dólares que eso exige.

Las fuentes gratuitas de datos suelen tener imprecisiones que pueden ser o no aceptables según el objetivo del usuario. En cualquier caso, es recomendable documentar fecha y hora de las descargas y conservar copias locales (Bailey et al., 2015).

2.6 Métodos de backtesting y evaluación de estrategias

Para que un backtest sea creíble es necesario evitar errores de diseño como el *look-ahead bias* y el *survivorship bias*. También resulta importante modelar de manera realista los costos de transacción y el *slippage*, ya que pueden cambiar de forma significativa el resultado de una estrategia (López de Prado, 2018).

Una práctica habitual para reducir el riesgo de sobreajuste consiste en separar los datos en conjuntos distintos para entrenamiento y prueba (validación *out-of-sample*), con esquemas simples de *holdout* o enfoques más dinámicos como el *walk-forward*. (Bailey et al., 2015) advierten que el holdout pierde confiabilidad cuando la estrategia se elige tras probar muchas configuraciones sobre la misma muestra, porque el método no contabiliza la cantidad de intentos previos. Ese riesgo se atenúa cuando los parámetros se fijan de antemano con valores estándar y no se optimizan sobre los datos. El análisis de sensibilidad sobre parámetros clave, como el valor de n en la HMA, finaliza esa separación.

Para la evaluación de resultados, la literatura propone reportar al menos un conjunto básico de métricas que incluya el retorno acumulado, el drawdown máximo, el porcentaje de operaciones ganadoras, el número total de operaciones y la duración media por operación. Estas medidas permiten evaluar la rentabilidad, el perfil de riesgo y la estabilidad de la estrategia (Hurst et al., 2017).

Deep et al. (2025) advierten que más del 90% de las estrategias académicas fallan al implementarse con capital real. Para atenuar este problema se proponen *frameworks* de

validación walk-forward con múltiples períodos de prueba out-of-sample que respetan la información disponible en cada momento y aplican costos de transacción realistas.

2.7 Plataformas web de backtesting orientadas a usabilidad

Riofrio López y Pinargote (2025) desarrollaron una plataforma web de backtesting orientada a usuarios no expertos con foco en mejorar la experiencia de uso. La validación se realizó con encuestas a diez operadores financieros donde el 70% reportó mejor comprensión del mercado.

La validación se basa en la percepción de los usuarios y no tanto en la calidad de los backtests. La mencionada encuesta mide si la herramienta es entendible, lo cual evidencia que la plataforma prioriza la educación por encima de los resultados. El artículo tampoco publica la documentación técnica del motor de simulación, así que no hay forma de saber si previene look-ahead bias ni si modela costos de transacción (López de Prado, 2018).

2.8 Integración de Análisis Técnico con Procesamiento de Lenguaje Natural

Solis Arellano (2025) desarrolló un sistema para criptomercados que integra análisis técnico con análisis de sentimiento de noticias y publicaciones utilizando un LLM² (LLaMA 3.1) para generar reportes contextualizados. El contexto no es el del mercado argentino, pero la integración es aplicable en lo técnico.

La integración expone dos limitaciones técnicas. Una es que los resultados generados por LLM no siempre son reproducibles entre ejecuciones, lo que dificulta la auditoría que exigen las buenas prácticas de backtesting (Bailey et al., 2015). La otra es el alineamiento temporal entre la publicación de la noticia y la rueda de decisión, crítico para evitar look-ahead bias.

2.9 Brechas identificadas

Existe escasez de estudios académicos que comparen explícitamente la HMA con conjuntos de medias clásicas, como EMA12/26 y SMA10/50/100, en mercados emergentes, y que además consideren la dolarización de series locales para una comparación libre del efecto inflacionario. La evidencia empírica disponible que favorece a la elección de la HMA proviene principalmente de fuentes no académicas, por lo que no puede considerarse concluyente sin

² LLaMA 3.1 es un *Large Language Model* (LLM) desarrollado por Meta. Estos modelos son entrenados con grandes volúmenes de texto y pueden generar y comprender lenguaje natural

una evaluación con análisis de sensibilidad de parámetros y validación fuera de muestra (Bailey et al., 2015).

Los antecedentes recientes tampoco cubren la brecha. Riofrio López y Pinargote (2025) se centran en usabilidad por sobre la calidad del backtest, y Solis Arellano (2025) opera sobre criptomercados, no sobre el mercado argentino con sus particularidades de CCL y *parking*.

El relevamiento muestra, además, que la dolarización de series locales para análisis técnico ya está disponible tanto en herramientas comerciales del mercado argentino como en trabajos académicos locales, de modo que no constituye en sí una novedad. Lo que no se halló, ni en la literatura ni en las herramientas relevadas, es una solución que combine la construcción de series dolarizadas por el tipo de cambio MEP, accesible al inversor minorista, con un backtesting reproducible, validado externamente contra el ADR y con controles documentados contra el sesgo de anticipación, sobre una arquitectura abierta y extensible. Se presenta entonces la oportunidad de aportar evidencia empírica local que contraste la HMA16 con sistemas de medias clásicas en el contexto argentino, dolarizando las series con el MEP y reportando costos y supuestos de ejecución, sobre una base reproducible y auditable.

Capítulo 3. Marco Teórico

3.1 Conceptos fundamentales de trading y análisis técnico

Murphy (2000) describe el análisis técnico como el estudio del comportamiento de los mercados a partir del registro histórico de precios y volúmenes, orientado a anticipar el rumbo de las cotizaciones. Su premisa es que el precio ya incorpora toda la información relevante, de manera que alcanza con examinar su trayectoria para reconocer oportunidades de operación. Esto lo diferencia del análisis fundamental, que estima el valor de un activo a partir de variables económicas y contables.

El enfoque se apoya en la Teoría de Dow, según la cual el mercado descuenta toda la información disponible y los precios se desplazan en tendencias que se sostienen hasta que aparece una señal clara de reversión. De esos principios derivan tanto las representaciones gráficas de la serie de precios como los indicadores que se construyen sobre ella, pensados para volver identificable el comportamiento del mercado (Murphy, 2000).

Un indicador técnico es una transformación matemática de la serie de precios que resalta alguna de sus propiedades, como la dirección o la fuerza de la tendencia. Dentro de ese conjunto, Murphy (2000) ubica a las medias móviles entre las más utilizadas, por la sencillez de su cálculo y su eficacia para atenuar las oscilaciones de corto plazo.

3.1.1 Medias móviles

Una media móvil es un promedio del precio de un activo sobre una ventana de tiempo definida y se recalcula a medida que entran nuevos datos, desplazándose junto con la serie. Lo que busca es suavizarla y al promediar, atenúa las oscilaciones de corto plazo y deja a la vista la tendencia de fondo (Murphy, 2000).

Su parámetro clave es el largo de la ventana. Una ventana corta sigue de cerca al precio y reacciona rápido, pero capta más ruido y dispara más señales falsas. Una larga entrega una curva más estable, al costo de retrasarse frente al precio actual. Ese balance es lo que diferencia a una media móvil de otra.

Las medias móviles se distinguen, además, por cómo reparten el peso entre las observaciones de la ventana. Las más habituales son la simple, la exponencial y la ponderada, que se desarrollan en los apartados siguientes. El esquema de ponderación define cuánto influye el dato más reciente frente al más antiguo y, con ello, el retraso del indicador (Murphy, 2000).

3.1.2 Media móvil simple (Simple Moving Average, SMA)

La media móvil simple (Simple Moving Average, SMA) es la más directa de todas: promedia los precios de cierre de una ventana de n períodos dando el mismo peso a cada uno. Se expresa como:

$$SMA_t = \frac{1}{n} \sum_{i=0}^{n-1} P_{t-i}$$

donde P_{t-i} es el cierre del período $t-i$ y n el largo de la ventana (Murphy, 2000, p. 226)

Su fuerte es la simplicidad de cálculo e interpretación. La debilidad nace de esa misma ponderación uniforme: como el dato más antiguo de la ventana pesa igual que el más reciente, la SMA tarda en reaccionar y arrastra un retraso (*lag*) proporcional al largo de la ventana. Cuanto más larga, más suave resulta la curva, pero más tarde llega la señal de un cambio de tendencia.

3.1.3 Media móvil exponencial (Exponential Moving Average, EMA)

La media móvil exponencial (Exponential Moving Average, EMA) surge para corregir la principal carencia de la simple: su lentitud. En lugar de pesar todos los datos por igual, asigna un peso que decrece de forma exponencial a medida que las observaciones se alejan en el tiempo, de modo que el cierre más reciente domina el resultado. Su forma recursiva es:

$$EMA_t = \alpha \cdot P_t + (1 - \alpha) \cdot EMA_{t-1}$$

donde el factor de suavizado α se calcula habitualmente como $\alpha = 2 / (n + 1)$, siendo n el largo del período (Murphy, 2000, pp. 227-228).

Al privilegiar lo reciente, la EMA gira antes que la SMA cuando el precio cambia de dirección, y por eso se recorta el retraso en detectar una tendencia nueva. Esa misma característica la vuelve más sensible al ruido, ya que reacciona a movimientos breves que la SMA filtraría. La elección entre ambas es un trade-off entre rapidez y estabilidad (Murphy, 2000, p. 228).

3.1.4 Media móvil de Hull (Hull Moving Average, HMA16)

La media móvil de Hull fue propuesta por Alan Hull en 2005 con un objetivo concreto: reducir de forma drástica el retraso propio de las medias tradicionales sin perder el suavizado de la curva. Para lograrlo combina medias móviles ponderadas anidadas y amortigua el suavizado empleando la raíz cuadrada del período en lugar del período completo (Hull, 2005).

El bloque de base es la media móvil ponderada (*Weighted Moving Average*, WMA), que asigna pesos linealmente crecientes a las observaciones más recientes. Por ejemplo, en una media de diez días, el último cierre se multiplica por diez, el anterior por nueve, y así sucesivamente, dividiendo luego por la suma de los pesos (Murphy, 2000, p. 227). Su forma general es:

$$WMA_n = \frac{\sum_{i=1}^n w_i P_i}{\sum_{i=1}^n w_i}, w_i = i$$

A partir de esa media ponderada, la HMA se construye en tres pasos:

$$HMA_n = WMA \left(2 WMA_{\frac{n}{2}} - WMA_n, \sqrt{n} \right)$$

La resta entre el doble de la WMA de período mitad y la WMA de período completo cancela gran parte del retraso, mientras que el suavizado final sobre la raíz cuadrada del período limpia el ruido que esa operación introduce. El resultado es una media bastante más reactiva que la SMA o la EMA del mismo período (Hull, 2005).

La variante empleada usa un período de 16, de donde proviene la denominación HMA16. El parámetro se fija a priori, esto es, se define antes de la evaluación empírica y no se ajusta sobre los datos del estudio. El motivo es que optimizar un parámetro contra los mismos datos con los que luego se lo evalúa introduce sobreajuste (*overfitting*), y fijarlo de antemano descarta ese riesgo (López de Prado, 2018, p. 11).

3.1.5 Generación de señales: cruces y puntos de giro

Una media móvil, por sí sola, no indica cuándo comprar o vender. La señal surge de la relación entre la media y el precio, o entre dos medias de distinto período. El mecanismo más frecuente es el cruce. Cuando el precio de cierre supera a la media, o una media rápida cruza por encima de una lenta, se interpreta una señal de compra, y el cruce inverso, una de venta (Murphy, 2000, p. 227).

Ese esquema depende, sin embargo, de la diferencia de retraso entre las series que se cruzan. Por eso el propio Hull no aconseja aplicarlo a la HMA. Como su retraso ya fue reducido al mínimo, el cruce pierde sentido, y en su lugar recomienda leer los puntos de giro de la curva, es decir, los cambios en la dirección de su pendiente, como señales de entrada y salida (Hull, 2005).

3.1.6 El volumen

El volumen mide la cantidad de operaciones realizadas sobre un activo en un determinado período y refleja la intensidad de la participación del mercado. Murphy (2000) lo describe como un indicador de confirmación, y de hecho lo considera el más importante de ese tipo (p. 186). Un movimiento de precios acompañado por volumen alto se interpreta como más sólido, porque confirma participación de inversores, mientras que un movimiento sostenido por volumen bajo sugiere una señal débil y más expuesta a revertirse.

Por esa función de confirmación, el volumen se usa como filtro para reducir la proporción de señales falsas. Una señal gana fuerza cuando se produce en un contexto de participación superior a la habitual, donde la referencia para juzgar ese nivel es el comportamiento reciente del propio volumen (Murphy, 2000, p. 190).

3.1.7 Patrones de velas japonesas

El gráfico de velas japonesas representa los mismos cuatro precios de cada rueda que un gráfico de barras, apertura, máximo, mínimo y cierre, pero los muestra mediante un cuerpo y dos sombras. El cuerpo abarca el rango entre la apertura y el cierre, y su color diferencia las ruedas alcistas de las bajistas según cuál de esos dos precios quede más alto. Las líneas finas por encima y por debajo del cuerpo, llamadas sombras, marcan el máximo y el mínimo de la jornada (Murphy, 2000, p. 322).

Determinadas configuraciones de cuerpo y sombras, ya sea en una sola vela o en secuencias de varias, se asocian con la continuación o la reversión de la tendencia, y se agrupan en patrones de continuidad y patrones de inversión (Murphy, 2000, p. 325).

3.2 Mercado de capitales argentino: acciones y ADRs

El mercado de capitales argentino se ordena a partir de la Ley 26.831 de 2012, de Mercado de Capitales, que reformuló la estructura institucional del sector y definió las funciones de sus actores. El organismo de regulación y supervisión de la oferta pública es la Comisión Nacional de Valores (CNV), y bajo su autorización funcionan los mercados de

negociación y las categorías de agentes que intermedian las operaciones, denominados Agentes de Liquidación y Compensación (ALyC). El mercado local donde se negocian las acciones argentinas es Bolsas y Mercados Argentinos (BYMA), constituido en 2017 a partir de la escisión del antiguo Mercado de Valores de Buenos Aires. La custodia y el movimiento de los valores negociables están a cargo de la Caja de Valores (Erpen, 2010, p. 79).

Entre los instrumentos que se negocian se distinguen las acciones, que representan una fracción del capital de una empresa (Erpen, 2010, p. 60), y los títulos públicos, instrumentos de deuda emitidos por el Estado (Erpen, 2010, p. 54). Existen además los Certificados de Depósito Argentinos (CEDEAR), que permiten negociar de forma local acciones de empresas del exterior (Erpen, 2010, p. 63). Por el contrario, las acciones de empresas argentinas pueden negociarse en mercados estadounidenses mediante American Depositary Receipts (ADR), títulos emitidos por un banco de ese país que representan un número determinado de acciones extranjeras (Berk y DeMarzo, 2008, p. 78).

3.3 Paridad entre acciones locales y ADRs

La relación entre una acción local y su ADR se apoya en un principio básico de valuación. La Ley del Precio Único establece que un mismo bien o título negociado de forma simultánea en distintos mercados competitivos debe tener el mismo precio en todos ellos, lo que equivale a la ausencia de oportunidades de arbitraje (Berk y DeMarzo, 2008, p. 60). El mecanismo que sostiene esa igualdad es el arbitraje, la operación de comprar el activo donde está más barato y venderlo donde está más caro, que al repetirse empuja las cotizaciones hasta eliminar la diferencia.

En el caso de una acción argentina, el ADR que cotiza en el exterior representa una cantidad fija de acciones locales, de modo que ambos instrumentos reflejan el valor de la misma empresa (Berk y DeMarzo, 2008, p. 78). Por la Ley del Precio Único sus cotizaciones deberían guardar una relación estable, y cualquier desvío persistente abriría una oportunidad de arbitraje que tendería a corregirlo. Dado que una se expresa en pesos y la otra en dólares, esa relación incorpora el tipo de cambio implícito entre ambas plazas.

3.3.1 El dólar MEP

El dólar MEP (Mercado Electrónico de Pagos), también conocido como dólar bolsa, es un tipo de cambio implícito que surge de operar un mismo título en dos monedas. Consiste en comprar un bono en pesos y venderlo en dólares con liquidación en plaza local, de manera que el cociente entre su precio en pesos y su precio en dólares determina el tipo de cambio

resultante. Su fundamento es la Ley del Precio Único, ya que un mismo activo expresado en dos monedas revela, a través de la relación entre sus precios, el tipo de cambio implícito entre ellas (Berk y DeMarzo, 2008, p. 78).

El dólar MEP se diferencia del contado con liquidación (CCL), o dólar cable, únicamente por el lugar de liquidación, ya que el primero liquida los dólares en plaza local y el segundo mediante transferencia a cuentas radicadas en el exterior (Matba Rofex, 2024). La brecha de precio entre ambos, conocida como canje, refleja el costo de mover los fondos hacia o desde el exterior. La Comisión Nacional de Valores (2025) regula la operatoria que da lugar a estos tipos de cambio.

Históricamente, esta operatoria estuvo sujeta a un plazo mínimo de permanencia en cartera, conocido como parking, que obligaba a mantener el bono un tiempo antes de venderlo en la otra especie. Para las personas humanas residentes, ese plazo se redujo de forma progresiva hasta su eliminación por la Resolución General N° 1062/2025 de la Comisión Nacional de Valores (2025).

3.3.2 Cálculo de la paridad teórica Local-ADR

La Ley del Precio Único permite expresar la equivalencia de precios entre una acción local y su ADR. Como cada ADR representa un número determinado de acciones locales, definido por el *ratio* de conversión, su precio en dólares debe igualar al de esas acciones llevadas a dólares mediante el tipo de cambio (Matba Rofex, 2024, p. 9). La paridad teórica puede escribirse entonces como:

$$P_{ADR} = \frac{P_{local}r}{TC}$$

donde P_{ADR} es el precio del ADR en dólares, P_{local} el precio de la acción local en pesos, r el ratio de conversión de acciones locales por ADR y TC el tipo de cambio en pesos por dólar.

El tipo de cambio que corresponde a esta equivalencia es el cable, dado que el ADR cotiza y liquida en el exterior. Cuando los precios observados se apartan de esta paridad, la diferencia se atribuye a las fricciones que limitan el arbitraje, como los costos de transacción, que acotan el rango dentro del cual los precios pueden diferir sin abrir una oportunidad de arbitraje (Berk y DeMarzo, 2008, p. 73).

Para el mercado argentino, esta brecha está documentada de forma empírica. Auguste et al. (2006) muestran que, bajo controles de cambio, el precio de una acción local convertido a dólares se aparta del de su ADR, y que esa diferencia refleja la expectativa de devaluación del mercado y el costo de eludir las restricciones cambiarias. Pasquariello (2008) agrega que

durante las crisis financieras la Ley del Precio Único se debilita en el mercado de ADRs de economías emergentes, y las desviaciones entre el ADR y su activo subyacente se amplían. Estos antecedentes dan el marco para interpretar la validación de las series sintéticas de este trabajo, donde cada serie en dólares MEP se contrasta contra el ADR real de la empresa: en condiciones normales ambas deberían seguirse de cerca, y los desvíos observados responden a las mismas fricciones que la literatura identifica

3.4 Conceptos de backtesting y evaluación de estrategias

El backtesting consiste en evaluar una estrategia de inversión sobre datos históricos, simulando las operaciones que habría generado con el objetivo de simular su desempeño pasado (López de Prado, 2018, p. 151). Su validez depende de que la simulación reproduzca con fidelidad las condiciones reales de operación y, sobre todo, de que no incorpore información que no habría estado disponible en el momento de cada decisión.

El primero de los sesgos que dificultan esa validez es el sesgo de anticipación (*look-ahead bias*), que ocurre cuando la simulación utiliza datos que en la práctica todavía no se conocían al momento de operar. Un backtest contaminado por este sesgo arroja un desempeño irrealmente favorable, ya que decide con información del futuro (López de Prado, 2018, p. 151).

López de Prado (2018) también describe el sobreajuste, que aparece cuando una estrategia se ajusta a los datos históricos hasta capturar su ruido particular en lugar de un patrón genuino. Una estrategia sobreajustada muestra resultados excelentes en el período de prueba y demuestra un menor rendimiento al enfrentar datos nuevos (pp. 153-154).

El sesgo de supervivencia (*survivorship bias*) surge al conformar el universo de análisis solo con los activos que siguen en actividad al final del período, dejando afuera los que dejaron de cotizar. Como los activos que desaparecieron suelen ser los de desempeño más bajo, su exclusión tiende a sobreestimar los resultados de la estrategia (López de Prado, 2018, p. 152).

La utilidad de una estrategia se evalúa por comparación con un punto de referencia. Murphy (2000) plantea esta cuestión mediante la Teoría del Paseo Aleatorio, que pone en duda que alguna técnica de pronóstico logre superar a una simple estrategia de comprar y retener (*buy and hold*), un referente pasivo que consiste en adquirir el activo y conservarlo durante todo el período (p. 42). Una estrategia activa solo se justifica si supera a este criterio básico de inversión.

