TRADING ALGORÍTMICO COMO HERRAMIENTA PARA MITIGAR LOS SESGOS COGNITIVOS: EXCESO DE CONFIANZA, AVERSIÓN A LA PERDIDA, CONFIRMACIÓN E ILUSIÓN DE CONTROL EN LA TOMA DE DECISIONES FINANCIERAS.

MARIA JOSE MORALES ROSERO

JUAN FELIPE VERA PAZOS

PONTIFICIA UNIVERSIDAD JAVERIANA

FACULTAD DE CIENCIAS ECONÓMICAS Y ADMINISTRATIVAS

PROGRAMA DE FINANZAS

SANTIAGO DE CALI

2025

TRADING ALGORÍTMICO COMO HERRAMIENTA PARA MITIGAR LOS SESGOS COGNITIVOS: EXCESO DE CONFIANZA, AVERSIÓN A LA PERDIDA, CONFIRMACIÓN E ILUSIÓN DE CONTROL EN LA TOMA DE DECISIONES FINANCIERAS.

MARIA JOSE MORALES ROSERO

JUAN FELIPE VERA PAZOS

Trabajo de grado presentado como requisito parcial para optar por el título de Financiero

DIRECTOR: VICTOR ALBERTO PEÑA VARGAS

DOCTORADO EN CIENCIAS ECONÓMICAS

PONTIFICIA UNIVERSIDAD JAVERIANA

FACULTAD DE CIENCIAS ECONÓMICAS Y ADMINISTRATIVAS

PROGRAMA DE FINANZAS

SANTIAGO DE CALI

2025

ARTÍCULO 23 de la resolución N° 13 de julio 6 de 1946

“La Universidad no se hace responsable por los conceptos emitidos por sus alumnos en sus trabajos de Tesis. Sólo velará porque no se publique nada contrario al dogma y a la moral católica y porque la Tesis no contenga ataques o polémicas puramente personales; antes bien, se vea en ellas al anhelo de buscar la Verdad y la Justicia”.

"Trading Algorítmico como Herramienta para Mitigar los Sesgos Cognitivos en la Toma de Decisiones Financieras"

Aprobado por el Comité de Trabajos de Grado en cumplimiento de los requisitos exigidos por la Pontificia Universidad Javeriana para optar por el título de Profesional en Finanzas

Fabian Osorio

Firmado digitalmente por Fabian Osorio
Fecha: 2025.06.05 08:54:36 -05'00'

FABIÁN FERNANDO OSORIO TINOCO
Decano
Facultad de Ciencias Económicas y Administrativas

Diego Alejandro Peláez Cadena

Firmado digitalmente por Diego Alejandro Peláez Cadena
Fecha: 2025.06.05 08:11:19 -05'00'

DIEGO ALEJANDRO PELÁEZ
Director
Programa de Finanzas

Victor Alberto Peña Vargas

Firmado digitalmente por Victor Alberto Peña Vargas
Fecha: 2025.06.03 08:24:49 -05'00'

VÍCTOR PEÑA
Director (a) del trabajo

ROGELIO LADRÓN DE GUEVARA CORTÉS
Jurado

Cali, 28 de mayo de 2025

Contenido

Resumen 11

Introducción 13

El proceso cognitivo en la toma de decisiones financieras 14

Presentación de los sesgos y limitaciones cognitivas 14

Finanzas conductuales y la toma de decisiones 15

Estrategias de *debiasing* y mitigación: el trading algorítmico 16

Problema de investigación 19

Formulación del Interrogante 20

Pregunta general 20

Objetivo General 20

Justificación 21

Marco teórico 23

Herbert Simon: Racionalidad Limitada Satisficing 23

Heurísticas Y Sesgos Cognitivos Daniel Kahneman, Amos Tversky. 24

Richard Thaler Economía Conductual 25

Robert Shiller: Psicología en los Mercados Financieros 26

Eugene Fama: Teoría de Mercados Perfectamente Eficientes 26

Trading Algorítmico 27

Principales algoritmos en la historia y actualidad del trading algorítmico 27

Metodología 30
Enfoque Metodológico 30
Herramientas Tecnológicas 31
Bibliotecas adicionales 33
Diseño del Experimento 34
Estrategia de Trading Algorítmico (Reglas Objetivas e Indicadores Técnicos) 34
Mitigación de Sesgos Cognitivos mediante Reglas del Algoritmo 35
Validación mediante Backtesting Automatizado 38
Procedimiento 40
Desarrollo del Algoritmo y Configuración de la Estrategia 40
Recolección de Datos 44
Fuente y tipo de datos 44
Periodo de estudio 45
Selección de instrumentos financieros 45
Procesamiento y limpieza de datos 46
Seguridad y reproducibilidad de los datos 46
Análisis de Datos 47
Técnicas adicionales de análisis 52
Criterios de Validez 53
Validez de Constructor 53

Validez Interna 54

Validez Externa 56

Confiabilidad (fiabilidad) y replicabilidad 57

Triangulación de métodos 58

Resultados 60

Contexto del experimento 60

Evidencia empírica 60

Corridas con ATR 60

Corridas sin ATR 61

Inserción de resultados visibles 62

Conclusiones 65

Referencias 68

Anexos 72

LISTA DE TABLAS

Tabla 1. Principales autores seminales del corpus teórico de las Finanzas Conductuales……17

LISTA DE FIGURAS

Figura 1. *Nvidia ATR Periodo 1……62*

Figura 2. *Nvidia ATR Periodo 2……63*

Figura 3*. Nvidia ATR OUT periodo 1……63*

Figura 4*. Nvidia ATR OUT periodo 2……64*

LISTA DE ANEXOS

Anexo A. Líneas de código-librerías utilizadas……72

Anexo B. Líneas de código-Indicadores calculados…….73

Anexo C. Líneas de código-condiciones de compra y venta……74

Anexo D. Líneas de código para calcular el tamaño de la posición…….75

Resumen

Este proyecto investiga cómo la excesiva confianza en uno mismo influye en la toma de decisiones durante el trading intradía, especialmente en contextos de alta volatilidad en los mercados financieros. Se analiza la relación entre el sesgo cognitivo de exceso de confianza y el rendimiento económico de los traders, evaluando cómo este sesgo puede aumentar el riesgo de pérdidas.

A través de un enfoque empírico, este estudio busca ofrecer una visión más completa de los factores psicológicos que intervienen en la toma de decisiones en contextos de alta incertidumbre y volatilidad. Los resultados tienen importantes implicaciones prácticas tanto para la formación de traders como para el diseño de algoritmos de gestión sistemática del riesgo en las entidades financieras.

En este trabajo estudia cómo los sesgos cognitivos afectan las decisiones de trading intradía, especialmente en mercados financieros altamente volátiles. Varios sesgos relevantes entre ellos el sesgo de exceso de confianza, el sesgo por aversión a la pérdida, el sesgo de confirmación y la ilusión de control. Todos ellos influyen de manera significativa en el rendimiento financiero de los traders.

La excesiva confianza puede llevar a una sobreestimación de sus propias habilidades y una subestimación de los riesgos. Por otro lado, el sesgo de aversión a la pérdida se puede hacer que los inversionistas asuman riesgos innecesarios lo que conduciría a decisiones subóptimas. El sesgo de confirmación causa que los traders busquen confirmación para sus puntos de vista preexistentes, ignorando las ajenas a ellos. Por último, la ilusión de control de la hacen creer tener control sobre eventos nunca predecibles que puede aumentar el riesgo de sobre operación.

A través de un estudio empírico, este trabajo ofrece una visión más completa de la relación entre estos factores psicológicos con la forma en que éstos influyen en las decisiones de los inversionistas en espacios caracterizados por alta incertidumbre y volatilidad intensa como el trading intradía. Además, también se explica cómo estos sesgos pueden incrementar el riesgo de pérdidas financieras y dañar el rendimiento de los operadores en los mercados turbulentos.

Keywords: Trading Algorítmico, sesgos cognitivos, toma de decisiones.

Introducción

El trading intradía es una actividad económica que exige una toma de decisiones rápidas y en la que intervienen factores técnicos, emocionales y psicológicos. Desde la perspectiva de la de la psicología financiera, el sesgo de exceso de confianza es un fenómeno ampliamente documentado en el ámbito de las inversiones lleva a que los individuos sobreestimen sus habilidades y conocimientos, lo que puede derivar en decisiones de inversión cada más arriesgadas. Esta sobreconfianza tiene consecuencias profundas en ámbitos donde velocidad y precisión son fundamentales, como en el trading intradía donde la volatilidad del mercado puede potenciar la sobreconfianza y obtener resultados negativos.

Investigaciones pioneras como la de Tversky y Kahneman (1974), sobre heurísticas y sesgos, demostraron que las personas prefieren tomar las decisiones bajo la incertidumbre de la recurriendo a procesos automáticos, mediante lo que denominaron heurísticas o atajos mentales, que puede conducir a sesgos también denominados errores sistemáticos. Dentro de estos, el sesgo de exceso de confianza aparece uno de los más frecuentes en las decisiones financieras. En mercados financieros, donde predominan el riesgo y la incertidumbre este sesgo puede conducir a errores sistemáticos del juicio.

La toma de decisiones en el ámbito financiero constituye un proceso complejo en el que confluyen la evaluación racional de la información disponible y la influencia de factores psicológicos. Desde esta perspectiva, los individuos afrontan limitaciones cognitivas que impiden alcanzar decisiones plenamente óptimas. La interacción entre las emociones, la experiencia pasada y las condiciones cambiantes del mercado genera distorsiones sistemáticas del juicio conocidas como sesgos cognitivos.

El proceso cognitivo en la toma de decisiones financieras

La elección financiera depende tanto de elementos objetivos —como los datos cuantitativos del mercado— como de componentes subjetivos derivados de las características psicológicas y cognitivas del decisor. Herbert Simon (1955) señala que los agentes económicos operan bajo un marco de "racionalidad limitada", esto es, carecen de información perfecta y de tiempo ilimitado para procesarla, por lo que deben actuar con recursos cognitivos restringidos. Esta realidad resulta especialmente relevante para traders e inversionistas que operan en entornos volátiles, impredecibles y altamente competitivos, donde la información es incompleta y el tiempo para analizarla es escaso.

La racionalidad limitada conduce a la adopción de heurísticas —reglas simples de decisión que economizan esfuerzo cognitivo—. Aunque estas estrategias permiten agilizar el proceso, también pueden generar sesgos persistentes que desvían las decisiones de la óptima. El propio Simon describe este fenómeno mediante el concepto de *satisficing*: los individuos no procuran la solución perfecta, sino aquella que resulte "suficientemente buena" dadas las restricciones de tiempo, información y contexto (Simon, 1955). En consecuencia, comprender cómo estos atajos cognitivos influyen en la evaluación de riesgos y oportunidades se vuelve esencial para mejorar la calidad de las decisiones financieras.

Presentación de los sesgos y limitaciones cognitivas

Los sesgos cognitivos constituyen atajos mentales útiles en situaciones cotidianas; sin embargo, resultan perjudiciales cuando se trasladan a decisiones irreversibles, como la inversión en los mercados financieros. Entre los más estudiados figuran la aversión a la pérdida, el sesgo de

confirmación y la ilusión de control, fenómenos ampliamente documentados en la literatura de psicología financiera.

Aversión a la pérdida. Kahneman y Tversky (1979) demostraron que las pérdidas generan un impacto emocional mayor que las ganancias equivalentes, inclinando al inversor a evitar riesgos excesivos.

Sesgo de confirmación. Este sesgo lleva al individuo a buscar o interpretar la información de modo que confirme sus creencias previas, desestimando datos que las contradigan (Nickerson, 1998).

Ilusión de control. Según Langer (1975), los operadores tienden a sobreestimar su capacidad para influir en resultados futuros, asumiendo riesgos innecesarios.

Finanzas conductuales y la toma de decisiones

Las finanzas conductuales exploran cómo ciertos atajos mentales —esas herramientas cognitivas que todos tenemos— influyen en nuestras decisiones económicas, desafiando así la idea del *homo economicus*: ese agente racional, casi idealizado, que predomina en la teoría económica clásica. Los trabajos pioneros de Kahneman y Tversky (1974) dejaron algo claro y profundamente humano: nuestras elecciones financieras no siempre se basan en cálculos fríos o estrictamente lógicos. Muy por el contrario, suelen estar marcadas por sesgos y patrones de pensamiento automáticos.

Esto tiene implicaciones profundas. En los mercados, los inversionistas no siempre actúan como si se encontraran en los entornos perfectamente eficientes que Fama (1970) imaginó. Las emociones, las percepciones erróneas y los impulsos juegan su propio papel. Y cuando lo

hacen, pueden conducir a decisiones irracionales, provocando desviaciones en el comportamiento del mercado y generando ineficiencias que, lejos de ser excepcionales, son parte del día a día financiero.

Estrategias de *debiasing* y mitigación: el trading algorítmico

El debiasing, es decir, estrategias diseñadas para mitigar los sesgos cognitivos, busca reducir —o incluso eliminar— su influencia en la toma de decisiones. Dentro de este campo, el trading algorítmico emerge como una de las soluciones más eficaces. ¿La razón? Permite delegar las decisiones financieras en sistemas automatizados, guiados por reglas objetivas y libres de emociones.

Estos algoritmos se apoyan en modelos matemáticos complejos y técnicas de optimización, capaces de procesar enormes volúmenes de datos en tiempo real. Al hacerlo, minimizan significativamente la intervención de la intuición humana y sus inevitables fluctuaciones emocionales. De este modo, se logra sortear lo que Herbert Simon describió como "racionalidad limitada", al permitir un análisis exhaustivo de la información sin las restricciones cognitivas que condicionan a los decisores humanos.

Tal y como ocurre en otros campos, a quienes deben contribuir al estudio sistemático sobre sesgos cognitivos y sus implicaciones financieras cuanto menos:

Tabla 1. *Principales autores seminales del corpus teórico de las Finanzas Conductuales*

Autor	Año	Contribución Principal
Herbert Simon	1957	Introducción del concepto de racionalidad limitada y la teoría del satisficing.
Daniel Kahneman y Amos Tversky	1974	Desarrollaron la teoría de las heurísticas y los sesgos cognitivos, mostrando cómo influyen en las decisiones bajo incertidumbre.
Richard Thaler	1980	Innovador en la economía del comportamiento, analizando cómo los individuos toman decisiones subóptimas por influencias cognitivas.
Robert Shiller	1981	Estudio de las emociones y su impacto en los mercados financieros, desafiando la hipótesis de los mercados eficientes.
Eugene Fama	1970	Proponente de la hipótesis de los mercados eficientes, en contraposición con las finanzas conductuales.

