



POLITÉCNICA



Universidad
Politécnica
de Madrid

**ETSI SISTEMAS
INFORMÁTICOS**

Bots de Trading en Mercados Financieros

Proyecto Fin de Grado

Grado en Ingeniería del Software

Autor:

Iván Nevado Martín

Tutores:

Carlos Camacho

Junio 2025

AGRADECIMIENTO

Quiero expresar mi más sincero agradecimiento a mi tutor por su guía y apoyo durante la elaboración de este Trabajo de Fin de Grado. Su experiencia y conocimiento han sido fundamentales para el desarrollo de este proyecto. También agradezco a mi familia y amigos por su constante apoyo y motivación a lo largo de este proceso.

Desde una edad temprana, mi interés por el mundo financiero se vio influenciado por conversaciones que tuve con mi padre. Recuerdo con especial cariño un día que fuimos a la montaña cuando tenía 13 años, y él me habló sobre temas que, en ese momento, parecían complejos pero fascinantes: los ETFs (*Exchange Traded Funds*), los fondos cotizados en bolsa y la inversión en acciones. Aunque las criptomonedas aún no tenían de la popularidad que tienen hoy en día, mi curiosidad por los mercados financieros ya empezaba a surgir. Mi padre recuerda cómo le hacía preguntas incesantes, intentando entender cómo funcionaba todo aquello. Desde entonces, mi familia me empezó a llamar "Tío Gilito", en referencia al personaje de dibujos animados obsesionado con el dinero, dado mi creciente interés por el valor del dinero y la inversión.

A lo largo de los años, esa curiosidad inicial fue creciendo, pero no fue hasta el año posterior a la pandemia cuando descubrí las criptomonedas, un sector que capturó por completo mi atención. Pasé largas horas investigando cómo funcionaban los *exchanges*, el concepto del *market cap*, y la tecnología subyacente, como la *blockchain*. Sin embargo, lo que más me atraía en aquel momento era entender cómo podría obtener beneficios a largo plazo comprando y manteniendo criptomonedas. Mi enfoque no era el trading activo, sino adquirir proyectos en los que creía firmemente tras un exhaustivo análisis, y esperar su revalorización.

No obstante, mi primera incursión en el mundo de las criptomonedas no fue como esperaba. Invertí alrededor de 2,000 euros justo antes de que comenzara un mercado bajista, lo que provocó que perdiera el 90% de mi inversión en pocos meses. Este revés me enseñó una lección fundamental: no arriesgar más dinero del que se está dispuesto a perder. Tras esta experiencia, decidí tomar un descanso.

Un año después, retomé mi viaje en el mundo financiero, pero esta vez centrándome en el trading. Tras estudiar múltiples estrategias y tomar cursos con los que considero algunos de los mejores en el campo, comencé a operar de manera manual. Este proceso me llevó a dar el siguiente paso en mi formación: aprender a automatizar mis estrategias para optimizar mi operativa.

RESUMEN

En este trabajo se aborda el estudio de los mercados financieros, la implementación de sistemas automatizados de trading, conocidos como bots de trading, y su optimización mediante algoritmos metaheurísticos. Estas herramientas han experimentado un auge significativo, permitiendo a inversores, tanto profesionales como particulares, participar en los mercados de manera más eficiente. Sin embargo, una gran cantidad de usuarios se adentra en los mercados sin contar con los conocimientos necesarios, lo que frecuentemente deriva en decisiones financieras erróneas y pérdidas económicas.

En primer lugar, se analizan las bases del trading en los mercados financieros, explicando el funcionamiento de estos entornos y las principales estrategias empleadas por los operadores. Posteriormente, el enfoque se desplaza hacia los bots de trading, profundizando en su funcionamiento y en la oferta actual disponible en el mercado. Se presta especial atención a la proliferación de bots que prometen rentabilidades atractivas, pero que frecuentemente utilizan datos manipulados o ajustados para aparentar mejores resultados. Estas prácticas suelen incluir la manipulación de marcos temporales y otras técnicas diseñadas para inflar artificialmente el rendimiento de los sistemas.

En la siguiente sección, se desarrolla un bot de trading propio, basado en estrategias conservadoras diseñadas específicamente para gestionar de manera efectiva el riesgo y minimizar las pérdidas. Además, se realiza una comparativa entre el rendimiento de diferentes estrategias, con el objetivo de evaluar la efectividad real de los bots de trading bajo diversas condiciones de mercado. El propósito principal es determinar si estas herramientas pueden ser útiles para los inversores interesados en automatizar sus operaciones.

Finalmente, se incorporan diferentes algoritmos de optimización en una de las estrategias para identificar la combinación de parámetros que ofrezca el mejor rendimiento. Este análisis busca mejorar la efectividad de las estrategias automatizadas y aportar un enfoque práctico al diseño de sistemas de trading automatizado.

ABSTRACT

This project addresses the study of financial markets and the implementation of automated trading systems, commonly known as trading bots. These tools have experienced significant growth, enabling both professional and retail investors to participate in the markets more efficiently. However, many users enter the markets without the necessary knowledge, often leading to poor financial decisions and economic losses.

First, the fundamentals of trading in financial markets are analyzed, explaining how these environments operate and the main strategies employed by traders. Subsequently, the focus shifts to trading bots, delving into their functionality and the current offerings available in the market. Special attention is given to the proliferation of bots that promise attractive returns but frequently rely on manipulated or adjusted data to appear more successful. These practices often include manipulating timeframes and other techniques designed to artificially inflate system performance.

In the next section, a proprietary trading bot is developed, based on conservative strategies specifically designed to effectively manage risk and minimize losses. Additionally, a comparison is conducted between the performance of different strategies to evaluate the real effectiveness of trading bots under various market conditions. The primary goal is to determine whether these tools can be useful for investors interested in automating their trades.

Finally, various optimization algorithms are applied to one of the strategies to identify the combination of parameters that delivers the best performance. This analysis seeks to enhance the effectiveness of automated strategies and provide a practical approach to designing automated trading systems.

ÍNDICE

RESUMEN.....	3
ÍNDICE	5
APÉNDICE	7
ÍNDICE FIGURAS.....	9
ÍNDICE TABLAS.....	9
1. INTRODUCCIÓN.....	10
2. ANTECEDENTE	11
2.1. LAS INVERSIONES.....	11
2.1.1. ¿Qué son las inversiones?.....	11
2.1.2. Tipos de inversiones	11
2.1.3. Riesgos asociados a las inversiones.....	14
2.1.4. Inversión vs Especulación.....	17
2.1.5. Tendencias actuales	17
2.2 LOS MERCADOS FINANCIEROS.....	18
2.2.1. ¿Qué son los mercados financieros?	18
2.2.2. ¿Cómo se mueven los precios?	18
2.2.3. Tipos de mercados financieros	21
2.2.4. Elementos de los mercados financieros	26
2.2.5. Evolución de los mercados financieros	31
2.2.6. Riesgos en los mercados financieros	31
2.3. EL TRADING	32
2.3.1. Tipos de análisis	32
2.3.2. Elementos que forman parte del Trading	35
2.4. BOTS DE TRADING	42
2.4.1. Evolución histórica de los Bots de Trading	42
2.4.2. Trading Automático vs Trading Manual	46
2.4.3. Indicadores técnicos utilizados en los Bots de Trading	48
2.4.4. Inteligencia Artificial en los Bots de Trading	50
2.4.5. Aspectos Éticos y Legales del Uso de Bots	52
2.4.6. Hazañas en el Trading Algorítmico	54
2.4.7. ¿Funcionan los bots de Trading?	57
3. DESARROLLO	58
3.1. CREANDO UN BOT DE TRADING.....	58
3.1.1. ¿Qué mercado escogemos?	58
3.1.2. ¿Qué Exchange operamos?	59

3.1.3. ¿Qué activo operamos?	60
3.1.4. ¿Qué temporalidad escogemos?	60
3.2. DESCRIPCIÓN DE LOS BOTS DE TRADING.....	61
3.2.1. Bot de EMA y RSI	61
3.2.2. Bot de EMA y Order Block	62
3.2.3. Bot de Fair Value Gap	63
3.2.4. Bot Multi Timeframe Analysis	64
3.3. IMPLEMENTACIÓN TÉCNICA	66
3.3.1. Herramientas y Lenguaje de Programación Utilizados	66
3.3.2. Indicadores Realizados	66
3.3.3. Backtesting de las estrategias	71
3.3.4. Implementación de las estrategias en Phemex	86
3.3.5. Implementación de las estrategias en Hyperliquid	99
3.4. ANÁLISIS DE RESULTADOS.....	112
3.4.1. Evaluación de la Efectividad de cada Bot	112
3.4.2. Comparación del Rendimiento de los Bots	113
3.5. OPTIMIZACIÓN DE LAS ESTRATEGIAS.....	114
3.5.1. ¿Por qué usar algoritmos de optimización?	114
3.5.2. Librerías de Optimización	114
3.5.3. Comparativa de Resultados	125
3.6. DESAFÍOS ENCONTRADOS DURANTE LA IMPLEMENTACIÓN	126
4. CONCLUSIONES FINALES	127
5. LINEAS FUTURAS	128
6. PLANIFICACION TEMPORAL Y PRESUPUESTO	129
6.1. PLANIFICACIÓN TEMPORAL	129
6.2. PRESUPUESTO.....	131
6.3. Impactos Sociales y Ambientales del Proyecto	131
6.4. Responsabilidad Ética y Profesional	131
7. BIBLIOGRAFÍA.....	132

APÉNDICES

ABREVIATURAS, UNIDADES Y ACRÓNIMOS

- **EMA:** Media Móvil Exponencial
- **RSI:** Índice de Fuerza Relativa
- **FVG:** Fair Value Gap
- **CEX:** Exchange Centralizado
- **DEX:** Exchange Descentralizado
- **API:** Interfaz de Programación de Aplicaciones
- **MiFID II:** Directiva de Mercados en Instrumentos Financieros
- **NFT:** Token No Fungible
- **ETF:** Exchange-Traded Fund (Fondo Cotizado en Bolsa)
- **OB :** Order Block
- **CD:** Certificado de Depósito
- **NYSE:** New York Stock Exchange (Bolsa de valores de Nueva York)
- **EEUU:** Estados Unidos
- **CME:** Chicago Mercantile Exchange (Mercado Mercantil de Chicago)
- **CNMV:** Comisión Nacional del Mercado de Valores
- **SEC:** Securities and Exchange Commission (Comisión de Valores y Bolsa de Estados Unidos)
- **FCA:** Financial Conduct Authority (Autoridad de Conducta Financiera)
- **HFT:** High-Frequency Trading (Comercio de Alta Frecuencia)
- **IA:** Inteligencia Artificial
- **RNN:** Recurrent Neuronal Network (Redes Neuronales Recurrentes)
- **NLP:** Natural Language Processing (Procesamiento del Lenguaje Natural)

GLOSARIO

- **Exchange Centralizado (CEX):** Plataforma de intercambio de criptomonedas gestionada por una entidad centralizada que almacena los fondos de los usuarios.
- **Exchange Descentralizado (DEX):** Plataforma de intercambio de criptomonedas donde los usuarios operan directamente desde sus billeteras, sin necesidad de intermediarios.
- **Bid:** Oferta o el precio que un comprador está dispuesto a pagar por un activo, como una acción, una divisa, un bono, o cualquier otro instrumento financiero.