Capítulo 4. Abordaje Metodológico

4.1 Tipo de investigación

Este trabajo es una intervención profesional de tipo aplicado, orientada a resolver un problema concreto con el diseño y la implementación de una solución informática. El producto central es una web funcional y reproducible, cuya construcción y documentación son el aporte del trabajo.

Sobre datos históricos del mercado argentino, el desarrollo suma una etapa de validación, encargada de verificar que el sistema produce salidas consistentes y trazables, con controles sobre los sesgos que afectan la validez de un backtesting.

4.2 Diseño de la plataforma

La plataforma se diseñó bajo la lógica de un pipeline modular, donde cada etapa procesa un conjunto de datos y persiste el resultado en archivos antes de pasar a la siguiente fase. Este desacoplamiento permite ejecutar cada módulo por separado y verificar su salida de manera independiente. De ahí salen la trazabilidad y la reproducibilidad que el trabajo exige.

4.2.1 Requisitos funcionales y no funcionales

El trabajo no tiene un cliente externo, así que el relevamiento de requisitos se apoyó en el rol de analista del autor y en las restricciones del propio dominio. Los requisitos no funcionales terminan pesando tanto como los funcionales, ya que la trazabilidad y la reproducibilidad son los criterios con los que se valida el sistema.

Requisitos funcionales

RF-01. Ingesta de datos: el sistema obtiene cotizaciones diarias de trece acciones del mercado local y de los bonos AL30 y AL30D desde la API de PPI. La carga histórica comienza el 4 de septiembre de 2020, diez ruedas antes del inicio de la evaluación, para dar ventana a los indicadores. Las actualizaciones posteriores son incrementales.

RF-02. Normalización: el sistema consolida las series descargadas en un esquema OHLCV uniforme, con una fecha y un registro por rueda.

RF-03. Generación de activos sintéticos en MEP: el sistema calcula el tipo de cambio MEP a partir del cociente entre AL30 y AL30D y expresa las series de cada acción en dólares MEP.

RF-04. Cálculo de indicadores: el sistema calcula la HMA16, las medias móviles simples SMA 10, 50 y 100, las exponenciales EMA 12 y 26, y la VMA20 sobre las series normalizadas.

RF-05. Generación de señales: el sistema deriva señales de tendencia, identifica patrones de velas y clasifica el volumen en categorías cualitativas a partir de los indicadores calculados.

RF-06. Backtesting: el sistema simula operaciones sobre las señales generadas y calcula métricas de desempeño por activo y por método.

RF-07. Visualización: el sistema expone las series, los indicadores y las señales en una interfaz web de acceso público.

Requisitos no funcionales

RNF-01. Trazabilidad: las salidas se persisten en CSV de texto plano. La corrida automática versiona en Git solo los crudos y los datos del sitio. El estado reproducible del trabajo se congela aparte, con un *tag* anotado y el *hash* SHA256 de los crudos. Cada corrida escribe un archivo de metadatos con la fecha y los parámetros del estudio.

RNF-02. Reproducibilidad: cada corrida estampa en el archivo de metadatos la versión del esquema de salida, el identificador del *commit* que la generó y su fecha de ejecución. Las versiones de las bibliotecas quedan fijadas en *requirements.txt*, de modo que un tercero pueda reconstruir el resultado desde un estado versionado.

RNF-03. Integridad temporal: el sistema impide el uso de información futura en la generación y la evaluación de señales. Una señal calculada con datos hasta el cierre de una rueda se ejecuta a la apertura de la rueda siguiente.

RNF-04. Modularidad: cada módulo mantiene una responsabilidad única. El cálculo de indicadores y la generación de señales permanecen separados para acotar el impacto de los cambios.

RNF-05. Incrementalidad: la ingesta evita la descarga y el reprocesamiento de datos ya almacenados mediante la deduplicación por fecha.

RNF-06. Portabilidad y costo de operación: la persistencia se resuelve con archivos y no requiere un servidor de base de datos. La interfaz se despliega como sitio estático.

RNF-07. Automatización: la actualización del pipeline puede programarse sin intervención manual.

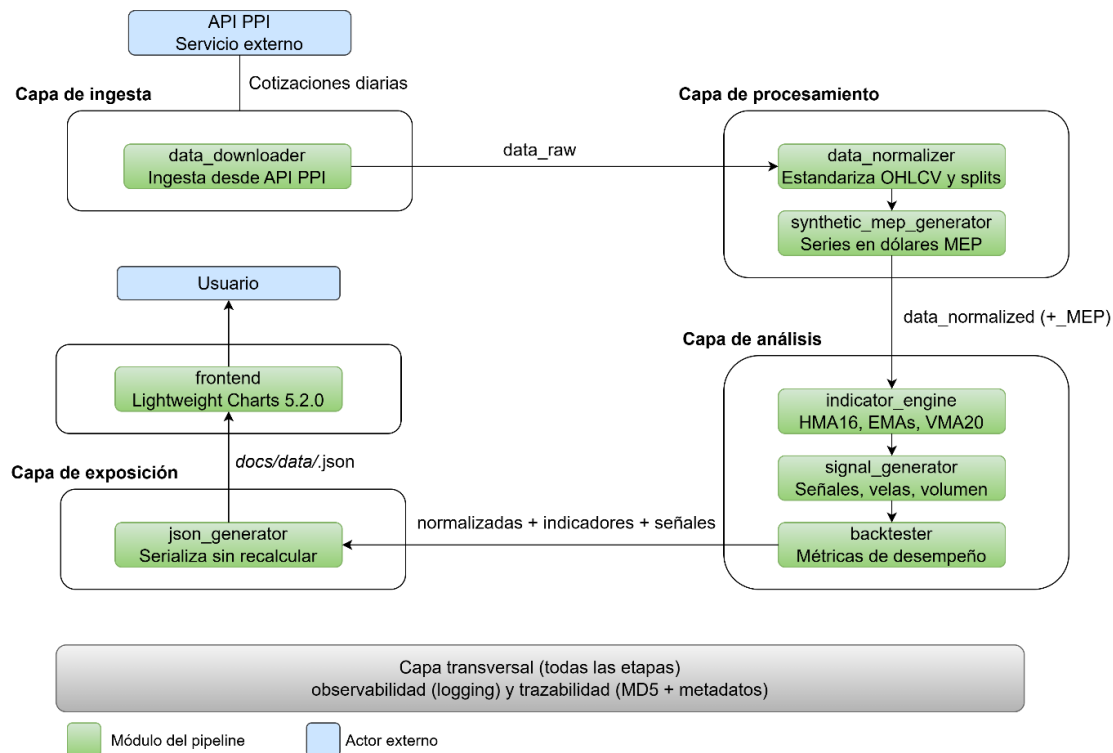
RNF-08. Observabilidad: el sistema registra el avance y los errores de cada ejecución mediante un esquema de *logging* unificado.

4.2.2 Arquitectura del sistema

La arquitectura se organiza como un pipeline de procesamiento por lotes, dividido en etapas independientes. Cada etapa lee los archivos que dejó la anterior, les aplica una transformación acotada y escribe su resultado en un directorio propio. Se ejecuta hasta completarse antes de pasar a la siguiente. El aislamiento entre etapas concreta la separación de responsabilidades de RNF-04 y permite volver a ejecutar una etapa sin recalculr las anteriores.

Figura 1

Arquitectura del pipeline



Nota. Elaboración propia.

El recorrido del dato sigue una secuencia fija. La ingesta obtiene las cotizaciones desde la API de PPI y las almacena como series crudas. La normalización las estandariza al esquema OHLCV y ajusta los precios por desdoblamientos de acciones (splits). La generación de tickers sintéticos toma esas series y calcula sus equivalentes en dólares MEP a partir del tipo de cambio implícito entre AL30 y AL30D. El cálculo de indicadores produce la HMA16, las medias simples y exponenciales y la VMA20, que la generación de señales interpreta para producir señales de tendencia, patrones de velas y una clasificación del volumen. El backtesting toma las señales generadas, simula las operaciones y escribe las métricas de desempeño en archivos JSON. Es la última etapa de procesamiento. El orden completo en que la integración continua llama cada módulo se detalla en el Anexo I.

El *frontend* permanece desacoplado del procesamiento. La interfaz web lee los archivos JSON ya calculados y solo los representa, sin ejecutar ningún cálculo en el navegador. Como el cálculo ya ocurrió, la web se publica de forma estática, sin un servidor detrás. Así se elimina el costo de *hosting* y se cumple con RNF-06.

Dos mecanismos atraviesan todas las etapas, la trazabilidad y la observabilidad. La trazabilidad sale de los propios archivos. Las salidas son CSV de texto plano versionados en Git, así que cualquier cambio en una serie aparece en el diff, y los datos crudos llevan un hash SHA256 anotado en el tag de la versión. La observabilidad corre por cuenta de un esquema de logging centralizado, que registra de forma independiente el avance y los errores de cada módulo.

4.2.3 Estrategia de persistencia de datos

La plataforma persiste sus datos en archivos planos. Las series se guardan en formato CSV y los metadatos en JSON, organizados en un directorio por cada etapa del pipeline. No se utiliza un motor de base de datos.

Con un universo acotado y series diarias desde septiembre de 2020, cada archivo entra holgadamente en memoria. El acceso es por lotes. Cada etapa lee un archivo completo, lo transforma y escribe el resultado, sin operaciones concurrentes ni consultas relacionales. Un motor de base de datos agregaría una dependencia de infraestructura y un costo de operación, sin resolver ningún problema real.

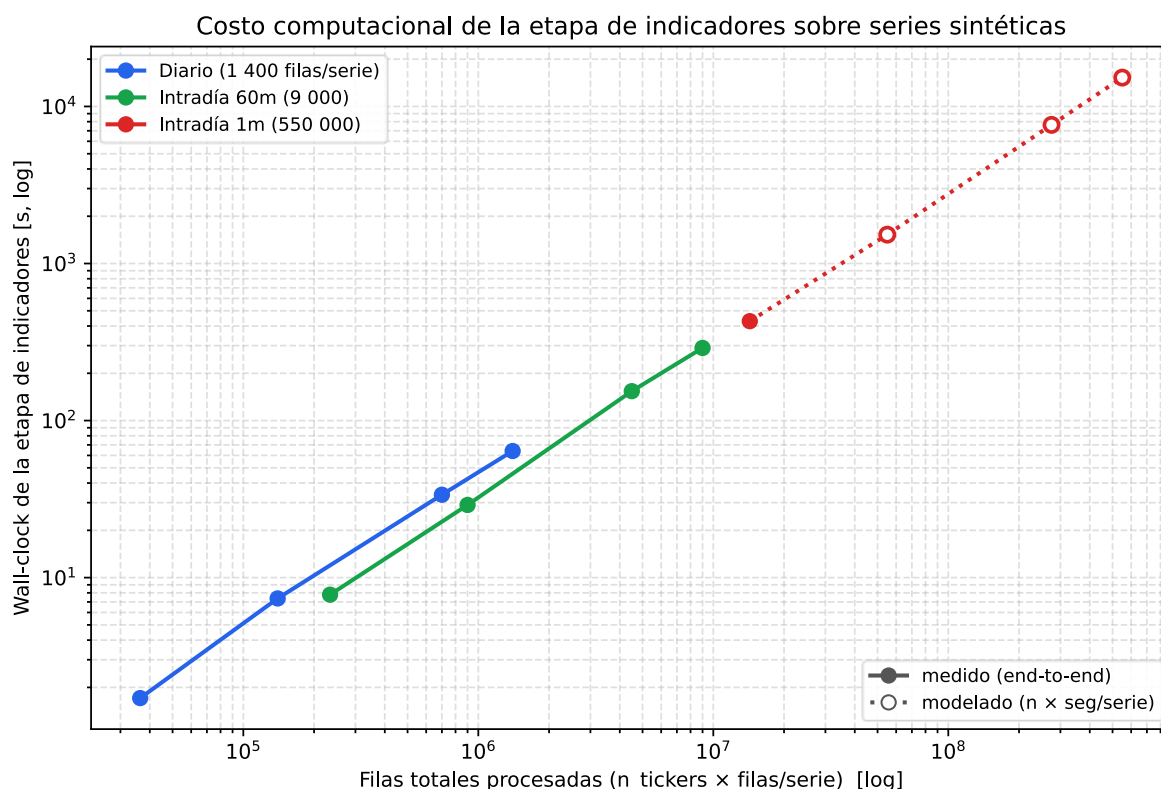
Un CSV es legible, se versiona en Git y admite el cálculo de un hash, lo que sostiene la trazabilidad declarada en RNF-01. Como tampoco requiere un servidor, encaja con la portabilidad que pide RNF-06.

El esquema y los tipos de cada campo no los controla el archivo plano por sí solo. Un motor de base de datos lo haría de entrada. Acá los valida el código, en la etapa de normalización, antes de que la serie siga su curso. Para ese volumen, montar un motor quedaría sobredimensionado. Aun sumando más años de datos diarios, las series siguen siendo livianas. El enfoque recién empezaría a quedar corto con granularidad intradiaria o con un universo de cientos de instrumentos, donde el número de filas por serie se multiplica y el costo de cómputo por corrida deja de rendir.

El diseño de archivos planos con procesamiento secuencial por serie se midió empíricamente para acotar su techo. El resultado descarta la memoria como barrera y ubica el límite en el tiempo de cómputo, dominado por el cálculo del HMA. El enfoque escala al universo diario actual y tolera cientos de instrumentos en granularidad diaria u horaria. El intradía de un minuto a gran escala es el punto donde deja de ser viable para una corrida diaria (Figura 2).

Figura 2

Costo computacional de la etapa de indicadores sobre series sintéticas



Nota. Costo computacional de la etapa de indicadores sobre series sintéticas. Tiempo y pico de memoria según granularidad y cantidad de instrumentos (escala log-log). Las celdas de 1 minuto a gran escala se proyectan por linealidad medida. Elaboración propia.

4.2.4 Diseño de la interfaz de usuario

La interfaz consume la salida del pipeline. Lee los archivos JSON publicados y los representa sin ejecutar ningún procesamiento en el navegador. Toda la lógica analítica se mantiene del lado del *backend*.

La información se organiza en tres secciones. El Panel de Señales da una vista de conjunto de todos los activos, con su estado resumido en una tabla. El Detalle del Activo grafica un instrumento con sus indicadores superpuestos. Eventos del Mercado, por último, reúne los hechos relevantes que pueden afectar a los distintos activos.

Los gráficos se construyen con Lightweight Charts v5.2.0, una biblioteca de visualización financiera de TradingView. La licencia MIT, permisiva y de código abierto, autoriza redistribuir la biblioteca. Por eso el repositorio incluye una copia local en vez de cargarla desde una red de distribución de contenidos (CDN). Con la versión fijada, el sitio se reconstruye sin depender de un servicio externo.

Fuera de los gráficos, la interfaz es HTML, CSS y JavaScript sin framework. La aplicación tiene tres secciones y un gráfico. Para algo así, un framework solo suma dependencias de npm y un paso de compilación que no aportan. El código fuente corre tal cual en el navegador, y levantar el sitio no requiere compilar nada.

El módulo que genera los JSON toma los CSV de indicadores y señales y los serializa por fecha sin reprocesarlos. La interfaz muestra lo que el pipeline ya emitió, sin recalcular nada y sin introducir un sesgo de anticipación en la etapa de visualización.

4.3 Recolección y preprocesamiento de datos

Antes de calcular indicadores, las series crudas pasan por una etapa de recolección y acondicionamiento. Así el análisis posterior trabaja sobre datos consistentes y corregidos.

4.3.1 Fuentes de datos

La fuente primaria de datos es la API de Portfolio Personal Inversiones (PPI), un broker autorizado por la Comisión Nacional de Valores (CNV). El acceso se hace con el cliente oficial ppi-client, mediante autenticación con dos claves, una pública y una privada. Las cotizaciones diarias las descarga el módulo `data_downloader.py`, que trabaja en dos modos. En la primera corrida ejecuta un *backfill* con la carga histórica completa. Después pasa a un modo incremental, donde pide solo los datos posteriores a la última fecha ya almacenada y no vuelve a descargar lo que el repositorio ya tiene.

La fuente entrega las series ya ajustadas por splits, y así se verificó. La inspección de los retornos diarios de las trece acciones sobre la ventana de estudio no arrojó discontinuidades compatibles con desdoblamientos no corregidos. Los únicos saltos superiores al 30% son alzas agrupadas en fechas de eventos electorales, como las ruedas posteriores a los comicios de noviembre de 2023 y octubre de 2025, movimientos de mercado reales y no factores nominales de split. Por eso `apply_split_adjustments()` queda disponible pero inactivo.

4.3.1.1 Criterios de calidad de los datos

La calidad de una fuente se evaluó con cuatro criterios. La integridad requiere que la serie no tenga ruedas faltantes ni huecos en los cierres. La consistencia pide un esquema uniforme, con un registro por rueda. La reproducibilidad es el criterio que más pesa acá: una misma descarga tiene que devolver siempre el mismo resultado, con ajustes por dividendos y splits trazables, de modo que una nueva descarga reproduzca la serie ya versionada. El cuarto criterio es la fuente misma, que tiene que ser un proveedor oficial, con una API estable y con mantenimiento.

4.3.1.2 Evolución y selección de la fuente de datos

PPI no fue la primera fuente. Se llegó a ella después de descartar dos alternativas por razones técnicas.

La primera prueba se hizo con `yfinance`. Como se apoya en el scraping de Yahoo Finance, sus cotizaciones en pesos del mercado local llegaban con desajustes horarios, ruedas omitidas en feriados argentinos y errores de conversión. El problema de fondo para este trabajo era otro. `yfinance` aplica sus propios ajustes por dividendos y splits sin explicar el criterio, así que la serie histórica cambiaba entre descargas y dejaba de coincidir con la ya versionada, lo que rompe la reproducibilidad que el trabajo necesita. Y al no ser una API oficial, cualquier cambio en el sitio de Yahoo rompe la librería y compromete el mantenimiento.

La segunda alternativa fue la API de InvertirOnline (IOL), descartada por restricciones de su arquitectura. La sesión era inestable, con una gestión de tokens que exigía renovaciones constantes y se caía seguido, lo que impedía un flujo de ingesta automatizado y desatendido. Su política de limitación de peticiones (rate limiting) bloqueaba las descargas secuenciales del backfill histórico e interrumpía el pipeline. Se sumaban lagunas en los cierres pasados y una estructura de respuesta menos predecible, que obligaba a recargar la lógica de limpieza en el código.

PPI resolvió esos puntos. Es un broker regulado por la CNV, con un cliente oficial mantenido y una API estable y documentada, lo que permitió construir un módulo de ingesta con control de errores y logging sólido. Entrega también datos confiables de las acciones locales y de los bonos que el cálculo del MEP necesita.

4.3.2 Universo de análisis

El universo son trece acciones del panel local, todas con ADR directo en Estados Unidos. El criterio de selección fue ese, que cotizaran también como ADR, porque una serie sintética en dólares MEP se puede validar contra un precio real, el del ADR, y así se comprueba que la dolarización es consistente. TXAR (Ternium Argentina) se evaluó y quedó afuera porque su instrumento en NYSE, TX, corresponde a Ternium S.A., la holding consolidada, no a un ADR de la filial argentina. A partir de cada acción se genera su equivalente sintético en MEP, lo que lleva el total a veintiséis series analizadas. Los bonos AL30 y AL30D no se analizan como activos, entran como insumo para calcular el tipo de cambio MEP. La paridad Local-ADR suele medirse con el CCL, que liquida en el exterior. La serie sintética usa el dólar MEP, que sale del bono AL30D, porque es la tasa a la que el minorista dolariza localmente y se calcula con datos del mismo mercado. Como MEP y CCL no siempre coinciden, la validación contra el ADR es aproximada.

El criterio pesa sobre la validación, no sobre el monitoreo. El pipeline puede ingestar y analizar un ticker del panel local que no cotice como ADR. Lo que esa serie no tendría es la verificación cruzada contra un precio externo en dólares. El universo se acotó a trece para que todas las series expuestas queden validadas, y no porque la arquitectura dependa de la existencia del ADR. Sumar instrumentos sin ADR es una cuestión de configuración del universo bajo el criterio de que esas series se expondrán sin la verificación cruzada mencionada en la sección 6.4.

4.3.3 Período de estudio y cantidad de ruedas

El período de estudio abarca desde el 14 de septiembre de 2020 hasta el 15 de mayo de 2026, alrededor de 1.383 ruedas bursátiles. El método SMA 10/50/100 necesita 100 ruedas de calentamiento (*warm-up*) para que las tres medias queden calculadas, así que sus señales accionables recién empiezan en febrero de 2021. La demora es propia del método SMA y entra en el diseño como tal.

4.3.4 Normalización y ajuste de datos

El módulo `data_normalizer.py` estandariza las series crudas a un esquema OHLCV uniforme, con un registro por rueda y columnas fijas para fecha, apertura, máximo, mínimo, cierre y volumen. Ya normalizadas, las series pasan por el ajuste por splits, esos desdoblamientos que incorporan un salto nominal de precio cuando una acción se divide. La corrección toma los eventos desde un archivo de configuración con los splits conocidos. El módulo escribe la serie normalizada en `data_normalized/csv/{ticker}_normalized.csv`.