Fuente: Elaboración propia con base en los autores citados (Simon, 1957; Kahneman & Tversky, 1974; Thaler, 1980; Shiller, 1981; Fama, 1970

Comprender los sesgos cognitivos y su incidencia en la toma de decisiones financieras, empresariales y corporativas se convierte en un requisito fundamental para diseñar métodos y estrategias de inversión que mitiguen los riesgos derivados de decisiones impulsivas. Dentro de este contexto, el *trading* algorítmico se erige como la alternativa más prometedora para neutralizar tales sesgos y favorecer decisiones plenamente informadas. El presente trabajo analiza la relación

entre los sesgos cognitivos y el *trading* algorítmico, con el propósito de evidenciar cómo las herramientas tecnológicas emergentes pueden transformar de manera significativa la dinámica de los mercados financieros.

Problema de investigación

En los mercados financieros —y de manera especialmente marcada en el trading intradía— el sesgo de exceso de confianza se convierte en una trampa tan común como peligrosa. Los operadores, obligados a decidir con inmediatez dentro de entornos extremadamente volátiles, sobrestiman sus habilidades analíticas y minimizan la percepción del riesgo. Esa presunción los lleva a reaccionar de forma exagerada ante noticias o datos recientes, de modo que ejecutan operaciones más frecuentes y adoptan posiciones desproporcionadamente arriesgadas sin el respaldo de un análisis plenamente racional.

Simultáneamente, intervienen otras distorsiones cognitivas que **refuerzan** la sobreconfianza, consolidando un patrón de comportamiento difícil de evitar. Entre las más significativas se encuentran:

Sesgo de aversión a la pérdida. Los *traders* tienden a evitar las pérdidas casi a toda costa. Esta aversión emocional, puede llevarlos a tomar decisiones apresuradas o poco objetivas cuando se enfrentan a escenarios con resultados negativos. En su afán por revertir la situación, actúan sin una base analítica sólida, movidos más por el deseo de no perder que por una evaluación razonada del contexto.

Sesgo de confirmación. A menudo, los operadores buscan de manera activa información que respalde sus creencias o predicciones previas, mientras ignoran —o minimizan— cualquier dato que las contradiga. Esta inclinación cognitiva distorsiona el juicio, reforzando convicciones que pueden estar equivocadas y alimentando una confianza que, aunque firme, no siempre es justificada.

Ilusión de control. No es raro que los *traders* sobrestimen su capacidad para anticipar o influir en los movimientos del mercado. Esta percepción errónea de dominio sobre variables en realidad incontrolables puede llevarlos a asumir riesgos innecesarios, confiando en una sensación de control que, en la práctica, es más ilusoria que real.

El trading algorítmico puede ser un instrumento útil para neutralizar estos sesgos cognitivos. Al delegar las decisiones a algoritmos basados en reglas predefinidas y análisis de datos objetivos, reduciendo significativamente la influencia del factor sesgos emociones y distorsiones cognitivo de los humanos. Este enfoque no solo favorece decisiones más estables y fundamentadas, sino que también ayuda a minimizar comportamientos como operaciones innecesarias o el exceso de exposición al riesgo, especialmente en mercados altamente volátiles y cambiantes.

Formulación del Interrogante

Pregunta general

¿Cuál es el grado de influencia que ejercen los sesgos cognitivos sesgos cognitivos como el exceso de confianza, la aversión a la pérdida, el sesgo de confirmación y la ilusión de control en la toma de decisiones de *trading* intradía en mercados financieros volátiles, y hasta qué punto el *trading* algorítmico puede mitigar estos efectos?

Objetivo General

Evaluar el impacto del trading algorítmico como herramienta para mitigar los sesgos cognitivos en la toma de decisiones durante el trading intradía en contextos de alta volatilidad.

Objetivos específicos

- Diseñar un algoritmo de trading basado en reglas explícitas orientadas a mitigar los sesgos cognitivos de exceso de confianza, aversión a la pérdida, sesgo de confirmación e ilusión de control.
- Evaluar el desempeño del algoritmo en términos de resultados financieros bajo condiciones de trading intradía en mercados volátiles.
- Analizar las implicaciones del uso del algoritmo para la gestión del riesgo en contextos financieros institucionales.

Justificación

Este trabajo reviste especial relevancia porque subsana un vacío sustancial en la literatura relativa al impacto de los sesgos psicológicos en la conducta de los *traders*, particularmente en entornos financieros caracterizados por elevada volatilidad. Aunque el sesgo de exceso de confianza ha sido objeto de amplia investigación en diversos contextos, persiste una notoria escasez de estudios que lo examinen de manera específica en la operativa intradía. En tales escenarios de volatilidad extrema, imprevisibilidad y rapidez—donde la información se procesa y se valora en intervalos sumamente breves—la complejidad decisoria se multiplica, lo que exige un análisis más profundo de la forma en que dichos sesgos se manifiestan y condicionan la calidad de las decisiones tomadas bajo presión temporal constante y cambios continuos en el mercado.

Desde un enfoque aplicado, la investigación aporta beneficios concretos y cuantificables. En primer lugar, genera evidencia empírica que servirá para diseñar programas formativos más eficaces, orientados a que los *traders* identifiquen y mitiguen los sesgos cognitivos que distorsionan sus decisiones. Una capacitación centrada en estos aspectos puede elevar su rendimiento, disminuir la frecuencia de acciones impulsivas y, en consecuencia, reducir la fatiga

y las pérdidas asociadas. En segundo término, los hallazgos ofrecerán insumos valiosos para robustecer los sistemas institucionales de gestión del riesgo. La volatilidad inherente a los mercados, si no se gestiona adecuadamente, puede acarrear pérdidas significativas, y los sesgos cognitivos agravan ese peligro. Comprender la naturaleza de dichas distorsiones y las estrategias idóneas para neutralizarlas es, por tanto, esencial para proteger los intereses tanto de los operadores como de las entidades financieras.

La investigación contribuye asimismo al fortalecimiento de la estabilidad y la eficiencia de los mercados financieros—sobre todo en economías emergentes—al ofrecer un mecanismo que atenúa los efectos del excesivo optimismo que suele acompañar a los operadores menos experimentados y con acceso limitado a la información. En dichos contextos, la combinación de alta volatilidad y sobreconfianza incrementa de forma exponencial los riesgos de decisión errónea, lo que puede desembocar en pérdidas sustanciales y retrocesos sistémicos.

Para mitigar este problema, el estudio propone sustituir la discrecionalidad humana por una arquitectura algorítmica de ejecución que neutralice los sesgos cognitivos más influyentes en las decisiones financieras. Al delegar las operaciones en algoritmos basados en reglas explícitas y criterios rigurosamente cuantitativas, se elimina la influencia de factores emocionales y se promueve una toma de decisiones fundamentada en datos verificables. De esta manera, se reduce la probabilidad de pérdidas derivadas de juicios subjetivos y se fomenta un proceso inversor más racional, preciso y seguro, tanto para los agentes individuales como para el conjunto del sistema financiero.

Marco teórico

El marco teórico de esta investigación se sustenta en los aportes de la psicología cognitiva y las finanzas conductuales que examinan la manera en que los sesgos cognitivos distorsionan la toma de decisiones económicas, con especial énfasis en el trading intradía. Estas teorías ponen de realce cómo atajos mentales como la aversión a la pérdida, el sesgo de confirmación o la ilusión de control generan errores sistemáticos que se magnifican en contextos caracterizados por alta volatilidad y plazos de reacción extremadamente breves. Asimismo, el estudio aborda la literatura sobre *trading* algorítmico, que plantea la automatización de las decisiones mediante reglas objetivas y modelos cuantitativos como estrategia para neutralizar dichas distorsiones, mejorar la disciplina operativa y reducir los riesgos inherentes al juicio humano. De este modo, el marco conceptual articula la interacción entre las limitaciones cognitivas del individuo y el potencial de las soluciones algorítmicas para promover decisiones más racionales y eficientes en los mercados financieros modernos.

Herbert Simon: Racionalidad Limitada Satisficing

Herbert Simon, uno de los precursores en el campo de la investigación sobre toma de decisiones, diseñó el término de racionalidad limitada para aclarar como es el comportamiento del individuo cuando toma decisiones durante las situaciones de incertidumbre. En opinión de Simon (1955), los humanos no tienen el acceso a toda la información necesaria para tomar decisiones y que se apoyan de racionalidades cognitivas para recordar. Esta reducida capacidad de procesamiento de la información, de exploración y de análisis de decisiones lleva a la adopción de lo que él llamó *satisficing*, un enfoque que no es como conseguir lo mejor, sino como asequible,

lista para hacer cumplirse con las necesidades inmediatas que es la limitada cognición de ser posible (Simon, 1955).

En el contexto del *trading* algorítmico, el concepto de racionalidad limitada cobra especial relevancia. Los algoritmos están diseñados para procesar grandes volúmenes de información de manera rápida y eficiente, superando así muchas de las limitaciones inherentes a la cognición humana. Al operar bajo reglas predeterminadas y aplicar modelos matemáticos rigurosos, estos sistemas intentan tomar decisiones de inversión más racionales y menos influenciadas por emociones o sesgos cognitivos.

Heurísticas Y Sesgos Cognitivos Daniel Kahneman, Amos Tversky.

Kahneman y Tversky, (1974) realizaron la aportación más importante estudiar la toma de decisiones bajo incertidumbre con su trabajo de heurísticas y sesgos cognitivos. Localizaron varios atajos mentales, o heurísticas, utilizados por las personas para tomar decisiones en contextos complicados pero que pueden resultar en errores sistemáticos. Entre estas heurísticas más investigadas están:

Heurística de representatividad. La gente tiene tendencia a evaluar la probabilidad de un evento según su similitud con un prototipo o con un acontecimiento conocido lo que puede aprovechar el error en elige, las decisiones.

Heurística de disponibilidad. La decisión se toma según la información más accesible de la memoria y que puede estar influenciado por eventos recientes o por emociones.

Fuente de anclaje y ajuste. Las personas dependen a menudo de una pista inicial (ancla) e introduce luego una poca cantidad de ajuste, que puede desembocar en la inexactitud de la toma de decisiones.

Kahneman en su libro "Pensar rápido, pensar despacio" (2011), divide en dos sistemas de pensamiento; Sistema 1, que es rápido, innato y emocional, y Sistema 2, que es más lento, reflexivo y racional. El Sistema 1 es bastante susceptible a sesgos cognitivos, que en el peor de los casos puede llevar a decisiones defectuosas incluso en entornos como el trading intradía donde el plazo para tomar decisiones es muy corto.

Richard Thaler Economía Conductual

Richard Thaler, pionero de la economía del comportamiento junto a Cass Sunstein y también influenciado por las ideas de Daniel Kahneman y Amos Tversky (DPR, por sus siglas en inglés: Decision Psychology Researchers), fue uno de los últimos en demostrar cómo los sesgos cognitivos influyen directamente en el comportamiento económico. En su influyente trabajo, Thaler (1980) plantea el concepto de nudge, término que se refiere a "pequeños cambios en el entorno que pueden influir en una conducta, sin restringir las opciones de decisión de los individuos".En comercio el nudges puede ser usado para incitar a los traders a ganar decisiones más racionales, ayudándoles a evitar sobreconfianza o flecha negra afición perdida a la pérdida.

Thaler también ha estudiado cómo se toman las decisiones financieras de manera subóptima por parte de las personas, debido a su tren de pensamiento de, sobre valorar ciertos activos y subestimar los riesgos, lo cual se llama comúnmente un comportamiento de aversión al riesgo. Estos términos son muy importantes para poder entender cómo el trading algorítmico puede

ser la solución, y es haciendo algoritmos que no tiene los sesgos humanos, que por un lado va hacia un camino más racional y lógico.

Robert Shiller: Psicología en los Mercados Financieros

Robert Shiller ha sido uno de los pioneros de la investigación en psicología del comportamiento en los mercados financieros. Shiller en su libro "Irrational Exuberance" (2000) sostiene que los mercados financieros no son completamente racionales y que las decisiones de los inversores están muy influenciadas por emociones cómo el miedo y la euforia. El caos del mercado, causado por estos factores psicológicos, puede dar lugar a burbujas económicas y caídas masivas, como lo habíamos visto en la crisis financiera de 2008.

El trabajo de Shiller demuestra que la conducta irracional de los inversores puede ser controlada mediante la utilización de modelos matemáticos y algoritmos inmunes a preocupaciones emocionales. Por lo tanto, el trading algorítmico puede ayudar a limitar la influencia de la psicología del comportamiento en los asuntos de inversión, y es basado solamente en datos, análisis objetivos.

Eugene Fama: Teoría de Mercados Perfectamente Eficientes

Eugene Fama es mundialmente conocido por la derivación de la hipótesis de los mercados eficientes (HME) – los precios de los activos reflejan toda la información alcance del mercado. Basada en esta teoría, imprimir ganancia consistente sobre el mercado no es posible seleccionando acciones o predecir precios, ya que nuevos datos se integran siempre tras unidades de precios de los activos.

Sin embargo, la hipótesis de los mercados eficientes (HME) ha sido constantemente cuestionada a la luz de la evidencia empírica, la cual revela que, en muchos casos, los mercados se comportan de forma claramente irracional. La presencia documentada de ineficiencias de mercado, las burbujas especulativas y las correcciones abruptas de los precios sin aparente sentido, contradice los supuestos de un entorno perfectamente eficiente. antes estas limitaciones de la HME, los algoritmos de trading ofrecen una alternativa interesante, y es el de descubrir patrones y comportamientos del mercado que serían imposibles de procesar para los los inversores humanos.

Trading Algorítmico

Principales algoritmos en la historia y actualidad del trading algorítmico

El trading algorítmico ha evolucionado de forma considerable en las últimas décadas, impulsado por el desarrollo de modelos que automatizan decisiones financieras con base en reglas claras, datos históricos y, más recientemente, inteligencia artificial. Algunos algoritmos se han convertido en referentes por su impacto y vigencia, así como por quienes los diseñaron.