- **Ask:** Precio al que un vendedor está dispuesto a vender un activo, como una acción, una divisa, un bono, entre otros.
- **Inflación:** Aumento generalizado y sostenido de los precios de los bienes y servicios en una economía durante un período prolongado.
- **Liquidez:** Facilidad con la que un activo puede ser convertido en efectivo sin afectar significativamente su precio.
- **Descentralización:** Proceso de distribuir el poder, la autoridad o el control desde una entidad central hacia varias entidades o niveles inferiores, con el objetivo de que las decisiones y las funciones no sean gestionadas únicamente desde un punto central, sino que sean asumidas de manera más distribuida.
- **Futuros:** Contratos financieros que obligan a comprar o vender un activo específico (como una mercancía, un índice bursátil, o una divisa) en una fecha futura, a un precio previamente acordado.
- **Opciones:** Instrumentos financieros derivados que otorgan a los compradores el derecho, pero no la obligación, de comprar o vender un activo subyacente (como acciones, futuros, divisas, etc.) a un precio específico (llamado precio de ejercicio o strike) en una fecha determinada o antes de ella.
- **Swaps:** Contratos financieros derivados en los que dos partes acuerdan intercambiar flujos de caja futuros, basados en una cantidad o valor determinado.
- **Forward:** Contrato financiero entre dos partes en el que acuerdan comprar o vender un activo en el futuro, a un precio determinado en el momento en que se firma el contrato.
- **Altcoin:** Cualquier criptomoneda que no sea Bitcoin. El término "altcoin" proviene de la combinación de las palabras "alternative" (alternativa) y "coin" (moneda), y se usa para referirse a las monedas digitales alternativas que han surgido después de la creación de Bitcoin, la primera criptomoneda.
- **Blockchain:** Tecnología que permite el registro seguro, transparente y descentralizado de información en una red de computadoras.
- **NFT:** Token digital que representa la propiedad o la autenticidad de un activo único en el mundo digital.
- **Token:** Activo digital que representa algo en particular dentro de un sistema o red, como un bien, un servicio, una propiedad o una unidad de valor.
- **P2P:** Tipo de interacción o transacción directa entre personas, sin la necesidad de intermediarios o entidades centrales como bancos, gobiernos o empresas.
- **Layering:** Estrategia o táctica utilizada en ciertos tipos de fraude financiero y lavado de dinero.
- **Spreads:** El spread es la diferencia entre el precio de compra (ask) y el precio de venta (bid) de un activo financiero en los mercados.
- **Deslizamiento:** El deslizamiento es la diferencia entre el precio esperado de una operación y el precio real al que se ejecuta.

ÍNDICE DE FIGURAS

- Figura 1:** Ejecución de la estrategia del Bot EMA y RSI
- Figura 2:** Ejecución de la estrategia del Bot EMA y Order Block
- Figura 3:** Ejecución de la estrategia del Bot EMA y Fair Value Gap
- Figura 4:** Ejecución de la estrategia del Bot Multi Timeframe Analysis
- Figura 5:** Indicador de detección de Order Blocks
- Figura 6:** Indicador de detección de Fair Value Gaps
- Figura 7:** Resultados de la estrategia EMA y RSI
- Figura 8:** Resultados de la estrategia EMA y Order Block
- Figura 9:** Resultados de la estrategia EMA y Fair Value Gap
- Figura 10:** Resultados de la estrategia Multi Timeframe Analysis
- Figura 11:** Resultado de la optimización de la estrategia usando Differential Evolution
- Figura 12:** Resultado de la optimización de la estrategia usando Genetic Algorithm
- Figura 13:** Resultado de la optimización de la estrategia usando PSO

ÍNDICE DE TABLAS

- Tabla 1:** Resultados del backtest de la estrategia EMA y RSI
- Tabla 2:** Resultados del backtest de la estrategia EMA y Order Block
- Tabla 3:** Resultados del backtest de la estrategia EMA y Fair Value Gap
- Tabla 4:** Resultados del backtest de la estrategia Multi Timeframe Analysis
- Tabla 5:** Resultados del backtest de las estrategias
- Tabla 6:** Resultados de la optimización de la estrategia usando Differential Evolution
- Tabla 7:** Resultados de la optimización de la estrategia usando Genetic Algorithm
- Tabla 8:** Resultados de la optimización de la estrategia usando PSO
- Tabla 8:** Comparativa de resultados de los Algoritmos de Optimización
- Tabla 9:** Comparativa de resultados de los Algoritmos de Optimización
- Tabla 10:** Tabla de Presupuesto

1. INTRODUCCIÓN

En la actualidad, los mercados financieros están cada vez más dominados por sistemas automatizados que toman decisiones de inversión basadas en algoritmos avanzados. Sin embargo, a pesar del crecimiento de estas herramientas, muchos inversores aún desconocen su funcionamiento o subestiman su potencial. Es común que se perciba el trading como una actividad exclusivamente manual, en la que el éxito depende de la intuición y la experiencia, cuando en realidad la automatización puede reducir los errores humanos y mejorar la rentabilidad.

El principal problema al que se enfrentan los traders es la influencia de factores emocionales en su operativa. El miedo a perder dinero puede llevar a cerrar operaciones antes de tiempo, mientras que la codicia puede hacer que se mantengan posiciones abiertas demasiado tiempo, arriesgando ganancias obtenidas. Estos sesgos psicológicos afectan incluso a *traders* experimentados, por lo que encontrar métodos para neutralizar su impacto es clave para mejorar la rentabilidad y la consistencia de las estrategias de inversión.

El trading algorítmico surge como una solución a este problema, ya que permite ejecutar estrategias basadas en reglas predefinidas sin la intervención humana. Gracias a la automatización, se pueden procesar grandes volúmenes de datos en milisegundos, identificar oportunidades de mercado y ejecutar operaciones con precisión, eliminando las limitaciones humanas como la fatiga, el estrés o la indecisión. Además, este enfoque permite realizar *backtesting* sobre datos históricos para evaluar la efectividad de una estrategia antes de aplicarla en el mercado real.

A lo largo de este proyecto se desarrollarán e implementarán diversas estrategias de trading automatizado con el objetivo de evaluar su rentabilidad a largo plazo y proponer mejoras que maximicen su efectividad. Se emplearán distintos métodos de optimización para encontrar configuraciones que permitan obtener el mayor beneficio con el menor riesgo posible. Entre las técnicas utilizadas, se explorarán tres de las más comunes en la optimización de estrategias: el algoritmo de evolución diferencial, los algoritmos genéticos y la optimización por enjambre de partículas.

Otro aspecto relevante del trading algorítmico es su capacidad para operar en diferentes marcos temporales y mercados. Mientras que un trader humano solo puede analizar un número limitado de activos y gráficos a la vez, un sistema automatizado puede monitorizar múltiples mercados simultáneamente, detectando patrones y oportunidades de inversión en tiempo real. Esta capacidad es especialmente útil en entornos de alta volatilidad, donde la velocidad en la toma de decisiones es fundamental.

El objetivo final de esta investigación es demostrar que la automatización no solo es una herramienta útil, sino una necesidad en el trading moderno. Al eliminar el componente emocional de la toma de decisiones y aprovechar el poder del análisis computacional, cualquier inversor puede mejorar su desempeño en los mercados financieros. A través de este estudio, se analizará el impacto del trading algorítmico en la rentabilidad y la gestión del riesgo, aportando una visión detallada de sus beneficios y de los desafíos que conlleva su implementación.

2. ANTECEDENTES

2.1. LAS INVERSIONES

2.1.1 ¿Qué son las inversiones?

Las inversiones consisten en destinar recursos, como dinero, tiempo o conocimiento, a un activo o proyecto con la expectativa de obtener un rendimiento o beneficio futuro. Estas actividades permiten a individuos y empresas incrementar su patrimonio, generar ingresos pasivos y proteger su poder adquisitivo frente a la inflación.

El concepto de inversión abarca una amplia variedad de áreas, que van desde la adquisición de bienes tangibles, como propiedades inmobiliarias o metales preciosos, hasta la compra de activos financieros, como acciones, bonos o productos derivados. Con el paso del tiempo, las inversiones han evolucionado significativamente, incorporando categorías innovadoras relacionadas con la tecnología, como las criptomonedas y los **NFTs** (tokens no fungibles) [\[1\]](#).

2.1.2. Tipos de Inversiones

1. Inversiones Financieras

Estas inversiones implican la compra de activos negociados en los mercados financieros. Algunos ejemplos incluyen:

- **Acciones:** Permiten a los inversores adquirir una participación en empresas, obteniendo rendimientos a través del aumento en el precio de las acciones y/o dividendos distribuidos.
- **Bonos:** Son instrumentos de deuda emitidos por gobiernos o empresas, ofreciendo rendimientos fijos. Son ideales para perfiles de inversión más conservadores.
- **ETFs y Fondos de Inversión:** Proporcionan diversificación al agrupar varios activos en un solo instrumento, reduciendo riesgos asociados con la exposición a un solo activo.
- **Derivados:** Instrumentos financieros complejos, como futuros y opciones, que permiten gestionar riesgos o especular sobre movimientos de precios. [\[2\]](#)



Fuente: [Inversiones Financieras](#)

Estas inversiones son atractivas debido a su liquidez y la posibilidad de generar rentabilidades significativas, aunque pueden conllevar volatilidad y riesgos.

2. Inversiones en Activos Reales

Este tipo de inversiones involucra la adquisición de bienes tangibles con valor intrínseco. Algunos ejemplos destacados son:

- **Propiedades Inmobiliarias**

La inversión en bienes raíces, como viviendas o locales comerciales, permite obtener ingresos recurrentes a través de alquileres y, además, puede generar ganancias a largo plazo gracias a la apreciación del valor del inmueble en el tiempo. Esta inversión combina flujo de efectivo con la posibilidad de revalorización del activo.



Fuente: [Propiedades Inmobiliarias](#)

- **Metales Preciosos**

El oro, la plata y el platino son considerados activos refugio porque tienden a mantener o incluso aumentar su valor durante periodos de incertidumbre económica o alta inflación. Esto se debe a su limitada oferta y a su aceptación universal como reserva de valor.



Fuente: [Metales Preciosos](#)

- **Bienes de Colección**

Obras de arte, automóviles clásicos o antigüedades pueden ser inversiones alternativas para diversificar el patrimonio. Su valor generalmente se basa en la rareza, la demanda del mercado y el estado de conservación, pero también pueden implicar riesgos debido a su baja liquidez y la dificultad para determinar su precio real.



Fuente: [Bienes de Colección](#)

Estas inversiones suelen ser vistas como una forma de protegerse contra la inflación y diversificar riesgos, aunque su liquidez es menor en comparación con las inversiones financieras.

3. Inversiones Alternativas



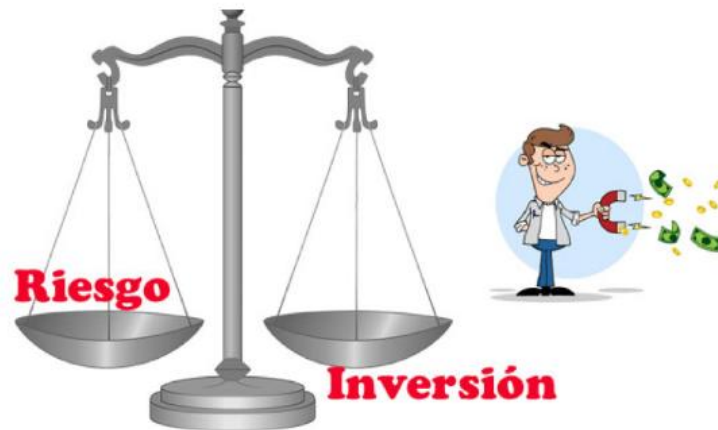
Fuente: [Inversiones Alternativas](#)

Son aquellas que no están directamente vinculadas a los mercados financieros tradicionales, ofreciendo oportunidades de diversificación y altos potenciales de rentabilidad, aunque suelen implicar mayor riesgo y especulación.

- **Criptomonedas:** Activos digitales como Bitcoin o Ethereum, caracterizados por su descentralización y potencial de crecimiento, aunque altamente volátiles y especulativos.
- **NFTs (Tokens no fungibles):** Representan propiedad digital única sobre arte, música o bienes virtuales, con un creciente interés debido a su vinculación con la tecnología *blockchain* y el metaverso.

- **Venture Capital y Equity Crowdfunding:** Permiten invertir en startups o empresas emergentes con alto potencial de crecimiento, aunque conlleva el riesgo de que muchas de estas empresas no logren consolidarse.