4.4 Implementación de indicadores técnicos

Sobre las series normalizadas, el módulo `indicator_engine.py` calcula los indicadores que alimentan la generación de señales.

La HMA16 se implementa con la fórmula de Hull sobre un *span* de 16. Se arma como una media móvil ponderada (WMA) de 4 períodos aplicada a la combinación $2 \cdot \text{WMA}(8) - \text{WMA}(16)$.

El experimento evalúa por separado dos familias de medias, una EMA de 12 y 26 períodos y una SMA de 10, 50 y 100 períodos. La SMA de 100 necesita 100 ruedas para quedar calculada, así que ese sistema recién produce señales una vez pasado ese calentamiento.

La VMA20 no mide precio. Es el cociente entre el volumen de la rueda y su media de 20 períodos, una lectura de cuánta fuerza acompaña al movimiento del día.

4.5 Diseño del experimento de evaluación

El experimento somete cada serie a un backtesting que reproduce, rueda por rueda, las decisiones que habría tomado cada estrategia. Mide el comportamiento de los métodos bajo reglas idénticas, sin que ninguna información futura se relacione con una decisión presente.

4.5.1 Estrategias de trading

El experimento compara cuatro estrategias. Tres generan señales a partir de indicadores y una funciona como referencia.

La estrategia HMA16 sigue la pendiente de la media. Cuando la HMA empieza a subir, la tendencia se vuelve alcista y se abre una posición. Cuando empieza a bajar, se vuelve bajista y se cierra.

La estrategia EMA usa el cruce entre la EMA de 12 y la de 26 períodos. La tendencia es alcista mientras la de 12 está por encima de la de 26, y bajista cuando queda por debajo.

La estrategia SMA usa la alineación de las medias simples de 10, 50 y 100 períodos. La tendencia es alcista cuando quedan ordenadas con la de 10 por encima de la de 50 y la de 50 por encima de la de 100, y bajista cuando se ordenan al revés. Cualquier orden parcial o mezclado se toma como neutro y no abre posición.

Buy & Hold no usa indicadores. Compra al inicio del período y mantiene hasta el final. Es una base de inversión de mínimo esfuerzo, y sirve para contrastar a las otras tres.

4.5.2 Parámetros de backtesting

Todas las estrategias se ejecutan con las mismas reglas. Una señal se calcula con el cierre de una rueda y se ejecuta recién a la apertura de la rueda siguiente. Ninguna decisión utiliza información que no estuviera disponible al momento de su ejecución. Este desfase es el control de sesgo de anticipación del experimento.

Una posición se cierra cuando aparece una señal contraria, o cuando se alcanza un tope de tenencia de 30 ruedas que fuerza la salida si esa señal no llega antes. El retorno de cada operación es la variación entre el precio de entrada y el de salida, ambos tomados a la apertura.

El backtest descuenta un costo fijo de 0,5% en cada punta, compra y venta, de modo que cada operación completa paga 1%. Es una referencia de las comisiones y derechos de mercado del minorista argentino. Sin ese descuento los retornos quedarían inflados.

El backtester no modela el plazo de permanencia (parking) que rigió sobre la operatoria del MEP durante buena parte del período de estudio, ni los costos de oportunidad asociados a ese plazo. La serie sintética en dólares MEP es un constructo analítico para comparar el desempeño relativo de las estrategias en una unidad libre del efecto inflacionario, no una réplica de la operatoria de conversión con sus tiempos de liquidación. El Capítulo 7 detalla esta simplificación junto con la no inclusión del slippage, dos factores a considerar al extrapolar estos resultados a un entorno de negociación real.

4.5.3 Métricas de evaluación

Sobre la curva de capital de cada estrategia se calculan el retorno acumulado, el retorno anualizado, que es la métrica principal de la comparación, la máxima caída (drawdown), la peor pérdida desde un pico hasta el valle siguiente, y el tiempo invertido, la proporción del período con posición abierta. A nivel de operación se miden el número de operaciones, la proporción que termina en ganancia, y el retorno promedio y mediano, que sirven para describir cómo opera cada estrategia y para evaluar los validadores.

Las métricas se calculan sobre el período completo y, por separado, sobre los tramos dentro de muestra (*in-sample*) y fuera de muestra (*out-of-sample*). Se reportan también para Buy & Hold, que sirve de piso de comparación.

4.5.4 Metodología de validación

La muestra se divide en orden cronológico. El primer 70 % de las ruedas forma el tramo *in-sample* y el 30 % restante el *out-of-sample*. El corte respeta el orden temporal, así que el período de validación queda siempre después del de referencia y nunca se evalúa el pasado con información futura.

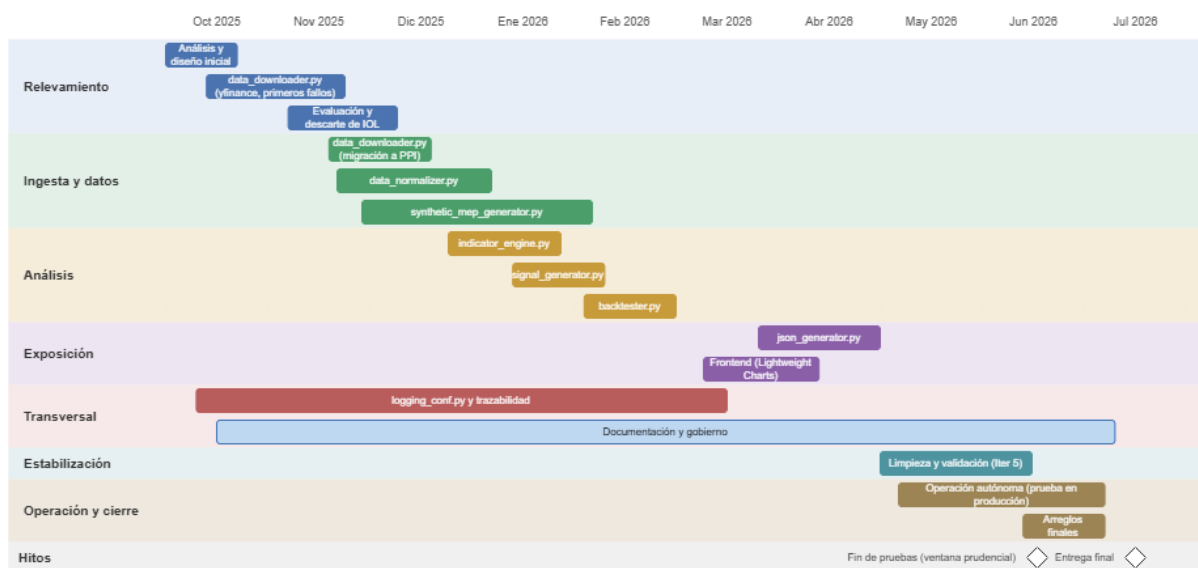
Las métricas se reportan asimismo sobre el período completo para no descartar datos en la lectura descriptiva. Ese corte total incluye el tramo *in-sample*, así que no es un resultado de validación y se presenta solo como resumen.

Los parámetros de cada método se fijaron de antemano y no se optimizaron sobre el tramo *in-sample*. El tramo fuera de muestra no opera como resguardo contra el *overfitting*, porque no hubo parámetros que ajustar. Lo que sí muestra es si cada método rinde parecido en un tramo que no entró en su cálculo.

Se usa un solo corte cronológico. El *walk-forward* y la validación cruzada purgada son más rigurosos, pero esta versión no los implementó y quedan para más adelante.

4.6 Planificación y cronograma del desarrollo

La Figura 3 muestra el cronograma de desarrollo de la plataforma, reconstruido del historial de commits del repositorio. Varias etapas se solaparon. El trabajo arrancó por la ingesta de datos, con el cambio de fuente de *yfinance* a *PPI*, y siguió con los módulos de análisis y la capa de exposición. El logging y la trazabilidad corrieron en paralelo durante todo el proyecto. Hoy el sistema opera solo, y las pruebas se cierran unas semanas antes de la entrega.

Figura 3*Cronograma de desarrollo de la plataforma**Nota.* Elaboración propia.

Capítulo 5. Desarrollo e Implementación de la Plataforma

5.1 Tecnologías utilizadas

El desarrollo se apoya en un conjunto acotado de herramientas. La Tabla 1 lista el *stack* real del repositorio, con las versiones fijadas en el código. Claude Code se usó como asistente de generación de código. El diseño, la revisión y la validación de cada componente, junto con la ejecución de las pruebas, estuvieron a cargo del autor. La asistencia operó sobre la implementación, no sobre las decisiones de arquitectura ni el análisis de resultados.

Tabla 1

Stack tecnológico de la plataforma

Capa	Componente	Versión	Función
Pipeline	Python	3.11	Runtime de los siete módulos en GitHub Actions
Pipeline	pandas	2.3.3	Series temporales, CSV, agregados
Pipeline	numpy	2.3.4	Cálculo numérico de indicadores (HMA, MMs)
Pipeline	ppi_client	1.2.4	Cliente de la API de PPI (datos BYMA)
Pipeline	signalrcorePPI	0.9.2	Transporte SignalR que requiere ppi_client
Pipeline	python-dotenv	1.2.1	Carga de credenciales PPI desde .env
Frontend	JavaScript	ES2020+ vanilla	Lógica del sitio (app.js), sin framework ni build
Frontend	Lightweight Charts	5.2.0	Gráficos financieros, vendorizada en docs/vendor/
Frontend	HTML5 / CSS3	propio	index.html, style.css (512 líneas)
Infra	GitHub Actions		Cron 0 21 * * 1-5 UTC (18 hs ART) + workflow_dispatch
Infra	GitHub Pages		Hosting estático desde main/docs
Datos	PPI (API)		Fuente de producción: acciones _BA y bonos AL30/AL30D para el MEP
Validación	yfinance	0.2.66	Descarga de ADR en USD, solo validación
Validación	requests	2.32.5	Fallback Stooq en la validación ADR
Validación	matplotlib	3.10.7	Figuras de la sección 6.4
Asistencia	Claude Code		Asistente de programación integrado al repositorio

Fuente: elaboración propia a partir de requirements.txt y daily-pipeline.yml del repositorio.

5.2 Módulos implementados

El pipeline se reparte en las cuatro capas operativas de la arquitectura. La ingesta corre un único módulo, `data_downloader.py`. El procesamiento agrupa `data_normalizer.py` y `synthetic_mep_generator.py`. El análisis concentra el grueso del cómputo, con `indicator_engine.py`, `signal_generator.py` y `backtester.py`. La capa web cierra con `json_generator.py`. Las subsecciones siguen ese orden de ejecución y describen cada módulo por su implementación, con sus entradas, sus salidas y los parámetros tal como están en el código. El fundamento de diseño de cada decisión está desarrollado en las secciones de arquitectura y no se replica acá.

5.2.1 Ingesta de datos

El módulo `data_downloader.py` resuelve la ingesta. En la primera corrida hace un backfill desde el 04/09/2020 (`START_DATE_BACKFILL`). Después lee la última fecha registrada en `data_raw/csv/{ticker}.csv` y le pide a la API de PPI solo el tramo que falta hasta el día actual. El cliente se autentica en `get_client()` con las claves `PPI_PUBLIC_KEY` y `PPI_PRIVATE_KEY`, que se cargan desde un archivo `.env` en local y desde los secrets del repositorio en la corrida automática. La descarga usa el timeframe "A-48HS" de PPI, y `process_data()` la consolida a un registro OHLCV por rueda agrupando por día. La validación por fecha evita duplicar ruedas ya guardadas. El universo a descargar sale de `study_config`, los bonos AL30 y AL30D más las trece acciones, y cada serie cruda se escribe en `data_raw/csv/{ticker}.csv`.

5.2.2 Normalización y ajuste

El módulo `data_normalizer.py` estandariza las series crudas. Recorre todos los CSV de `data_raw/csv/` y lleva cada uno al esquema `NORMAL_COLUMNS` (date, open, high, low, close, volume, ticker), con un registro por rueda. La función `apply_split_adjustments()` multiplica los precios anteriores a un desdoblamiento por el factor correspondiente, tomado de `src/config/splits.json`. Ese archivo no está en el repositorio, así que en la práctica no se ajusta ningún ticker. La cotización de PPI ya llega ajustada, y el mecanismo queda disponible pero inactivo. Cada serie se escribe en `data_normalized/csv/{ticker}_normalized.csv`, en texto plano y versionada en Git.

5.2.3 Generación de series sintéticas MEP

El módulo `synthetic_mep_generator.py` expresa cada acción en dólares MEP. La función `calculate_base_mep()` lee los cierres de `AL30_BA` y `AL30D_BA` ya normalizados, los une por fecha con un *inner join* y calcula el cociente diario entre ambos. Con esa referencia, el script recorre las acciones locales, divide sus valores de apertura, máximo, mínimo y cierre por el tipo de cambio de esa misma rueda, y guarda el resultado en `data_normalized/csv/{ticker}_MEP_normalized.csv`. Como el cruce es por fecha, la serie sintética solo existe para las ruedas en que cotizaron a la vez el par de bonos y la acción.

5.2.4 Cálculo de indicadores

El módulo `indicator_engine.py` calcula los indicadores y no genera señales. Esa división deja el cálculo numérico aislado del que lo interpreta. Recorre los CSV de `data_normalized/csv/` (las series locales y sus versiones MEP) y le agrega a cada uno sus indicadores. La HMA de span 16 se arma con la función auxiliar `wma()`, según la construcción de Hull definida en 3.1.4. Las medias simples de 10, 50 y 100 ruedas usan `rolling(window).mean()`. Las exponenciales de 12 y 26 usan `ewm(span, adjust=False).mean()`, que arranca en NaN durante el período de calentamiento. La relación de volumen `VMA20` es el volumen del día sobre su media móvil de 20 ruedas (`calculate_vma20_ratio`). Cada salida se escribe en `data_indicators/csv/{ticker}_indicators.csv`, en texto plano y versionada en Git.

5.2.5 Generación de señales

El módulo `signal_generator.py` traduce los indicadores numéricos en condiciones lógicas de mercado. El script procesa los archivos del paso previo y genera tres columnas de tendencia independientes. La función `calculate_hma_trend()` toma la dirección del precio del signo de la pendiente de la HMA16. `calculate_ema_crossover_trend_12_26()` define el estado por la posición relativa de las dos medias exponenciales. `calculate_sma_crossover_trend_10_50_100()` exige la alineación estricta de las tres medias simples, marca alcista o bajista solo cuando quedan ordenadas y neutro ante cualquier mezcla. Las tres lógicas pasan por una máquina de estados interna (`_trend_state_machine()`) que mantiene la posición vigente hasta que aparece una confirmación de cambio, así el sistema no vuelve a decidir lo mismo en cada rueda. Las salidas se insertan en sus respectivas columnas de control (`T_hma16`, `T_ema12_26` y `T_sma10_50_100`). En paralelo, el módulo categoriza la relación de volumen en cinco niveles de actividad operativa y detecta patrones de velas tradicionales como envolventes, dojis o marubozus. El volumen y las configuraciones de velas

operan exclusivamente como validadores descriptivos. Su uso como confirmación de la tendencia se evalúa en una etapa posterior dentro del backtester. La salida final se almacena en `data_signals/csv/{ticker}_signals.csv`.

5.2.6 Backtesting

El módulo `backtester.py` simula la ejecución operativa y arma las métricas de rendimiento. Toma las señales y los indicadores de las etapas previas, sin los bonos que sirven de referencia cambiaria. `run_backtest()` recorre las columnas de tendencia y abre, mantiene o cierra posiciones según el estado de cada estrategia, contrastando los resultados frente a la estrategia pasiva ejecutada en `run_buy_and_hold()`.

Los costos se descuentan con la constante `COST_PER_TRADE`, que aplica el cargo de la operación completa. El parámetro `max_hold` acota la permanencia máxima en mercado y fuerza la salida cuando se cumple el tope.

El sesgo de anticipación se controla en tres puntos del flujo. Una señal detectada al cierre de una rueda se ejecuta al precio de apertura de la siguiente. Las condiciones de volumen y de forma de velas se evalúan en la rueda donde nace la señal, no en la de ejecución, así ningún dato de la rueda de ejecución entra en la decisión. La mediana de volumen se calcula sobre una ventana expansiva corrida con `shift()`, así el dato del día en análisis no entra en su propia métrica de control.

La segmentación en los tramos dentro y fuera de muestra usa filtros cronológicos sobre la fecha de entrada de cada operación. `compute_validator_significance()` mide el efecto de los filtros de confirmación con pruebas de hipótesis de dos proporciones y de diferencia de medias, y escribe las matrices resultantes en texto plano en el directorio público.

5.2.7 Exposición y visualización web

El módulo `json_generator.py` une la capa de datos con la de exposición. Recoge los archivos de salida del backtesting, los indicadores y las señales, y los serializa en las estructuras que consume la interfaz web. Transforma formatos sin alterar los valores ya calculados. Las funciones de generación cruzan la información con uniones por izquierda (`left join`) por fecha y descartan los registros del período de calentamiento. Otras auxiliares convierten los valores no disponibles (NaN) a tipos válidos para el estándar web y ajustan la precisión decimal de los coeficientes de correlación.

El frontend es un sitio estático y desacoplado. La lógica corre en app.js con JavaScript estándar, sin entornos de desarrollo ni pasos de compilación. La aplicación representa las estructuras de datos y no hace ningún cálculo analítico del lado del cliente. Los gráficos se delegan en la biblioteca Lightweight Charts, integrada localmente en el repositorio para no depender de una red externa. La aplicación llama a la API del componente para dibujar los gráficos de velas, las líneas de tendencia y los marcadores de posición, y los reparte en los tres paneles de la interfaz.

5.3 Desafíos técnicos y soluciones

La implementación dejó varios problemas técnicos con diagnóstico y solución documentados. El más estructural apareció al graficar las señales. Hacía falta mostrar los marcadores de operación en una banda inferior, separada de la escala de precio, y la versión 4 de Lightweight Charts no soportaba paneles apilados de forma nativa. Se resolvió migrando a la versión 5, que incorporó los paneles como primitiva. Los marcadores pasaron a un panel propio (índice 1) sin interferir con el precio. La decisión queda registrada en el Anexo III.

La representación gráfica de la HMA introdujo una complejidad más leve. La línea debía ser una sola y cambiar de color según el estado de tendencia, pero aparecía como dos líneas paralelas. La primera versión usaba tres series fijas (verde, roja, gris) con whitespace points para apagar los días del estado inactivo, y falló por una razón concreta. Lightweight Charts une los puntos consecutivos de una LineSeries aunque sus fechas no sean contiguas, así que los puntos lejanos de un mismo color se enlazaban con rectas. La solución fue dibujar un segmento por tramo de color, con una LineSeries por segmento y un punto de empalme en cada transición.

El mismo gráfico tenía un segundo inconveniente de escala. Al activar las medias móviles, su rango se sumaba al del precio y lo comprimía, sobre todo por la SMA100 al inicio del período. Se corrigió pasando autoscaleInfoProvider: () => null a cada serie de indicador, para que no contribuyan al cálculo del rango. El costo aceptado es que una media fuera del rango visible del precio se recorta, algo que solo ocurre durante el calentamiento.

En la capa de datos el problema fue de consistencia entre lo planificado y lo real. Al marcar los indicadores en el detalle del activo no se dibujaba nada. El JSON por ticker traía solo el OHLC base y el frontend leía campos que no existían. El esquema asumido no coincidía con lo que el generador producía, y la diferencia recién apareció al inspeccionar el JSON publicado. La solución fue que gen_ticker() uniera por fecha los CSV de indicadores y señales antes de serializar, emitiendo todos los campos.

El escenario final afectó a la sección de eventos. Al acceder al detalle del activo con anterioridad a dicha sección, esta se visualizaba vacía. La explicación inicial la atribuyó a una condición de carrera, pero la causa real era otra: un guard de control de flujo que salteaba el render cuando los datos ya estaban en caché. Se quitó ese guard, se hizo idempotente el registro de filtros y se cortaron los fetches concurrentes. Lo que destrabó el problema fue corregir el diagnóstico, no el síntoma.

5.4 Reproducibilidad y trazabilidad

La trazabilidad se apoya en dos mecanismos que operan en todas las etapas. Las salidas son CSV de texto plano, así que cualquier cambio en una serie aparece en el diff de Git y queda en el historial del repositorio. Sobre los datos crudos, que son la entrada de todo el pipeline, se calcula un hash SHA256 que se registra en el mensaje del tag de la versión. Ese hash sella la integridad de los insumos y permite confirmar que un tercero parte exactamente de los mismos datos de origen. Un motor de base de datos, en cambio, produce un binario cuyo hash varía por detalles internos ajenos al dato, lo que volvería frágil esta verificación. La organización en archivos planos, justificada en 4.2.2 por el volumen y el patrón de acceso, es la misma que abarata la trazabilidad.