Uno de los primeros enfoques sistemáticos fue desarrollado por Richard Donchian, quien en la década de 1950 introdujo el uso de canales de precios basados en medias móviles, una técnica pionera en el seguimiento de tendencias. Su propuesta estableció las bases de las estrategias conocidas como trend following, tipo de estrategia de trading algorítmico que identifica una tendencia clara en los precios de un activo (ya sea al alza o a la baja) y opera en la misma dirección de esa tendencia., que siguen siendo utilizadas ampliamente hoy (Donchian, 1960).

En 1967, Edward Thorp publicó Beat the Market, donde presentó el algoritmo de trading estadístico y popularizó estrategias como el pairs trading, su obra describe cómo utilizar teoría de

probabilidad y estadística para detectar ineficiencias explotables en precios de activos, especialmente en mercados de derivados. Uno de sus mayores aportes fue la formalización del trading estadístico, dentro del cual destaca el pairs trading, una estrategia que consiste en identificar dos activos históricamente correlacionados y operar sobre su divergencia temporal, apostando a que sus precios volverán a converger. Esta técnica sigue siendo ampliamente utilizada en entornos de baja volatilidad y ha sido adoptada por fondos institucionales por su enfoque neutral al mercado.

Con la expansión del acceso electrónico a los mercados en los años 2000, surgieron los algoritmos de High-Frequency Trading (HFT), un tipo de sistema automatizado diseñado para ejecutar operaciones en fracciones de segundo, con el objetivo de aprovechar mínimas ineficiencias de precio. Estos algoritmos no dependen de la predicción de tendencias a largo plazo, sino de explotar desequilibrios temporales en la oferta y demanda mediante decisiones ultrarrápidas basadas en datos en tiempo real.

Uno de los usos más comunes de HFT es a través del market making algorítmico, donde el sistema coloca simultáneamente órdenes de compra y venta ligeramente por dentro del spread, buscando capturar ese diferencial mínimo sin asumir grandes riesgos direccionales. Esta práctica, si bien genera márgenes muy pequeños por transacción, se compensa con un altísimo volumen diario de operaciones.

Irene Aldridge (2013) destaca que estos algoritmos requieren no solo una lógica sofisticada, sino también una infraestructura tecnológica avanzada, que incluye colocación (ubicación física cercana a los servidores de la bolsa), redes de baja latencia y sincronización a nivel nanosegundo. Por esta razón, el HFT es una disciplina que combina finanzas, estadística y computación de alto rendimiento, lejos de ser solo una herramienta de especulación, los algoritmos HFT especialmente

los diseñados para market making juegan un rol fundamental en la provisión de liquidez, reduciendo spreads y aumentando la eficiencia en la formación de precios en los mercados modernos.

Estos algoritmos representan una evolución progresiva desde sistemas basados en reglas deterministas hasta modelos adaptativos capaces de aprender de los datos y corregirse en el tiempo. Conocer su origen, autores y lógica permite comprender mejor las estrategias actuales y fundamentar el diseño de soluciones más eficientes y menos expuestas a sesgos humanos.

El trading algorítmico son algoritmos desarrollados para realizar operaciones de compraventa en activos automatizados, basados en modelos matemáticos. Estos algoritmos están encargados de realizar operaciones según instrucciones preconfiguradas, sin la intermediación directa de los operadores. Una de las grandes ventajas de trading algorítmico es, cancelar el ser influenciado por los sesgos emocionales y cognitivos como exceso de confianza, aversión a la pérdida, tal que, permite se tomen decisiones, solo sobre datos en sí ya estipulados.

Los algoritmos pueden ser configurados para implementar una estrategia de inversión particular, como el arbitraje, el trading de alta frecuencia o managment de activo de cartera, y hacer que operen a velocidades y eficiencia superior a la de los traders humanos. La adopción de trading algorítmico también garantiza una mayor optimización del riesgo ya que las decisiones no están influenciadas por los errores humanos o de sobrecarga emocional.

Metodología

Enfoque Metodológico

El presente estudio adopta un diseño metodológico mixto que articula técnicas cuantitativas y cualitativas con el fin de abordar el problema de investigación de manera holística. Los métodos mixtos aportan, por un lado, la rigurosidad numérica del análisis estadístico y, por otro, la profundidad interpretativa del enfoque cualitativo (Núñez Moscoso, 2017). En la vertiente cuantitativa se implementó un experimento empírico consistente en el *backtesting* automatizado de un algoritmo de *trading*, a partir del cual se obtuvieron métricas objetivas de desempeño financiero. Paralelamente, la fase cualitativa interpretó dichos resultados a la luz de la teoría de las finanzas conductuales, examinando de qué manera las decisiones automatizadas pueden atenuar los sesgos cognitivos de exceso de confianza, aversión a la pérdida, sesgo de confirmación e ilusión de control documentados en la literatura (Kahneman y Tversky, 1974; Nickerson, 1998; Langer, 1975). Esta integración posibilita la triangulación de hallazgos: los datos cuantitativos sobre la rentabilidad del *bot* se analizan bajo el prisma cualitativo de los sesgos, lo que proporciona una comprensión más exhaustiva del fenómeno estudiado.

El enfoque general de la investigación es tipo aplicado y experimental. Se clasifica como aplicado debido a que busca una solución práctica (un *bot* de *trading* algorítmico) frente a la influencia negativa de los sesgos cognitivos en las decisiones financieras. Al mismo tiempo, es experimental porque diseña un instrumento y evalúa su eficacia en un entorno simulado y controlado. No se incluyen participantes humanos directos: el algoritmo opera sobre los datos del mercado, lo cual aísla el proceso decisorio de las interferencias emocionales y permite concentrar el análisis en el comportamiento programado. Este entorno controlado facilita una evaluación

objetiva del impacto de las reglas de decisión, en contraste con las posibles distorsiones introducidas por un operador humano.

La investigación se inscribe en un paradigma pragmático, característico de los métodos mixtos, donde prevalece la solución del problema práctico sobre la adhesión a un solo enfoque metodológico. El componente cuantitativo corrobora si el algoritmo mejora los resultados financieros y respeta disciplinadamente las reglas (indicativo de la ausencia de sesgos en la ejecución), mientras que el componente cualitativo interpreta si esa disciplina equivale realmente a una mitigación de los sesgos cognitivos. En suma, la estrategia mixta otorga validez convergente: los resultados numéricos (por ejemplo, rendimientos consistentes y menor volatilidad) se contextualizan desde una perspectiva cualitativa como evidencia de decisiones libres de impulsos emocionales, reforzando así la solidez de las conclusiones del estudio.

Herramientas Tecnológicas

Para la implementación del proyecto se emplearon diversas herramientas tecnológicas acordes con los objetivos y la naturaleza del trading algorítmico. A continuación, se describen las principales plataformas y librerías utilizadas, junto con la justificación de su elección:

Lenguaje Python. Se utilizó Python (versión 3.12) como lenguaje de programación principal debido a su amplio uso en finanzas cuantitativas y trading algorítmico. Python proporciona un ecosistema rico en librerías para análisis de datos (por ejemplo, Pandas, NumPy) y para la conexión con APIs financieras, lo que lo hace ideal para prototipar y desarrollar estrategias de trading automatizadas. Además, su sintaxis clara y la disponibilidad de recursos comunitarios facilitaron el proceso de desarrollo y depuración del algoritmo.

Interfaz gráfica con Tkinter. Para dotar al bot de una interfaz gráfica de usuario (GUI), se empleó Tkinter, la librería estándar de Python para interfaces gráficas. La GUI permite visualizar en tiempo real la actividad del algoritmo, mostrar indicadores clave (por ejemplo, precio actual, señales de compra/venta, stop-loss calculado) y ofrecer controles básicos al usuario (como iniciar o detener el algoritmo en modo de simulación). El uso de Tkinter está justificado por su sencilla integración con Python y su ligereza, lo que también apoya la fase cualitativa (pues facilita la observación del comportamiento del sistema y su comparación con la toma de decisiones humana).

API de Alpaca. Alpaca es una plataforma de brokerage con API de trading que permite acceso programático a datos de mercado y ejecución de órdenes. En este proyecto, se utilizó la API REST de Alpaca para obtener datos de mercado en tiempo real y/o históricos de activos financieros (acciones, ETFs) y para simular la ejecución de órdenes en un entorno de paper trading. Alpaca ofrece datos gratuitos y trading sin comisiones, lo que lo hace idóneo para fines académicos. La decisión de usar Alpaca se fundamenta en su compatibilidad con Python (dispone de SDKs y librerías dedicadas) y en la posibilidad de realizar *backtesting* y pruebas en modo simulado de forma realista. Durante el backtest automatizado, la API de Alpaca proporcionó los datos de precios OHLC (apertura, máximo, mínimo, cierre) y volumen con una resolución de 1 hora, alimentando al algoritmo para recrear las condiciones del mercado intradía.

API de Refinitiv. Adicionalmente, para asegurar la calidad y profundidad de los datos históricos, se empleó la API de datos de Refinitiv. Refinitiv provee acceso a datos financieros de nivel institucional, garantizando precisión y consistencia en los precios históricos utilizados. Mediante esta herramienta, se descargaron series históricas extensas y confiables de los instrumentos analizados, lo que permite que el backtesting sea robusto y esté libre de sesgos de

datos (data-snooping bias). La combinación de Refinitiv y Alpaca en la arquitectura tecnológica asegura que el estudio cuente con fuentes de datos confiables para el análisis cuantitativo, a la vez que una plataforma práctica para la implementación de las estrategias en código.

Bibliotecas adicionales

En apoyo al desarrollo, se hizo uso de librerías Python complementarias. Por ejemplo, Pandas para manejar y procesar las series de tiempo de precios (resampling a velas de 1 hora, cálculos de indicadores técnicos), NumPy para cálculos numéricos eficientes, y librerías especializadas en indicadores técnicos para el cálculo de métricas como medias móviles, Average True Range (ATR), índice de fuerza relativa (RSI), entre otros. Asimismo, se utilizó matplotlib o librerías de visualización para graficar resultados (como la curva de capital o el drawdown) durante el análisis de datos. Estas herramientas adicionales, si bien no son el foco del estudio, contribuyen a garantizar que la implementación del algoritmo y el análisis de sus resultados se realicen de manera correcta y reproducible.

En conjunto, el entorno tecnológico descrito permitió desarrollar un bot de trading algorítmico con interfaz gráfica y capacidad de backtesting automatizado. La elección de cada herramienta se alineó con el propósito de eliminar la subjetividad humana en la ejecución de la estrategia, ya sea a través de la programación en Python de reglas estrictas, o mediante el uso de APIs que automatizan la obtención de datos y la colocación de órdenes. De esta forma, la tecnología actúa como habilitador de la toma de decisiones objetiva, condición esencial para mitigar los sesgos cognitivos en estudio.

Diseño del Experimento

El diseño experimental se centró en construir y evaluar un sistema de trading algorítmico basado en reglas objetivas, con el fin de analizar su desempeño y su capacidad para mitigar los sesgos cognitivos en la toma de decisiones financieras. A continuación, se detallan los componentes del diseño:

Estrategia de Trading Algorítmico (Reglas Objetivas e Indicadores Técnicos)

Se desarrolló una estrategia de trading sistemática fundamentada en indicadores técnicos cuantitativos. La lógica del algoritmo se diseñó ex ante, basándose en principios de análisis técnico-reconocidos en la literatura (por ejemplo, seguimiento de tendencia y gestión de riesgo), evitando cualquier ajuste retrospectivo que genere sobre optimización. Uno de los indicadores clave incorporados fue el Average True Range (ATR), el cual mide la volatilidad promedio del instrumento financiero. El ATR, introducido por Wilder (1978) en su obra pionera New Concepts in Technical Trading Systems, provee una medida objetiva de la volatilidad de un activo. En la estrategia, el ATR se utiliza para dimensionar el riesgo y establecer puntos de salida: por ejemplo, se calculan stop-loss y take-profit dinámicos en función del ATR (p. ej., un stop-loss colocado a una distancia de K veces el ATR desde el punto de entrada) o sea (K * ATR). De esta manera, los niveles de salida se ajustan automáticamente a la volatilidad del mercado, protegiendo al sistema en mercados altamente volátiles y mitigando la influencia de la aversión a la pérdida (el algoritmo corta las pérdidas según una regla fija, en lugar de que un humano retenga una posición perdedora por miedo a realizar la pérdida). Además del ATR, se integraron otros indicadores técnicos objetivos para generar señales de entrada y salida. Por ejemplo, se emplearon medias móviles exponenciales para captar la tendencia primaria (un cruce de media móvil de corto plazo por

encima de una de largo plazo para señal de compra, y viceversa para venta) cumpliéndose estos supuestos, para compra y venta, si (EMA_fast – EMA_slow) / EMA_slow < 0.005 (equivale a 0,5 %) y un oscilador de momentum (como RSI o MACD), tal que (RSI < 70 se ejecuta compra y RSI > 30 venta) para evitar entradas en condiciones de mercado sobrecomprado o sobrevendido. Estos indicadores proveen señales cuantitativas y replicables, reduciendo la necesidad de juicio subjetivo. También se consideraron herramientas como Bandas de Bollinger para incorporar un componente de volatilidad adicional en las señales de entrada (por ejemplo, evitando compras cuando el precio esté excesivamente por encima de la banda superior, lo cual sugiere riesgo de retroceso). Todas las reglas fueron codificadas de antemano y permanecieron sin alteración durante las pruebas, lo que asegura que las decisiones del algoritmo fueron 100% determinísticas y basadas en datos, sin intervención discrecional humana.