2.1.3. Riesgos asociados a las inversiones



Fuente: [Riesgos asociados a las inversiones](#)

Cada tipo de inversión conlleva riesgos específicos que los inversores deben considerar antes de comprometer sus recursos:

Cada tipo de inversión implica riesgos específicos que deben ser cuidadosamente evaluados antes de comprometer recursos:

- **Riesgo de Mercado:** Variaciones en los precios de los activos debido a factores económicos, políticos o sociales que afectan la oferta y la demanda.
- **Riesgo de Liquidez:** Dificultad para vender un activo rápidamente sin incurrir en pérdidas significativas, especialmente en mercados con baja actividad.
- **Riesgo Específico:** Factores únicos relacionados con una empresa, sector o activo, como el mal desempeño de una compañía o problemas en una industria concreta.
- **Riesgo Regulator:** Impacto de cambios normativos o legales en ciertos tipos de inversiones, como restricciones gubernamentales en criptomonedas o modificaciones en las reglas para el trading algorítmico. [3]

1. Riesgo Bajo



Fuente: [Riesgo Bajo](#)

Inversiones diseñadas para preservar el capital, ideales para perfiles conservadores o inversores con horizontes temporales cortos. Ejemplos destacados:

- **Bonos Gubernamentales:** Emitidos por los gobiernos, ofrecen retornos estables y predecibles. Su nivel de riesgo varía según la calificación crediticia del país emisor. Por ejemplo, bonos emitidos por países con economías sólidas, como Estados Unidos o Alemania, son considerados de bajo riesgo, mientras que bonos de países en desarrollo pueden tener mayor riesgo.
- **Certificados de Depósito (CDs):** Productos financieros ofrecidos por bancos que aseguran un rendimiento fijo durante un plazo establecido.
- **Letras del Tesoro:** Instrumentos de deuda a corto plazo emitidos por el gobierno, generalmente con un vencimiento inferior a un año.

Estas inversiones suelen ofrecer tasas de retorno menores, pero son altamente seguras y adecuadas para proteger el capital frente a la **inflación**.

2. Riesgo Medio



Fuente: [Riesgo Medio](#)

Inversiones con un equilibrio entre riesgo y retorno, adecuadas para inversores moderados con objetivos a medio o largo plazo. Ejemplos incluyen:

- **Acciones de Grandes Empresas:** También conocidas como "*blue chips*", estas empresas consolidadas suelen tener flujos de ingresos estables y menor volatilidad en comparación con startups o empresas emergentes.
- **Fondos Diversificados o ETFs:** Agrupan múltiples activos, lo que reduce el riesgo asociado con la inversión en una sola empresa o sector. Por ejemplo, el índice S&P 500, que incluye 500 de las principales empresas de Estados Unidos, ha demostrado históricamente retornos promedio superiores al 10% anual.
- **Bienes Raíces:** La inversión en propiedades puede generar ingresos estables a través de alquileres, además de la apreciación del valor del inmueble a lo largo del tiempo.

Estas opciones son ideales para quienes buscan equilibrar seguridad y crecimiento en su portafolio.

3. Riesgo Alto



Fuente: [Riesgo Alto](#)

Inversiones con alta volatilidad y posibilidad de retornos significativos, pero también con un elevado riesgo de pérdida. Estas inversiones son recomendadas para inversores con tolerancia al riesgo alta y horizontes temporales amplios. Ejemplos:

- **Criptomonedas:** Activos digitales como Bitcoin o Ethereum, conocidos por su alta volatilidad y retornos potencialmente altos en periodos cortos, aunque también implican riesgos elevados debido a su falta de regulación y sensibilidad a las noticias del mercado.
- **Startups y Venture Capital:** Invertir en empresas emergentes ofrece el potencial de retornos exponenciales si la empresa tiene éxito, aunque la tasa de fracaso en este tipo de negocios puede ser alta.
- **Materias Primas Especulativas:** Activos como petróleo, gas o productos agrícolas pueden experimentar fluctuaciones drásticas en precio debido a factores geopolíticos o climáticos.

Estas inversiones requieren un análisis cuidadoso, experiencia en el mercado y, preferiblemente, diversificación para mitigar los riesgos asociados.

Relación entre Riesgo y Horizonte Temporal

La relación entre el riesgo asumido y el horizonte temporal es fundamental en las estrategias de inversión:

- **Corto Plazo (<3 años):** Inversiones de bajo riesgo son preferibles, ya que ofrecen estabilidad y reducen la probabilidad de pérdidas en un periodo limitado.
- **Mediano Plazo (3-10 años):** Se pueden considerar inversiones de riesgo medio, aprovechando el tiempo suficiente para recuperar posibles caídas del mercado.
- **Largo Plazo (>10 años):** Los activos de mayor riesgo, como acciones o criptomonedas, pueden generar retornos significativos, ya que el horizonte extendido permite amortiguar la volatilidad y beneficiarse del crecimiento compuesto.

2.1.4. Inversión vs Especulación

Es importante diferenciar entre inversión y especulación:

- **Inversión:** Busca un crecimiento sostenible y razonable a largo plazo. Implica un análisis fundamentado de los activos seleccionados, como comprar acciones de una empresa con sólidos fundamentos financieros y expectativas de crecimiento estable en el tiempo.
- **Especulación:** Se centra en obtener ganancias rápidas aprovechando movimientos de precios a corto plazo. Suele implicar mayores niveles de riesgo y volatilidad, como operar en el mercado de criptomonedas esperando subidas rápidas de precio.



Fuente: [Inversión vs Especulación](#)

2.1.5. Tendencias actuales



Fuente: [Tendencias actuales](#)

En los últimos años, la globalización y los avances tecnológicos han transformado el panorama de las inversiones. La digitalización ha facilitado el acceso a los mercados financieros para inversores minoristas, mientras que la innovación ha introducido nuevos activos, como criptomonedas y plataformas de trading social. Aunque estas tendencias democratizan las inversiones, también presentan retos regulatorios y mayores riesgos.

En este contexto dinámico, la planificación financiera cuidadosa y la diversificación son esenciales para gestionar riesgos y maximizar el rendimiento. Además, factores como el perfil de riesgo y el horizonte temporal del inversor determinan las estrategias de inversión, que pueden clasificarse en diversas categorías.

2.2. LOS MERCADOS FINANCIEROS

2.2.1. ¿Qué son los mercados financieros?

Los mercados financieros son espacios, físicos o virtuales, donde se intercambian activos financieros como acciones, bonos, divisas, materias primas y derivados. Su función principal es facilitar la transferencia eficiente de capital entre quienes lo necesitan (empresas, gobiernos) y quienes lo poseen (inversores).

Además de proporcionar liquidez, los mercados financieros impulsan el crecimiento económico al canalizar los ahorros hacia inversiones productivas. Gobiernos y empresas pueden financiarse a través de deuda pública o privada, respectivamente.

Se dividen en mercados regulados y no regulados. Los **regulados** están supervisados por entidades gubernamentales o financieras que garantizan transparencia y protección al inversor, mientras que los **no regulados** ofrecen mayor flexibilidad, pero también mayores riesgos.

Ejemplos incluyen bolsas de valores como **NYSE** y **Euronext**, el mercado de bonos y el mercado de divisas (**Forex**). [\[4\]](#) [\[5\]](#)

2.2.2. ¿Cómo se mueven los precios?

Los precios de los activos financieros fluctúan continuamente como resultado de la interacción entre compradores y vendedores, en función de la oferta y la demanda. Hay varios factores que influyen en el movimiento de los precios en los mercados financieros: [\[6\]](#) [\[7\]](#)

OFERTA Y DEMANDA

$P_{\text{mercado}} = P_{\text{equilibrio}}$



Fuente: [OFERTA Y DEMANDA](#)

- **Demanda:** Representa la cantidad de un activo que los inversores desean comprar a un precio determinado. Si muchos inversores quieren adquirir un activo, su precio tenderá a subir.
- **Oferta:** Es la cantidad de un activo que está disponible para ser vendido en el mercado. Si hay una gran cantidad de un activo disponible para la venta, pero poca demanda, el precio tenderá a bajar.

El precio de mercado de un activo se determina por el equilibrio entre oferta y demanda en un momento dado. Cuando la demanda aumenta, por ejemplo, debido a un informe económico positivo, los precios tienden a subir. Por el contrario, una disminución de la oferta, como la causada por restricciones en la producción, también puede generar un aumento en los precios. Estas variaciones reflejan los ajustes necesarios para equilibrar las fuerzas del mercado.

FACTORES EXTERNOS



Fuente: [FACTORES EXTERNOS](#)

Los precios de los activos financieros también están influenciados por diversos factores externos, entre ellos:

- **Políticas monetarias:** Cambios en las tasas de interés, establecidos por los bancos centrales, afectan directamente el costo del dinero y, por ende, la valoración de activos financieros.
- **Eventos geopolíticos:** Conflictos, acuerdos internacionales o sanciones económicas pueden generar incertidumbre en los mercados y causar fluctuaciones en los precios.
- **Publicación de resultados financieros:** Los reportes trimestrales o anuales de las empresas, que reflejan su desempeño, impactan la percepción de los inversores y el precio de sus acciones.

VOLATILIDAD

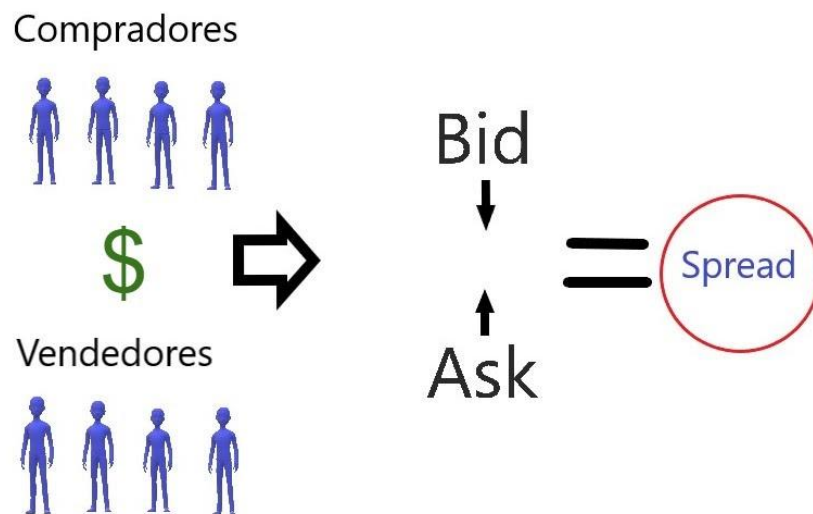


Fuente: [VOLATILIDAD](#)

La volatilidad mide la magnitud de las variaciones en el precio de un activo durante un período de tiempo. Un mercado con alta volatilidad muestra fluctuaciones de precios más amplias y rápidas, lo que implica mayores riesgos para los inversores, pero también potenciales oportunidades de ganancias.

Los inversores suelen buscar activos con alta volatilidad cuando buscan aprovechar los movimientos rápidos del mercado, aunque esto conlleva un riesgo adicional.

LIQUIDEZ



Fuente: [LIQUIDEZ](#)

La liquidez se refiere a la facilidad con la que un activo puede comprarse o venderse en el mercado sin causar cambios significativos en su precio. Un mercado es considerado líquido cuando hay un alto volumen de compradores y vendedores activos, lo que permite realizar transacciones de manera rápida, eficiente y con bajos costos de transacción.

Por ejemplo, el mercado de divisas (Forex) es uno de los mercados más líquidos del mundo, ya que opera las 24 horas con un volumen masivo de operaciones. En contraste, los bienes raíces suelen ser menos líquidos, ya que requieren más tiempo y esfuerzo para encontrar un comprador o vendedor y cerrar una transacción.

MARKET MAKERS



Fuente: [MARKET MAKERS](#)

Los *market makers* (creadores de mercado) son instituciones, como bancos o firmas de trading, o individuos que facilitan la liquidez en los mercados financieros al estar constantemente dispuestos a comprar y vender un activo. Su objetivo principal es garantizar que siempre exista suficiente oferta y demanda, permitiendo que las transacciones se realicen de forma rápida y con precios estables.

Los market makers obtienen ganancias a través del diferencial (**spread**) entre el precio de compra (**bid**) y el precio de venta (**ask**). Por ejemplo, si un market maker compra un activo a \$100 y lo vende a \$101, su ganancia es de \$1 por unidad.

Aunque son esenciales para el buen funcionamiento de los mercados, en mercados menos regulados o con menor transparencia, su actividad puede influir significativamente en los precios, generando volatilidad o manipulaciones en ciertas circunstancias.