La reproducibilidad del cómputo descansa en versiones fijas. El requirements.txt pinnea pandas 2.3.3 y numpy 2.3.4, y la versión de referencia del intérprete es Python 3.11, la que fija la integración continua que genera los artefactos publicados. El universo son veintiséis series, las trece acciones locales y sus trece equivalentes MEP. Las acciones y los dos bonos que alimentan el MEP son datos crudos versionados en data_raw, quince archivos CSV en total. Las trece series MEP no se guardan como crudos, las deriva el pipeline a partir de esos insumos. Las salidas son texto plano y el cómputo es determinista sobre las versiones fijadas. Regenerar desde los mismos crudos reproduce los agregados del backtest y los JSON que consume el sitio, ambos versionados, y un desvío en cualquiera de ellos aparecería en el diff. La afirmación tiene alcance acotado. Vale para el intérprete de referencia y las librerías pinneadas, y se comprueba sobre los valores reportados, no como identidad de bytes entre entornos de ejecución distintos. Las capas intermedias (normalización, indicadores, señales) son derivados regenerables y no se versionan, porque su corrección queda cubierta por la de las salidas que dependen de ellas. La verificación no exige rehacer la descarga. Los crudos quedan congelados en el repositorio y su SHA256 está en el mensaje del tag, así que un tercero confirma primero que los insumos no se modificaron y recién después regenera.

La trazabilidad alcanza también a la capa de publicación. El generador de JSON no recalcula nada (5.2.7), serializa lo que el pipeline emitió, de modo que los valores que muestra el sitio son exactamente los que produjo el pipeline. La validación contra los ADR reales también es reproducible. Las series de yfinance se descargaron una sola vez y permanecen congeladas como CSV crudos en el repositorio. El script de validación consulta primero el caché local y va a la fuente externa solo cuando el archivo no existe. Rehacer la comparación desde el estado versionado no requiere conexión. La ventana de la serie ADR cierra el 14 de mayo de 2026, una rueda antes que las series locales, porque el parámetro de fin del descargador es exclusivo. El cotejo se resuelve por intersección de ruedas comunes. El desfase recorta un día del rango evaluado. No introduce anticipación, dado que la serie de referencia termina antes, nunca después, que la serie evaluada. Toda la cadena, desde la descarga cruda hasta el sitio publicado, queda versionada en Git. El estado que produjo estos resultados quedó marcado con el tag v1.1-tf, sobre el commit 7cb8d13, con la ventana de análisis cerrada al 15 de mayo de 2026. El código y los datos residen en el repositorio público <https://github.com/sebams81/analisis-tecnico-arg>, y la interfaz se publica en <https://sebams81.github.io/analisis-tecnico-arg/>. El estado congelado se archiva además en Zenodo bajo el identificador persistente <https://doi.org/10.5281/zenodo.21288495>, de modo que la verificación no depende de la permanencia del repositorio de origen. La verificación del hash y la regeneración operan sobre ese tag, no sobre la rama principal. La automatización diaria sigue agregando ruedas a los datos crudos, de modo que el hash SHA256 anclado corresponde al estado congelado del tag y solo coincide al posicionarse en él. Un tercero hace *checkout* de ese punto y reproduce la corrida completa.

5.5 Marco de gobernanza

El sistema se sostiene en procedimientos documentados y no necesita la intervención continua del autor. El marco de gobernanza que sigue cubre cinco dominios, los datos, el código, la operación del sistema desplegado, la curaduría de eventos fundamentales y el riesgo. Los procedimientos operativos detallados están en el Anexo III. Aquí se fijan las políticas.

5.5.1 Gobierno de datos

El gobierno de los datos parte de una fuente única de verdad. Toda la información de mercado del pipeline de producción proviene de la API de PPI. La única excepción es la descarga puntual de yfinance para la validación contra los ADR, que queda congelada en el repositorio y por fuera del pipeline (5.4).

La política de calidad es el conjunto de controles anti-look-ahead bias que se aplican en cada etapa del pipeline, descritos en 4.5.4 y 5.2.6. El ciclo de vida del dato va de la serie cruda a la publicación, y cada etapa deja un artefacto verificable, de modo que cualquier valor del sitio es trazable hasta una rueda bursátil concreta (5.4). PPI se cita como proveedora de los datos y su uso se restringe a fines académicos.

5.5.2 Gobierno del código

El código vive en un repositorio público versionado con Git, lo que deja todo el historial de decisiones disponible para auditoría. Los cambios de código siguen un flujo formal. Se trabajan en ramas de funcionalidad (*feature branches*) que se integran a la rama principal mediante pull requests. Las actualizaciones diarias de datos, en cambio, las commitea de forma automática el bot de integración continua, por fuera de ese flujo. Los mensajes de commit se escriben en español y explican el motivo de cada cambio. El marco de licenciamiento separa tres planos. El código se publica bajo licencia MIT, la documentación y las figuras de autoría propia bajo CC BY 4.0, y los datos de mercado permanecen bajo los términos de PPI, citados y de uso académico.

5.5.3 Gobierno operativo

El sistema opera de forma autónoma. La integración continua de GitHub Actions corre el pipeline una vez por día hábil, a las 18 de Argentina, después del cierre del mercado local, y deja además un disparador manual disponible. El rol de operador no está atado a una persona. Los procedimientos están documentados en el Anexo III, de modo que cualquiera con acceso al repositorio puede sostener el sistema. Ante una corrida fallida, la recuperación consiste en volver a disparar el workflow de forma manual. Como la ingesta es incremental e idempotente (lee la última fecha registrada y pide solo el diferencial), repetir una corrida no duplica ni corrompe la serie. Las credenciales de la API de PPI se guardan en los secrets del repositorio y nunca en el código.

5.5.4 Gobierno de la curación

El panel de eventos fundamentales no es una señal analizada ni testeada. Es un *feed* curado con un proceso documentado en el Anexo III. Ese proceso fija las fuentes admitidas (instituciones reconocidas como la CNV y BYMA, medios financieros y los canales de relación con inversores de cada empresa), qué califica como evento relevante y el tipo con el que se clasifica cada evento (macroeconómico, electoral, resultados, deuda, regulatorio o dividendo, entre otros). Cada evento incluye la URL a su fuente original como atribución obligatoria. Con eso la curaduría deja de ser una selección personal y pasa a ser un procedimiento auditable.

El panel clasifica los eventos por tipo y no por su impacto esperado sobre el precio. Una versión inicial asignaba a cada evento una etiqueta de impacto, pero esa etiqueta es incompatible con el enfoque del trabajo. Atribuirle a un evento una dirección esperada equivale a emitir un pronóstico fundamental. El análisis técnico, tal como lo plantea Murphy (2000), se apoya en que el precio ya incorpora la información disponible y lee su trayectoria, no anticipa el efecto de una causa externa. Rotular el impacto agrega además un juicio retrospectivo, porque la dirección "correcta" solo se conoce después de observar la reacción del mercado. Es el tipo de sesgo que López de Prado (2018) advierte evitar. Por eso el panel registra el tipo de evento, un dato objetivo y verificable, y no su impacto, que sería interpretación.

Ese mismo criterio marca el límite de lo que el panel afirma. Un evento entra por ser pertinente y verificable, no por su efecto esperado. El sistema no dice hacia dónde va el precio. Avisa que viene una fecha sensible, o que algo pasó fuera de horario, y muestra el hecho con su fuente y su tipo. Así el usuario llega con tiempo para decidir qué hacer antes de la rueda siguiente.

El panel cumple un rol secundario, de demostración, sin peso analítico. Existe porque el análisis técnico por sí solo no captura los movimientos de origen fundamental, algo que se discute en 6.3 y se retoma en el Capítulo 7. Como no participa del cómputo y funciona solo como señalización para el usuario, su cobertura temporal no está atada a la ventana congelada del backtest y puede extenderse más allá del corte de análisis sin afectar la reproducibilidad de los resultados. La automatización de la curaduría desde fuentes oficiales queda planteada como línea de trabajo futuro (7.5), no como componente validado.

5.5.5 Gobierno del riesgo

El sistema depende de tres servicios externos, cada uno con su modo de falla. La API de PPI es la fuente de datos. Si deja de responder o cambia su esquema, el pipeline no puede actualizarse. GitHub Actions orquesta la ejecución diaria y GitHub Pages aloja la interfaz pública. Una interrupción en estos servicios degrada la actualización o la visualización temporal del sistema, pero no compromete la integridad de los datos históricos preexistentes.

Las mitigaciones frente a esos riesgos son acotadas. La ingesta es idempotente, así que una corrida fallida se reintenta sin corromper la serie (5.5.3). El pipeline procesa los tickers de forma independiente. Si una etapa falla para un ticker, el error se registra y la corrida continúa con los demás. El generador de JSON no recalcula nada (5.2.7), de modo que un error en la capa de publicación no puede alterar los valores que produjo el pipeline. La recuperación ante una corrida fallida está documentada en el Anexo III y se reduce a volver a disparar el workflow.

El trabajo reconoce de forma explícita los riesgos que no mitiga. Un cambio mayor en la API de PPI o la discontinuación de un servicio externo requeriría intervención manual, con la remediación documentada en el Anexo III.

5.6 Resiliencia operativa

La arquitectura estática hace al sistema resiliente por construcción. No hay backend, base de datos ni autenticación, así que ningún componente de servidor puede caerse. Si el pipeline o la API de PPI fallan, el sitio sigue sirviendo los últimos datos publicados. La falla degrada a datos de un día viejo, no a una caída. La recuperación de una corrida fallida es el disparo manual e idempotente descrito en 5.5.3.

La contracara es la propagación de los cambios. Tras la publicación, la CDN de GitHub Pages y la caché del navegador pueden demorar la actualización, y la interfaz muestra por un rato el estado anterior. Es una limitación conocida del hosting estático. El usuario la resuelve forzando la actualización del navegador, y publicar los archivos con un hash en el nombre, para que la caché no interfiera, queda como mejora futura. El sistema se actualiza una vez por día, no en tiempo real, lo que alcanza para una herramienta de monitoreo del inversor minorista que no pretende ser una terminal de operación.

Capítulo 6. Resultados y Análisis

6.1 Resultados del backtesting

El backtest corre sobre las 26 series del estudio, las 13 acciones locales y sus 13 versiones sintéticas en dólares MEP. El período va del 14/09/2020 al 15/05/2026, alrededor de 1.383 ruedas. Cada estrategia opera con un costo de transacción del 1% por operación completa y se mide contra comprar y mantener (Buy & Hold), que funciona como referencia. Los tres cortes temporales (período completo, muestra de entrenamiento o In-Sample, muestra de validación o Out-of-Sample) se informan en mediana entre los tickers de cada universo. La mecánica completa del experimento está en 4.5. El universo local y el MEP se reportan por separado. En pesos los retornos están inflados por la devaluación del período, así que la lectura económica del desempeño es la del universo en dólares. La métrica que gobierna la comparación es el retorno anualizado.

6.1.1 HMA16

HMA16 es la más reactiva de las tres estrategias y la que más opera, entre 75 y 88 operaciones por ticker según el universo. Su retorno anualizado en mediana figura en la Tabla 2.

Tabla 2

Retorno anualizado de HMA16 por universo y corte

Universo	Estrategia	Total	IS	OOS
Local (peso)	HMA16	23,2%	38,5%	-6,0%
Local (peso)	Buy & Hold	98,3%	141,2%	31,8%
MEP (dólar)	HMA16	-4,1%	0,1%	-16,6%
MEP (dólar)	Buy & Hold	28,6%	33,5%	23,7%

Nota. Mediana del retorno anualizado entre los 13 tickers del universo. Buy & Hold se incluye como referencia.

En pesos rinde positivo en el período completo y en entrenamiento, pero se da vuelta a negativo en la validación. En dólares rinde negativo o nulo en los tres cortes, con el peor número en el tramo de validación. En todos los casos queda muy por debajo del *benchmark*, que en mediana rinde cerca de 98% anual en pesos y 28% en dólares en el período completo. El contraste detallado contra Buy & Hold va en 6.2, igual que el efecto del costo, al que esta estrategia está más expuesta por su frecuencia de operación.

6.1.2 Medias móviles clásicas

Agrupar el cruce EMA 12/26 y la alineación SMA 10/50/100. Las dos operan bastante menos que HMA16, alrededor de 39 y 25 operaciones por ticker respectivamente. El retorno anualizado en mediana figura en la Tabla 3.

Tabla 3

Retorno anualizado de las medias clásicas por universo y corte

Universo	Estrategia	Total	IS	OOS
Local (peso)	EMA 12/26	40,3%	53,6%	9,0%
Local (peso)	SMA 10/50/100	56,8%	75,1%	13,7%
Local (peso)	Buy & Hold	98,3%	141,2%	31,8%
MEP (dólar)	EMA 12/26	-0,3%	-4,4%	10,2%
MEP (dólar)	SMA 10/50/100	-6,3%	-7,7%	5,0%
MEP (dólar)	Buy & Hold	28,6%	33,5%	23,7%

Nota. Mediana del retorno anualizado entre los 13 tickers del universo. Buy & Hold se incluye como referencia.

En pesos las dos rinden más que HMA16 en los tres cortes, y SMA es la de mayor retorno nominal. En dólares ambas quedan en torno a cero o levemente negativas en el período completo y en entrenamiento, y recién en la validación muestran retorno positivo, con EMA cerca del 10% y SMA del 5% en mediana. Aun con eso, ninguna alcanza al benchmark en ningún corte ni universo. El único tramo donde alguna se acerca es la validación en dólares, una ventana corta y por eso sensible al punto final. Las últimas ruedas del período, de baja, movieron las medianas de *Buy & Hold*. Esto se retoma en 6.2 y 6.3.

6.2 Análisis comparativo de desempeño

La sección anterior mostró el retorno de cada estrategia por separado. Falta compararlas contra la estrategia pasiva en tres dimensiones, la tasa de éxito por activo, el riesgo asumido y el impacto del costo de operar. La pregunta es si alguna estrategia, tomada como sistema completo, supera a Buy & Hold en algún corte o universo.

No lo hace. En el período completo ninguna de las tres estrategias supera al benchmark en ninguno de los 13 tickers, en los dos universos, y en entrenamiento el cuadro es casi idéntico. Las pocas victorias aparecen solo en la validación, y siempre son minoría, a lo sumo 4 tickers de 13 (Tabla 4).

El argumento habitual a favor de estas estrategias es que, al salir en las bajas, reducen el riesgo. Los datos no lo respaldan. En pesos, las tres estrategias tienen una caída máxima peor que la de Buy & Hold, que con este universo es la estrategia de menor drawdown. En dólares el cuadro es apenas mejor para las estrategias. Solo SMA queda marginalmente por debajo del benchmark, y HMA y EMA siguen peor. Ni la estrategia más conservadora consigue una protección clara, en el mejor caso empata. El tiempo invertido ronda entre el 44% y el 65% del período según método y universo, contra el 100% de Buy & Hold, y esa exposición parcial no se traduce en menos drawdown. La lectura ajustada por riesgo se retoma en 6.3 (Tabla 5).

El costo agranda una diferencia que ya existe sin él. Con el retorno medido a tres niveles de costo (0%, 0,5% y 1% por operación completa), la penalización crece con la frecuencia de operación. HMA16, con sus 75 a 88 operaciones por ticker, pierde alrededor de 20 puntos de retorno anual al pasar de 0% a 1%, mientras que SMA, con 25 operaciones, pierde menos de la mitad de eso. Lo que importa es que, aun sin costo, la mediana de las tres estrategias queda por debajo de Buy & Hold en todos los cortes, incluida la validación. El costo profundiza una desventaja que ya está en el retorno bruto (Tabla 6).

El resultado es el mismo se mire por donde se mire. Las estrategias no superan a Buy & Hold ni en retorno ni en riesgo, y las comisiones solo empeoran el cuadro. Por qué pasa esto, y si sirven al menos como salida ante un desplome, se discute en 6.3.

Tabla 4

Tickers que superan a Buy & Hold por método, universo y corte

Universo	Método	Total	IS	OOS
Local (peso)	HMA16	0	0	2
Local (peso)	EMA 12/26	0	0	1
Local (peso)	SMA 10/50/100	0	0	4
MEP (dólar)	HMA16	0	1	0
MEP (dólar)	EMA 12/26	0	0	4
MEP (dólar)	SMA 10/50/100	0	0	0

Nota. Conteo sobre los 13 tickers de cada universo. Un ticker cuenta si su retorno anualizado supera al de Buy & Hold en esa misma celda.

Tabla 5*Drawdown máximo (mediana) por método, universo y corte*

Universo	Método	Total	IS	OOS
Local (peso)	HMA16	-48,4	-46,5	-43,8
Local (peso)	EMA 12/26	-49,7	-43,9	-42,6
Local (peso)	SMA 10/50/100	-41,6	-38,2	-35,3
Local (peso)	Buy & Hold	-38,1	-37,7	-38,0
MEP (dólar)	HMA16	-66,8	-53,5	-59,0
MEP (dólar)	EMA 12/26	-61,2	-61,0	-35,4
MEP (dólar)	SMA 10/50/100	-52,3	-52,4	-43,1
MEP (dólar)	Buy & Hold	-53,1	-50,6	-52,7

Nota. Mediana del drawdown máximo entre los 13 tickers de cada universo, en porcentaje. El valor negativo indica la caída pico a valle.

Tabla 6*Retorno anualizado por método, universo y corte según el costo*

Universo	Método	Corte	0%	0.5%	1%
Local (peso)	HMA16	Total	44,4	33,5	23,2
Local (peso)	HMA16	IS	62,7	50,1	38,5
Local (peso)	HMA16	OOS	9,6	1,6	-6,0
Local (peso)	EMA 12/26	Total	50,9	45,5	40,3
Local (peso)	EMA 12/26	IS	65,4	59,3	53,6
Local (peso)	EMA 12/26	OOS	17,1	13,0	9,0
Local (peso)	SMA 10/50/100	Total	64,4	60,6	56,8
Local (peso)	SMA 10/50/100	IS	84,7	79,8	75,1
Local (peso)	SMA 10/50/100	OOS	17,6	15,6	13,7
MEP (dólar)	HMA16	Total	9,0	2,5	-4,1
MEP (dólar)	HMA16	IS	14,6	7,5	0,1
MEP (dólar)	HMA16	OOS	-3,8	-10,4	-16,6
MEP (dólar)	EMA 12/26	Total	5,9	2,8	-0,3
MEP (dólar)	EMA 12/26	IS	2,4	-1,0	-4,4
MEP (dólar)	EMA 12/26	OOS	16,1	13,1	10,2
MEP (dólar)	SMA 10/50/100	Total	-2,7	-4,5	-6,3
MEP (dólar)	SMA 10/50/100	IS	-3,8	-5,8	-7,7
MEP (dólar)	SMA 10/50/100	OOS	8,6	6,9	5,0

Nota. Mediana del retorno anualizado entre los 13 tickers. Los valores de referencia de Buy & Hold figuran en las Tablas 2 y 3.

6.3 Discusión de los resultados

Queda por explicar por qué ninguna de las estrategias supera a Buy & Hold, ni siquiera en bruto. Las causas son dos y las dos vienen del diseño mismo de las señales, no del costo.

La primera es el tiempo que se pasa afuera del mercado. Las tres estrategias tienen el capital en efectivo cerca de la mitad del período, y en una tendencia de fondo alcista cada rueda afuera es retorno que se pierde. El capital que espera entre operaciones no rinde, y en pesos además se desvaloriza contra la inflación.

La segunda es el retraso de las señales. Una media móvil reacciona después de que el precio ya se movió. En la práctica, la estrategia vende cuando la caída ya ocurrió y vuelve a comprar cuando el rebote ya pasó. En un mercado que se recupera rápido, en forma de V, esa demora deja a la estrategia afuera justo durante el tramo que paga. Por eso los dos resultados de las secciones anteriores son el mismo fenómeno. El retorno es más bajo y la protección no aparece porque la salida tardía no evita la caída y la reentrada tardía no aprovecha la suba.

El alcance de esta conclusión tiene un límite. El período estudiado es mayormente alcista y con caídas seguidas de recuperación, y en un mercado así salir nunca conviene. En un escenario distinto, de baja sostenida y sin rebote rápido, el valor de salir a tiempo podría aparecer. El período no contiene ese escenario, así que la conclusión vale para el tipo de mercado observado y no se generaliza a cualquier mercado.

6.3.1 La media como filtro de desastre

Si las estrategias no funcionan como sistema completo de operación, queda una pregunta más acotada. ¿Sirven al menos para salir en los desplomes? Para responderla se probó usar la media solo como alarma de salida sobre una posición pasiva, en lugar de operar de ida y vuelta. Sobre una base de Buy & Hold se midió una alarma con cada una de las cuatro variantes (HMA16, EMA 12/26, SMA triple y precio por debajo de la SMA100), comparándola contra Buy & Hold puro con el ratio Calmar, que pone el retorno anualizado sobre la caída máxima.