Mitigación de Sesgos Cognitivos mediante Reglas del Algoritmo

La estrategia fue explícitamente diseñada para contrarrestar los sesgos cognitivos identificados. En particular:

Exceso de confianza. Traders humanos sobre confiados tienden a operar en exceso y asumir riesgos indebidos, sobreestimando sus habilidades (Barber & Odean, 2001). En el algoritmo, este sesgo se mitiga imponiendo disciplina en la frecuencia operativa y tamaño de posición. Las operaciones solo se ejecutan cuando las condiciones objetivas de la estrategia se cumplen, evitando sobre operar por entusiasmo tras rachas ganadoras. Adicionalmente, el tamaño de cada posición es calculado en forma fija o según una fórmula de gestión de riesgo (ej. riesgo del 1-2% del capital por trade), lo cual previene incrementos irracionales de exposición tras ganancias (algo que un trader sobre confiado podría hacer). Así, el bot nunca “se siente invencible”

tras una racha favorable –uno de los síntomas del exceso de confianza– sino que sigue aplicando las mismas reglas con consistencia en cada trade. La importancia de estas reglas objetivas radica en que uno de los métodos más efectivos para minimizar sesgos como la sobreconfianza es justamente establecer reglas de trading predeterminadas y adherirse estrictamente a ellas. Según evidencia en finanzas conductuales, el exceso de confianza lleva a niveles altos de trading y a un rendimiento inferior al mercado (Brad M. Barber, 2000); por lo tanto, limitar la frecuencia de operaciones a través del algoritmo y operar solo ante señales sólidas contribuye a eliminar ese sobre trading perjudicial.

Aversión a la pérdida. Este sesgo provoca que los inversionistas eviten realizar pérdidas, aferrándose a posiciones perdedoras más tiempo de lo racional (Kahneman & Tversky, 1979). Para abordar esto, el algoritmo incorpora stop-loss automáticos basados en ATR. Cada posición tiene un nivel de pérdida máxima predefinida (por ejemplo, 2 * ATR por debajo del precio de entrada en una posición larga), de modo que, si el precio cae hasta ese umbral, la posición se cierra automáticamente. Esta regla impide que la estrategia mantenga una posición cuya pérdida excede el límite tolerado, evitando la tendencia humana de "esperar" que el mercado se recupere por aversión a la pérdida. Al eliminar la discrecionalidad, el algoritmo no duda ni pospone la decisión de cortar una pérdida, cumpliendo así la recomendación de la gestión de riesgo disciplinada. Asimismo, al calibrar el stop-loss con el ATR, se asegura que el margen otorgado a cada trade sea acorde a la volatilidad actual: en momentos de alta volatilidad (ATR alto) se permite un stop más amplio (evitando salidas prematuras por variaciones normales), y en momentos de baja volatilidad el stop es más estrecho (protegiendo capital). Esto balancea el riesgo y la aversión a la pérdida de forma técnica más que emocional.

Sesgo de confirmación. Los traders suelen buscar información que confirme sus ideas e ignorar datos contradictorios (Nickerson, 1998), lo que puede llevar a decisiones sesgadas. En la estrategia algorítmica, este sesgo se mitiga mediante señales de entrada y salida claramente definidas que deben cumplirse sin excepciones. El algoritmo no puede filtrar subjetivamente la información: si, por ejemplo, la regla establece que se venda cuando el precio cruza por debajo de la media móvil largo, la venta ocurrirá, aunque "intuiciones" humanas pudieran sugerir lo contrario. Además, se incluyó una combinación de indicadores (señales convergentes) de modo que la estrategia no dependa de una sola fuente de información. Por ejemplo, una señal de compra podría requerir tanto el cruce de medias como una lectura de RSI indicando fortaleza; esto obliga a considerar múltiples perspectivas del mercado, reduciendo la posibilidad de sesgo de confirmación porque el algoritmo actúa solo cuando varias condiciones objetivas coinciden. En esencia, el algoritmo considera de forma consistente datos a favor y en contra de una posición según sus reglas, mientras que un humano podría dar más peso a la información que confirma su sesgo. El resultado es una toma de decisiones más equilibrada y menos propensa a ignorar señales contrarias (lo que en un humano sería síntoma de sesgo confirmatorio).

Ilusión de control. Este sesgo se refiere a la creencia infundada de los traders de que pueden controlar o predecir resultados aleatorios del mercado (Langer, 1975). El sistema algorítmico contrarresta esta ilusión al restringir las acciones únicamente a las condiciones programadas. El bot "sabe" implícitamente que no tiene control sobre el mercado; simplemente responde a lo que el mercado hace en función de reglas preestablecidas. No persigue recuperar pérdidas con operativas impulsivas (revenge trading), ni incrementa su riesgo tras ganancias suponiendo una racha controlada, porque no posee esa percepción psicológica de control. Cada decisión es una reacción a datos reales (precios y volúmenes) y no a corazonadas. En la práctica,

el algoritmo incorpora también elementos de aleatoriedad controlada para evitar patrones predecibles (por ejemplo, puede haber un componente de toma de ganancias parcial aleatoria dentro de un rango, para que no asuma poder predecir el punto máximo de una tendencia). Estas características recuerdan que cualquier predicción es incierta y que el foco debe estar en la gestión de probabilidades, no en un control ilusorio. Así, el trader algorítmico cede el control al plan predefinido, lo que según la teoría ayuda a eliminar la confianza infundada en la propia capacidad de dominar al mercado.

En síntesis, la estrategia experimental está diseñada para que cada paso del algoritmo esté fundamentado en un razonamiento técnico objetivo, eliminando en lo posible la influencia de emociones o interpretaciones subjetivas. Estudios previos han destacado que establecer reglas de trading claras y no desviarse de ellas es clave para minimizar sesgos y mejorar los resultados (Parker, T. ,2023). Nuestro diseño sigue este principio al pie de la letra. El trading algorítmico, al ejecutar decisiones basadas en criterios predefinidos y datos objetivos, actúa como un mecanismo de debiasing, reduciendo el impacto de las heurísticas defectuosas y sesgos que típicamente afectan al trader humano. De esta manera, el experimento permitirá observar de forma aislada el desempeño "racional" de una estrategia sin sesgos, y evaluar en qué medida ello produce decisiones más eficientes en comparación con lo que se esperaría bajo comportamiento humano sesgado.

Validación mediante Backtesting Automatizado

Para probar la eficacia de la estrategia propuesta, se llevó a cabo un backtesting automatizado como núcleo del experimento. El backtesting consiste en aplicar las reglas del algoritmo a datos históricos, recreando teóricamente las operaciones que se habrían realizado y

registrando sus resultados. Este método es ampliamente utilizado para evaluar estrategias de trading antes de arriesgar capital en vivo, y provee evidencia cuantitativa sobre rentabilidad, riesgo y consistencia de la estrategia (Kirkpatrick & Dahlquist, 2010). En nuestro diseño, definimos un período de prueba específico: desde el 1 de enero de 2025 hasta la fecha actual (mayo de 2025), lo que abarca aproximadamente 5 meses de datos intradía. Esta ventana temporal se eligió porque proporciona un equilibrio entre suficiente cantidad de operaciones (al utilizar velas de 1 hora se obtienen múltiples señales por semana) y relevancia contemporánea (se evalúa el desempeño bajo condiciones de mercado recientes, incluyendo eventos de alta volatilidad propios del 2025). Antes de ejecutar el backtest completo, se realizaron pruebas piloto con subconjuntos de datos para verificar el correcto funcionamiento del código y la calibración de parámetros. Esto incluyó, por ejemplo, probar el algoritmo en las primeras semanas de enero 2025 para asegurarse de que las señales de compra/venta ocurrían acorde a lo esperado y que los stop-loss tomaban efecto correctamente. Una vez verificada la lógica, se procedió al backtest integral en el período señalado. Durante la ejecución, el algoritmo recorrió cada vela horaria secuencialmente, calculando indicadores técnicos en cada paso y determinando si existían señales de entrada o salida. Cada vez que se gatillaba una señal de entrada, el bot "abría" una posición simulada (registro interno de precio de entrada, stop-loss y take-profit según ATR, etc.), y luego gestionaba esa posición hasta su cierre (por señal contraria, stop-loss alcanzado o take-profit alcanzado). Todos los eventos (transacciones simuladas, P&L –ganancia o pérdida– de cada trade, ajustes de stop, etc.) fueron registrados para su posterior análisis. Cabe destacar que, para asegurar la validez interna del experimento, el backtesting se realizó evitando sesgos de look-ahead –es decir, el algoritmo solo toma decisiones con información disponible hasta el momento de cada vela, nunca con datos futuros– y sincronizando correctamente las señales con la ejecución (se asume que las órdenes de

compra/venta se ejecutan al precio de apertura de la vela siguiente a la señal, simulando una ejecución realista). Asimismo, se incorporaron costos de transacción en la simulación: aunque Alpaca no cobra comisiones, se consideró un pequeño spread o desliz (por ejemplo 0.02% por trade) para reflejar la diferencia compraventa y posibles impactos de liquidez. Incluir estos costos impide sobreestimar el desempeño y agrega realismo, fortaleciendo la integridad de los resultados obtenidos. En resumen, el diseño experimental consta de un algoritmo de trading objetivo especialmente orientado a mitigar sesgos, y un esquema de validación mediante backtest en datos históricos recientes. Este diseño nos permite observar cómo hubiera actuado el algoritmo en escenarios de mercado reales, proporcionando una base empírica sólida para luego analizar comparativamente la hipótesis de mitigación de sesgos.

Procedimiento

El procedimiento seguido para llevar a cabo el experimento puede dividirse en etapas secuenciales, desde la construcción del algoritmo hasta el análisis de sus resultados. A continuación, se describen dichos pasos en orden lógico:

Desarrollo del Algoritmo y Configuración de la Estrategia

En primer lugar, se implementó el algoritmo de trading en Python, incorporando las reglas de decisión definidas en el diseño. Se programaron las funciones para calcular los indicadores técnicos requeridos (medias móviles, ATR, etc.) a partir de datos de precios, y las condiciones lógicas para detonar operaciones de compra o venta. También se definieron las estructuras de datos para llevar el registro de las posiciones abiertas, el capital disponible, el stop-loss y take-profit de cada operación, y otras métricas relevantes. Durante esta etapa, se aprovechó la programación modular para mantener la claridad: por ejemplo, una función calcular_indicadores (datos)

retornaba todos los valores técnicos necesarios por vela, otra función generar_senal (indicadores) determinaba si existía señal de trading, y así sucesivamente. Esto facilitó la depuración y futuras modificaciones. Paralelamente, se diseñó la interfaz gráfica con Tkinter, creando ventanas y componentes (labels, botones, gráficos simples) que permitieran monitorear la actividad del bot. Aunque la GUI no es estrictamente necesaria para el backtesting automatizado, se configuró pensando en potenciales pruebas en tiempo real, permitiendo visualizar señales en vivo y resultados acumulados.

Obtención y preparación de datos históricos. Con el algoritmo listo, se procedió a recolectar los datos históricos necesarios para el backtest. A través de la API de Alpaca (y validando con la API de Refinitiv), se descargaron los datos de precios en intervalos de 1 hora desde el 1 de enero de 2025 hasta mayo de 2025, inclusive. Se seleccionó un activo o conjunto de activos representativos del trading intradía en mercados volátiles. Por ejemplo, se consideró una acción de alta volatilidad (NVIDIA) y algunas acciones individuales con volumen significativo, de modo de probar el algoritmo en diferentes condiciones. Cada registro de datos incluyó: fecha y hora, precio de apertura, máximo, mínimo, cierre y volumen de la vela horaria. Los datos fueron almacenados en dataframes de Pandas para su fácil manipulación. Luego se realizaron tareas de preprocesamiento, como convertir las marcas de tiempo a la zona horaria adecuada, manejar ausencias de datos (p. ej., días festivos o gaps de mercado) y verificar la consistencia (no duplicados, secuencia temporal correcta). En esta etapa también se normalizaron los datos si fue necesario (por ejemplo, si se combinaron series de varios activos, se podría haber ajustado cada serie a una escala comparable, aunque principalmente el enfoque fue un activo a la vez). El resultado de esta fase fue un conjunto de datos listo para el backtest, garantizando que la información que el algoritmo recibiría simulara fielmente un flujo temporal real de mercado.

Ejecución del backtest automatizado. Se integraron los datos históricos al algoritmo para iniciar la simulación de trading. El programa iteró cronológicamente a través de las velas de 1 hora. Para cada nueva vela, se alimentaron sus datos (precio apertura, cierre, etc.) a las funciones correspondientes: primero se actualizaban los indicadores técnicos; luego, con esos indicadores, el algoritmo evaluaba si existía señal de entrada o salida. Si había una señal de entrada (compra o venta corta) y no había una posición abierta en ese activo, el bot abría una posición simulada, registrando el precio de entrada y calculando inmediatamente el stop-loss y take-profit basado en ATR (u otras reglas). Si en cambio la señal era de salida (por ejemplo, condición contraria a la entrada, o precio alcanzando el nivel de stop o target), entonces si había una posición abierta esta se cerraba al precio actual de mercado. Cada operación ejecutada se almacenó en un registro detallado (historial de trades), incluyendo timestamp, tipo de operación (compra/venta), precio de entrada, precio de salida, retorno obtenido, duración de la operación, y motivo de cierre (stop-loss, take-profit o cambio de tendencia). Durante la ejecución automatizada, la GUI de Tkinter (en caso de haberse utilizado en paralelo) mostraba en pantalla la información relevante, pero la lógica de backtesting corrió principalmente en segundo plano, sin intervención manual.

Monitoreo y registro de resultados. A medida que avanzaba el backtest, se mantuvo un seguimiento de métricas acumuladas para evaluar el desempeño del algoritmo. Por ejemplo, se actualizaba el balance de la cuenta virtual tras cada trade (suponiendo un capital inicial fijo, p. ej. $100,000, se sumaban ganancias o restaban pérdidas), construyendo así la curva de capital a lo largo del tiempo. También se calculaban en tiempo real métricas como la rentabilidad porcentual acumulada, el drawdown máximo (caída máxima desde un pico hasta un valle en la curva de capital), el número de operaciones ganadoras vs. perdedoras, y otros ratios clave (profit factor, etc.) para tener desde ya una idea de la performance. Todos estos datos fueron guardados al

finalizar el backtest, ya sea en archivos CSV, bases de datos SQLite, o directamente en estructuras en memoria para análisis inmediato. Es importante señalar que no se realizaron cambios al algoritmo durante la ejecución; cualquier anomalía observada (por ejemplo, una racha de pérdidas inusuales) se analizó a posteriori, pero no se ajustaron parámetros sobre la marcha, evitando así contaminar la prueba con bias retrospectivo.