2.2.3. Tipos de mercados financieros

Los mercados financieros pueden clasificarse en varios tipos según los activos que se negocian y las características de las transacciones. A continuación, se describen los principales tipos de mercados:

MERCADO DE VALORES



Fuente: [MERCADO DE VALORES](#)

En el mercado de valores se negocian acciones, que representan participaciones en empresas. Este mercado permite a las empresas obtener financiamiento al emitir acciones y a los inversores participar en la propiedad y beneficios de las empresas.^[8]

- **Ejemplo:** Bolsa de Nueva York (NYSE), Nasdaq.
- **Características:**
 - Alta liquidez en las bolsas principales.
 - Transparencia regulada por entidades financieras.
 - Movimientos influenciados por resultados empresariales, eventos macroeconómicos y políticas gubernamentales.
 - Herramientas como índices bursátiles (S&P 500, Dow Jones) para medir el rendimiento general del mercado.

MERCADO DE BONOS



Fuente: [MERCADO DE BONOS](#)

El mercado de bonos es donde se negocian instrumentos de deuda, como bonos corporativos, bonos del estado y otros títulos de deuda. Los bonos son esencialmente préstamos que los inversores realizan a emisores a cambio de pagos de intereses regulares.

- **Ejemplo:** Mercado de bonos del Tesoro de EE. UU.
- **Características:**
 - Menor riesgo comparado con las acciones, pero con rendimientos potencialmente más bajos.
 - Se utilizan para la diversificación de carteras y como cobertura frente a riesgos.
 - Instrumentos a corto, mediano y largo plazo con diferentes niveles de riesgo (bonos de grado de inversión frente a bonos basura).

MERCADO DE DIVISAS (FOREX)



Fuente: [MERCADO DE DIVISAS \(FOREX\)](#)

El mercado Forex (*Foreign Exchange*) es el mayor mercado financiero del mundo, con un volumen de negociación diario que supera los \$6.6 billones. Aquí se intercambian divisas, facilitando el comercio y la inversión internacionales.

- **Ejemplo:** Mercado global de Forex, accesible mediante brokers y plataformas electrónicas.
- **Características:**
 - Operativo las 24 horas del día, 5 días a la semana.
 - Alta volatilidad, lo que lo hace atractivo para traders especulativos.
 - Utilizado por empresas y gobiernos para gestionar riesgos de tipo de cambio.
 - Principales pares de divisas: EUR/USD, USD/JPY, GBP/USD.

MERCADO DE MATERIAS PRIMAS



Fuente: [MERCADO DE MATERIAS PRIMAS](#)

En este mercado se negocian productos básicos, como metales preciosos (oro, plata), energía (petróleo, gas natural) y productos agrícolas (trigo, café). Las materias primas son fundamentales para la economía mundial.

- **Ejemplo:** Chicago Mercantile Exchange (CME).
- **Características:**
 - Dividido en dos categorías principales: materias primas duras (extraídas o minadas) y materias primas blandas (agrícolas o ganaderas).
 - Los precios suelen estar influenciados por eventos geopolíticos, desastres naturales y cambios climáticos.
 - Se utilizan contratos de futuros para gestionar riesgos asociados a la volatilidad de los precios.

MERCADO DE DERIVADOS



Fuente: [MERCADO DE DERIVADOS](#)

Este mercado incluye contratos cuyo valor depende de un activo subyacente, como **futuros**, **opciones**, **swaps** y **forwards**. Los derivados son utilizados tanto para la especulación como para la cobertura de riesgos financieros.

- **Ejemplo:** Eurex (plataforma europea de derivados).
- **Características:**
 - Permiten **apalancamiento**, lo que amplifica tanto las ganancias como las pérdidas.
 - Pueden estar vinculados a una variedad de activos, como índices bursátiles, divisas o materias primas.
 - Importantes para estrategias avanzadas de inversión y gestión de riesgos.

MERCADO DE CRIPTOMONEDAS



Fuente: [MERCADO DE CRIPTOMONEDAS](#)

Este mercado es relativamente nuevo y se centra en el intercambio de monedas digitales como Bitcoin, Ethereum y otras **altcoins**. Ha ganado popularidad por su potencial de altas ganancias, pero también presenta un alto nivel de riesgo.

- **Ejemplo:** Binance, Coinbase.
- **Características:**
 - Opera las 24 horas, todos los días del año.
 - Descentralización en gran medida, aunque algunas plataformas están reguladas en países específicos.
 - Alta volatilidad y riesgo de fraude debido a la naturaleza no regulada de muchas criptomonedas.
 - Uso creciente de tecnología blockchain y contratos inteligentes para facilitar transacciones.

MERCADO DE SEGUROS



Fuente: [MERCADO DE SEGUROS](#)

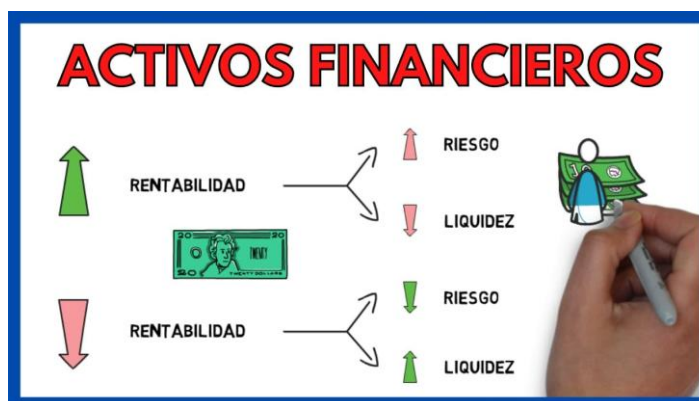
Aunque menos visible, este mercado negocia pólizas de seguro como instrumentos financieros, permitiendo a individuos y empresas gestionar riesgos frente a eventos inesperados.

- **Ejemplo:** Empresas aseguradoras como Allianz, Axa, y mercados secundarios de reaseguro como Lloyd's of London.
- **Características:**
 - Los seguros son esenciales para la gestión de riesgos en áreas como salud, vida, propiedad y responsabilidad.
 - Las pólizas se pueden negociar como instrumentos financieros en mercados secundarios especializados.
 - Los reaseguradores ayudan a las aseguradoras a gestionar grandes riesgos compartiendo la carga financiera.

2.2.4. Elementos de los mercados financieros

Los mercados financieros están compuestos por varios elementos clave que hacen posible su funcionamiento eficiente. Cada uno de ellos desempeña un rol fundamental en la dinámica de estas complejas estructuras económicas:

ACTIVOS FINANCIEROS



Fuente: [ACTIVOS FINANCIEROS](#)

Los activos financieros son instrumentos que representan un derecho económico, como la propiedad de un bien, un flujo de ingresos futuro o la posibilidad de realizar intercambios en los mercados financieros. Estos activos incluyen productos tradicionales como acciones, bonos, divisas y derivados, que son comprados y vendidos en los mercados financieros.

En los últimos años, han surgido nuevas categorías como los activos digitales, entre ellos las criptomonedas y los tokens no fungibles (NFTs), que han abierto nuevas oportunidades y planteado desafíos para los inversores.

INTERMEDIARIOS FINANCIEROS



Fuente: [INTERMEDIARIOS FINANCIEROS](#)

Los intermediarios financieros son entidades que facilitan las transacciones entre compradores y vendedores en los mercados financieros, conectando a quienes necesitan financiación con quienes tienen capital para invertir. Su papel es esencial para garantizar la eficiencia y estabilidad de los mercados. Algunos ejemplos son:

- **Bancos de inversión:** Ayudan a las empresas a recaudar capital mediante la emisión de acciones y bonos, además de asesorar en procesos de fusiones y adquisiciones.
- **Brokers:** Actúan como intermediarios entre los inversores y el mercado, ejecutando órdenes de compra y venta de activos financieros. El auge de plataformas de trading online, como **Robinhood**, ha democratizado el acceso al mercado, permitiendo a pequeños inversores operar de manera directa y sencilla.
- **Fondos de inversión:** Agrupan el capital de múltiples inversores para diversificar el riesgo y obtener acceso a una variedad de activos financieros. Estos fondos pueden ser administrados de forma activa o pasiva, y hoy en día, herramientas avanzadas como la inteligencia artificial y los algoritmos están transformando la gestión de carteras, haciéndola más eficiente y basada en datos.

REGULADORES



Fuente: [REGULADORES](#)

Los reguladores son instituciones encargadas de garantizar el correcto funcionamiento de los mercados financieros y proteger a los inversores. Su principal objetivo es promover la transparencia, la equidad y la estabilidad en el sistema financiero.

En un entorno dinámico y en constante evolución, los reguladores han adoptado normativas específicas para abordar desafíos emergentes, como el comercio de criptomonedas, el auge de las *Fintech* [9] y la implementación de estándares ESG (ambientales, sociales y de gobernanza).

Ejemplos de reguladores incluyen:

- **Comisión Nacional del Mercado de Valores (CNMV)** en España, que supervisa los mercados y las instituciones financieras para garantizar la protección de los inversores.
- **Securities and Exchange Commission (SEC)** en EE. UU., responsable de hacer cumplir las leyes sobre valores y proteger a los inversores frente a prácticas fraudulentas.
- **Financial Conduct Authority (FCA)** en el Reino Unido, que regula las empresas de servicios financieros y asegura la competencia efectiva en el mercado.

Estas instituciones desempeñan un papel crucial al equilibrar la innovación y la seguridad de los inversores, especialmente en sectores emergentes como los activos digitales. Además, su labor ayuda a mantener la confianza del público en los mercados financieros, fomentando un crecimiento económico sostenible.

INFRAESTRUCTURA DEL MERCADO



Fuente: [INFRAESTRUCTURA DEL MERCADO](#)

La infraestructura del mercado se refiere a los sistemas, plataformas tecnológicas y procesos que permiten el funcionamiento eficiente de los mercados financieros. Incluye las bolsas de valores, donde se compran y venden activos financieros, y los sistemas de compensación y liquidación, que aseguran que las transacciones se completen correctamente entre las partes involucradas.

Además, tecnologías emergentes como *blockchain* están transformando esta infraestructura al proporcionar métodos más transparentes, seguros y eficientes para registrar y procesar transacciones. Por ejemplo, las bolsas tradicionales como NYSE o Nasdaq operan con plataformas avanzadas que procesan millones de transacciones por

segundo, mientras que redes *blockchain*, como Ethereum, están habilitando nuevas formas de intercambio mediante contratos inteligentes.

INVERSORES



Fuente: [INVERSORES](#)

Los inversores son agentes fundamentales en los mercados financieros, ya que aportan el capital necesario para que estos funcionen. Se dividen principalmente en dos categorías:

- **Inversores minoristas:** Individuos que invierten sus ahorros personales en activos financieros, como acciones, bonos o fondos. En los últimos años, su participación ha crecido significativamente gracias al acceso a plataformas de trading online y a la influencia de las redes sociales. Un ejemplo destacado fue el caso de GameStop, donde los inversores minoristas coordinaron estrategias colectivas que impactaron el precio de la acción.
- **Inversores institucionales:** Entidades como fondos soberanos, fondos de pensiones, aseguradoras y gestores de activos. Estos manejan grandes volúmenes de capital y tienen un impacto considerable en los mercados debido a sus decisiones de inversión estratégicas y a largo plazo.

Ambos tipos de inversores desempeñan un papel crucial en los mercados financieros, ya que aportan liquidez, estabilizan precios y contribuyen al crecimiento económico.

FACTORES EXTERNOS QUE AFECTAN A LOS MERCADOS



Fuente: [FACTORES EXTERNOS QUE AFECTAN A LOS MERCADOS](#)

Los mercados financieros están influenciados por una amplia variedad de factores, tanto macroeconómicos como globales, que afectan el comportamiento de los precios y la confianza de los inversores. Entre los más relevantes se encuentran:

- **Políticas monetarias:** Las decisiones de los bancos centrales, como ajustes en las tasas de interés o programas de compra de activos, pueden influir directamente en la liquidez del mercado y en el costo del crédito. Por ejemplo, un aumento en las tasas de interés suele fortalecer la moneda local pero puede afectar negativamente a los mercados de acciones.
- **Inflación:** Un aumento sostenido en los precios puede disminuir el poder adquisitivo y afectar las decisiones de inversión, mientras que niveles controlados de inflación suelen ser favorables para el crecimiento económico.
- **Conflictos geopolíticos:** Eventos como guerras, sanciones económicas o tensiones internacionales pueden generar incertidumbre y volatilidad en los mercados, especialmente en sectores sensibles como el petróleo o los metales preciosos.
- **Pandemias:** Como ocurrió con la COVID-19, las pandemias pueden afectar drásticamente la economía global, interrumpir cadenas de suministro y alterar las proyecciones de crecimiento económico.[\[10\]](#)

Estos factores pueden generar volatilidad, incertidumbre o incluso crear oportunidades de inversión, dependiendo de cómo los mercados interpreten y reaccionen a los eventos globales.