Ninguna de las cuatro alarmas mejora el Calmar respecto del benchmark, en ningún corte ni universo, 0 de 24 combinaciones. Incluso en el caso más favorable, la alarma queda por debajo. Las alarmas sí amortiguan la caída en el medio del desplome, porque salen antes del piso y registran un drawdown menor mientras dura la baja. Pero pierden la recuperación. Cuando el precio vuelve a su pico previo, todas terminan con menos riqueza que haber aguantado, porque la reentrada tardía deja el capital afuera del rebote.

El resultado se sostuvo al mirar el peor episodio de caída de cada ticker, elegido de forma objetiva por la caída máxima de Buy & Hold, el escenario más exigente para un filtro anti-desastre. Incluso ahí las alarmas salen bastante antes del piso, entre 37 y 165 ruedas según el universo, y reducen la caída intermedia, pero cuando el benchmark recupera su pico terminan por debajo en riqueza acumulada, entre el 63% y el 95% de la de B&H. La falla está en la reentrada, no en la salida. La alarma que menos pierde es la de la media triple, por ser la que menos sale. La peor es HMA16.

Dos aclaraciones metodológicas. La protección de un filtro de salida solo rendiría en una baja sostenida sin recuperación rápida, un escenario que este período no contiene, así que el resultado está acotado a la muestra. Y estas pruebas de robustez estaban definidas de antemano. No se agregaron indicadores ni se ajustaron parámetros para buscar un resultado favorable, de modo que no hay selección a posteriori.

6.3.2 Corrección de look-ahead bias en los validadores

La plataforma incorpora dos validadores que buscan confirmar las señales, el volumen relativo (VMA20) y los patrones de velas. Se evaluó si los *trades* confirmados por cada validador rinden mejor que los no confirmados, mirando la tasa de aciertos y el retorno por operación.

La primera medición del validador de volumen mostraba una ventaja amplia a favor de los trades confirmados. Al revisarla apareció un sesgo de anticipación (look-ahead bias). El criterio leía el volumen del propio día de entrada, que recién se conoce al cierre, mientras la operación se ejecuta a la apertura siguiente. Cuando el criterio se ajusta para leer el volumen de la barra de la señal, la ventaja casi desaparece. Sobre los 3.783 trades clasificables del agregado, la tasa de aciertos de los confirmados queda en 43,1% contra 40,5% de los no confirmados, un *spread* de 2,6 puntos que no alcanza significancia (prueba z de dos proporciones, $p \approx 0,12$). El diferencial tampoco aparece en la rentabilidad. El retorno medio por operación es prácticamente igual entre ambos grupos, incluso una fracción a favor de los no confirmados (diferencia de $-0,19$ puntos, t de Welch $p \approx 0,73$). Desagregado por método, el único caso donde el spread de aciertos llega a ser significativo es HMA16 aislado ($+4,7$ puntos, $p \approx 0,03$), pero ese diferencial de frecuencia tampoco se traduce en más retorno. El validador de volumen no mejora el resultado económico de las operaciones.

Con los patrones de velas el resultado fue otro. La ventaja aparente era casi enteramente circular, porque la vela del día de entrada incorpora el movimiento de ese mismo día, que ya forma parte del resultado del trade. Una vez corregido el sesgo, el spread de aciertos queda en

4,6 puntos pero con un p-valor de 0,37, sin significancia. La cobertura además es baja, solo el 10,7% de los trades cae en el grupo alineado. El validador de velas no resiste la corrección.

Los dos validadores empezaban con una ventaja inflada por usar datos del futuro. Corregida esa violación a las reglas, ninguno sostiene una ventaja estadística en el agregado. El de velas se cae ya en la tasa de aciertos, y el de volumen retiene a lo sumo un diferencial de frecuencia menor y no significativo, que tampoco se traduce en rentabilidad. El ejercicio deja además un registro concreto, con datos del propio trabajo, de cómo un look-ahead bias no controlado infla resultados, y es la razón por la que la regla de ejecutar a la apertura siguiente se aplica de forma pareja en toda la cadena.

Hasta acá, las estrategias no le ganan a Buy & Hold como sistema de operación ni como filtro de salida, y los validadores tampoco suman. El análisis técnico opera sobre el precio y el volumen, y por construcción no incorpora información de origen fundamental. Un cambio regulatorio, un resultado trimestral, un evento macroeconómico o una elección pueden mover una cotización por causas que ningún indicador de precio anticipa.

Esa limitación es la que justifica el panel de eventos fundamentales descrito en 5.5.4. El panel no predice ni explica esos movimientos. Marca, sobre la misma serie de precios, fechas que ya se sabían sensibles, las agendadas y las de algún hecho ocurrido con el mercado cerrado, donde queda margen hasta la apertura. Son cosas que el análisis técnico no ve venir. No se lo evaluó de forma cuantitativa ni entra en el backtesting, así que queda por fuera de lo que mide este capítulo. Sirve para que el usuario llegue con tiempo a esas fechas, y para dejar en claro que la lectura técnica es una dimensión más y no alcanza por sí sola para explicar cómo se mueve un activo.

6.4 Validación visual de la serie sintética MEP

El estudio trabaja sobre series sintéticas en dólares MEP, construidas a partir del precio local y el tipo de cambio implícito en el AL30/AL30D. Esta sección verifica que esa construcción representa de forma fiel el valor en dólares de cada acción. La verificación se apoya en una referencia externa e independiente, el ADR que las trece empresas cotizan en Estados Unidos, que es la misma acción negociada directamente en dólares. La comparación se hace acción por acción. Se toma el ADR, se lo divide por su ratio, la cantidad de acciones locales que representa, para dejarlo en dólares por acción, y se lo superpone contra la serie sintética MEP a lo largo del período.

Los precios de los ADR se tomaron de yfinance. La fuente es apropiada acá por ser una referencia independiente del mercado estadounidense, con datos de alta calidad para instrumentos líquidos de NYSE y NASDAQ, a diferencia de las limitaciones señaladas en 4.3.1 para las acciones locales. Las series se descargaron una vez y quedaron congeladas en el repositorio bajo el tag v1.0-adr, de modo que la validación se reproduce con independencia de la disponibilidad futura de la fuente. El tag fija el commit de la validación, lo que alcanza para reproducirla. A diferencia de la cadena principal (5.4), esta capa no incorpora un hash de integridad de los datos, porque es una verificación complementaria que no alimenta al backtest.

Las dos series se mueven casi igual. La correlación media entre sintético y ADR es de 0,998 sobre los trece nombres, sin desfase, ambas son contemporáneas y ninguna adelanta a la otra. La distancia entre las dos, medida como desviación porcentual media, separa al universo en dos grupos. Doce acciones quedan agrupadas entre 3,7% y 4,4%, con una mediana de 4,0%. Esa distancia no es ruido ni error de construcción. Es el desvío estructural entre el MEP, calculado por el AL30, y el dólar implícito en el ADR, los dos tipos de cambio que no siempre coinciden, según se explica en 4.3.2. El ratio empírico lo confirma. Si se calcula cuántas acciones locales valdría cada ADR a partir de los precios observados, el resultado reproduce el ratio nominal dentro de ese mismo 3% en las doce agrupadas. GGAL da 9,70 contra 10 e YPF 0,97 contra 1. La conversión a dólares está bien hecha.

La excepción es CRES, con una desviación del 12,4%, el triple que el resto. La brecha no fue pareja. Se concentró entre 2021 y 2023, con el ADR cotizando por debajo de la paridad, y desde 2024 las dos series se vuelven a juntar, como se ve en la Figura 9. La construcción de la serie no es el problema, en los otros doce nombres reproduce la paridad dentro del 4%. Lo que aparece acá es un descuento que el mercado le aplicó a este papel en ese tramo y que el arbitraje no cerró, algo esperable en el ADR menos líquido del panel. La paridad entre mercados funciona bien en los papeles líquidos y se afloja en los que operan poco, y CRES es la muestra de eso dentro del universo.

Las Figuras 4 a 9 muestran la superposición para tres acciones elegidas para cubrir el rango, con dos vistas por acción, la comparación en dólares por acción y la misma serie con el precio local agregado, indexadas a base 100. YPF es el caso canónico, ratio 1, el ADR más líquido y la mediana exacta del ranking de desvíos, con 4,0% (Figuras 4 y 5). GGAL es el de mejor ajuste, 3,7% (Figuras 6 y 7). CRES es el outlier, donde la brecha del 12% se ve a simple vista (Figuras 8 y 9). En la versión en dólares la serie sintética MEP y el ADR se superponen. En la de base 100 se agrega el precio local en pesos, que se dispara por la devaluación del período mientras las dos series en dólares se siguen entre sí.

La Tabla 7 reúne los trece nombres. La validación corre sobre el universo entero y no sobre tres casos elegidos, lo que descarta que se hayan mostrado solo los que dan bien. El estudio necesita una serie en dólares consistente para comparar estrategias, y la validación muestra que la tiene.

Tabla 7

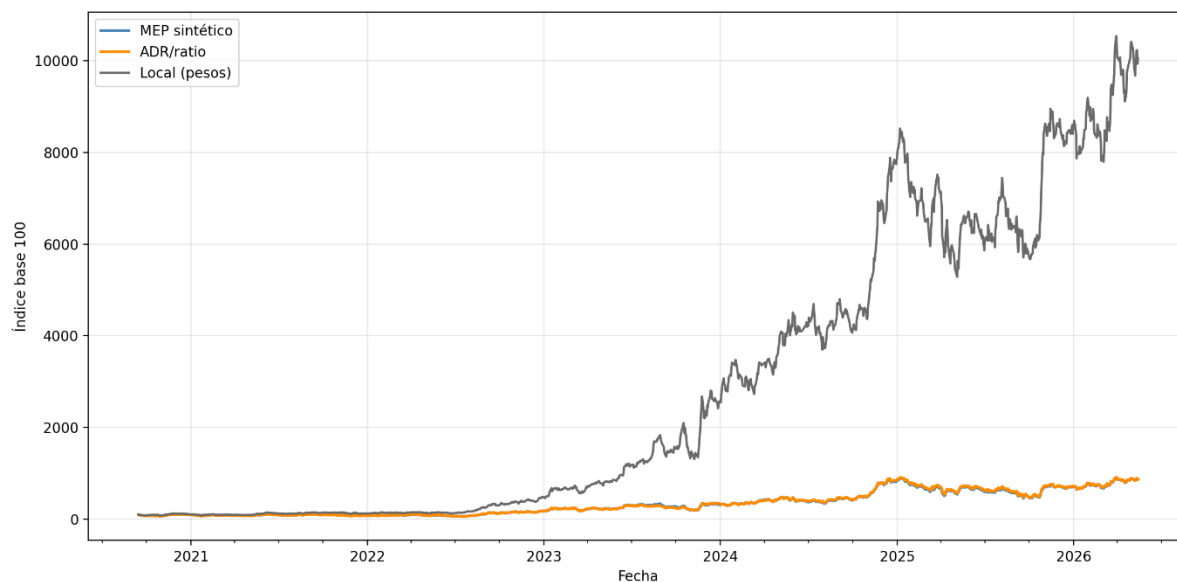
Validación de la serie sintética MEP contra el ADR real, por acción

Acción local	ADR	Ratio	Desvío
GGAL	GGAL	10	3,7%
BBAR	BBAR	3	3,7%
PAMP	PAM	25	3,8%
EDN	EDN	20	3,9%
TGSU2	TGS	5	3,9%
TECO2	TEO	5	3,9%
YPFD	YPF	1	4,0%
BMA	BMA	10	4,0%
CEPU	CEPU	10	4,1%
SUPV	SUPV	5	4,1%
LOMA	LOMA	5	4,2%
IRSA	IRS	10	4,4%
CRES	CRESY	10	12,4%

Nota. Desviación media entre la serie sintética MEP y el ADR real, ordenada de menor a mayor. La correlación media es de 0,998 y el desfasaje es cero en las trece. Fuente yfinance. El desvío del racimo (3,7% a 4,4%) corresponde al spread entre el MEP y el dólar implícito en el ADR, explicado en 4.3.2. CRES se aparta por una brecha de paridad propia del activo, descrita en el texto.

Figura 4*YPF, MEP sintético y ADR*

Nota. Ambas series se expresan en dólares por acción local. El ADR se divide por su ratio de conversión. Elaboración propia a partir de datos de yfinance.

Figura 5*YPF en base 100, las tres series*

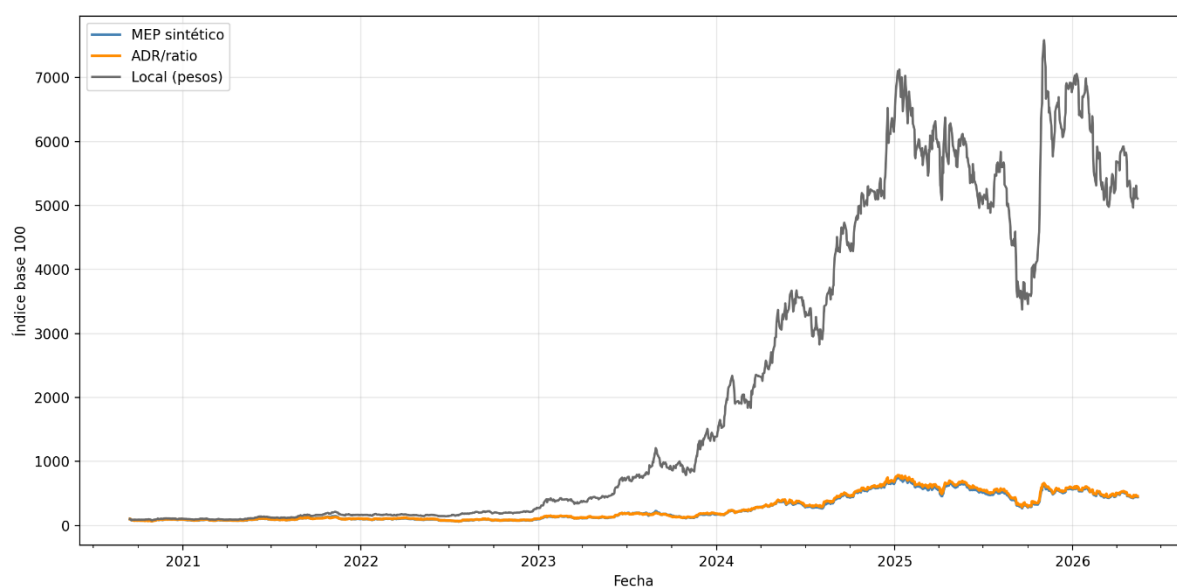
Nota. Serie sintética MEP, ADR y precio local en pesos, indexadas a 100 en la primera rueda común. Elaboración propia a partir de datos de yfinance.

Figura 6
GGAL, MEP sintético y ADR

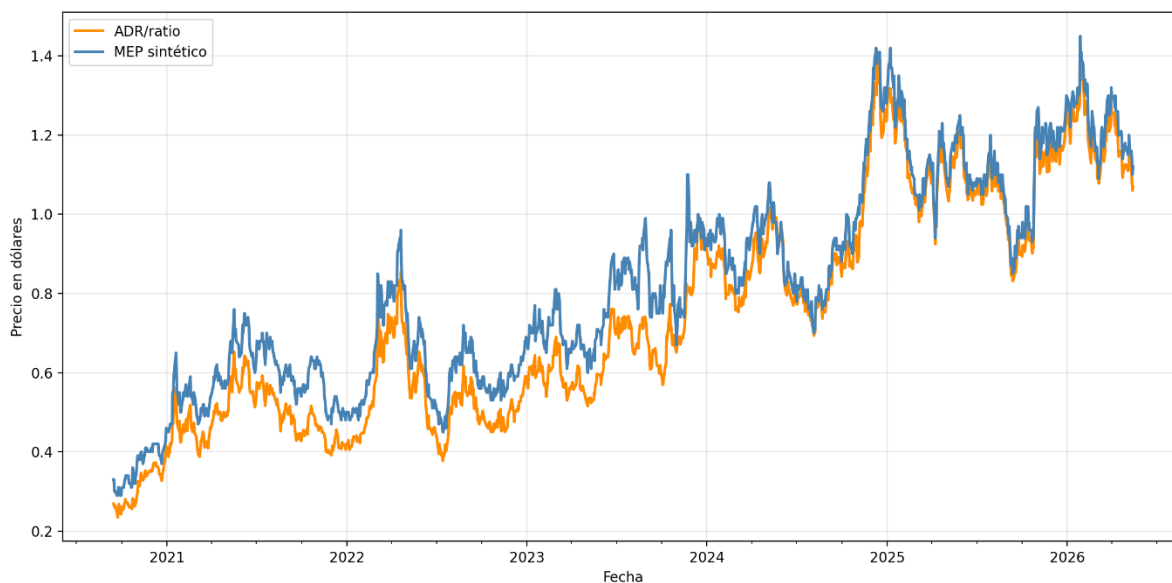


Nota. Ambas series se expresan en dólares por acción local. El ADR se divide por su ratio de conversión. Elaboración propia a partir de datos de yfinance.

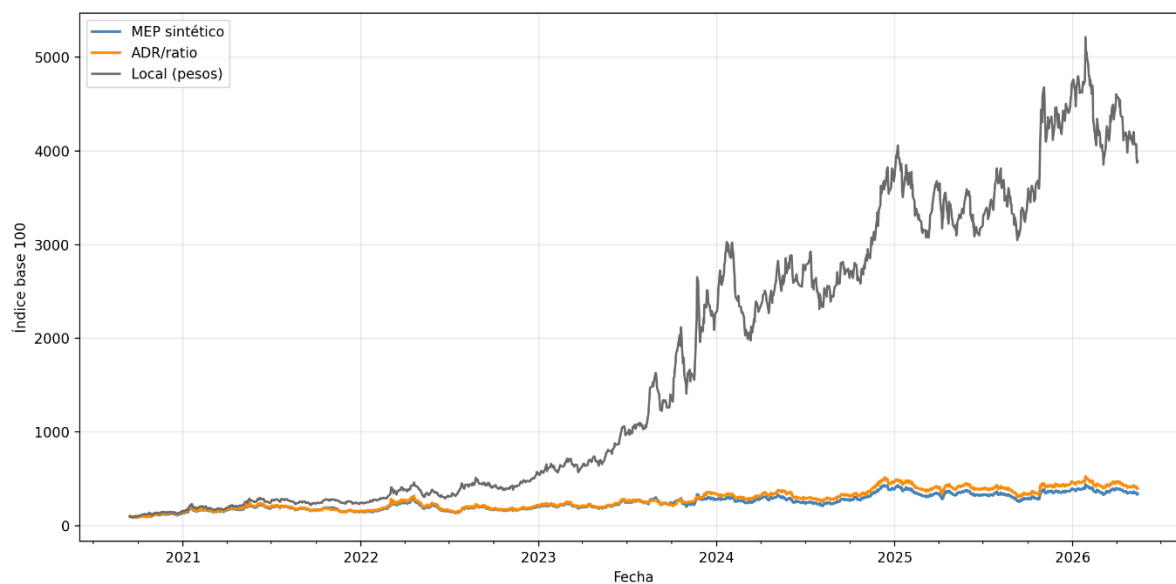
Figura 7
GGAL en base 100, las tres series



Nota. Serie sintética MEP, ADR y precio local en pesos, indexadas a 100 en la primera rueda común. Elaboración propia a partir de datos de yfinance.

Figura 8*CRES, MEP sintético y ADR*

Nota. Ambas series se expresan en dólares por acción local. El ADR se divide por su ratio de conversión. Elaboración propia a partir de datos de yfinance.

Figura 9*CRES en base 100, las tres series*

Nota. Serie sintética MEP, ADR y precio local en pesos, indexadas a 100 en la primera rueda común. Elaboración propia a partir de datos de yfinance.

Capítulo 7. Conclusiones

7.1 Conclusiones principales

La intervención dejó una plataforma desplegada y en funcionamiento, y todo lo que calcula puede volver a generarse desde los datos crudos. El propósito era evaluar señales técnicas sobre acciones argentinas dejando rastro de cada paso, sin adelantarse con información futura. Cumplirlo pesa más que cualquier número de rentabilidad que la plataforma pueda producir.

En lo técnico, el aporte es que la cadena completa, de la cotización cruda al resultado del backtest, corre sin intervención manual. El control anti-look-ahead no depende del cuidado de quien programó. Ninguna señal usa datos posteriores a la rueda en que se genera, la operación recién se ejecuta en la apertura de la rueda siguiente, y las dos cosas se pueden comprobar en el código.

Los resultados de la evaluación fueron uniformes. Ninguna de las tres estrategias (HMA16, EMA 12/26 y SMA 10/50/100) supera a comprar y mantener, en ninguno de los tres cortes ni en los dos universos. El resultado incluso se mantiene al quitar las comisiones de la simulación, con lo cual la falla no está en el costo de operar sino en las señales mismas. El universo MEP es el que mejor refleja el resultado económico real, porque al medir en dólares descuenta el efecto de la devaluación. Ahí las tres estrategias quedan cerca de cero o en números negativos.