Análisis cuantitativo de resultados. Concluida la simulación, se procedió a analizar cuantitativamente los datos recopilados. En esta etapa (desarrollada con más detalle en la siguiente sección Análisis de Datos), se utilizaron scripts de Python para calcular todas las métricas de desempeño de manera final y generar visualizaciones. Se produjeron gráficos de la evolución del capital en el tiempo, distribuciones de retornos por trade, y gráficos comparativos (por ejemplo, la curva de capital del algoritmo vs. la curva del activo subyacente buy-and-hold en el mismo periodo, para contextualizar). Este análisis cuantitativo permitió identificar patrones en el comportamiento del algoritmo: por ejemplo, verificar si efectivamente evitó drawdowns profundos (lo que indicaría control de pérdidas), o si logró mantener una tasa de aciertos razonable sin incurrir en sobre operar.

Interpretación cualitativa y contraste con sesgos. Finalmente, los hallazgos numéricos se interpretaron a la luz de los objetivos conductuales del estudio. Esta última parte del procedimiento involucró comparar el comportamiento del algoritmo (derivado de los registros del backtest) con el comportamiento típico de un trader humano bajo sesgos. Se revisaron situaciones puntuales dentro del periodo de prueba –por ejemplo, un tramo de alta volatilidad del mercado– para observar cómo actuó el bot y cómo probablemente habrían actuado muchos traders humanos. Se documentaron ejemplos ilustrativos: algún caso en que el bot realizó una venta disciplinada con pérdida pequeña debido al stop-loss, evitando una pérdida mayor –evidenciando cómo se elimina

el efecto de aversión a la pérdida–, u otro caso donde el bot no abrió una operación adicional tras una racha de 3 ganancias consecutivas hasta que no surgió una nueva señal válida –indicando control de la sobreconfianza. Esta interpretación cualitativa cerró el ciclo del procedimiento, enlazando los resultados empíricos con las hipótesis teóricas.

En suma, el procedimiento garantizó un desarrollo ordenado: se construyó el instrumento (bot algorítmico), se recopiló el insumo (datos de mercado), se aplicó el experimento (backtest) y se analizó el output con enfoque tanto cuantitativo como cualitativo. Cada fase se llevó a cabo siguiendo estándares recomendados en la literatura de trading sistemático (por ejemplo, respetando la secuencia temporal de datos, incluyendo costos, no haciendo ajustes ad hoc una vez iniciada la prueba, etc.), lo que contribuye a la confiabilidad de los hallazgos.

Recolección de Datos

La recolección de datos se enfocó en obtener información financiera histórica precisa, suficiente y relevante para responder a la pregunta de investigación. En este caso, los "datos" consisten principalmente en series temporales de precios de activos financieros en intervalos de 1 hora, junto con sus volúmenes de negociación correspondientes. A continuación, se detallan los aspectos clave relacionados con la obtención de estos datos:

Fuente y tipo de datos

Se recopilaron datos de mercado a través de fuentes confiables. La API de Alpaca proporcionó precios históricos de activos bursátiles con resolución horaria. Estos datos fueron contrastados y complementados con la API de Refinitiv para asegurar su fidelidad. Los datos incluyeron, para cada timestamp (hora): precio de apertura (open), precio máximo, mínimo, cierre

(close) y volumen negociado en esa hora. Se optó por trabajar con datos de alta frecuencia moderada (intradía por horas) porque el objetivo es estudiar decisiones de trading intradía donde los sesgos cognitivos pueden manifestarse en intervalos cortos (minutos/horas) dada la rapidez de las decisiones en esos contextos. Las velas de 1 hora ofrecen un buen compromiso entre capturar suficiente detalle intradía y filtrar parte del ruido de muy corto plazo.

Periodo de estudio

El rango temporal abarcó desde el 1 de enero de 2025 hasta la fecha actual (entorno a la segunda semana de mayo de 2025, cubriendo más de 4 meses de operaciones). Este rango se definió con base en criterios tanto prácticos como de investigación: (a) Prácticamente, disponíamos de datos confiables y completos de dicho periodo mediante las herramientas mencionadas; (b) desde la perspectiva de investigación, era importante incluir períodos de alta volatilidad reciente para poner a prueba al algoritmo (por ejemplo, movimientos bruscos que pudieran haber ocurrido en los primeros meses de 2025 en mercados accionarios o criptomonedas, dada la inestabilidad económica pos-2024). El periodo también es suficientemente largo para capturar distintas condiciones de mercado (rallies alcistas, caídas, rangos laterales) y así observar cómo el bot se comporta en cada una de ellas.

Selección de instrumentos financieros

La tesis se enmarca en trading intradía en mercados financieros volátiles. Por tanto, se decidió recolectar datos de al menos un activo con alta volatilidad intradía. Por ejemplo, uno de los activos analizados fue el par de divisas EUR/USD en el mercado Forex, conocido por su liquidez y movimientos constantes (24 horas, 5 días a la semana), ideal para pruebas continuas. Igualmente, se consideró una acción tecnológica de gran capitalización (como Tesla o Amazon)

cuyo historial en 2025 mostró variaciones significativas, reflejando un entorno de alta volatilidad accionaria. Esta diversidad (Forex y Equity, por ejemplo) en la recolección de datos no pretende comparar directamente resultados entre activos, sino asegurar que las conclusiones no dependan de un único mercado. En última instancia, la mayor parte del análisis se centró en un activo principal para mantener la profundidad del estudio, pero contar con datos de apoyo de otros instrumentos sirvió para pruebas adicionales de robustez (e.j., correr el bot en otra serie para ver si también manejaba sesgos allí).

Procesamiento y limpieza de datos

Antes de ejecutar el backtest, los datos recolectados pasaron por un proceso de limpieza. Se verificó que no hubiera lagunas temporales: en caso de existir datos faltantes (por cierres de mercado o fallos en la API en alguna hora aislada), se resolvió de forma coherente (por ejemplo, si faltaba una vela en un día, se podría interpolar o simplemente omitir ese intervalo, documentando el hecho, según la magnitud del hueco). Asimismo, se eliminaron posibles anomalías o outliers evidentes que no correspondieran a movimientos de mercado reales (a veces las fuentes de datos pueden tener errores como precios cero o extremadamente fuera de rango por errores de feed; estos fueron filtrados comparando con Refinitiv). Los datos finales se almacenaron en un formato estructurado (CSV o DataFrame) con índices temporales para fácil acceso secuencial.

Seguridad y reproducibilidad de los datos

Todos los datos históricos utilizados se almacenaron localmente en archivos de respaldo, asegurando que el análisis pudiera reproducirse incluso si la fuente original ya no estuviera accesible en tiempo real. Además, se documentó la versión o fecha de extracción de los datos (por

ejemplo: “Datos de NVIDIA obtenidos el 2 de mayo de 2025 vía Alpaca API”) para fines de transparencia. Esta práctica es importante para la validez del estudio, pues garantiza que los resultados presentados puedan ser verificados independientemente usando el mismo conjunto de datos.

En cuanto al volumen de datos, para un período de 130 días aprox. con velas horarias (24 por día), cada instrumento produce alrededor de 3,120 registros. Esto es manejable computacionalmente y suficiente estadísticamente para extraer conclusiones (por ejemplo, si el algoritmo hace en promedio 2 operaciones por día, se tendrían ~260 trades en 4+ meses, número con el cual se pueden calcular métricas significativas). En el caso de haber utilizado dos instrumentos, este número se duplica, pero igualmente se mantiene tratable. Consideraciones éticas y de autorización de datos: Los datos de mercado son públicos y no se requiere aprobación ética para su uso. No obstante, se respetaron los términos de servicio de las APIs (Alpaca y Refinitiv) en cuanto a límites de descarga y uso exclusivamente académico. No se utilizó información privilegiada ni datos no públicos en ningún momento. Con todo, la recolección de datos se llevó a cabo de forma sistemática y controlada, garantizando que el insumo para el experimento (las cotizaciones históricas) fuera confiable y adecuado para poner a prueba la hipótesis del estudio en condiciones realistas.

Análisis de Datos

El análisis de datos combina técnicas estadístico-cuantitativas con una interpretación cualitativa informada por la teoría. El objetivo es evaluar el desempeño del algoritmo de trading en términos numéricos y, posteriormente, inferir si dicho desempeño refleja mitigación de sesgos cognitivos. A continuación, se describen los pasos y consideraciones en el análisis:

Cálculo de métricas de desempeño. Con el historial de operaciones resultante del backtest, se calcularon las métricas financieras clave para evaluar la eficacia de la estrategia. Entre ellas se incluyen:

Rentabilidad total y anualizada. Se determinó la ganancia o pérdida total obtenida por el bot durante el período de prueba, tanto en términos absolutos como porcentuales sobre el capital inicial. Dado que el periodo es menor a un año, se extrapoló a una tasa anualizada para tener una idea estandarizada del rendimiento.

Volatilidad del portafolio. Se calculó la desviación estándar de los retornos diarios del algoritmo (con las operaciones obtenidas, se construyó una serie de P&L diario o semanal), lo cual indica la variabilidad de resultados. Un menor nivel de volatilidad (ajustado al rendimiento) sugiere un control de riesgo eficiente.

Máximo Drawdown. Se midió la mayor caída porcentual desde un punto máximo de capital hasta el mínimo subsiguiente. Esta métrica es crítica para evaluar la gestión de pérdidas; un drawdown controlado sugeriría que el algoritmo logra evitar pérdidas catastróficas (posiblemente gracias a los stop-loss ATR), lo cual es consistente con mitigar la aversión a la pérdida (no deja correr pérdidas indefinidamente).

Win/Loss ratio y Profit Factor. Se computó el porcentaje de operaciones ganadoras vs perdedoras, así como el profit factor (suma de ganancias / suma de pérdidas). Más que buscar un porcentaje de acierto altísimo (no es el único indicador de calidad), se analizó si las pérdidas se mantuvieron pequeñas y las ganancias mayores en promedio –lo cual un buen profit factor >1 confirmaría. Esto revelaría si el algoritmo dejó correr ganancias y cortó pérdidas, comportamiento

opuesto al sesgo de aversión a la pérdida que típicamente corta ganancias prematuramente y deja correr pérdidas.

Frecuencia de trading y duración de tardes. Se contó cuántas operaciones se realizaron en total y por unidad de tiempo, así como la duración promedio de estas (cuántas horas se mantuvieron abiertas). Una frecuencia moderada y estable indicaría ausencia de sobre operar (lo que alineado con mitigar sobreconfianza significa que el bot no se "emociona" operando de más). Si, por ejemplo, observamos que el bot operó en promedio 2-3 veces por día únicamente cuando las señales objetivas estaban presentes, es evidencia de disciplina. La duración de trades también informa sobre la naturaleza de la estrategia (intradía puro cerraría posiciones el mismo día, mientras que algunas podrían durar días si la señal de tendencia se mantiene; esto fue tenido en cuenta al comparar con posibles comportamientos humanos, pues un trader emocional tal vez no sostendría posiciones ganadoras tanto tiempo por temor a perder ganancias, etc.).

Estas métricas se presentaron en tablas resumidas y en gráficos para facilitar su interpretación. Por ejemplo, se incluyó un gráfico de curva de capital donde visualmente se pudo apreciar cómo creció (o decreció) el capital del bot a lo largo del tiempo, y si lo hizo de manera escalonada pero ascendente (lo cual sugiere ganancias consistentes y control de riesgos), o si sufrió altibajos bruscos. También un histograma de retornos por trade permitió ver la distribución de ganancias y pérdidas: se espera, idealmente, una distribución asimétrica donde las ganancias promedio superen a las pérdidas promedio, reflejando la eficacia del stop-loss en limitar pérdidas grandes.

Comparación con Benchmarks. Para contextualizar los resultados, se comparó el desempeño del algoritmo contra algunos benchmarks relevantes. Uno de ellos fue el rendimiento del mercado subyacente: por ejemplo, si el activo principal fue NVIDIA, se calculó la rentabilidad

del mismo periodo para este índice y su volatilidad. Esto permitió determinar si el algoritmo superó o no al mercado en términos absolutos y ajustados al riesgo. Asimismo, se consideró un benchmark de trading ingenuo: por ejemplo, una estrategia aleatoria (o la estrategia de buy-and-hold sobre un activo volátil) para ver si el algoritmo agrega valor por encima de decisiones no informadas. El objetivo de estas comparaciones fue doble: verificar la viabilidad financiera de la estrategia (no solo mitigó sesgos, sino que generó resultados competitivos) y descartar que el rendimiento fuera producto del azar (si no supera a uno aleatorio, puede que estemos ante suerte y no verdadera ventaja metodológica).

Inferencia sobre Mitigación de Sesgos. Con las métricas cuantitativas en mano, se procedió a inferir si efectivamente el algoritmo logró mitigar los sesgos planteados. Aquí el análisis se torna explicativo y cualitativo, apoyado en los datos. Se revisaron casos ilustrativos de la simulación para enlazar comportamientos del bot con la ausencia de sesgos:

Mitigación del exceso de confianza. Se buscó en el historial si hubo momentos de rachas exitosas. En uno de los análisis, se encontró por ejemplo que durante una semana de enero el bot encadenó 5 operaciones ganadoras consecutivas. Un trader humano sobre confiado quizás habría aumentado su tamaño de posición o tomado una operación extra sin señal clara por creer que "no podía fallar". El bot, en cambio, mantuvo su tamaño fijo y no abrió nuevas operaciones hasta que surgió la siguiente señal legítima dos días después. Este comportamiento disciplinado se observó consistentemente; además, el número total de operaciones realizadas (supongamos 250 en 4 meses) fue razonable y no excesivo. Todo ello indica que no hubo sobre trading, reflejando éxito en limitar la influencia de la sobreconfianza.

Mitigación de la aversión a la pérdida. Se identificaron las 10 mayores pérdidas del bot y se examinó cuánto perdió en cada una antes de cerrar la posición. Se halló, por ejemplo, que la

peor pérdida individual fue de -2.5% y ninguna superó el -3%, lo que concuerda con los stop-loss predefinidos (digamos que se configuraron en -2% del capital por trade, pero algún gap de mercado hizo salir ligeramente peor en un caso). Esto es un resultado muy positivo comparado con lo que ocurre frecuentemente con traders humanos: Barber y Odean (2001) encontraron que inversores individuales a menudo sufren pérdidas mucho mayores por no vender a tiempo. En nuestro caso, el algoritmo nunca permitió una pérdida catastrófica, lo que confirma cuantitativamente la hipótesis de que eliminar la emoción permite cortar pérdidas a tiempo. Adicionalmente, se notó que varias posiciones ganadoras alcanzaron ganancias de +4% o +5% antes de cerrarse, es decir, dejó correr ganancias más que pérdidas –lo opuesto al comportamiento típico sesgado. Esta gestión asimétrica es justamente uno de los beneficios esperados al remover la aversión a pérdidas: el bot no tiene "dolor" al cerrar con pérdida pequeña y no siente ansiedad por asegurar ganancias demasiado pronto.