2.2.5. Evolución de los mercados financieros

La digitalización ha transformado los mercados financieros, permitiendo operaciones en tiempo real y la implementación de trading algorítmico. Esto ha mejorado la eficiencia y reducido los costos, pero también ha introducido nuevos riesgos, como ataques cibernéticos y volatilidad causada por algoritmos de alta frecuencia.

Además, la digitalización ha permitido la integración global de los mercados financieros, facilitando el acceso a inversores en todo el mundo y fomentando la aparición de mercados emergentes. Esto ha ampliado las oportunidades de inversión y diversificación, aunque también ha incrementado la exposición a riesgos globales como crisis financieras internacionales.

La tecnología **blockchain** está transformando la infraestructura de los mercados financieros, ofreciendo mayor transparencia, velocidad y reducción de costos en transacciones. Paralelamente, las **fintech** han revolucionado los servicios financieros, introduciendo soluciones innovadoras como préstamos entre pares (P2P) y sistemas de pagos digitales.

Por otro lado, la aparición de activos digitales, como las criptomonedas, ha redefinido los conceptos tradicionales de inversión, generando nuevos mercados. Sin embargo, estos activos presentan desafíos regulatorios y riesgos asociados a su alta volatilidad y falta de estandarización.

Si bien la digitalización ha mejorado la eficiencia, también ha incrementado los riesgos emergentes. Entre ellos destacan la dependencia tecnológica, el riesgo de interrupciones operativas debido a ataques cibernéticos, la manipulación del mercado por algoritmos mal diseñados y la alta volatilidad por el trading de alta frecuencia.

2.2.6. Riesgos en los mercados financieros

Los principales riesgos asociados a los mercados financieros incluyen:

- **Riesgo sistémico:** Impacto que afecta a todo el mercado debido a fallos en una institución importante o una crisis generalizada. Este tipo de riesgo se intensificó durante la crisis financiera de 2008, cuando la quiebra de Lehman Brothers tuvo repercusiones globales.
- **Riesgo de mercado:** Cambios adversos en los precios de los activos debido a fluctuaciones en la oferta y la demanda, así como por eventos externos como recesiones económicas o cambios en las tasas de interés.
- **Riesgo de crédito:** La posibilidad de que una contraparte incumpla con sus obligaciones financieras. Esto es especialmente relevante en préstamos, bonos corporativos y derivados financieros.
- **Riesgo de liquidez:** La falta de capacidad de un mercado o una institución para convertir activos en efectivo sin afectar significativamente su precio. Este riesgo suele intensificarse en momentos de crisis financiera, cuando la liquidez se reduce drásticamente.
- **Riesgo regulatorio y legal:** Cambios en las normativas o leyes que afectan a los mercados financieros o productos específicos. Un ejemplo reciente es la

regulación de los criptoactivos, que busca reducir riesgos pero también genera incertidumbre entre los inversores.

- **Riesgo geopolítico:** Eventos como conflictos armados, tensiones comerciales o sanciones internacionales que alteran los mercados globales. La guerra en Ucrania, por ejemplo, ha afectado los precios de la energía y los productos básicos, generando inestabilidad en varios mercados.
- **Riesgo tecnológico:** Dependencia de sistemas tecnológicos avanzados que pueden fallar debido a interrupciones técnicas o ataques cibernéticos. Los ataques **ransomware** dirigidos a instituciones financieras son un ejemplo claro de este tipo de riesgo.

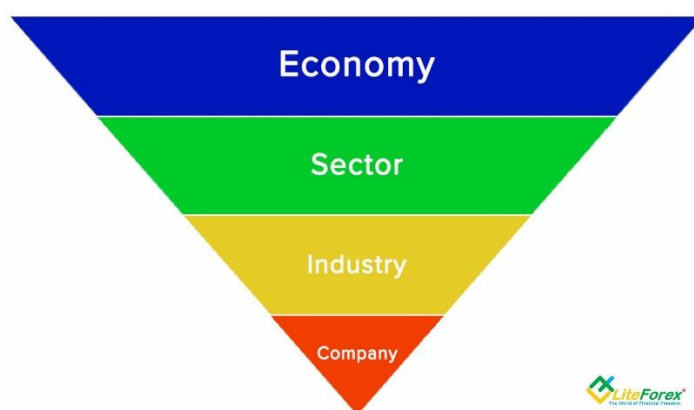
2.3. TRADING

El trading es la compra y venta de activos financieros en los mercados con el objetivo de obtener un beneficio a corto o medio plazo. A diferencia de la inversión a largo plazo, los *traders* suelen buscar explotar las fluctuaciones diarias o semanales de los precios de los activos, como acciones, divisas, materias primas y criptomonedas. Además de las estrategias tradicionales, el avance de las tecnologías ha permitido la proliferación del trading automatizado y algorítmico, que utiliza algoritmos complejos para ejecutar operaciones en milisegundos.

2.3.1. Tipos de análisis

Para tomar decisiones de trading, existen tres enfoques principales de análisis que los *traders* utilizan para predecir los movimientos de los precios. [\[11\]](#)

ANÁLISIS FUNDAMENTAL



Fuente: [ANÁLISIS FUNDAMENTAL](#)

El análisis fundamental se centra en evaluar el valor intrínseco de un activo financiero examinando los factores económicos, financieros y de negocio que pueden influir en su precio.

- **Empresas:** Para acciones, el análisis fundamental incluye el estudio de informes financieros (como los estados de resultados, balance general, flujo de caja), las proyecciones de crecimiento de la empresa, los dividendos, las ganancias y el entorno macroeconómico.
- **Factores macroeconómicos:** Para divisas y otros activos, se consideran variables como las tasas de interés, la inflación, el crecimiento económico y las políticas gubernamentales.

Un ejemplo sería un *trader* que compra acciones de una empresa basándose en un sólido crecimiento de las ganancias y un buen pronóstico económico.

ANÁLISIS TÉCNICO



Fuente: [ANÁLISIS TÉCNICO](#)

El análisis técnico se basa en el estudio de los movimientos históricos del precio y los volúmenes de negociación para predecir futuros movimientos de los precios. Este enfoque asume que los precios siguen patrones y que el historial de precios refleja toda la información disponible.

- **Gráficos de precios:** Se utilizan gráficos para analizar el comportamiento pasado del precio.
- **Indicadores técnicos:** Los traders técnicos se apoyan en herramientas que calculan distintos indicadores, los cuales se explicarán posteriormente, para identificar tendencias y puntos de entrada o salida. Estas herramientas incluyen las medias móviles, el RSI (Índice de Fuerza Relativa), el MACD y las bandas de Bollinger.

Por ejemplo, un trader puede identificar un "soporte" en el gráfico y decidir comprar cuando el precio de una acción alcanza ese nivel.

ANÁLISIS FUNDAMENTAL VS ANÁLISIS TÉCNICO



Fuente: [ANÁLISIS FUNDAMENTAL VS ANÁLISIS TÉCNICO](#)

Aquí puedes hacer una comparación entre ambos enfoques:

- **Horizonte temporal:** El análisis fundamental suele utilizarse para inversiones a medio y largo plazo, mientras que el análisis técnico es más común en el trading a corto plazo.
- **Fuente de información:** El análisis fundamental se basa en datos económicos, financieros y empresariales, mientras que el análisis técnico se enfoca únicamente en el comportamiento del precio y los volúmenes de negociación.
- **Complementarios:** A menudo, los traders combinan ambos análisis. Por ejemplo, un análisis fundamental puede indicar que una acción está infravalorada, pero el *trader* espera a que el análisis técnico confirme un punto de entrada adecuado.

ANÁLISIS CUANTITATIVO



Fuente: [ANÁLISIS CUANTITATIVO](#)

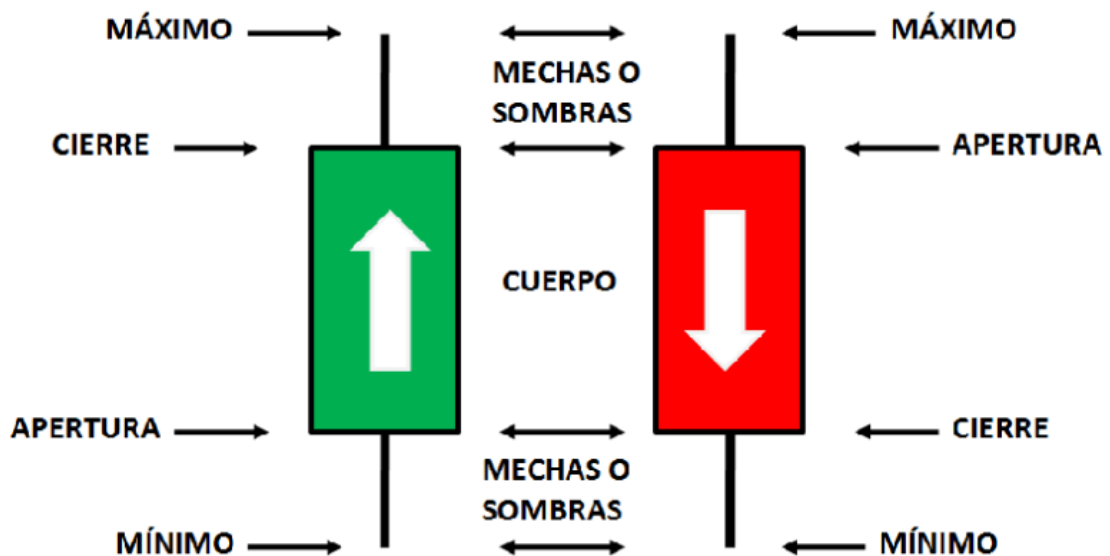
Este enfoque utiliza modelos matemáticos y estadísticas para identificar patrones y tomar decisiones de trading. Es común en el trading algorítmico y de alta frecuencia (HFT).

- **Herramientas:** Machine learning, análisis de datos históricos y simulaciones (*backtesting*).
- **Ejemplo:** Un modelo cuantitativo podría identificar correlaciones entre activos aparentemente no relacionados para explotar oportunidades de arbitraje.

2.3.2. Elementos que forman parte del trading

Para llevar a cabo operaciones de trading exitosas, existen varios elementos que son fundamentales. Algunos de ellos son:

LAS VELAS



Fuente: [LAS VELAS](#)

Las velas japonesas son una representación gráfica del comportamiento del precio de un activo financiero durante un periodo de tiempo determinado. Cada vela muestra cuatro precios clave:

- **Apertura:** El precio al inicio del periodo.
- **Cierre:** El precio al final del periodo.
- **Máximo:** El precio más alto alcanzado durante el periodo.
- **Mínimo:** El precio más bajo alcanzado.

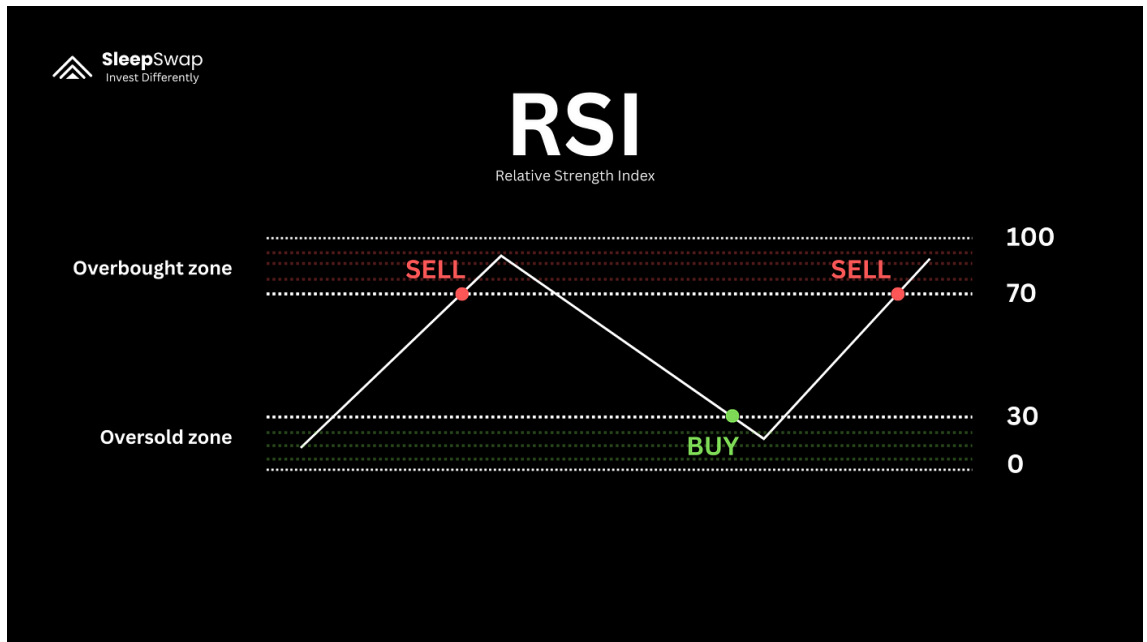
Las velas pueden ser alcistas (cuando el cierre es superior a la apertura) o bajistas (cuando el cierre es inferior a la apertura). Los patrones formados por grupos de velas son utilizados por los *traders* técnicos para predecir cambios en la tendencia.