La plataforma también sirvió para probar una variante que suele defenderse más, usar las medias no para operar sino como alarma de salida ante una caída. Se corrieron las veinticuatro combinaciones posibles y en ninguna el resultado mejoró al de quedarse comprado. Los datos del capítulo 6 muestran dónde falla. La venta llega antes del piso pero la recompra llega tarde, con el rebote ya consumido, y en este mercado esa demora costó más que la caída que se quería evitar.

Que el análisis técnico no le haya ganado a comprar y mantener en este período no va en contra del trabajo. El objetivo era construir una herramienta que pudiera medir eso de forma controlada, y la medición se hizo, quedó documentada y puede repetirla cualquiera que clone el repositorio.

7.1.1 El caso de HMA16

HMA16 fue el caso de prueba principal. Por construcción reacciona antes que una media simple o exponencial, y esa velocidad era el atributo a evaluar. En la práctica jugó en contra. Genera entre 75 y 88 operaciones por ticker en promedio, según el universo, cada una paga comisión y muchas son señales falsas. Como alarma de salida tampoco rindió, fue la peor de las variantes probadas por el mismo motivo, sale del mercado y vuelve a entrar demasiadas veces.

Reaccionar más rápido no fue una ventaja en este período. El período tampoco fue neutral, fue mayormente alcista y las caídas se recuperaron rápido, y en un mercado así salir antes de tiempo siempre cuesta. En una baja larga o en un mercado sin dirección clara HMA16 podría dar otro resultado. Con estos datos no se puede afirmar ni descartar. La plataforma queda lista para repetir la prueba cuando ese escenario aparezca en los datos, que es lo dicho al principio del capítulo.

7.2 Cumplimiento de objetivos

El objetivo general se cumplió. La plataforma monitorea y evalúa señales de varios sistemas de medias móviles, con la validación por patrones de velas incorporada, y compara el desempeño en acciones locales contra sus versiones sintéticas en dólares MEP.

Los cuatro objetivos particulares también se cumplen. El pipeline de siete módulos ingesta, normaliza, calcula indicadores y exporta resultados, y cada corrida registra su configuración, las versiones de librerías y un hash por archivo. La web presenta las señales por ticker para un usuario no experto. El backtester evaluó HMA16 contra las otras medias móviles y contra comprar y mantener.

La reproducibilidad se resolvió por verificación de archivos. Desde el checkout del tag se confirma el SHA256 de los datos crudos, y al volver a correr el pipeline los resultados se contrastan contra los agregados de backtest versionados, que son las salidas que el trabajo reporta. Las capas intermedias se rehacen en cada corrida y no se guardan. Es una decisión de diseño, si las salidas finales coinciden, lo intermedio también tuvo que coincidir.

El objetivo era evaluar el desempeño, y eso se logró. Otra cuestión es si el resultado fue favorable o no. Que la evaluación haya concluido que las estrategias no le ganan al mercado no es un objetivo incumplido. La plataforma cumplió su cometido y mostró un resultado respaldado por datos. Una herramienta que solo pudiera confirmar resultados favorables no serviría para evaluar.

Queda un punto más del objetivo general del Capítulo 1. La arquitectura se diseñó modular y documentada para que otros puedan tomarla y probar sus propios métodos. El trabajo cierra como intervención, con la plataforma funcionando y la evaluación hecha, y a la vez sirve de base para seguir. Del lado de sistemas eso significa extender el pipeline, sumar indicadores o cambiar la granularidad. Del lado financiero, probar otras estrategias sobre períodos y mercados que esta muestra no cubre.

7.3 Contribuciones del trabajo

La contribución central es la plataforma. El trabajo entrega un pipeline modular sobre archivos planos. Corrido de nuevo sobre los mismos datos crudos y con las versiones de librerías fijadas en requirements.txt, reproduce los resultados que el trabajo reporta. Persistir en CSV versionado en Git, y no en una base de datos, es lo que permite verificar la salida y rastrear cualquier valor publicado en la web hasta una rueda concreta, sin herramientas adicionales. Se suman los controles anti-look-ahead metidos en el propio código, la señal se calcula al cierre de una rueda y la orden se ejecuta a la apertura de la siguiente, sin excepciones. También el marco de gobernanza de cinco dominios y la arquitectura estática, que no tiene servidores que mantener ni costo de operación. Como aporte metodológico aparte, el trabajo documenta un sesgo de anticipación que se detectó en los propios validadores. Los primeros números mostraban que el filtro de volumen mejoraba las señales, pero esa mejora venía de que el validador miraba datos que en la práctica todavía no existían. Corregido el error, esa mejora no se sostuvo, la diferencia a favor de las señales filtradas dejó de ser significativa. Quedó registrado con datos propios y también cuenta como aporte.

La serie sintética en dólares MEP también es un aporte, y quedó validada. La comparación contra el ADR real de cada empresa dio una correlación media de 0,998 con un desvío cercano al 4%, es decir que la serie construida replica el precio que efectivamente se negocia en dólares. Ningún antecedente revisado había dolarizado series locales por MEP ni verificado el resultado contra el ADR. Ese método queda disponible para cualquier trabajo que estudie series argentinas en dólares.

Del lado financiero queda un dato que no existía para este mercado. HMA16, comparada contra las medias clásicas y contra comprar y mantener sobre series dolarizadas, no le gana al benchmark. El alcance es acotado, trece acciones, un solo tipo de mercado y una sola partición de validación, así que la evidencia sirve como punto de partida, pero los resultados no son concluyentes. El rigor del trabajo no depende de generalizar este hallazgo. Depende de

que el camino completo, desde la descarga cruda hasta el número publicado, se puede reproducir y auditar.

7.4 Limitaciones

La limitación más seria está en cómo se validó. Los datos se partieron una sola vez, cerca de 70/30 por fecha, cuando la literatura recomienda repetir la prueba sobre varias ventanas fuera de muestra. Tampoco se probó qué pasa si se cambia el parámetro de la HMA o los demás valores que quedaron fijos por diseño. Contra el sobreajuste juegan dos cosas. Los parámetros son los valores estándar de la literatura y no se tocaron después de ver los resultados, y el corte respeta el orden del tiempo, el tramo de validación siempre queda después del tramo con el que se compara, así que por ahí no se filtra información futura. Aun así, la falta de ventanas múltiples no queda compensada. Las pocas veces que una estrategia le gana al benchmark en la validación en dólares ocurren en una ventana corta, y el resultado cambia si se mueve la fecha de corte, así que no hay que leerlas como señal.

El backtester no modela slippage. Descuenta un costo fijo del 1% por operación completa como referencia del minorista, pero el deslizamiento entre el precio de la señal y el de ejecución real queda fuera. En las acciones de menor liquidez ese deslizamiento existe y no está descontado, así que los retornos simulados quedan por encima de los que un inversor habría obtenido.

El resultado está acotado al tipo de mercado observado. El período, mayormente alcista y de recuperaciones rápidas, no contiene una baja sostenida sin rebote. Ese es justo el escenario donde un filtro de salida podría rendir, y donde la reactividad de HMA16 quedó sin ponerse a prueba. La conclusión vale para esta muestra, pero no para cualquier mercado.

Quedan otras limitaciones de menor peso. Los datos de mercado salen de una sola fuente, la API de PPI. El MEP y el CCL no siempre coinciden, así que la validación contra el ADR es aproximada. CRES queda aparte, con un desvío del 12,4% cuando el resto ronda el 4%, por una causa propia del instrumento que se discute en el capítulo 6. La web se actualiza una vez por día, no en tiempo real, y no ejecuta operaciones. El panel de análisis fundamental entró como componente de demostración y su capacidad predictiva no se midió, porque medirla sobre los mismos eventos que lo definen sería circular.

7.5 Líneas de trabajo futuro

Rigor de la evaluación. Lo primero es cambiar el esquema de validación. Hoy los datos se parten una sola vez, y la mejora es pasar a un walk-forward que repita la prueba sobre varias ventanas fuera de muestra. Después, un análisis de sensibilidad que no se limite al período de la HMA y cubra también los parámetros que hoy están fijos por decisión, las ventanas de las medias, el tope de tenencia de treinta ruedas y el umbral de costo. Sin esas dos pruebas no se puede afirmar que las conclusiones no dependan de una configuración puntual. Incorporar además el slippage al modelo de costos dejaría los retornos simulados más cerca de los que un inversor obtiene al ejecutar sus órdenes en el mercado.

Alcance del estudio financiero. El límite está en la muestra. Ampliar el universo más allá de las trece acciones con ADR y, sobre todo, extender el período hasta cubrir una baja sostenida sin rebote rápido, el escenario que esta muestra no tiene. Eso pondría a prueba la conclusión central, que hoy está atada a un mercado mayormente alcista, y de paso a la reactividad de HMA16. Un barrido del período de la HMA por activo y por tipo de mercado, en lugar del 16 fijo, abriría la pregunta de si existe una calibración mejor que la estándar.

Evolución de la plataforma. La arquitectura de archivos planos rinde para el volumen actual. El benchmark de la Figura 2 muestra que el límite está en el tiempo de cómputo y no en la memoria, y que casi todo ese tiempo se va en el cálculo de la HMA, hoy resuelto con `rolling().apply()` de pandas, que llama una función de Python por cada ventana. Reescribir ese cálculo con numpy lo aceleraría uno o dos órdenes de magnitud, y conviene hacerlo antes que cambiar la forma de guardar los datos. Migrar a una base de datos o a un formato columnar recién tendría sentido si, aun vectorizado, el cálculo no llegara a escala intradiaria sobre cientos de instrumentos. A más corto plazo conviene publicar los archivos con un hash en el nombre, para que el navegador no muestre datos viejos que le quedaron en el caché, y sumar una segunda fuente de datos de mercado por si la API de PPI falla o cambia. También queda pendiente que un indicador nuevo pueda agregarse por configuración, sin tocar el código del pipeline. Y falta separar el manifiesto de fuentes del directorio de datos crudos. La validación contra los ADR reescribe ese archivo dentro del directorio sobre el que se calcula el hash de integridad, y por eso el Anexo III exige verificar el hash antes de correr nada. Movidado el manifiesto, esa restricción de orden desaparece.

Panel fundamental. El panel entró como demostración y es donde más se puede crecer. Primero hay que formalizar cómo se seleccionan y cargan los eventos, y recién después diseñar una forma de medir su valor como señal que no sea circular, es decir, que no lo evalúe sobre

los mismos eventos que lo definen. Con eso resuelto, el paso siguiente sería detectar eventos de forma automática con procesamiento de lenguaje natural, como hacen varios de los antecedentes revisados.

Anexos

Anexo I. Pseudocódigo y diagramas

Este anexo documenta la lógica interna de la plataforma en dos formas complementarias. El pseudocódigo describe cada uno de los siete módulos del pipeline en el orden en que los ejecuta la integración continua, con un vocabulario en castellano pensado para que la lógica se entienda sin leer el código fuente. Los diagramas representan la secuencia de ejecución y la máquina de estados que genera las señales. El diagrama de arquitectura del pipeline se presenta en el cuerpo del trabajo, en la sección de arquitectura del sistema.

I.1 Pseudocódigo de los módulos

Módulo 1: Ingesta → data_downloader.py

Obtiene las cotizaciones de la API de PPI. En la primera corrida hace un backfill histórico; en las siguientes descarga solo el tramo faltante y lo consolida a un registro OHLCV por rueda.

```

FUNCIÓN run_ingestion():
    cliente ← get_cliente()                # autentica con claves PPI de entorno
    PARA CADA (ticker, label) EN (BONOS + ACCIONES):
        SI existe data_raw/csv/{label}.csv:
            desde ← última_fecha_registrada + 1 día    # incremental
        SINO:
            desde ← 2020-09-04                    # backfill inicial
        crudo ← cliente.descargar(ticker, desde, hoy)
        ohlcv ← process_data(crudo)                # consolida a un registro diario por rueda
        serie ← concatenar_y_deduplicar_por_fecha(previo, ohlcv)
        escribir data_raw/csv/{label}.csv

```

Módulo 2: Normalización → data_normalizer.py

Estandariza las series crudas a un esquema OHLCV uniforme. El ajuste por splits queda disponible pero inactivo, porque PPI ya entrega la cotización ajustada.

```

FUNCIÓN main():
    splits ← cargar_splits.json              # en la práctica vacío: PPI ya entrega ajustado
    PARA CADA archivo EN data_raw/csv/*.csv:
        df ← read_raw_ohlcv(ticker, splits)    # esquema OHLCV estándar, tipos coercitivos,
                                                # dropna(close), consolidado diario
        df ← apply_split_adjustments(df, ticker, splits)  # inactivo si no hay config
        df ← redondear_OHLC(df, 2)
        escribir data_normalized/csv/{ticker}_normalized.csv

```

Módulo 3: Conversión a MEP → synthetic_mep_generator.py

Construye la serie del tipo de cambio MEP a partir del cociente AL30 / AL30D y expresa cada acción en dólares. El uso de uniones internas define el universo efectivo: una serie MEP existe solo en las ruedas con cotización simultánea del par de bonos y de la acción.

```

FUNCIÓN calculate_base_mep():
    al30 ← leer data_normalized/csv/AL30_BA_normalized.csv
    al30d ← leer data_normalized/csv/AL30D_BA_normalized.csv
    base ← merge(al30, al30d, POR date, TIPO inner) # solo ruedas con ambos bonos
    SI base está vacía: DEVOLVER None
    base.mep_value ← base.al30_close / base.al30d_close
    DEVOLVER base[date, mep_value] # 1383 ruedas MEP-construibles

FUNCIÓN main():
    base ← calculate_base_mep()
    SI base es None: ABORTAR (sin bonos no hay MEP)
    PARA CADA ticker EN TICKERS_ACCIONES: # 13 acciones
        accion ← leer {ticker}_normalized.csv
        df ← merge(accion, base, POR date, TIPO inner) # intersección acción n MEP
        PARA CADA col EN [open, high, low, close]:
            df[col] ← redondear(df[col] / df.mep_value, 2)
        renombrar sufijo _BA → _MEP
        escribir data_normalized/csv/{ticker}_MEP_normalized.csv

```

Módulo 4: Indicadores → indicator_engine.py

Calcula los indicadores sobre las series normalizadas y no genera señales. Esa separación deja el cálculo numérico aislado del que lo interpreta.

```

FUNCIÓN process_ticker(ticker, archivo):
    df ← leer archivo; validar columnas requeridas; ordenar por fecha
    df.hma16 ← calculate_hma(close, span=16) # 2·WMA(n/2) - WMA(n), suavizado √n
    df.sma10/50/100 ← media_móvil_simple(close, ventana) # NaN durante warm-up
    df.ema12/26 ← media_móvil_exp(close); primeras (span-1) filas ← NaN
    df.vma20_ratio ← volume / SMA20(volume)
    escribir data_indicators/csv/{ticker}_indicators.csv

FUNCIÓN main():
    PARA CADA archivo EN data_normalized/csv/*_normalized.csv:
        process_ticker(...)

```

Módulo 5: Señales → signal_generator.py

Traduce los indicadores en señales de tendencia mediante una máquina de estados compartida por los tres métodos. Cada método difiere solo en la función que define el estado alcista o bajista de cada barra. El módulo agrega además la categoría de volumen y el patrón de vela.

```

FUNCIÓN maquina_estados_tendencia(n, obtener_estado):
    trend ← ["Neutra"]; prev ← None; up_count ← 0; down_count ← 0
    PARA i DESDE 1 HASTA n-1:
        state ← obtener_estado(i, prev)
        SI state es INVÁLIDO:
            trend.agregar("Neutra"); prev ← None; up_count ← 0; down_count ← 0; CONTINUAR
        SI prev es None:                                # primera barra válida
            SI state="up": trend.agregar("Compra Temprana"); up_count ← 1
            SINO SI state="down": trend.agregar("Venta Temprana")
            SINO: trend.agregar("Neutra")
        SINO SI state="up" Y prev≠"up":                  # giro alcista
            up_count ← 1; down_count ← 0; trend.agregar("Compra Temprana")
        SINO SI state="down" Y prev≠"down":              # giro bajista
            down_count ← 1; up_count ← 0; trend.agregar("Venta Temprana")
        SINO:                                            # continuación
            SI state="up":
                SI 0<up_count<2: up_count++; trend.agregar("Compra Confirmada")
                SINO: trend.agregar("Señal Alcista")
            SINO SI state="down":
                SI 0<down_count<2: down_count++; trend.agregar("Venta Confirmada")
                SINO: trend.agregar("Señal Bajista")
            SINO: trend.agregar(trend[último])          # arrastra estado vigente
        prev ← state
    DEVOLVER trend

# Las 3 tendencias comparten la máquina y difieren solo en obtener_estado:
#   HMA16      → "up"/"down" según signo de la pendiente del HMA
#   EMA 12/26  → "up"/"down" según signo de (ema12 - ema26)
#   SMA 10/50/100 → "up" si s10>s50>s100; "down" si s10<s50<s100; INVÁLIDO si mezclado

FUNCIÓN process_ticker(ticker, archivo):
    df ← leer archivo; validar 14 columnas
    df.T_hma16      ← maquina_estados_tendencia(n, estado_hma)
    df.T_ema12_26   ← maquina_estados_tendencia(n, estado_ema)
    df.T_sma10_50_100 ← maquina_estados_tendencia(n, estado_sma)
    df.vma20_cat     ← categorize_vma20(vma20_ratio)      # 6 niveles: Muy Bajo ... Muy Alto
    df.candle_pattern ← identify_candle_pattern(df)       # Engulfing/Doji/Marubozu por geometría
    escribir data_signals/csv/{ticker}_signals.csv

```

Módulo 6: Backtest y significancia → backtester.py

Simula operaciones sobre las señales y calcula las métricas de desempeño. El control anti-look-ahead está embebido en el desfase de la operación: la señal se lee al cierre de una rueda y se ejecuta a la apertura de la siguiente. Incluye los tests de significancia resueltos en forma cerrada, sin dependencias externas.

```

FUNCIÓN run_backtest(indicadores, señales, columna_señal, max_hold=30):
    trades ← []; posición ← None
    PARA i DESDE 0 HASTA n-2:                                # nunca la última barra
        señal ← norm_sig(señales[columna_señal][i])          # normaliza a {-1, 0, +1}
        SI posición es None Y señal = +1:
            entrada ← open[i+1]                                # entra a la APERTURA de la rueda siguiente
            posición ← abrir(entrada, fecha[i+1])
        SINO SI posición abierta:
            SI señal = -1 O barras_en_posición ≥ max_hold:
                salida ← open[i+1]                              # sale también a la apertura siguiente
                trade ← cerrar(posición, salida) - COST_PER_TRADE
                trades.agregar(trade); posición ← None
    DEVOLVER trades

# El desfase i → i+1 es el control anti-look-ahead: la señal se lee al cierre de i
# y recién se opera al open de i+1. Ninguna decisión usa datos de su propia rueda.

FUNCIÓN main():
    PARA CADA ticker × método EN {HMA16, EMA_12_26, SMA_10_50_100}:
        trades ← run_backtest(...)
        filtrar trades POR STUDY_START ≤ entrada ≤ STUDY_END
        marcar in_sample SI entrada ≤ STUDY_CUTOFF (2024-08-30), SINO out_of_sample
        resumen ← summarize_trades(trades)                    # win_rate, retorno, drawdown (total/IS/OOS)
    run_buy_and_hold(...)                                     # benchmark: 1 trade inicio→fin
    compute_validator_significance(...)                       # z de proporciones + t de Welch (sin scipy)
    escribir data_public/backtests/*.csv