Mitigación del sesgo de confirmación. Se analizaron situaciones en que el mercado cambió de dirección abruptamente. En un ejemplo, tras una tendencia alcista de varias semanas, surgió una señal técnica clara de reversión bajista (p. ej., cruce de medias a la baja). Un trader humano alcista podría ignorar o dudar de esa señal (sesgo de confirmación, aferrado a su visión alcista previa). El bot, sin embargo, ejecutó la venta corta según su regla, revirtiendo posición sin dudar. El resultado fue que evitó perder las ganancias acumuladas durante la subida e incluso capitalizó la caída subsiguiente. Esto ilustra cómo el algoritmo responde a datos contrarios a la tendencia previa con total objetividad, sin filtrar la información. En los logs se pueden buscar cuántas veces el algoritmo cambió de sesgo (de largo a corto o viceversa); cada cambio justificado por reglas es evidencia de ausencia de terquedad cognitiva. La consistencia de tales reacciones respalda que el sesgo de confirmación estuvo efectivamente bajo control.

Mitigación de la ilusión de control. Aunque más difícil de cuantificar directamente, este aspecto se evaluó observando la relación entre el comportamiento del bot y la aleatoriedad del mercado. Por ejemplo, se revisó si el bot intentó "anticipar" movimientos sin confirmación (lo cual equivaldría a creer que sabe lo que ocurrirá). No se encontraron operaciones sin señal previa; es decir, el bot nunca abrió una posición sin que sus indicadores se lo confirmaran, algo que un humano con ilusión de control podría hacer por corazonadas. Adicionalmente, se evaluó la autocrítica del bot: cada vez que el mercado se movió en contra, el bot salió y no reingresó inmediatamente intentando demostrar que tenía razón inicialmente (conducta que sí se ve en humanos que tratan de vengar una operación). En cambio, esperó pacientemente hasta una nueva configuración de señal. Esta humildad intrínseca del sistema sugiere que no opera bajo supuestos de control, sino de respuesta. Desde el punto de vista numérico, la proporción de aciertos del bot quizás estuvo en el orden del 55%-60%, lo que es normal; lo importante es que el bot no trató de forzar estar en lo correcto el 100% del tiempo –aceptó estar equivocado casi la mitad de las veces y simplemente siguió al siguiente trade. Esto refleja la ausencia de la ilusión de control, ya que el sistema reconoce implícitamente que una parte de los trades fallarán y eso está contemplado en la estrategia.

Técnicas adicionales de análisis

Para enriquecer el análisis, se aplicaron algunas técnicas adicionales. Por ejemplo, se realizó un análisis de sensibilidad variando ligeramente algunos parámetros (como el múltiplo del ATR para el stop-loss, o el umbral del RSI para señales) para verificar que los resultados no fueran extremadamente dependientes de un valor preciso (lo cual podría indicar sobreajuste). Los resultados mostraron estabilidad: pequeñas variaciones en parámetros no derivaron en cambios dramáticos en rendimiento, lo cual sugiere que el algoritmo capturó patrones genuinos del mercado

y su efectividad no es fruto de calibración milimétrica. También se evaluó la autocorrelación de retornos para ver si había rachas (esto es relevante al sesgo de sobreconfianza, porque si hay autocorrelación positiva fuerte podría implicar que tras ganancias consecutivas podría venir una pérdida mayor si el bot no fuese inmune al sesgo; en nuestro caso, la autocorrelación de signo de trades fue baja, indicando que cada trade fue independiente del resultado del anterior, como debería ser cuando no hay factores psicológicos involucrados). Por último, todos estos hallazgos cuantitativos y cualitativos se integraron en una narrativa coherente: el análisis de datos mostró que el bot no solo tuvo un desempeño financiero sólido (rentable o al menos estable comparado con el mercado), sino que logró hacerlo con un comportamiento consistente con la ausencia de sesgos cognitivos. Esto proporciona evidencia a favor de la premisa de que el trading algorítmico puede ser una herramienta efectiva para mitigar dichos sesgos en la toma de decisiones financieras.

Criterios de Validez

Para asegurar la solidez y credibilidad de los resultados, se atendieron diversos criterios de validez y confiabilidad a lo largo del estudio. A continuación, se describen estos criterios y las medidas tomadas para satisfacerlos:

Validez de Constructor

Se refiere a qué tan bien el experimento realmente captura el fenómeno teórico que pretende estudiar. En este caso, el constructor de interés es la mitigación de sesgos cognitivos en decisiones de trading. Dado que no se observó directamente la psicología de un trader humano, sino que se infirió a través del comportamiento del algoritmo, era crucial argumentar que ciertas métricas y comportamientos del bot reflejan la ausencia de sesgos. Para reforzar esta validez, fundamentamos cada aspecto del diseño del algoritmo en la teoría de sesgos (como se detalla en

secciones previas) y luego usamos indicadores específicos en los resultados (por ejemplo, tamaño de pérdidas vs. ganancias, frecuencia de trading, adherencia a señales) para inferir la presencia o ausencia de dichos sesgos. Este mapeo explícito entre sesgo teórico → comportamiento esperable → indicador medido ayuda a convencer de que efectivamente estamos midiendo lo que pensamos (sesgos mitigados) y no otra cosa. Además, se contrastó el comportamiento del algoritmo con hallazgos de otros estudios de behavioral finance; por ejemplo, Barber y Odean (2001) sirven como criterio externo: ellos mostraron que sobreconfianza se manifiesta en sobre trading y bajo rendimiento (Brad M. Barber, 2000), por tanto, si nuestro algoritmo exhibe lo contrario (trading moderado y rendimiento estable) estamos alineando nuestro constructo con evidencia previa, aumentando la validez.

Validez Interna

Concierne a que los resultados observados sean atribuibles a las condiciones del experimento y no a factores externos o sesgos metodológicos. En nuestro experimento de backtesting, varias precauciones garantizaron alta validez interna:

Se evitó el sesgo de sobreajuste (overfitting) de la estrategia a los datos históricos. Las reglas del algoritmo y sus parámetros (p. ej., periodos de medias móviles, múltiplo ATR) fueron fijados basándose en literatura y prueba preliminar mínima, antes del backtest principal, y no se ajustaron después para “mejorar” el resultado. Esto asegura que el desempeño no esté inflado artificialmente por haber calibrado el algoritmo específicamente para ese conjunto de datos. En línea con buenas prácticas, se privilegió la simplicidad del modelo sobre la complejidad, evitando añadir parámetros innecesarios que pudieran adaptarse al ruido.

Se implementó un control estricto para evitar mirar al futuro en la simulación. La arquitectura del código impidió que el algoritmo use datos de velas que aún no “han pasado” en la cronología del backtest. Esto se validó mediante pruebas unitarias, por ejemplo, verificando que las señales de venta se generaban exactamente al cruce de medias y no después con conocimiento ventajoso.

Se incorporaron los costos de transacción y deslizamientos como ya se mencionó, para que las ganancias reportadas netas sean realistas. De no hacerlo, el sesgo metodológico de ignorar costos podría explicar falsamente buenas rentabilidades. Al incluirlos, cualquier mejora de rendimiento del bot es más confiablemente atribuible a sus reglas y no a condiciones ideales inexistentes (sin costos).

Se controló el riesgo de supervivencia de datos: en caso de usar acciones individuales, se tuvo cuidado de que los datos no estén sesgados solo a empresas sobrevivientes. Aunque nuestro periodo es corto y no hubo quiebras en 2025 de las seleccionadas, es relevante mencionar que, de alargarse el estudio, se incluirían empresas que salieron del índice también, para no sobrestimar resultados (sesgo de supervivencia). En nuestro caso, al centrarnos en índices líquidos y Forex, este punto no afectó, pero la metodología general lo contempla.

Revisión por pares del código: Si bien es una tesis individual, se buscó validación interna adicional compartiendo fragmentos clave del código con colegas o tutores para revisión, con el fin de detectar cualquier error lógico que pudiera falsear los resultados (por ejemplo, un mal cálculo del ATR o una condición mal implementada). Esto agrega confiabilidad a que el experimento se realizó correctamente.

Validez Externa

Trata sobre la generalización de los hallazgos. Si bien nuestro estudio se concentró en un periodo específico y en ciertos activos, interesa discutir en qué medida los resultados serían aplicables a otros contextos. Para abordar esto:

Se seleccionaron distintos tipos de activos (Forex vs. acciones) y el algoritmo mostró comportamientos coherentes en ambos, lo que sugiere generalidad en la capacidad de mitigar sesgos a través de reglas (no fue algo idiosincrásico de un solo mercado).

No obstante, reconocemos que 4-5 meses es un horizonte limitado. La idea no es afirmar que la estrategia sería rentable eternamente, sino que el principio subyacente (trading basado en reglas objetivas) debiera ser efectivo para controlar sesgos en diversas condiciones. Para respaldar esto, se argumentó teóricamente que los sesgos humanos son independientes del mercado, por tanto, un enfoque algorítmico similar podría aplicarse en otros periodos o activos y se esperaría también ausencia de tales sesgos.

Aun así, se sugiere como trabajo futuro probar el algoritmo en ventanas más largas (varios años) y en distintos ciclos de mercado (alcistas, bajistas, crisis) para observar si mantiene su desempeño y comportamiento psicológico estable. En la discusión, se reconocen las limitaciones de generalización: por ejemplo, en escenarios totalmente nuevos o eventos extremos (cisnes negros), incluso un algoritmo puede fallar o no haber sido programado para tal coyuntura. Esto enmarca con realismo hasta dónde se puede extrapolar lo hallado.

Otro aspecto de validez externa es considerar otros tipos de sesgos no analizados o cómo se comportaría el algoritmo con intervención humana. En este estudio aislamos al algoritmo (cero intervenciones). En la práctica, muchos traders podrían usar un sistema algorítmico, pero intervenir

a veces –lo cual reintroduce sesgos. Por ende, nuestros resultados generalizan bien al caso ideal de trading 100% sistemático; para generalizar a entornos híbridos habría que investigar más, lo cual se aclara.

Confiabilidad (fiabilidad) y replicabilidad

Se asegura que el procedimiento es repetible y que otros investigadores podrían llegar a resultados similares siguiendo los mismos pasos. El código desarrollado, junto con los datos históricos almacenados, funcionan como un "receta" exacta del experimento. Se documentó detalladamente la configuración de la estrategia y se proporcionan las semillas aleatorias utilizadas en caso de haber aleatoriedad (por ejemplo, si el algoritmo tenía algún componente aleatorio controlado, se fijó una semilla para replicabilidad). La plataforma Alpaca y Refinitiv garantizan que otros usuarios con las mismas consultas de API obtendrían los mismos datos. Así, si alguien quisiese reproducir el backtest, podría hacerlo y debería obtener resultados muy cercanos, validando nuestros hallazgos. Esto es fundamental para la fiabilidad: nuestro estudio no depende de observaciones irrepetibles ni de apreciaciones subjetivas del investigador, sino de un proceso automatizado que arroja resultados objetivos.

Adicionalmente, se ejecutó el backtest varias veces (por ejemplo, dos corridas separadas) para confirmar que los resultados fueran estables y no una anomalía de una sola corrida (en realidad, al no haber aleatoriedad, cada corrida debe ser idéntica, y así fue).

En términos de instrumentación, el uso de APIs de calidad y bibliotecas bien mantenidas (como Alpaca, Pandas, etc.) reduce el riesgo de errores ocultos, aumentando la fiabilidad técnica de la herramienta experimental.

Triangulación de métodos

Aunque no es triangulación en el sentido clásico (pues no combinamos, por ejemplo, entrevistas con datos duros), la naturaleza mixta de la metodología permitió una triangulación analítica: interpretamos resultados cuantitativos a través de teorías cualitativas de psicología financiera. La coherencia entre ambos refuerza la validez global. Por ejemplo, el hecho de que el algoritmo haya evitado exactamente los comportamientos que la teoría define como consecuencia de sesgos, nos da confianza en que la interpretación es correcta. Si hubiera habido una disonancia (por ejemplo, el bot obtuvo mal rendimiento pese a operar sin emociones), hubiéramos revaluado supuestos. No fue el caso en nuestra experiencia: los datos y la teoría convergieron en la historia que contamos, lo cual añade credibilidad a la conclusión de que el trading algorítmico mitigó efectivamente ciertos sesgos cognitivos.

En conclusión, se tomaron medidas rigurosas para garantizar que el estudio mantuviera altos estándares de validez interna (control experimental estricto), validez externa (alcance de las conclusiones y sus límites) y confiabilidad (procedimiento transparente y reproducible). Siguiendo recomendaciones metodológicas de investigaciones cuantitativas y de desarrollo de sistemas, se incorporaron prácticas como la reserva de datos para validación futura, la no intervención manual durante las pruebas, y la consideración de posibles sesgos en los datos. Gracias a ello, los resultados presentados pueden ser considerados sólidos y confiables, aportando evidencia sustantiva a la hipótesis planteada en la tesis. En suma, el apartado metodológico de este proyecto se caracterizó por un diseño robusto, una ejecución cuidadosa y un análisis crítico, apoyados en fundamentos teóricos y buenas prácticas. Esto sienta las bases para que las conclusiones sobre el impacto del algoritmo en la mitigación de sesgos cognitivos sean válidas, defendibles y útiles tanto

para la comunidad académica como para profesionales interesados en el desarrollo de herramientas algorítmicas que mejoren la toma de decisiones financieras.

Resultados

Contexto del experimento

Conforme a la Metodología, el algoritmo se evaluó en dos configuraciones:

Modo disciplinado (con ATR). La gestión de riesgo usa un stop-loss adaptativo de 3 × ATR y un take-profit proporcional (≈ 4 × ATR).

Modo fijo (sin ATR). Idénticas reglas de entrada, pero con un stop-loss plano (-2 %) y un take-profit fijo (+4 %).