INDICADORES

Los indicadores técnicos son fórmulas matemáticas aplicadas al precio, volumen o volatilidad para ayudar a los *traders* a interpretar el mercado y tomar decisiones de compra o venta. Algunos de los más populares son:

- **RSI (Índice de Fuerza Relativa):** Es un indicador técnico que mide la velocidad y magnitud de los movimientos de precio para identificar si un activo está en condiciones de sobrecompra o sobreventa. Se calcula en una escala de 0 a 100:
 - Un valor por encima de **70** generalmente indica que el activo está sobrecomprado y podría experimentar una corrección o retroceso en su precio.
 - Un valor por debajo de **30** sugiere que el activo está sobrevendido y podría rebotar al alza.

El RSI es utilizado principalmente para detectar posibles puntos de giro en el mercado, aunque su interpretación varía dependiendo del contexto del mercado (tendencia alcista, bajista o lateral). Los niveles de 50 también se consideran importantes, ya que pueden reflejar el equilibrio entre compradores y vendedores.



Fuente: [RSI](#)

- **Media Móvil Exponencial:** La EMA es un indicador técnico que suaviza las fluctuaciones de precios y ayuda a identificar tendencias. Se calcula usando los **precios de cierre**, dando mayor peso a los más recientes, lo que la hace más sensible a los cambios del mercado que la Media Móvil Simple (SMA). ¿Para qué sirve?
 - **Identificar tendencias:** Si la EMA sube, indica una tendencia alcista; si baja, una bajista.
 - **Señales de compra/venta:** Los cruces entre EMAs rápidas (corto plazo) y lentas (largo plazo) suelen indicar oportunidades de entrada o salida.
 - **Soportes y resistencias dinámicas:** El precio a menudo rebota en EMAs clave como las de 50 o 200 periodos.

La EMA es una herramienta basada en precios para detectar movimientos del mercado y anticipar cambios de tendencia.



Fuente: [Media Móvil Exponencial](#)

- **MACD (Media Móvil de Convergencia/Divergencia):** Es un indicador técnico que muestra la relación entre dos medias móviles de un activo, generalmente una media móvil exponencial (EMA) de 12 y 26 períodos. El MACD se calcula restando la EMA de 26 períodos de la EMA de 12 períodos. El resultado es una línea llamada **línea MACD**. Además, se incluye una **línea de señal**, que es la media móvil exponencial de 9 períodos del MACD.
 - **Cruce de la línea MACD y la línea de señal:** Cuando la línea MACD cruza por encima de la línea de señal, puede indicar una señal de compra, mientras que cuando cruza por debajo, puede ser una señal de venta.
 - **Histograma del MACD:** Muestra la diferencia entre la línea MACD y la línea de señal, ayudando a visualizar la fuerza de la tendencia. Un histograma creciente sugiere que la tendencia es fuerte, mientras que un histograma decreciente indica que la tendencia está perdiendo fuerza.

El MACD es especialmente útil para identificar cambios en la dirección de la tendencia, así como para confirmar tendencias existentes.



Fuente: [MACD](#)

GESTIÓN DEL RIESGO

La **gestión del riesgo** es uno de los pilares más importantes en el trading, ya que protege el capital y permite mantenerse en el mercado a largo plazo. Para lograrlo, los traders deben establecer reglas claras que limiten sus pérdidas. Algunas estrategias

Mastering Risk Management in Trading



clave incluyen:

- **Stop Loss:** Una orden automática que cierra una posición si el precio alcanza un nivel predefinido, limitando pérdidas antes de que aumenten.
- **Diversificación:** Invertir en diferentes activos o mercados para reducir el impacto de una mala operación. Por ejemplo, combinar acciones, bonos y divisas en una cartera.
- **Tamaño de la posición:** Asegurarse de que cada operación representa solo una fracción del capital total, considerando cuánto se está dispuesto a arriesgar en cada *trade*.

La gestión del riesgo no solo minimiza pérdidas, sino que también mejora la disciplina y aumenta las probabilidades de éxito a largo plazo.

GESTIÓN DE LAS OPERACIONES

La gestión de las operaciones consiste en optimizar y supervisar las posiciones abiertas para maximizar beneficios y minimizar riesgos. Un trader puede ajustar sus posiciones según las condiciones del mercado utilizando herramientas como:

- **Trailing Stop:** Un *stop loss* dinámico que sube junto con el precio cuando este se mueve a favor del *trader*. Ayuda a asegurar ganancias ya obtenidas mientras permite que la operación siga abierta en caso de que el precio continúe subiendo.
- **Toma de beneficios:** Establecer puntos específicos para cerrar una operación exitosa y asegurar los beneficios, evitando el riesgo de una reversión inesperada en el mercado.

Una buena gestión de las operaciones garantiza que las decisiones sean estratégicas, no emocionales, y contribuye al éxito a largo plazo.

PSICOLOGÍA EN EL TRADING

La psicología juega un papel crucial en el trading. Las decisiones de los inversores humanos están constantemente influenciadas por emociones como el miedo, la codicia o la incertidumbre. Estas emociones, aunque naturales, pueden llevar a tomar decisiones irracionales que comprometan el éxito de una estrategia de inversión.

El uso de bots de *trading* ha revolucionado la forma en que los inversores enfrentan estos desafíos psicológicos. Al automatizar las decisiones basándose en datos objetivos, los bots eliminan muchos de los sesgos de comportamiento que afectan negativamente al rendimiento en los mercados financieros. Sin embargo, el impacto psicológico del uso de bots no es completamente positivo, ya que también puede generar dependencia y una desconexión del inversor con el mercado. [\[12\]](#) [\[13\]](#) [\[14\]](#)

Sesgos de comportamiento

Los bots de trading están diseñados para operar sin emociones, siguiendo reglas predefinidas o algoritmos optimizados. Esto les permite evitar algunos de los sesgos cognitivos más comunes que afectan a los inversores humanos, tales como:

1. Sesgo de confirmación:

- Los inversores tienden a buscar información que confirme sus creencias existentes, ignorando datos contradictorios.
- Los bots, en cambio, analizan objetivamente todos los datos disponibles, eliminando este tipo de parcialidad.

2. Aversión a las pérdidas:

- Los humanos suelen aferrarse a posiciones perdedoras, esperando que el mercado se recupere, lo que puede agravar las pérdidas.
- Los bots ejecutan órdenes de *stop loss* sin cuestionamientos emocionales, limitando el riesgo de pérdidas mayores.

3. Euforia y sobreconfianza:

- En mercados alcistas, los inversores humanos pueden caer en un exceso de confianza, realizando operaciones impulsivas o sin análisis adecuado.
- Los bots mantienen una disciplina constante, independientemente de las condiciones del mercado.

Ejemplos de cómo las emociones afectan negativamente a las inversiones



Fuente: [Psicología del Mercado](#)

1. El pánico durante las caídas del mercado:

- Durante el "Flash Crash" de 2010, muchos inversores humanos vendieron en pánico al ver desplomes repentinos en los precios, consolidando pérdidas. Los bots, por otro lado, reaccionaron de manera

más sistemática, ajustándose rápidamente a las condiciones del mercado.[\[15\]](#)

2. La codicia durante una burbuja:

- En la burbuja de las puntocom de finales de los años 90, los inversores compraban acciones tecnológicas sin fundamentos sólidos, impulsados por la emoción de "no perderse" las ganancias. Este comportamiento agrandó la burbuja, que finalmente estalló.

3. Parálisis por análisis:

- Los inversores a menudo se sienten abrumados al enfrentar demasiada información y terminan sin tomar decisiones. Los bots resuelven este problema procesando grandes volúmenes de datos de manera eficiente y tomando decisiones en fracciones de segundo.

El impacto psicológico en los inversores que confían plenamente en bots de trading



Fuente: [Impacto Psicológico de los Bots de Trading](#)

Si bien los bots eliminan las emociones del proceso de trading, su uso también tiene consecuencias psicológicas que pueden afectar a los inversores de maneras inesperadas: [\[16\]](#)

1. Dependencia excesiva:

- Al delegar completamente las decisiones en los bots, algunos inversores dejan de involucrarse en la comprensión de las dinámicas del mercado. Esto puede ser problemático si el bot falla, ya que el inversor no tendrá las habilidades necesarias para intervenir o ajustar la estrategia.
- Ejemplo: Un inversor puede confiar ciegamente en un bot diseñado para operar en mercados alcistas, sin darse cuenta de que el algoritmo no está preparado para condiciones bajistas, lo que puede resultar en pérdidas significativas.

2. Desconexión emocional:

- Aunque los bots eliminan el estrés asociado al trading, esto puede llevar a una actitud indiferente hacia las inversiones. En lugar de ver el *trading*

como un proceso activo, algunos inversores podrían considerarlo una actividad pasiva, perdiendo el sentido de responsabilidad sobre sus propias finanzas.

3. Falsa seguridad:

- La confianza en que los bots son infalibles puede llevar a los inversores a asumir riesgos innecesarios o a no realizar pruebas rigurosas de las estrategias que implementan.

Balance entre automatización y control humano



Fuente: [Balance entre automatización y control humano](#)

El uso de bots no debería implicar una desvinculación total del inversor con el mercado. Para maximizar su efectividad, es crucial encontrar un equilibrio entre la automatización y el control humano:

- **Supervisión activa:** Aunque los bots toman decisiones automáticas, los inversores deben supervisar regularmente su desempeño y realizar ajustes cuando sea necesario.
- **Comprensión del algoritmo:** Es fundamental que los usuarios comprendan las reglas y limitaciones del bot que utilizan, en lugar de confiar ciegamente en él.
- **Capacitación continua:** Los inversores deben mantenerse informados sobre las tendencias del mercado y las nuevas tecnologías en trading automatizado.

2.4. BOTS DE TRADING

Los bots de *trading* son programas informáticos diseñados para ejecutar operaciones en los mercados financieros de manera automática, siguiendo reglas predefinidas o algoritmos complejos. Estos bots han transformado la manera en que se realiza el trading, permitiendo una mayor eficiencia, velocidad y precisión en las operaciones. [\[17\]](#)

2.4.1. Evolución histórica de los Bots de Trading

Los bots de *trading* han evolucionado de forma significativa desde sus inicios. En las primeras etapas, estos programas seguían reglas predefinidas por los operadores

humanos. Por ejemplo, los bots del siglo XX empleaban simples algoritmos de análisis técnico para automatizar la ejecución de órdenes basadas en indicadores clásicos como el promedio móvil o el índice de fuerza relativa (RSI).