```

Módulo 7: Serialización → json_generator.py

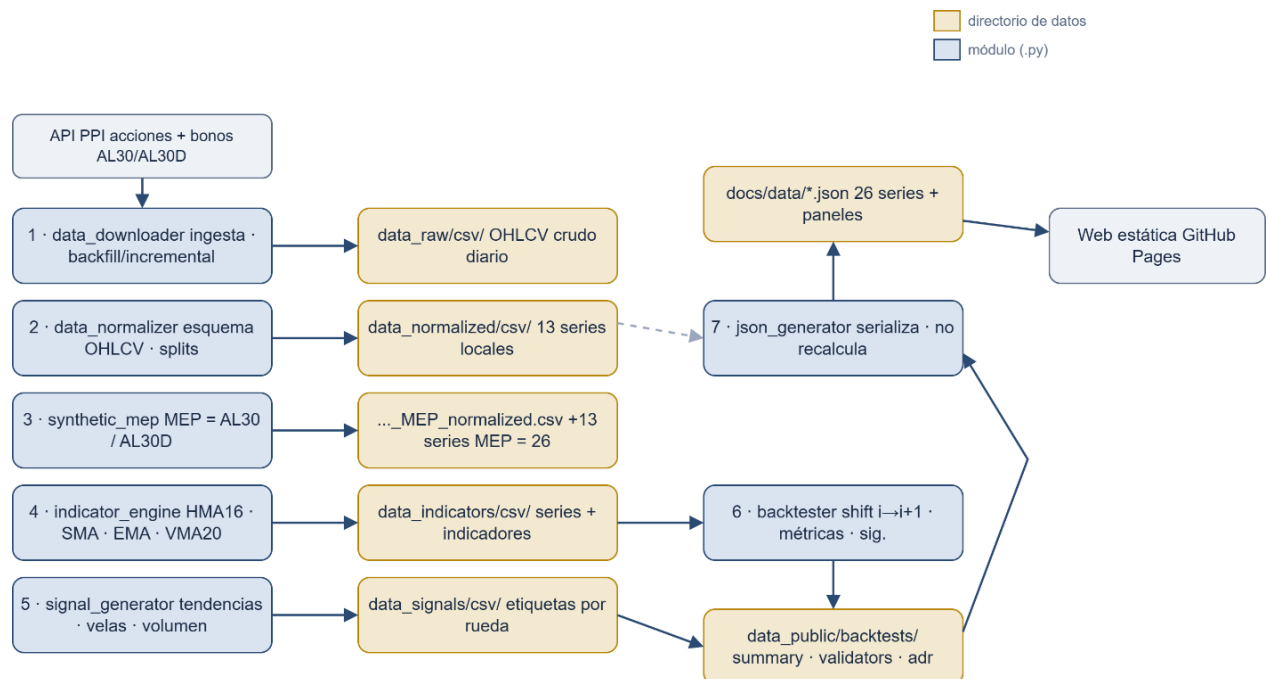
Serializa a JSON lo que el pipeline ya produjo, sin recalcular. Los archivos resultantes son los que consume el sitio estático.

```
# Standalone (docstring lo marca on-demand), pero el workflow lo invoca como paso final.
FUNCIÓN main():
    gen_meta()          # _meta.json global: fechas de estudio, tickers_count, métodos, costo
    gen_summary()       # resumen de backtest por ticker/método
    gen_validators()    # efectividad y significancia de validadores
    gen_local_vs_mep()  # comparación par local ↔ MEP
    gen_fundamentals()  # panel de eventos curado
    gen_daily_panel()   # señales por rueda, filtrado desde warmup_end_date
    PARA CADA ticker EN _all_tickers():          # 13 _BA + 13 _MEP = 26 series
        gen_ticker(ticker) # merge inner indicadores n señales → {ticker}.json
    # el generador no recalcula: serializa lo que el pipeline ya produjo
```

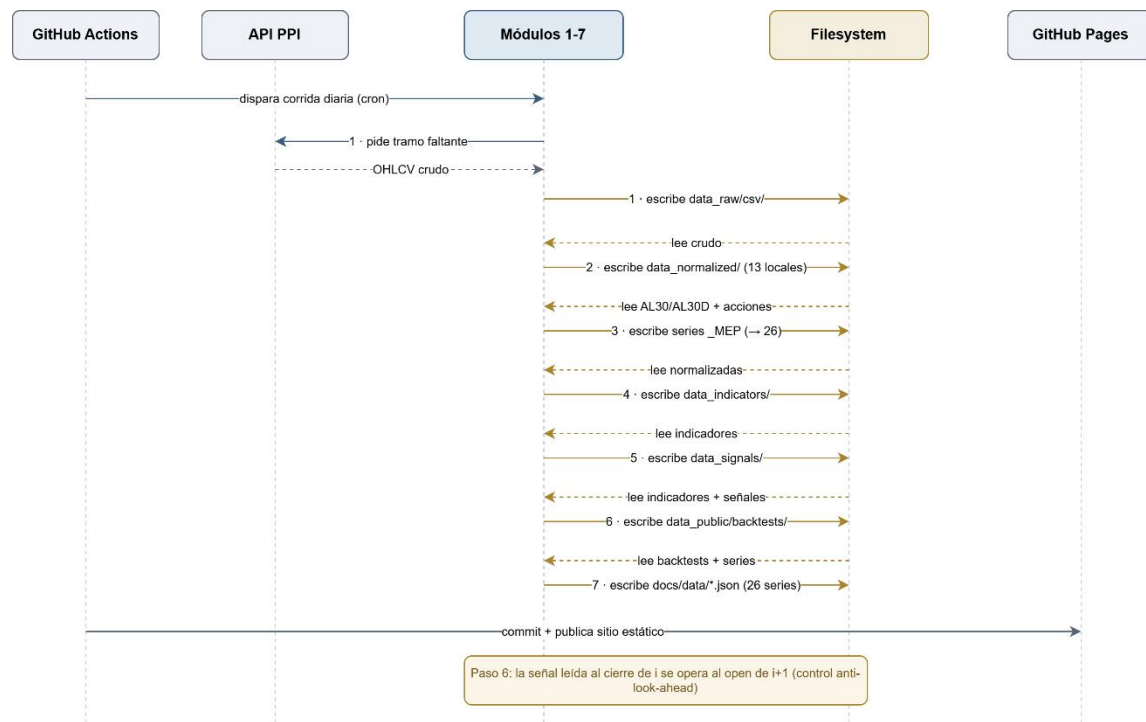
I.2 Diagramas

Figura 10

Flujo de datos entre módulos del pipeline

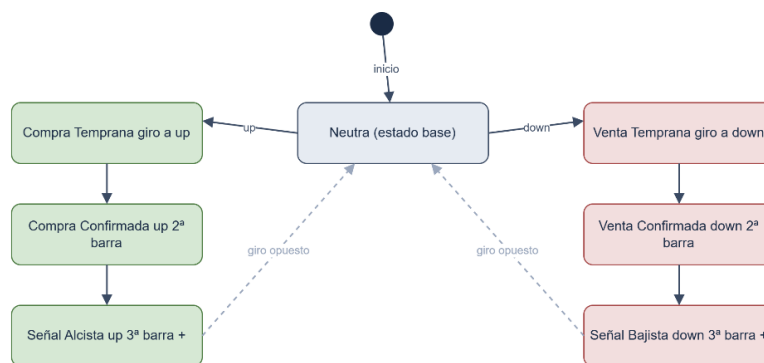


Nota. Elaboración propia a partir de daily-pipeline.yml y los módulos del repositorio.

Figura 11*Secuencia de ejecución del pipeline*

Nota. Elaboración propia a partir de daily-pipeline.yml.

La máquina emite la etiqueta de tendencia de cada barra. Un giro lleva al estado Temprana, la continuación pasa a Confirmada y luego a la señal sostenida. Una entrada inválida devuelve a Neutra y reinicia los contadores.

Figura 12*Máquina de estados de señales*

Entrada inválida (SMA mezclada o NaN de warm-up) → vuelve a Neutra y reinicia contadores.

Nota. Elaboración propia a partir de signal_generator.py

Anexo II. Interfaz web: capturas y esquema JSON

Este anexo documenta la capa de publicación. La primera parte muestra las tres vistas de la interfaz web. La segunda describe el esquema de los archivos JSON que el pipeline genera y que el sitio consume sin recalcular. Cada valor visible en pantalla se corresponde con un campo del esquema.

II.1 Interfaz web

El sitio se organiza en tres pestañas. Cada figura muestra una.

Figura 13

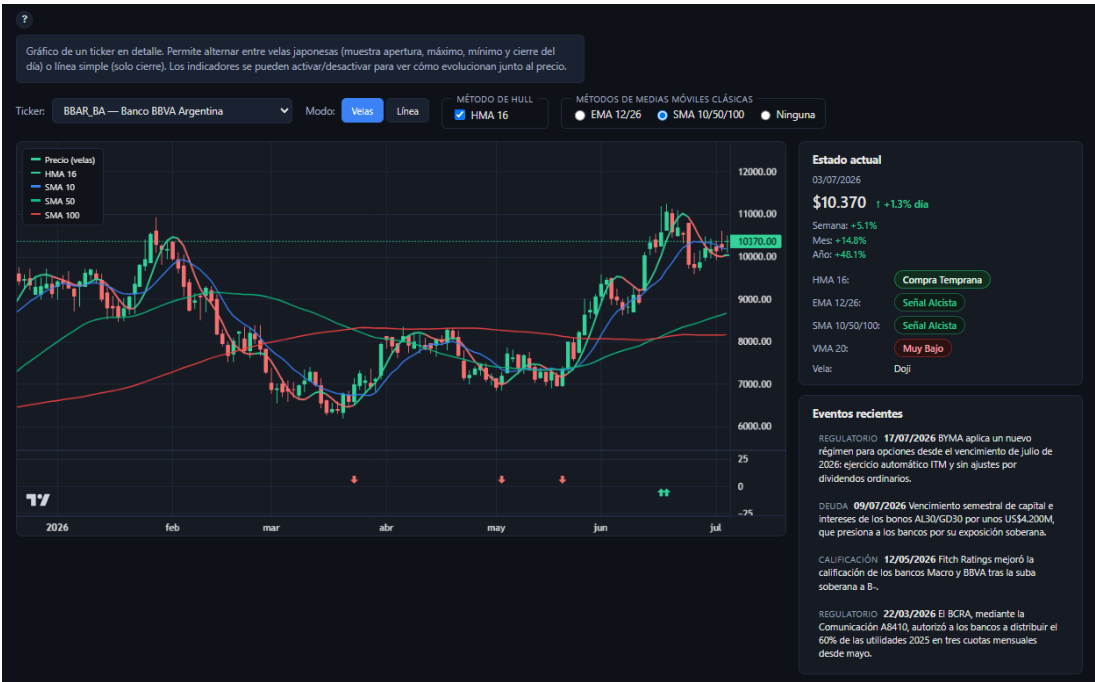
Panel de señales del día

Última rueda: 03/07/2026 Tickers: 26/26 ▼ Fecha: 03/07/2026 ☐							
#	TICKER	MERCADO	HMA 16	EMA 12/26	SMA 10/50/100	VMA 20	VELA DEL DÍA
1	BBAR	Local	Compra Temprana	Señal Alcista	Señal Alcista	Muy Bajo	Doji
2	BBAR	MEP	Señal Bajista	Señal Alcista	Señal Alcista	Muy Bajo	Doji
3	BMA	Local	Compra Temprana	Señal Alcista	Señal Alcista	Muy Bajo	Doji
4	BMA	MEP	Señal Bajista	Señal Alcista	Señal Alcista	Muy Bajo	Doji
5	CEPU	Local	Compra Confirmada	Señal Alcista	Neutra	Muy Bajo	—
6	CEPU	MEP	Señal Bajista	Señal Bajista	Señal Bajista	Muy Bajo	Doji
7	CRES	Local	Compra Temprana	Señal Alcista	Neutra	Muy Bajo	—
8	CRES	MEP	Señal Bajista	Señal Bajista	Señal Bajista	Muy Bajo	—
9	EDN	Local	Compra Temprana	Señal Bajista	Neutra	Muy Bajo	—
10	EDN	MEP	Señal Bajista	Señal Bajista	Señal Bajista	Muy Bajo	—
11	GGAL	Local	Compra Temprana	Señal Alcista	Señal Alcista	Muy Bajo	—
12	GGAL	MEP	Compra Temprana	Señal Alcista	Señal Alcista	Muy Bajo	—
13	IRSA	Local	Compra Confirmada	Señal Alcista	Neutra	Muy Bajo	—
14	IRSA	MEP	Señal Bajista	Señal Alcista	Neutra	Muy Bajo	—
15	LOMA	Local	Compra Confirmada	Señal Alcista	Señal Alcista	Muy Bajo	—
16	LOMA	MEP	Señal Bajista	Señal Alcista	Señal Alcista	Muy Bajo	—
17	PAMP	Local	Compra Confirmada	Señal Alcista	Señal Alcista	Muy Bajo	Doji
18	PAMP	MEP	Señal Bajista	Señal Bajista	Señal Bajista	Muy Bajo	Doji
19	SUPV	Local	Compra Temprana	Señal Alcista	Neutra	Muy Bajo	—
20	SUPV	MEP	Señal Bajista	Señal Alcista	Neutra	Muy Bajo	—
21	TECO2	Local	Señal Alcista	Señal Alcista	Señal Alcista	Muy Bajo	Doji
22	TECO2	MEP	Señal Bajista	Señal Bajista	Neutra	Muy Bajo	Doji
23	TGSU2	Local	Señal Bajista	Venta Confirmada	Neutra	Muy Bajo	—
24	TGSU2	MEP	Señal Bajista	Señal Bajista	Señal Bajista	Muy Bajo	—
25	YPFD	Local	Compra Temprana	Señal Bajista	Neutra	Muy Bajo	—
26	YPFD	MEP	Señal Bajista	Señal Bajista	Neutra	Muy Bajo	—

Nota. Elaboración propia. Vista alimentada por daily_panel.json.

Figura 14

Detalle del activo



Nota. Elaboración propia. Vista alimentada por ticker_{TICKER}.json.

Figura 15

Eventos del mercado

Men: Todos los meses Tickers: 14/14 ▼

FECHA	TICKERS	EVENTO	FUENTE
17/07/2026	GGAL, YPF, BMA, PAMP, BBAR	REGULATORIO BYMA aplica un nuevo régimen para opciones desde el vencimiento de julio de 2026: ejercicio automático ITM y sin ajustes por dividendos ordinarios.	Ver fuente
09/07/2026	GGAL, BMA, BBAR, SUPV	DEUDA Vencimiento semestral de capital e intereses de los bonos AL30/GD30 por unos US\$4.200M, que presiona a los bancos por su exposición soberana.	Ver fuente
01/07/2026	PAMP	OPERATIVO Pampa Energía proyecta alcanzar 28.000 barriles diarios en Rincón de Aranda hacia mediados de 2026, desde los 24.000 actuales.	Ver fuente
30/06/2026	YPFD	OPERATIVO YPF prevé la puesta en operación del revamp de la planta de tratamiento de gas Loma La Lata durante el segundo trimestre de 2026.	Ver fuente
03/06/2026	BMA	DIVIDENDO Banco Macro puso a disposición la segunda cuota de dividendo en efectivo, en cumplimiento de la Comunicación A8410 del BCRA.	Ver fuente
14/05/2026	PAMP	DEUDA Pampa Energía reabrió su bono internacional 2037 por US\$500M con un spread de 315bps, récord en su historia.	Ver fuente
13/05/2026	CEPU	RESULTADOS Central Puerto reportó Q1 2026 con ingresos por US\$249M, EBITDA de US\$120M y ganancia neta de US\$142M.	Ver fuente
13/05/2026	TGSU2	REGULATORIO El Gobierno aprobó el ingreso de TGS al RIGI para ampliar el Gasoducto Perito Moreno por US\$550M.	Ver fuente
12/05/2026	BMA, BBAR	CALIFICACIÓN Fitch Ratings mejoró la calificación de los bancos Macro y BBVA tras la suba soberana a B-.	Ver fuente
07/05/2026	YPFD	RESULTADOS YPF reportó Q1 2026 con EBITDA récord de US\$1.594M y superó los 200.000 barriles diarios de shale.	Ver fuente
06/05/2026	SUPV	RESULTADOS Grupo Supervielle reportó pérdida neta de ARS\$17,1B en Q1 2026, NPL en 5,6% y un plan de reducción de personal del 9%.	Ver fuente
06/05/2026	TGSU2	RESULTADOS Transportadora de Gas del Sur reportó Q1 2026 con utilidad neta +12,5% interanual; el negocio no regulado ya representa el 54% de su EBITDA.	Ver fuente
05/05/2026	LOMA	RESULTADOS Loma Negra reportó Q1 2026 con EBITDA en dólares +11,5% interanual, utilidad neta +44% y suba de la acción del 6,7%.	Ver fuente
31/03/2026	EDN	REGULATORIO El ENRE oficializó un aumento tarifario de Edenor del 2,04% vigente desde el 1 de abril.	Ver fuente
22/03/2026	GGAL, BMA, BBAR, SUPV	REGULATORIO El BCRA, mediante la Comunicación A8410, autorizó a los bancos a distribuir el 60% de las utilidades 2025 en tres cuotas mensuales desde mayo.	Ver fuente
05/03/2026	GGAL	RESULTADOS Grupo Galicia reportó pérdida neta de ARS\$83.544M en Q4 2025 y morosidad en 8,2%.	Ver fuente
10/02/2026	CRES	RESULTADOS Cresud reportó una ganancia de ARS\$193.932M en el primer semestre fiscal 2026 (revertió la pérdida del año previo), explicada principalmente por los resultados de IRSA.	Ver fuente
03/02/2026	IRSA	RESULTADOS IRSA presentó su balance semestral del ejercicio fiscal 2026, abriendo la temporada de resultados de las empresas argentinas.	Ver fuente
15/01/2026	TECO2	DEUDA Telecom Argentina colocó Obligaciones Negociables por US\$600M a 2036 en el mercado internacional.	Ver fuente

Nota. Elaboración propia. Vista alimentada por fundamentals.json.

II.2 Esquema de los archivos JSON

El pipeline genera 32 archivos JSON, 6 globales (uno por corrida) y 26 por serie (13 acciones en sus dos mercados). El sitio los lee tal como se emiten. Las señales se guardan como texto y los valores numéricos usan fracciones ($0.01 = 1\%$). Cada estructura se describe abajo con un fragmento representativo. Las dos de mayor complejidad, el resumen por serie y el archivo por serie, se detallan además campo por campo en las Tablas 8 y 9.

Archivos globales

_meta.json — objeto único con las fechas del estudio, el universo y los parámetros del backtest.

```
{ "pipeline_run_date": "2026-07-02T22:06:21+00:00",
  "study_start_date": "2020-09-14", "study_cutoff_date": "2024-08-30",
  "study_end_date": "2026-05-15", "warmup_end_date": "2021-02-10",
  "tickers_count": 26,
  "methods": ["HMA16", "EMA_12_26", "SMA_10_50_100", "B&H"],
  "cost_per_trade": 0.01 }
```

summary.json — arreglo de 26 entradas, una por serie, con la foto del último día y las métricas por método. Su estructura se detalla en la Tabla 8.

```
{ "ticker": "PAMP_BA", "mercado": "BA", "last_date": "2026-07-01",
  "signals": { "HMA16": "Señal Bajista", "EMA_12_26": "Señal Alcista",
    "SMA_10_50_100": "Señal Alcista" },
  "vma20_cat": "Bajo", "candle": "Engulfing_Baj",
  "metrics": { "HMA16": { "total": { "n_trades": 91, "win_rate": 0.428,
    "cumulative_return": 1.006, "max_drawdown": -0.402 },
    "is": { ... }, "oos": { ... } },
    "EMA_12_26": { ... }, "SMA_10_50_100": { ... }, "B&H": { ... } } }
```

validators.json — arreglo de 78 entradas (26 series por 3 métodos). Mide si el volumen y la vela mejoran el desempeño, mediante los conteos y retornos de cada grupo (confirmado / no confirmado, alineado / no alineado / neutral).

local_vs_mep.json — arreglo de 13 entradas, una por par de acciones. Compara la señal de la serie local con su equivalente MEP. Contiene el coeficiente de correlación de Pearson redondeado a seis decimales.

```
{ "local_ticker": "BBAR_BA", "mep_ticker": "BBAR_MEP",
  "correlation": 0.688634, "avg_lag": 0.091937765205 }
```

fundamentals.json — arreglo de eventos macroeconómicos y sectoriales, curados a mano. El proceso de curación se detalla en el Anexo de operación.

```
{ "id": 1, "fecha": "2025-10-26",
  "tickers_afectados": ["MERCADO"], "sector": "",
  "fuente": "https://www.boletinoficial.gob.ar/detalleAviso/primera/325597/20250520",
  "evento": "Elecciones legislativas nacionales", "tipo": "eleccion" }
```

daily_panel.json — objeto indexado por fecha. Cada clave es una rueda y su valor es el arreglo de las 26 series con su estado ese día. La estructura de cada entrada es la foto de summary.json sin las métricas.

```
"2026-07-01": [
  { "ticker": "BBAR_BA", "mercado": "BA",
    "signals": { "HMA16": "Señal Bajista", "EMA_12_26": "Señal Alcista",
      "SMA_10_50_100": "Señal Alcista" },
    "vma20_cat": "Muy Bajo", "candle": null, "last_date": "2026-07-01" }
]
```

Archivos por serie

ticker_{TICKER}.json — un archivo por serie (26 en total). Contiene la serie temporal con precios, indicadores y señales, más el detalle de las operaciones por método. Alimenta el gráfico de detalle. Su estructura se detalla en la Tabla 9.

```
{ "ticker": "GGAL_BA",
  "ohlcv": [ { "time": "2026-07-01", "open": 7810.0, "high": 7900.0,
    "low": 7510.0, "close": 7685.0, "volume": 17359713990,
    "hma16": 7667.19, "ema12": 7813.15, "ema26": 7617.16,
    "sma10": 7919.0, "sma50": 7065.9, "sma100": 6941.35,
    "vma20_ratio": 0.65, "vma20_cat": "Muy Bajo",
    "T_hma16": "Señal Bajista", "T_ema12_26": "Señal Alcista",
    "T_sma10_50_100": "Señal Alcista", "candle": null } ],
  "trades": { "HMA16": [ { "entry_date": "2020-10-02", "entry_price": 108.0,
    "exit_date": "2020-10-08", "exit_price": 107.55, "return": -0.0141,
    "holding_days": 4, "entry_vma20": 0.6, "vma20_confirm": null,
    "entry_candle_pattern": null, "candle_aligned": null,
    "is_in_sample": true } ],
    "EMA_12_26": [ ... ], "SMA_10_50_100": [ ... ] } }
```

Tabla 8*Esquema de summary.json (resumen por serie)*

Campo	Tipo	Representa
ticker	date	Serie
mercado	str null	Mercado: BA o MEP
last_date	str null	Última rueda de la serie
signals.{método}	str null	Señal vigente del último día por método
vma20_cat	str null	Categoría de volumen del último día
candle	str null	Patrón de vela del último día
metrics.{método}. {total is oos}.n_trades	int	Cantidad de operaciones
metrics.....win_rate	dec null	Proporción de operaciones ganadoras (0–1)
metrics.....cumulative_return	dec null	Retorno acumulado (fracción)
metrics.....max_drawdown	dec null	Máxima caída (fracción, ≤ 0)

Nota. Elaboración propia. Los sufijos is y oos corresponden a las particiones dentro y fuera de muestra.