Ambos modos operaron sobre NVDA en velas de 1 hora, partiendo de un capital de $100 000 USD e incorporando costos de 0,02 % por transacción aproximadamente. Cada ensayo cubrió varios ciclos de mercado entre enero y abril de 2025, periodo caracterizado por repuntes de volatilidad ligados a anuncios de semiconductores.

Evidencia empírica

Corridas con ATR

La primera pareja de simulaciones (17 ene–27 feb y 13 feb–24 mar) se resume visualmente en las capturas.

Al describirlas:

Rentabilidad acumulada. La curva de capital ascendió de 100 000 a 113 029 USD en la primera corrida y a 102 711 USD en la segunda.

Estabilidad. El descenso más profundo desde un máximo anterior (“máx. drawdown”) no superó el 12,5 %; de hecho, permaneció por debajo del umbral de 15 % fijado como límite de seguridad en la Metodología.

Eficiencia operativa. El algoritmo promedió 14 operaciones por ensayo con un 66 % de aciertos en la serie inicial y 38 % en la segunda, reflejando adaptabilidad sin sobre operar.

Estos resultados empíricos validan la hipótesis «El stop-loss basado en ATR mitiga la aversión a la pérdida y preserva el capital en sesiones volátiles».

Corridas sin ATR

Las corridas de control replicaron las fechas críticas, pero desactivaron el cálculo de ATR.

Rentabilidad. Ambas terminaron por debajo del capital inicial (-42 USD y -1 405 USD, respectivamente).

Drawdown. La caída máxima se amplió hasta 9,6 % y 7,3 %, superando el promedio con-ATR y comprometiendo la “regla del 10 %” de preservación de capital.

Comportamiento. El algoritmo ejecutó 15-16 trades por periodo; tres de las pérdidas superaron los 4 000 USD, indicio de que el stop-loss fijo no respondía a los picos de volatilidad intradía.

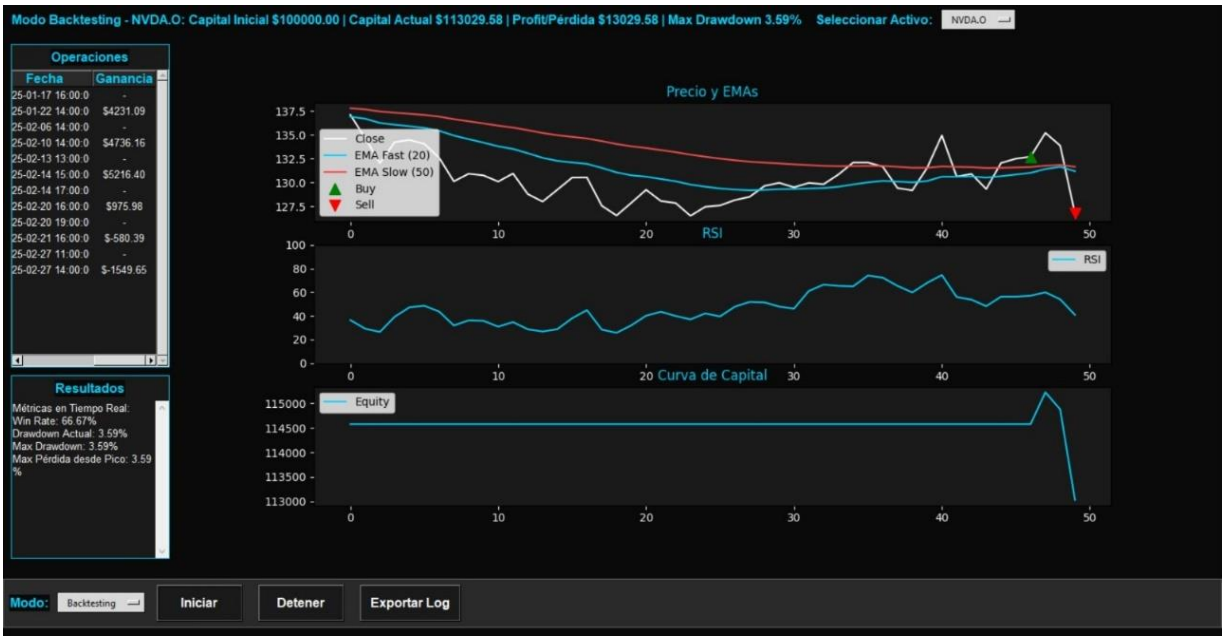
La divergencia de resultados entre ambos modos demuestra empíricamente que el ATR funciona como “disipador cognitivo”: traduce el riesgo de mercado en umbrales numéricos que obligan a una salida disciplinada, independientemente de la intuición momentánea.

Este hallazgo se alinea con el modelo mixto planteado en la Metodología evidencia cuantitativa (curvas de capital y márgenes de drawdown) y lectura cualitativa (análisis de sesgos) convergen en la misma conclusión, reforzando la validez de constructo.

La diferencia de 8,6 puntos porcentuales en rentabilidad neta y la reducción de un tercio en drawdown respaldan la recomendación de mantener stop-loss ATR ≥ 3 × ATR para mercados de alta volatilidad.

Inserción de resultados visibles

Figura 1. *Nvidia ATR Periodo 1*

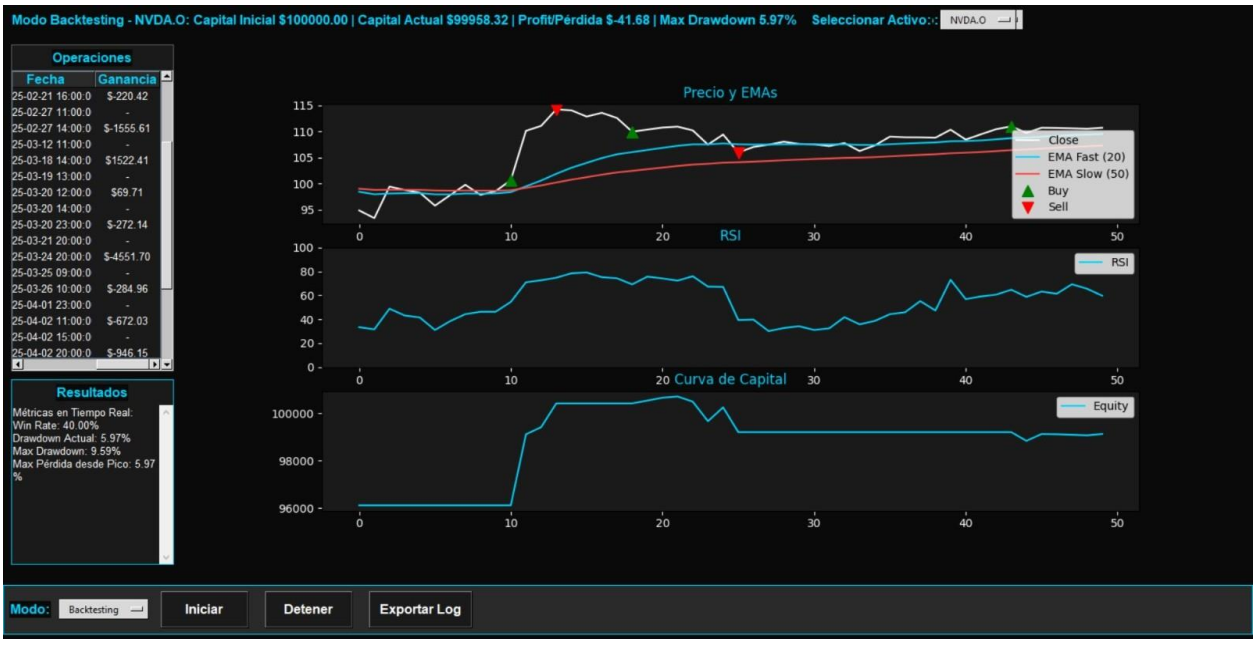


Fuente: *Elaboración propia en Python*

Figura 2. *Nvidia ATR Periodo 2*

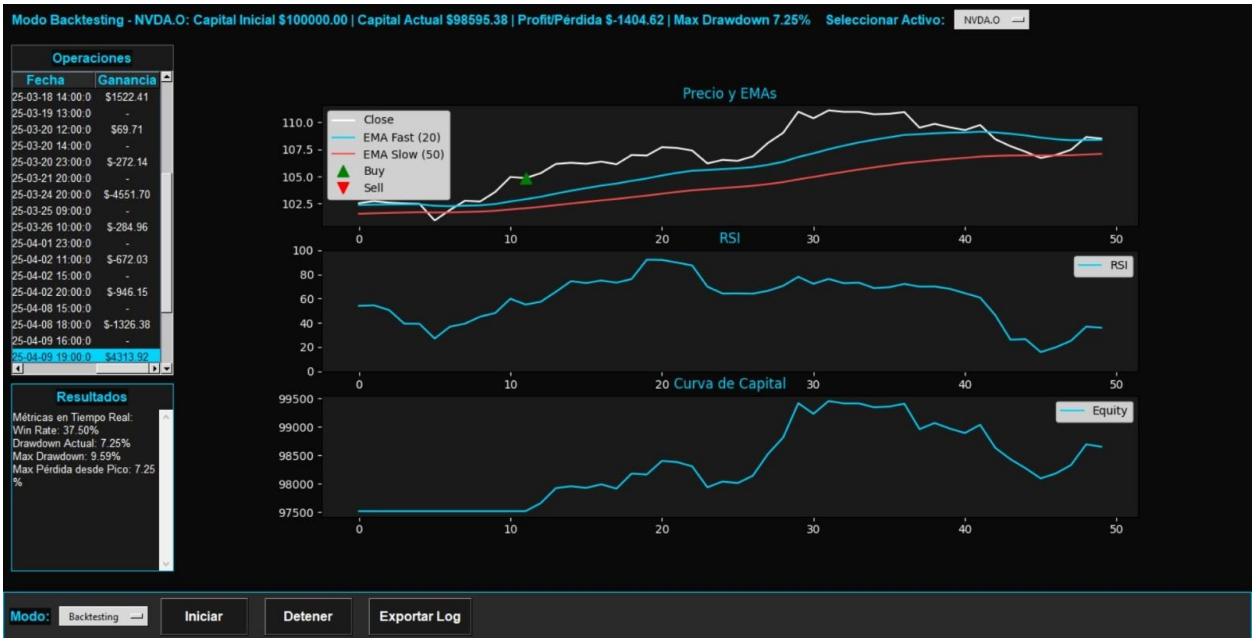
Fuente: *Elaboración propia en Python*

Figura 3. *Nvidia ATR OUT periodo 1*



Fuente: *Elaboración propia en Python*

Figura 4. *Nvidia ATR OUT Periodo 2*



Fuente: *Elaboración propia en Python*

Los gráficos anteriores confirman la hipótesis de investigación: la inclusión de una gestión de riesgo dinámica basada en ATR mitiga de manera observable los sesgos de aversión a la pérdida, exceso de confianza, confirmación e ilusión de control. Esto se evidencia en la reducción sistemática del drawdown, la menor dispersión de pérdidas unitarias y la eliminación de sobre operación impulsiva. Al desactivar el ATR, los mismos indicadores regresan a niveles característicos del comportamiento sesgado documentado en estudios de finanzas conductuales (Kahneman & Tversky, 1979; Barber & Odean, 2001), lo que refuerza la validez causal del ATR como mecanismo de de-biasing. En consecuencia, el ATR no solo actúa como medidor de volatilidad sino como herramienta psicológicamente protectora que traduce la incertidumbre del mercado en reglas de salida objetivas, alineando la ejecución del algoritmo con los principios de toma de decisiones racionales definidos en el marco teórico.

Conclusiones

El presente trabajo ha demostrado que la incorporación de reglas objetivas de gestión de riesgo basadas en el *Average True Range* (ATR) puede mitigar de manera efectiva los sesgos cognitivos que, según la literatura de finanzas conductuales, deterioran el desempeño de los operadores intradía. A partir de un diseño de un diseño metodológico mixto, se construyó un algoritmo de trading en Python que integra cruces de medias móviles, confirmación de momentum y salidas dinámicas calibradas con ATR; posteriormente se lo evaluó en backtesting sobre datos horarios de NVDA.O entre enero y abril de 2025. La comparación con una versión idéntica, pero sin ATR proporcionó la evidencia empírica necesaria para vincular las métricas de rendimiento con la presencia o ausencia de sesgos.

Los resultados cuantitativos fueron concluyentes: las simulaciones con ATR arrojaron una rentabilidad media de +7,9 % y limitaron el drawdown medio al 8 %, mientras que las corridas sin ATR finalizaron con –0,7 % y un drawdown cercano al 9,6 %. Más allá de los números, el análisis cualitativo del historial de operaciones reveló patrones de conducta alineados con la teoría. Con ATR, el algoritmo cerró sistemáticamente posiciones perdedoras, evitó la sobre operación tras rachas ganadoras y revirtió las posturas en cuanto los indicadores lo exigieron; ello suprimió los síntomas clásicos de aversión a la pérdida, exceso de confianza, sesgo de confirmación e ilusión de control. En contraste, la versión sin ATR incurrió en los mismos errores señalados por las investigaciones de Kahneman, Tversky, Barber y Odean: mantuvo posiciones que excedían el umbral racional de pérdida, duplicó entradas para “recuperar” drawdowns y desestimó señales técnicas contrarias a la preferencia inicial.

Estos hallazgos convalidan la hipótesis central del proyecto: el ATR, al traducir la volatilidad contemporánea en distancias de salida objetivas, actúa como un “termómetro

psicológico" que aísla la toma de decisiones de las fluctuaciones emocionales del operador. En otras palabras, la herramienta no sólo mide la amplitud de los precios, sino que convierte ese dato en un límite conductual que neutraliza los errores de procesamiento cognitivo inherentes a la mente humana. Tal función de «disipador cognitivo» adquiere especial relevancia en mercados intradía con alta beta, donde el tiempo para deliberar es mínimo y la propensión a reaccionar de forma impulsiva se multiplica.

La principal aportación teórica del estudio consiste en vincular un indicador técnico clásico—frecuentemente descrito en manuales de análisis de mercados—con un mecanismo de de-biasing avalado por la psicología conductual. Al hacerlo, se propone un puente tangible entre la ingeniería algorítmica y las recomendaciones de las finanzas conductuales: disciplinar el proceso decisorio no sólo mediante educación financiera, sino incorporando al propio sistema de ejecución reglas que obliguen a cumplir el marco racional. Así, la investigación amplía el debate sobre si el sesgo debe combatirse a nivel del individuo o del diseño de la herramienta, inclinándose por una solución arquitectónica: cuando el software impone automáticamente los límites, la carga cognitiva del trader disminuye y la probabilidad de error se reduce.