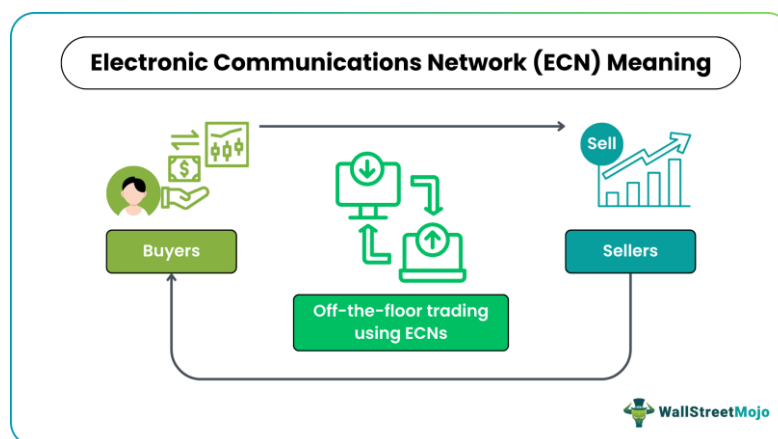
A medida que la tecnología informática y el acceso a datos de mercado en tiempo real mejoraron, los bots de *trading* fueron volviéndose más sofisticados. Los avances en velocidad de ejecución gracias a las redes de fibra óptica o la **colocalización** (la ubicación física de los servidores cerca de las bolsas), permitieron el desarrollo de bots de alta frecuencia. Este tipo de bots realizan miles de operaciones por segundo, generando pequeñas ganancias por operación, pero a una escala masiva.

En las últimas décadas, la inclusión de la inteligencia artificial (IA) y el aprendizaje automático (*machine learning*) ha revolucionado aún más el trading automatizado. Los bots de trading ya no dependen únicamente de reglas fijas, sino que pueden aprender de los datos históricos y ajustar sus estrategias en función de la dinámica del mercado.[\[18\]](#)

Décadas de 1970 y 1980: Los Primeros Pasos del Trading Automatizado

En estas décadas, los primeros sistemas automatizados surgieron con la introducción de computadoras personales y sistemas electrónicos. Algunas innovaciones clave fueron:

- **Primeros sistemas automatizados:** Estos permitían la ejecución de órdenes basadas en criterios predefinidos, facilitando tareas como el balanceo de carteras y la automatización básica de estrategias técnicas.
- **Electronic Communication Networks (ECNs):** Las ECNs revolucionaron el trading electrónico, permitiendo a los inversores conectarse directamente y ejecutar operaciones sin intermediarios tradicionales.



Fuente: [ECNs](#)

Un caso emblemático de esta época fue el *Lunes Negro* de 1987, cuando una venta masiva automatizada agrandó la caída del mercado, mostrando tanto el potencial como los riesgos de estos sistemas en una era de tecnología emergente.

Década de 1990: Refinamiento y Expansión

En los años 90, los algoritmos de trading comenzaron a consolidarse como una herramienta indispensable en los mercados financieros. Durante este período:

- Las estrategias seguían basándose en indicadores técnicos como el cruce de medias móviles (**MA**), el índice de fuerza relativa (**RSI**) y el oscilador estocástico.
- Los sistemas automatizados, aunque todavía rudimentarios, se implementaban en servidores locales, sin acceso a datos en tiempo real, lo que los hacía menos reactivos a las condiciones del mercado.

Este período también marcó el inicio del **program trading**, un sistema ampliamente utilizado para equilibrar carteras y realizar **arbitrajes** entre índices y futuros.



Fuente: [Program Trading](#)

Década de 2000: El Auge del Trading de Alta Frecuencia (HFT)

Con el inicio del nuevo milenio, los avances tecnológicos impulsaron el trading de alta frecuencia (HFT), un tipo de trading automatizado que permite ejecutar miles o incluso millones de operaciones por segundo. Este enfoque aprovecha pequeñas ineficiencias del mercado para generar ganancias.

Elementos clave de esta evolución:

- **Infraestructuras avanzadas:** Redes de fibra óptica y la colocalización, donde los servidores de las empresas se ubican cerca de las bolsas para minimizar la latencia.
- **Plataformas electrónicas:** Herramientas como NYSE Arca y Nasdaq TotalView facilitaron la ejecución masiva y eficiente de órdenes.



Fuente: [Trading Alta Frecuencia](#)

Empresas como Virtu Financial y Citadel Securities lideraron el sector, empleando algoritmos avanzados para arbitraje estadístico, detección de órdenes ocultas y *market making* automatizado.

Décadas de 2010 y 2020: Inteligencia Artificial e Innovación Avanzada

En las últimas dos décadas, los bots de trading han alcanzado niveles de sofisticación nunca antes imaginados gracias a la integración de:

- **Inteligencia Artificial (IA):** Los sistemas actuales ya no dependen de reglas fijas, sino que analizan volúmenes de datos masivos y se adaptan en tiempo real.
- **Técnicas avanzadas:**
 - Modelos predictivos basados en redes neuronales recurrentes (**RNN**) para identificar patrones en series temporales. Un estudio publicado en *Procedia Computer Science* describe la construcción de un modelo utilizando RNN, específicamente Long Short-Term Memory (LSTM), para predecir futuros movimientos en el mercado de valores. [\[19\]](#)
 - Procesamiento de lenguaje natural (**NLP**) para analizar noticias financieras, redes sociales y tendencias en tiempo real. Un artículo en *European Business Magazine* discute cómo el NLP está transformando el análisis y la predicción de tendencias del mercado al utilizar datos no estructurados como publicaciones en redes sociales, noticias y reseñas de clientes. [\[20\]](#)
 - **Deep Reinforcement Learning**, que permite a los bots aprender de sus propias operaciones mediante simulaciones de mercado. [\[21\]](#)



Fuente: [Inteligencia Artificial](#)

Ejemplos destacados incluyen los bots de fondos de cobertura como Two Sigma y Renaissance Technologies, que emplean modelos matemáticos e inteligencia artificial para identificar oportunidades de trading imperceptibles para operadores humanos.

Desafíos éticos y regulatorios: Los avances también han traído problemas, como:

- **Spoofing:** Colocar órdenes falsas para manipular precios.
- **Amplificación de volatilidad:** Bots que intensifican movimientos del mercado en periodos de alta volatilidad.

2.4.2. Trading Automático vs Trading Manual



Fuente: [Trading Automático vs Trading Manual](#)

El trading puede realizarse de dos maneras principales: manual y automático. Cada enfoque tiene sus propias ventajas y desventajas, y la elección entre uno u otro depende de los objetivos, recursos y preferencias del trader. [\[22\]](#)

TRADING MANUAL

Implica que el trader toma decisiones de compra y venta basándose en análisis y observaciones personales.

- **Ventajas:**
 - **Flexibilidad:** Permite adaptarse rápidamente a eventos inesperados y cambios en el mercado.
 - **Control Total:** El trader tiene control directo sobre cada operación y puede ajustar estrategias en tiempo real.
 - **Desarrollo de Habilidades:** Promueve el aprendizaje y la comprensión profunda de los mercados.

- **Desventajas:**
 - **Emociones:** Las decisiones pueden verse afectadas por emociones como el miedo y la codicia, lo que puede llevar a errores.
 - **Tiempo y Esfuerzo:** Requiere una dedicación constante para monitorear los mercados y ejecutar operaciones.
 - **Limitaciones Humanas:** Es difícil procesar grandes cantidades de datos rápidamente, lo que puede limitar la capacidad de aprovechar oportunidades de mercado.

TRADING AUTOMÁTICO

Utiliza programas informáticos (bots) para ejecutar operaciones automáticamente según reglas y algoritmos predefinidos.

- **Ventajas:**
 - **Velocidad y Eficiencia:** Los bots pueden ejecutar operaciones en milisegundos, aprovechando oportunidades que serían imposibles de captar manualmente.
 - **Disciplina y Consistencia:** Elimina las decisiones impulsivas, siguiendo estrictamente las reglas establecidas.
 - **Capacidad de Procesamiento:** Puede analizar y procesar grandes volúmenes de datos simultáneamente.

- **Desventajas:**
 - **Dependencia Tecnológica:** Requiere una infraestructura tecnológica robusta y puede ser vulnerable a fallos técnicos.
 - **Falta de Flexibilidad:** Los bots siguen reglas estrictas y pueden no adaptarse bien a condiciones de mercado altamente volátiles o inesperadas.

- **Costos Iniciales:** Desarrollo y mantenimiento de bots sofisticados pueden ser costosos.

COMPARACIÓN Y COMPLEMENTARIEDAD

- **Integración de ambos enfoques:** Muchos traders utilizan una combinación de trading manual y automático para beneficiarse de la flexibilidad humana y la eficiencia de los bots.
- **Ejemplo:** Un trader puede utilizar bots para ejecutar operaciones rutinarias y aprovechar pequeñas oportunidades, mientras reserva el trading manual para estrategias más complejas.

2.4.3. Indicadores Técnicos Utilizados en Bots de Trading

Actualmente, existe una amplia variedad de indicadores diseñados para ayudar a identificar si el precio de un activo se encuentra en una tendencia alcista o bajista, proporcionando señales que sugieren una mayor probabilidad de que el precio suba o baje.

Los bots de trading suelen apoyarse en estos indicadores para tomar decisiones automatizadas de compra o venta, maximizando así las oportunidades del mercado.

En este trabajo se analizan algunos de los indicadores más utilizados en la industria: las medias móviles exponenciales (**EMA**), el índice de fuerza relativa (**RSI**), los order blocks (**OB**) y los fair value gaps (**FVG**). A continuación, se detallan sus características y funcionamiento. [\[23\]](#)

MEDIAS MÓVILES EXPONENCIALES (EMA)

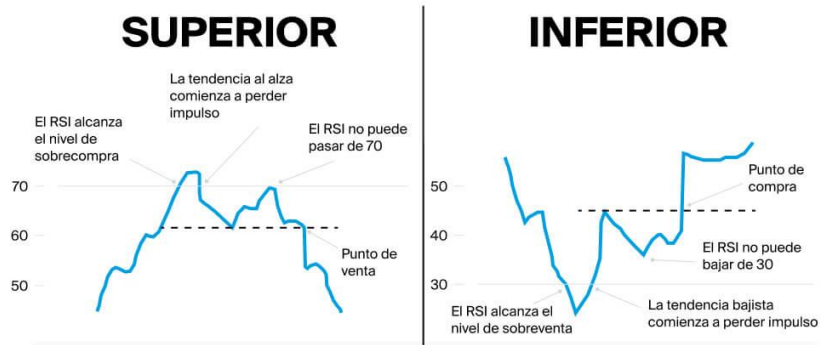


Fuente: [EMA](#)

La EMA es un tipo de media móvil que da más peso a los precios recientes, lo que la hace más sensible a los cambios de precio en comparación con la media móvil simple (SMA). La EMA es ampliamente utilizada para identificar la dirección de la tendencia y posibles puntos de entrada y salida en el mercado.

En este trabajo, se ha utilizado la EMA de **100 períodos** como filtro para determinar la tendencia general del mercado. Si el precio está por encima de la EMA, se considera una tendencia alcista, y si está por debajo, una tendencia bajista.

ÍNDICE DE FUERZA RELATIVA (RSI)



Fuente: [RSI](#)

El RSI es un oscilador de "momentum" que mide la velocidad y el cambio de los movimientos de precios. Oscila entre 0 y 100 y se utiliza para identificar condiciones de sobrecompra o sobreventa en el mercado. Un valor de RSI por encima de **70** generalmente indica una condición de sobrecompra, mientras que un valor por debajo de **30** indica una condición de sobreventa.

En este trabajo, el RSI se utiliza junto con la EMA para identificar momentos óptimos para comprar o vender.

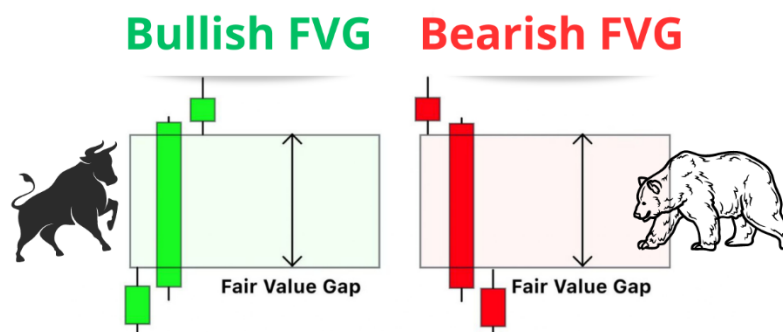
ORDER BLOCKS



Los order blocks son áreas en el gráfico donde se ha producido una acumulación significativa de órdenes de compra o venta, típicamente por parte de grandes instituciones financieras. Estas zonas representan niveles de soporte y resistencia institucionales donde se esperan reacciones del precio. El **pivote alcista** se forma si el mínimo de esa vela es menor que los mínimos de sus **3 velas** a la izquierda y sus 3 velas a la derecha. El **pivote bajista** se forma si el máximo de esa vela es mayor que los máximos de sus 3 velas a la izquierda y sus **3 velas** a la derecha. Los order blocks se utilizan para identificar puntos de entrada y salida en el mercado.