Tabla 9*Esquema de ticker_{TICKER}.json (serie completa)*

Campo	Tipo	Representa
ticker	string	Serie
ohlcv	array	Serie temporal diaria
ohlcv[].time	date	Fecha de la barra
ohlcv[].open/high/low/close	dec null	Precios de la rueda
ohlcv[].volume	int null	Volumen operado
ohlcv[].hma16 ... sma100	dec null	Indicadores técnicos
ohlcv[].vma20_ratio	dec null	Relación volumen sobre su media de 20
ohlcv[].vma20_cat	str null	Categoría de volumen
ohlcv[].T_{método}	str null	Señal por método ese día
ohlcv[].candle	str null	Patrón de vela
trades.{método}	array	Operaciones del método (B&H no)
trades[].entry_date / exit_date	date	Fechas de entrada y salida
trades[].entry_price / exit_price	dec	Precios de entrada y salida (apertura i+1)
trades[].return	dec null	Retorno neto de la operación (con costo)
trades[].holding_days	int	Días de tenencia
trades[].entry_vma20	dec null	Volumen en la barra de la señal
trades[].vma20_confirm	bool null	Volumen sobre la mediana expandida
trades[].entry_candle_pattern	str null	Patrón de vela en la señal
trades[].candle_aligned	bool null	Vela alineada con la dirección
trades[].is_in_sample	bool null	Operación dentro o fuera de muestra

Nota. Elaboración propia. El campo is_in_sample separa la partición de validación.

Anexo III. Manual de operación y reproducción

Este anexo reúne tres bloques dirigidos a un lector técnico que clona el repositorio, el manual de operación del pipeline, el gobierno del panel de eventos y el protocolo de verificación del estado congelado. El manual describe cómo correr y mantener la plataforma. El gobierno documenta cómo se administra la única fuente editada a mano del sistema. El protocolo especifica cómo un tercero confirma que trabaja sobre los mismos datos que este trabajo reporta.

III.1 Manual de operación

III.1.1 Ejecución del pipeline

No existe un script orquestador único. El orden lo define el flujo de integración continua, que ejecuta los siete módulos como comandos independientes, en secuencia, desde la raíz del repositorio. Cada módulo consume la salida del anterior.

<code>python -m src.ingestion.data_downloader</code>	<code># 1. Ingesta PPI</code>	<code>-> data_raw/csv/</code>
<code>python -m src.processing.data_normalizer</code>	<code># 2. Normalización</code>	<code>-> data_normalized/csv/</code>
<code>python -m src.processing.synthetic_mep_generator</code>	<code># 3. Series MEP</code>	<code>-> data_normalized/csv/</code>
<code>(*_MEP)</code>		
<code>python -m src.analysis.indicator_engine</code>	<code># 4. Indicadores</code>	<code>-> data_indicators/csv/</code>
<code>python -m src.analysis.signal_generator</code>	<code># 5. Señales</code>	<code>-> data_signals/csv/</code>
<code>python -m src.analysis.backtester</code>	<code># 6. Backtest + métricas</code>	<code>-> data_public/backtests/</code>
<code>python -m src.web.json_generator</code>	<code># 7. JSON del frontend</code>	<code>-> docs/data/</code>

Nota. El módulo 7 se declara autónomo en su documentación interna, pero el flujo lo invoca igual como paso final. En una corrida manual conviene ejecutarlo siempre al cierre para regenerar el directorio del sitio.

III.1.2 Dependencias e instalación

La versión de referencia del intérprete es Python 3.11, la misma que fija la integración continua. La instalación es estándar, sin pasos de compilación ni servicios externos: un entorno virtual y las dependencias con versión fijada.

```
python -m venv .venv
source .venv/bin/activate          # Windows: .venv\Scripts\activate
pip install -r requirements.txt
```

Contenido íntegro del requirements.txt, con todas las versiones fijadas:

```
numpy==2.3.4
pandas==2.3.3
ppi_client==1.2.4
python-dotenv==1.2.1
signalrcorePPI==0.9.2
```

III.1.3 Credenciales

Solo el módulo de ingesta requiere credenciales, porque es el único que consume una API externa. Las claves nunca se versionan: en local se cargan desde un archivo .env ignorado por Git, y en la integración continua se inyectan como secretos del repositorio.

Tabla 10

Variables de entorno requeridas por el módulo de ingesta

Variable	Para qué sirve
PPI_PUBLIC_KEY	Clave pública de la API de Portfolio Personal Inversiones. Autentica el cliente de datos de mercado.
PPI_PRIVATE_KEY	Clave privada de PPI, par de la anterior en el login del cliente.

Nota. Elaboración propia. El archivo .env no está versionado; no hay exposición de credenciales en el repositorio.

III.1.4 Disparo de una corrida

El flujo se dispara de dos formas. De manera programada, un cron lo ejecuta los días hábiles a las 21:00 UTC (18:00 de Argentina). De manera manual, bajo demanda. La corrida opera sobre la rama principal; si detecta cambios, el bot los publica y eso vuelve a desplegar el sitio. En local, la corrida es siempre manual. El commit es condicional: si no hay cambios, la corrida termina sin tocar el repositorio.

III.1.5 Recuperación ante fallas

El pipeline aísla los errores por unidad de trabajo. Cada módulo captura las fallas por ticker o por archivo y continúa con el resto, sin abortar la corrida. La ingesta además protege las descargas vacías de un día no hábil, que solo se registran sin modificar los archivos existentes. Como cada módulo escribe en su propio directorio, una falla en un ticker no corrompe los resultados ya escritos.

El flujo no define reintentos automáticos: si un paso falla de forma global, la corrida falla y la recuperación es volver a dispararla, de forma manual o en la corrida programada del día siguiente. El diseño incremental de la ingesta favorece esa recuperación, porque la corrida siguiente descarga por sí sola las ruedas pendientes.

III.2 Procedimiento de curaduría del panel de eventos

El panel de eventos fundamentales es la única fuente del sistema editada a mano. Todo el resto del pipeline es determinista y se deriva de los datos de mercado. El panel, en cambio, depende de un archivo curado manualmente. Esa diferencia obliga a un gobierno explícito, documentado aquí en su procedimiento, con su fundamento en la sección 5.5.4.

III.2.1 Alcance y encuadre

El panel cumple un rol secundario, no analítico. No alimenta el backtest ni la generación de señales, es una capa de señalización desacoplada del cómputo. Esa separación es deliberada y arrastra una consecuencia metodológica. El panel marca fechas sensibles que se pueden conocer antes de que el precio se mueva. Algunas están agendadas con tiempo, como un comicio o el día en que una empresa publica su balance. Otras son hechos que pasaron con el mercado ya cerrado, y ahí la ventana llega hasta la apertura siguiente. Nunca se usa información futura. Cada marca apunta a una fuente que ya existía antes del hecho, o en el mismo momento, y no a una nota escrita después, el decreto que convoca la elección, el aviso que anticipa cuándo sale el balance, una medida difundida fuera de hora. Que el panel quede afuera del backtest no responde a un sesgo por neutralizar. Es que no es una señal de trading a la que se le mida poder predictivo. Es una capa de aviso para el usuario y no entra en el cómputo, así que ninguna métrica de este trabajo se apoya en qué eventos se cargaron.

El feed no está balanceado, y conviene dejarlo dicho. La mitad de las entradas son de alcance mercado. El resto se reparte de manera despareja entre las empresas, algunas aparecen dos veces y bastantes una sola, cuando no ninguna. El grueso de esa cola es bancario, y pesa sobre todo un vencimiento soberano que toca a varios bancos en la misma fecha. Filtrar el panel por un activo puntual devuelve entonces pocas entradas, y por algunos casi ninguna. No es una falla de la interfaz. El feed refleja un mercado donde los hechos que mueven a estos papeles son en su mayoría macroeconómicos y afectan al conjunto, algo consistente con lo que se discute en 6.3. Esa densidad baja por empresa es un rasgo de cobertura del feed curado, y el panel sigue desacoplado del backtest, sin ninguna métrica que dependa de él. Su ampliación se plantea en 7.5.

III.2.2 Fuente y flujo de edición

La fuente editable es un archivo JSON crudo que el módulo de serialización procesa y publica. Se edita siempre el archivo de origen, nunca la salida. El flujo tiene tres pasos, editar el origen, volver a ejecutar el módulo de serialización y verificar la salida publicada.

Cada evento del archivo de origen tiene la siguiente estructura, con los campos que se cargan a mano:

```
{
  "fecha": "2025-10-26",
  "tickers_afectados": ["MERCADO"],
  "tipo": "eleccion",
  "evento": "Elecciones legislativas nacionales",
  "fuente": "https://www.boletinoficial.gob.ar/detalleAviso/primera/325597/20250520"
}
```

Tabla 11

Campos de un evento en el archivo de origen

Campo	Criterio de carga
fecha	Fecha del evento señalado (comicio, publicación de balance, vencimiento o shock fuera de horario), en formato AAAA-MM-DD.
tickers_afectados	Tickers afectados, sin sufijo de mercado. Se usa MERCADO si el evento es transversal.
tipo	Categoría del evento (por ejemplo, macro, resultados, regulatorio).
evento	Descripción textual del hecho, en términos objetivos y sin juicio de impacto.
fuelle	URL o referencia de origen. Si se omite, el generador la fija como manual.

Nota. El identificador secuencial y el sector se agregan de forma automática; no se cargan a mano.

III.2.3 Criterio de admisión y auditoría

Para que un evento entre tiene que cumplir cuatro cosas. Su fecha se tiene que poder conocer antes de que el precio reaccione, ya sea porque estaba agendada de antemano o porque el hecho pasó fuera del horario de mercado y dejó margen hasta la apertura. Tiene que ser verificable contra una fuente que ya estuviera disponible en ese momento, no una nota posterior. Tiene que ser pertinente para los activos del universo. Y la descripción tiene que ser objetiva, sin adelantar qué efecto va a tener sobre el precio. Registrar la dirección esperada lo convertiría en un pronóstico, que es justo lo que se evita al no rotular impacto. La auditoría es directa. Como cada evento conserva su fuente, un tercero puede rastrear y contrastar cada entrada contra su origen. El detalle del criterio y su fundamento metodológico están en la sección 5.5.4.

III.2.4 Extensión futura

La curación manual es sostenible para el volumen actual y suficiente para el rol demostrativo del panel. Su automatización desde fuentes oficiales queda planteada como línea de trabajo futuro, no como componente de esta entrega. Esa evolución se describe en la sección 7.5.

III.3 Protocolo de verificación del estado congelado

El protocolo permite a un tercero confirmar que trabaja sobre los mismos datos que este trabajo reporta. La verificación es de tipo archivístico, comprueba que los insumos son idénticos a los sellados, no que un cómputo se regenere bit a bit. Esta distinción es deliberada y se explica en la sección 5.4.

III.3.1 Qué garantiza y qué no

El hash SHA256 sella los datos crudos, que son la entrada de todo el pipeline. Si el hash coincide, se garantiza que los insumos son exactamente los que produjeron los resultados. Lo que el hash no hace es certificar por sí mismo la salida final. La equivalencia de resultados se sostiene sobre el determinismo del pipeline y las versiones fijadas de las dependencias, no sobre el hash. El hash prueba el punto de partida, y la reproducibilidad del recorrido descansa en el entorno pinneado.

III.3.2 Procedimiento

1. Posicionarse en el estado congelado mediante el tag.
2. Recalcular el hash de los datos crudos con la misma receta y compararlo contra el valor sellado en el mensaje del tag.
3. Si coincide, regenerar los resultados desde ese estado ejecutando los pasos finales del pipeline.

Paso 1, posicionamiento. El repositorio público es <https://github.com/sebams81/analisis-tecnico-arg> y el sitio publicado, <https://sebams81.github.io/analisis-tecnico-arg/>. Sobre un clon nuevo: `git clone https://github.com/sebams81/analisis-tecnico-arg.git`

El mismo estado se encuentra depositado en Zenodo con el identificador persistente <https://doi.org/10.5281/zenodo.21288495>, alternativa de descarga directa si el repositorio de origen no estuviera disponible

```
cd analisis-tecnico-arg
git fetch --tags origin
git checkout v1.1-tf # estado congelado: 26 series, datos hasta 2026-05-15
```

Paso 2, verificación del hash. El *digest* se calcula sobre el árbol de trabajo materializado, de modo que el resultado depende de la conversión de fin de línea vigente durante el checkout. El repositorio no incluye `.gitattributes` en este tag, por lo que la conversión queda determinada por la configuración de quien clona y debe fijarse de forma explícita antes de comparar.

```
git config core.autocrlf true # materialización CRLF, la del digest sellado
git checkout v1.1-tf
git tag -n99 v1.1-tf # muestra el hash esperado en el mensaje
git ls-files data_raw | LC_ALL=C sort | xargs sha256sum | sha256sum
# esperado: 46d5f326d4caf6cee4dd0e1fb7171093c2502146a94c58b0643b9425b6f46988 (29 archivos)
```

El mismo estado, materializado con fin de línea LF (`core.autocrlf false`), arroja `b45c463abdf3bb04528f5815364fbcd6165f2782b51c48c74578d3e5da3fbc91`. Ambos digests corresponden a idénticos objetos en la base de datos de Git y difieren solo en la representación en disco. Verifica quien reproduce cualquiera de los dos valores bajo la configuración correspondiente.

Paso 3, regeneración de resultados. No hace falta volver a descargar. Los datos crudos están versionados en el tag, de modo que la cadena se rehace desde ellos ejecutando los pasos 2 a 7. Las capas intermedias son derivados regenerables y no se versionan, así que el paso 2 de normalización y el 3 de construcción de las series MEP forman parte de la regeneración.

```
python -m src.processing.data_normalizer # paso 2
python -m src.processing.synthetic_mep_generator # paso 3
python -m src.analysis.indicator_engine # paso 4
python -m src.analysis.signal_generator # paso 5
python -m src.analysis.backtester # paso 6
python -m src.web.json_generator # paso 7
```

III.3.3 Advertencia sobre fin de línea

El hash es byte-exacto, de manera que registra los caracteres de fin de línea tal como quedan escritos en disco. Git almacena los blobs normalizados a LF y los convierte al materializar el checkout, así que la configuración de `core.autocrlf` determina cuál de los dos digests se obtiene sobre el mismo estado congelado. La verificación exige fijar esa configuración antes del checkout y contrastar contra el digest que le corresponde, según la equivalencia declarada en III.3.2. El cálculo del SHA256 requiere un entorno con utilidades POSIX, como Git Bash o WSL. El texto plano versionado hace el hash estable y auditable, y ese mismo formato deja el archivo expuesto a la normalización de línea del sistema operativo. La sensibilidad al fin de línea es la contraparte de esa elección. Hay otra precaución, esta vez de orden. El script de validación contra los ADR regenera el manifiesto de fuentes alojado en el directorio de datos crudos, así que la comprobación del digest se hace antes de ejecutar cualquier corrida, según la secuencia de pasos ya establecida.

Referencias

- AnalisisTecnico.com.ar. (s. f.). *Herramientas profesionales de análisis técnico para el mercado argentino*. Recuperado el 5 de julio de 2026, de <https://analisistecnico.com.ar>
- Auguste, S., Dominguez, K. M. E., Kamil, H., & Tesar, L. L. (2006). Cross-border trading as a mechanism for implicit capital flight: ADRs and the Argentine crisis. *Journal of Monetary Economics*, 53(7), 1259–1295.
<https://doi.org/10.1016/j.jmoneco.2005.05.012>
- Bailey, D. H., Borwein, J. M., López de Prado, M., & Zhu, Q. J. (2015). *The probability of backtest overfitting* [Versión revisada].
<https://www.davidhbailey.com/dhbpapers/backtest-prob.pdf>
- Berk, J., & DeMarzo, P. (2008). *Finanzas corporativas*. (J. Enríquez Brito, Trad.). Pearson Educación.
- Comisión Nacional de Valores. (15 de abril de 2025). Resolución General N° 1062/2025: Eliminación del plazo mínimo de tenencia para operaciones de valores negociables aplicable a personas humanas residentes.
<https://www.boletinoficial.gob.ar/detalleAviso/primera/324045/20250415>
- Deep, G., Deep, A., & Lamptey, W. (2025). *Interpretable hypothesis-driven trading: A rigorous walk-forward validation framework for market microstructure signals*. arXiv. <https://arxiv.org/abs/2512.12924>
- Ellareddy, S. (2025). *Fast vs. slow: Evaluating short- and long-term moving average strategies on SPY*. SSRN.
https://papers.ssrn.com/sol3/papers.cfm?abstract_id=5430117
- Erpen, M. (2010). *Mercado de capitales: Manual para no especialistas*. Temas Grupo Editorial / Instituto Argentino de Mercado de Capitales.
- Hull, A. (2005). *How to reduce lag in a moving average*. <https://alanhull.com/hull-moving-average>
- Hurst, B., Ooi, Y. H., & Pedersen, L. H. (2017). A century of evidence on trend-following investing. *The Journal of Portfolio Management*, 44(1), 15–29.
<https://doi.org/10.3905/jpm.2017.44.1.015>
- Jiménez-Méndez, E. R., & Álvarez-Lamus, G. K. (2024). Estrategias de trading para el bono del Tesoro estadounidense usando análisis técnico y Bloomberg Professional Services. *Revista CEA*, 10(22), e2634. <https://doi.org/10.22430/24223182.2634>

- Ley 26.831 de 2012. Mercado de Capitales. 27 de diciembre de 2012. Boletín Oficial N° 32.552. <https://servicios.infoleg.gob.ar/infolegInternet/anexos/205000-209999/206592/norma.htm>
- Lo, A. W., Mamaysky, H., & Wang, J. (2000). Foundations of technical analysis: Computational algorithms, statistical inference, and empirical implementation. *The Journal of Finance*, 55(4), 1705–1765. <https://doi.org/10.1111/0022-1082.00265>
- López de Prado, M. (2018). *Advances in financial machine learning*. Wiley.
- Matba Rofex. (12 de julio de 2024). *CCL-MtR — Metodología*. <https://www.matbarofex.com.ar/MetodologiaIndiceCCLMtR>
- Murphy, J. J. (2000). *Análisis técnico de los mercados financieros*. Gestión 2000.
- Ospina-Holguín, J. H., & Padilla-Ospina, A. M. (2025). Reinforcement learning meets technical analysis: Combining moving average rules for optimal alpha. *Cogent Economics & Finance*, 13(1). <https://doi.org/10.1080/23322039.2025.2490818>
- Pasquariello, P. (2008). The anatomy of financial crises: Evidence from the emerging ADR market. *Journal of International Economics*, 76(2), 193–207. <https://doi.org/10.1016/j.jinteco.2008.04.001>
- Riofrio López, L. E., & Pinargote, V. J. (2025). Desarrollo de una plataforma web para realizar backtesting de estrategias de inversión en el mercado de valores. *Ciencia Latina Revista Científica Multidisciplinar*, 9(5), 433–456. https://doi.org/10.37811/cl_rcm.v9i5.19682
- Sánchez, M. S. (2024). Cuando las inversiones se popularizan. Finanzas digitales e inversores amateurs en Argentina. *Estudios Sociológicos de El Colegio de México*, 42, e2531. <https://doi.org/10.24201/es.2024v42.e2531>
- Sarasa-Cabezuelo, A. (2023). Development of a backtesting web application for the definition of investment strategies. *Knowledge*, 3(3), 414–431. <https://doi.org/10.3390/knowledge3030028>
- Solis Arellano, M. S. (2025). *Sistema automatizado de adquisición y análisis dinámico de datos para el soporte a decisiones en criptomercados* [Tesis de maestría, CETYS Universidad]. <https://repositorio.cetys.mx/bitstream/60000/1942/1/Tesis-Selene-Solis.pdf>
- Vintimilla, F. (2021). *Relación Merval en USD y riesgo país: Correlación y testeo de una estrategia de trading* [Trabajo final de maestría, Universidad del CEMA]. <https://ucema.edu.ar/sites/default/files/2023-07/Vintimilla%20Federico%20-%20Tesina%20Final%20-%20MAF%202021.pdf>