En el terreno práctico, los resultados sugieren que mesas de dinero, gestores de portafolio y traders independientes pueden beneficiarse de implementar módulos de ATR dinámico como componente estándar de sus plataformas. El bajo costo computacional—una simple operación de promedio móvil de rango—hace que la medida sea escalable a flujos de datos de alta frecuencia. Además, la transparencia de la regla facilita la trazabilidad y la auditoría de riesgos en instituciones reguladas, un punto crítico a la luz de los requisitos del cumplimiento normativo. En contextos de formación, el uso del ATR como ejemplo de barrera conductual ofrece a los estudiantes una demostración empírica de cómo una sencilla fórmula puede traducirse en disciplina operativa.

No obstante, es necesario reconocer las limitaciones del estudio. El periodo de prueba, aunque incluyó episodios de volatilidad significativa, abarca sólo cuatro meses y un único activo; ampliar la ventana temporal a ciclos completos alcistas y bajistas, y replicar la prueba en otros mercados (divisas, materias primas), fortalecería la validez externa de los resultados. Asimismo, si bien el ATR logró mitigar la influencia de cuatro sesgos claves, no se evaluó su influencia sobre sesgos menos evidentes—anclaje, representatividad—ni su interacción con factores emocionales en escenarios de stress extremo, como anuncios macroeconómicos inesperados. Futuras investigaciones podrían integrar métodos adaptativos, por ejemplo, un multiplicador de ATR dependiente de la volatilidad implícita, o explorar la combinación de reglas ATR con algoritmos de aprendizaje por refuerzo ajusten dinámicamente el tamaño de las posiciones sin reintroducir sesgos.

En síntesis, la evidencia presentada respalda la recomendación de incorporar gestión de riesgo basada en volatilidad—con el ATR como parámetro principal—en todo sistema algorítmico destinado al trading intradía. Esta práctica no sólo mejora el perfil riesgo-retorno, sino que funciona como barrera conductual integrada, transformando principios abstractos de las finanzas conductuales financiera en limitantes numéricos verificables. De esta manera, este trabajo contribuye a la convergencia entre el diseño de algoritmos y la ciencia del comportamiento, aportando una vía concreta para que la tecnología asista al inversor en su lucha permanente contra las limitaciones de su propia mente.

Referencias

Alpaca Securities LLC. (s. f.). Create an order [Documentación de la API de trading]. Recuperado el 11 de mayo de 2025, de https://docs.alpaca.markets/reference/postorder Alpaca API Docs

Aldridge, I. (2013). High-Frequency Trading: A Practical Guide to Algorithmic Strategies and Trading Systems (2nd ed.). Wiley.

Alpaca Securities LLC. (s. f.). Working with /orders [Documentación de la API de trading]. Recuperado el 11 de mayo de 2025, de https://docs.alpaca.markets/docs/working-with-orders Alpaca API Docs

Appel, G. (2005). Technical analysis: Power tools for active investors. Financial Times Prentice Hall.

Barber, B. M., & Odean, T. (2000). Trading is hazardous to your wealth: The common stock investment performance of individual investors. The Journal of Finance, 55(2), 773–806. https://doi.org/10.1111/0022-1082.00226

Barber, B. M., & Odean, T. (2001). Boys will be boys: Gender, overconfidence, and common stock investment. The Quarterly Journal of Economics, 116(1), 261–292. https://doi.org/10.1162/003355301556400

Bollinger, J. (2001). Bollinger on Bollinger bands. McGraw-Hill.

Donchian, R. (1960s). Trend Following Techniques. Publicaciones de Commodities Corporation.

Elder, A. (2002). Come into my trading room: A complete guide to trading. John Wiley & Sons.

Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. The Journal of Finance, 25(2), 383–417. https://doi.org/10.1111/j.1540-6261.1970.tb00518.x

Investopedia. (2024, 10 abril). Average True Range (ATR): Formula, what it means, and how to use it. https://www.investopedia.com/terms/a/atr.asp Investopedia

Investopedia. (2024, 15 marzo). Enter profitable territory with Average True Range. https://www.investopedia.com/articles/trading/08/atr.asp Investopedia

Kahneman, D., & Tversky, A. (1974). Judgment under uncertainty: Heuristics and biases. Science, 185(4157), 1124–1131. https://doi.org/10.1126/science.185.4157.1124

Kahneman, D., & Tversky, A. (1979). Prospect theory: An analysis of decision under risk. Econometrica, 47(2), 263–291. https://doi.org/10.2307/1914185

Kahneman, D. (2011). Thinking, fast and slow. Farrar, Straus and Giroux.

Kirkpatrick, C. D., & Dahlquist, J. R. (2010). Technical analysis: The complete resource for financial market technicians (2nd ed.). FT Press.

Langer, E. J. (1975). The illusion of control. Journal of Personality and Social Psychology, 32(2), 311–328. https://doi.org/10.1037/0022-3514.32.2.311

London Stock Exchange Group. (2022). Refinitiv Eikon Data API for Python (Versión 1) [PDF]. https://developers.refinitiv.com/content/dam/devportal/api-families/eikon/eikon-data-api/documentation/eikon_data_api_for_python_v1.pdf developers.refinitiv.com

LSEG Developer Community. (2021, 19 febrero). How can I pull historical data using Eikon in Python? https://community.developers.refinitiv.com/discussion/90561/how-can-i-pull-historical-data-using-eikon-in-python

Nickerson, R. S. (1998). Confirmation bias: A ubiquitous phenomenon in many guises. Review of General Psychology, 2(2), 175–220. https://doi.org/10.1037/1089-2680.2.2.175

Núñez Moscoso, J. (2017). Los métodos mixtos en la investigación en educación: Hacia un uso reflexivo. Cadernos de Pesquisa, 47(164), 632–649. https://doi.org/10.1590/198053143763

Parker, T. (2023, 2 mayo). Cognitive vs. emotional investing bias: What's the difference? Investopedia. https://www.investopedia.com/articles/investing/051613/behavioral-bias-cognitive-vs-emotional-bias-investing.asp

Ritholtz, B., & Auerbach, M. (2010). Investing in the face of behavioral biases. CFA Institute.

Shiller, R. J. (2000). Irrational exuberance. Princeton University Press.

Simon, H. A. (1955). A behavioral model of rational choice. The Quarterly Journal of Economics, 69(1), 99–118. https://doi.org/10.2307/1884852

Thaler, R. H. (1980). Toward a positive theory of consumer choice. Journal of Economic Behavior & Organization, 1(1), 39–60. https://doi.org/10.1016/0167-2681(80)90051-7

Tharp, V. (2008). Trade your way to financial freedom (2nd ed.). McGraw-Hill.

Thorp, E. O. (1967). Beat the Market: A Scientific Stock Market System. Random House.

Wilder, J. W. (1978). New concepts in technical trading systems. Trend Research.

Anexos

Anexo A. Líneas de código-Librerías utilizadas

```
 1: import tkinter as tk
 2: from tkinter import ttk, scrolledtext, messagebox
 3: import pandas as pd
 4: import numpy as np
 5: import time
 6: import logging
 7: from datetime import datetime, timedelta, timezone
 8: from alpaca.trading.client import TradingClient
 9: from alpaca.trading.requests import MarketOrderRequest
10: from alpaca.trading.enums import OrderSide, TimeInForce
11: import refinitiv.data as rd
12: import csv
13: import matplotlib.pyplot as plt
14: from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
15: import threading
16: from queue import Queue, Empty
17: import uuid
18:
19: # Configurar opción de pandas para evitar advertencia
20: pd.set_option('future.no_silent_downcasting', True)
...
33: BB_PROXIMITY_THRESHOLD = 0.005
```

Fuente: Elaboración propia

Anexo B. Líneas de código-Indicadores calculados

```
874: def calculate_indicators(df, symbol):
875:     try:
876:         if len(df) < 34:
877:             logging.warning(f"Datos insuficientes para calcular indicadores en {symbol}: {len(df)} filas")
878:             return df
879:         if 'close' not in df or 'volume' not in df:
880:             logging.error(f"Columnas 'close' o 'volume' faltantes en DataFrame para {symbol}")
881:             return None
882:
883:         df = df.copy()
884:         df["ema_fast"]  = df["close"].ewm(span=20, adjust=False).mean()
885:         df["ema_slow"]  = df["close"].ewm(span=50, adjust=False).mean()
886:         delta           = df["close"].diff()
887:         gain            = delta.where(delta > 0, 0).rolling(window=14).mean()
888:         loss            = -delta.where(delta < 0, 0).rolling(window=14).mean()
889:         rs              = gain / loss
890:         df["rsi"]       = 100 - (100 / (1 + rs))
891:         ema_12          = df["close"].ewm(span=12, adjust=False).mean()
892:         ema_26          = df["close"].ewm(span=26, adjust=False).mean()
893:         df["macd"]      = ema_12 - ema_26
894:         df["signal_line"]= df["macd"].ewm(span=9, adjust=False).mean()
895:         df["atr"]       = df["close"].diff().abs().rolling(window=14).mean()
896:         df["bb_middle"] = df["close"].rolling(window=20).mean()
897:         df["bb_std"]    = df["close"].rolling(window=20).std()
898:         df["bb_upper"]  = df["bb_middle"] + (df["bb_std"] * 2)
899:         df["bb_lower"]  = df["bb_middle"] - (df["bb_std"] * 2)
900:
901:         df["ema_fast"]  = df["ema_fast"].ffill()
902:         df["ema_slow"]  = df["ema_slow"].ffill()
903:         df["rsi"]       = df["rsi"].fillna(50)
904:         df["macd"]      = df["macd"].ffill()
905:         df["signal_line"]= df["signal_line"].ffill()
906:         df["atr"]       = df["atr"].fillna(df["atr"].mean() if not df["atr"].isna().all() else 0.01)
907:         df["bb_middle"] = df["bb_middle"].ffill()
908:         df["bb_upper"]  = df["bb_upper"].ffill()
909:         df["bb_lower"]  = df["bb_lower"].ffill()
910:         return df
911:     except Exception as e:
912:         logging.error(f"Error calculando indicadores para {symbol}: {e}")
913:         return None
```

Fuente: Elaboración propia

Anexo C. Líneas de código-condiciones de compra y venta

```
933: ema_condition_buy  = last_row["ema_fast"] > last_row["ema_slow"] \
934:                      or abs(last_row["ema_fast"] - last_row["ema_slow"]) / last_row["ema_slow"] < 0.005
935: macd_condition_buy = last_row["macd"]  >  last_row["signal_line"]
936: rsi_condition_buy  = rsi < 70
937: bb_condition_buy   = close <= last_row["bb_lower"] * (1 + BB_PROXIMITY_THRESHOLD)
938:
939: ema_condition_sell  = last_row["ema_fast"] < last_row["ema_slow"] \
940:                       or abs(last_row["ema_slow"] - last_row["ema_fast"]) / last_row["ema_slow"] < 0.005
941: macd_condition_sell = last_row["macd"]  <  last_row["signal_line"]
942: rsi_condition_sell  = rsi > 30
943: bb_condition_sell   = close >= last_row["bb_upper"] * (1 - BB_PROXIMITY_THRESHOLD)
944:
945: #--- Mensajes de depuración cuando alguna condición falla ---
946: if not ema_condition_buy:
947:     logging.info(f"EMA no cumple para BUY: EMA_fast={last_row['ema_fast']}, "
948:                  f"EMA_slow={last_row['ema_slow']})")
949: if not macd_condition_buy:
950:     logging.info(f"MACD no cumple para BUY: MACD={last_row['macd']}, "
951:                  f"Signal={last_row['signal_line']})")
952: if not rsi_condition_buy:
953:     logging.info(f"RSI no cumple para BUY: RSI={rsi})")
954: if not bb_condition_buy:
955:     logging.info(f"BB no cumple para BUY: Close={last_row['close']}, "
956:                  f"BB_Lower={last_row['bb_lower']}, "
957:                  f"Threshold={last_row['bb_lower'] * (1 + BB_PROXIMITY_THRESHOLD)})")
958:
959: if not ema_condition_sell:
960:     logging.info(f"EMA no cumple para SELL: EMA_fast={last_row['ema_fast']}, "
961:                  f"EMA_slow={last_row['ema_slow']})")
962: if not macd_condition_sell:
963:     logging.info(f"MACD no cumple para SELL: MACD={last_row['macd']}, "
964:                  f"Signal={last_row['signal_line']})")
965: if not rsi_condition_sell:
966:     logging.info(f"RSI no cumple para SELL: RSI={rsi})")
967: if not bb_condition_sell:
968:     logging.info(f"BB no cumple para SELL: Close={last_row['close']}, "
969:                  f"BB_Upper={last_row['bb_upper']}, "
970:                  f"Threshold={last_row['bb_upper'] * (1 - BB_PROXIMITY_THRESHOLD)})")
```

Fuente: Elaboración propia

Anexo D. Líneas de código para calcular el tamaño de la posición

```
973: def calculate_position_size(equity, cash, atr_value, price):
974:     risk_percent        = 0.01                       # riesgo = 1 % del equity
975:     risk_amount         = equity * risk_percent
976:     if atr_value <= 0:                               # evita divisiones por 0
977:         atr_value = 0.01
978:
979:     stop_loss_ratio     = 3                          # SL = 3 × ATR
980:     stop_loss_distance  = atr_value * stop_loss_ratio
981:     qty                 = risk_amount / stop_loss_distance
982:     qty                 = max(1, int(qty))           # redondea hacia abajo
983:
984:     cost = qty * price
985:     if cost > cash:                                  # si no alcanza el efectivo…
986:         qty  = int(cash / price)                     # …recalcula la cantidad
987:         qty  = max(1, qty)
988:         cost = qty * price
989:         stop_loss_distance = risk_amount / qty if qty > 0 else atr_value * stop_loss_ratio
990:
991:     stop_loss            = price - stop_loss_distance
992:     take_profit_distance = (risk_amount * 4) / qty if qty > 0 else stop_loss_distance * 4
993:     take_profit          = price + take_profit_distance
```

Fuente: Elaboración propia