FAIR VALUE GAPS (FVG)

How to Identify Fair Value Gaps



Fuente: [Fair Value Gaps](#)

Los FVG son **imbalance**s en el precio que se producen cuando existe una fuerte presión de compra o venta, lo que hace que el precio se mueva rápidamente en una dirección sin negociar en ciertos niveles. Estos huecos suelen ser rellenados por el precio en el futuro, ofreciendo oportunidades para entrar en el mercado. Se forman mediante **3 velas consecutivas** y el imbalance se encontrará en la segunda vela.

Un **FVG alcista** se forma cuando el máximo de la primera vela es menor que el mínimo de la tercera vela creando un gap en la segunda vela indicando fuerte presión alcista.

Un **FVG bajista** se forma cuando el mínimo de la primera vela es mayor que el máximo de la tercera vela creando un gap en la segunda vela indicando fuerte presión bajista. Los fair value gaps se utilizan para identificar puntos de entrada y salida en el mercado.

2.4.4. Inteligencia Artificial en los Bots de Trading

La incorporación de Inteligencia Artificial (IA) ha revolucionado el trading automático, permitiendo a los bots aprender y adaptarse a las condiciones cambiantes del mercado.

[24] [25]

APLICACIONES DE LA IA EN BOTS DE TRADING

- **Machine Learning:** Permite que los bots aprendan de datos históricos y patrones de mercado para mejorar sus predicciones y estrategias.
- **Deep Learning:** Utiliza redes neuronales profundas para analizar datos complejos y no lineales, capturando relaciones que serían difíciles de detectar con métodos tradicionales.
- **Procesamiento de Lenguaje Natural (NLP):** Facilita la interpretación de noticias, informes financieros y redes sociales para incorporar información cualitativa en las decisiones de trading.
- **Análisis Predictivo:** Herramientas de IA pueden prever movimientos de precios futuros basándose en tendencias y señales históricas.

VENTAJAS DE LA IA EN BOTS DE TRADING

- **Adaptabilidad:** Los bots pueden ajustar sus estrategias en tiempo real según nuevas informaciones y cambios en el mercado.
- **Precisión Mejorada:** La capacidad de analizar grandes volúmenes de datos con mayor precisión reduce el riesgo de errores humanos.
- **Optimización Continua:** Los algoritmos de IA pueden optimizar y refinar sus propias reglas basándose en el rendimiento pasado y en nuevas tendencias.

DESAFÍOS Y CONSIDERACIONES

- **Overfitting:** Existe el riesgo de que los modelos de IA se ajusten demasiado a los datos históricos, perdiendo efectividad en condiciones futuras no previstas.
- **Transparencia y Explicabilidad:** Los modelos de IA, especialmente los de deep learning, pueden ser cajas negras, dificultando la comprensión de cómo se toman las decisiones.
- **Requerimientos Técnicos:** Implementar y mantener sistemas de IA requiere conocimientos avanzados en programación, estadística y ciencia de datos.

Ejemplos de Bots de Trading con IA

- **Kavout:** Utiliza *machine learning* para analizar datos y generar señales de trading basadas en patrones detectados.
- **Numerai:** Plataforma que utiliza inteligencia colectiva de científicos de datos para crear modelos de predicción de mercado.

2.4.5. Aspectos Éticos y Legales del Uso de Bots

El uso de bots de trading plantea una serie de cuestiones éticas y legales que deben ser consideradas para garantizar un funcionamiento justo y transparente de los mercados financieros. Si bien estos sistemas han mejorado la eficiencia y la liquidez en los mercados, también han generado debates sobre su impacto en la equidad, la regulación y la ética del *trading*. [26]

ASPECTOS ÉTICOS

- **Transparencia:** Es crucial que los *traders* y desarrolladores sean transparentes sobre cómo funcionan sus bots y qué estrategias emplean.
- **Equidad en el Mercado:** Los bots, especialmente los de alta frecuencia, pueden crear una ventaja injusta para aquellos con acceso a tecnología avanzada, lo que puede marginar a los *traders* minoristas.
- **Impacto en la Estabilidad del Mercado:** Un uso excesivo de bots de *trading* puede aumentar la volatilidad y el riesgo de eventos de mercado extremos, como el "*flash crash*".
- **Responsabilidad:** Determinar quién es responsable en caso de errores o pérdidas causadas por bots automatizados puede ser complicado.

ASPECTOS LEGALES

- **Regulación del Trading Algorítmico:** Muchos países han implementado regulaciones específicas para el trading automatizado, exigiendo a los *traders* cumplir con ciertos estándares y reportar actividades.
- **Protección de Datos:** Los bots que utilizan datos personales o sensibles deben cumplir con las leyes de protección de datos, como el GDPR en Europa.
- **Prevención de Manipulación del Mercado:** Las leyes buscan prevenir prácticas como el "*spoofing*" y el "*layering*", donde los bots colocan órdenes falsas para manipular los precios de mercado.
- **Licencias y Cumplimiento:** Las empresas que desarrollan y operan bots de trading pueden necesitar licencias específicas y deben asegurarse de cumplir con todas las normativas aplicables.

EJEMPLOS DE REGULACIONES ACTUALES

La creciente influencia de los bots ha llevado a reguladores de todo el mundo a introducir normativas específicas para mitigar sus riesgos. Entre las regulaciones más destacadas se encuentran:

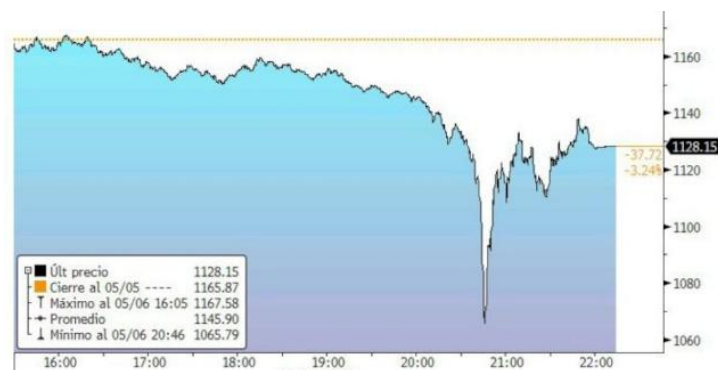


Fuente: [Normativa Financiera](#)

- **MiFID II (Unión Europea):** Implementada en 2018, esta directiva exige a los participantes del mercado que utilicen algoritmos de trading registrados y sometidos a pruebas rigurosas antes de su implementación. Además, los operadores deben garantizar que sus bots no generen una volatilidad excesiva ni comportamientos manipulativos. También impone requisitos de transparencia en las operaciones automatizadas y obliga a registrar todas las actividades para futuras auditorías.
- **Dodd-Frank Act (Estados Unidos):** Introducida en 2010, esta ley regula múltiples aspectos del sistema financiero, incluyendo la supervisión del trading algorítmico. Entre sus disposiciones, se incluye la obligación de reportar actividades sospechosas y mantener registros detallados de todas las operaciones realizadas por algoritmos.

CASOS DOCUMENTADOS DE USO MALINTENCIONADO DE BOTS

El uso de bots con fines malintencionados ha generado varios escándalos en los últimos años. Algunos de los casos más relevantes incluyen:



Fuente: [Flash Crash](#)

- **El Flash Crash de 2010:** El 6 de mayo de 2010, el mercado estadounidense sufrió una caída abrupta y temporal, con el índice Dow Jones perdiendo casi 1,000 puntos en minutos. Posteriormente, se descubrió que el operador Navinder Singh Sarao utilizó bots para realizar "**spoofing**", colocando grandes órdenes falsas para manipular los precios. Esto desestabilizó el mercado y aumentó la desconfianza hacia el HFT.

- **Manipulación de Divisas:** Algunos bots han sido diseñados para influir artificialmente en el mercado de divisas (**Forex**), manipulando spreads y ejecutando operaciones en puntos estratégicos para alterar temporalmente los precios. Este tipo de actividades ha llevado a multas significativas para bancos y empresas que toleraron estas prácticas.
- **Casos Recientes:** Empresas menos reguladas han sido acusadas de emplear algoritmos que deliberadamente manipulan mercados pequeños o menos líquidos para obtener beneficios, dañando a los inversores minoristas.

DEBATE SOBRE LA EQUIDAD EN EL USO DE BOTS

Los bots han introducido una capa de complejidad en los mercados financieros, lo que ha dado lugar a preguntas sobre la equidad:

1. **Diferencias entre Inversores Institucionales y Minoristas:** Los inversores institucionales, que cuentan con acceso a tecnología avanzada y recursos financieros significativos, pueden implementar bots más sofisticados, obteniendo ventajas claras sobre los minoristas. Esto genera una brecha tecnológica que puede percibirse como injusta.
2. **Deshumanización del Trading:** Los bots han eliminado en gran medida la toma de decisiones humanas en algunos sectores del mercado. Aunque esto mejora la eficiencia, también plantea preocupaciones sobre la pérdida de juicio humano, especialmente en situaciones excepcionales o de crisis.
3. **Efectos de los Bots en la Volatilidad:** Aunque los defensores argumentan que los bots mejoran la liquidez, también existe evidencia de que pueden amplificar movimientos de precios durante periodos de incertidumbre, aumentando la volatilidad del mercado.

CONSIDERACIONES PARA LOS DESARROLLADORES Y TRADERS

- **Cumplimiento Normativo:** Asegurarse de que los bots cumplen con todas las leyes y regulaciones aplicables.
- **Ética en el Diseño:** Incorporar principios éticos en el desarrollo de bots, como la equidad y la transparencia.
- **Monitoreo y Supervisión:** Mantener un monitoreo constante de los bots para detectar y corregir comportamientos no deseados o ilegales.

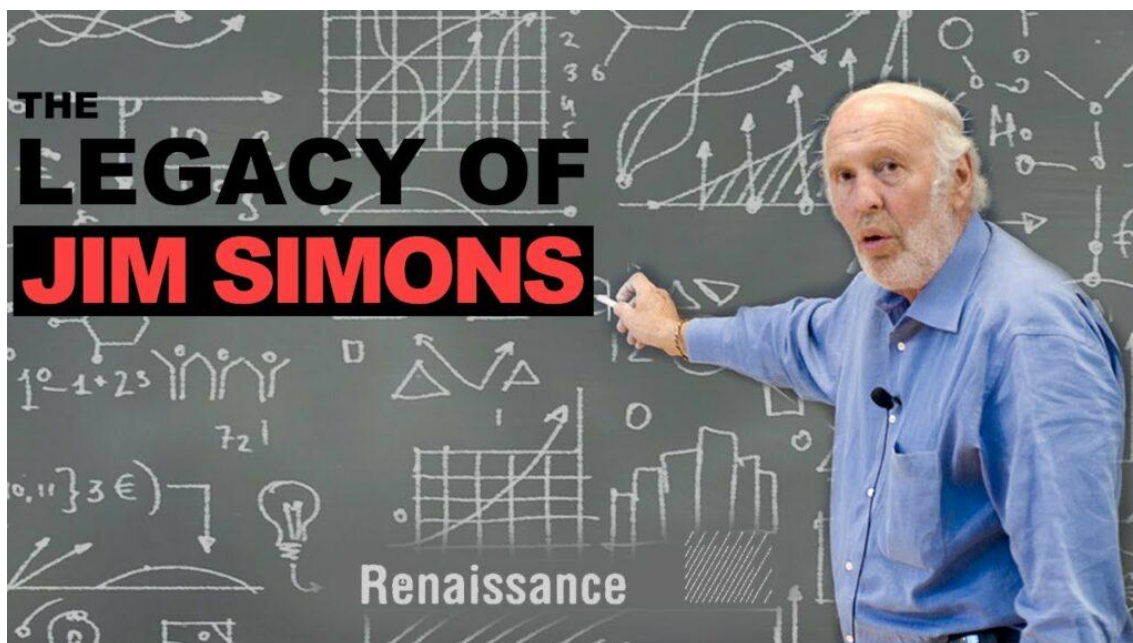
2.4.6. Hazañas en el trading algorítmico

El **trading algorítmico** ha revolucionado el panorama financiero al permitir a *traders* y fondos de inversión aprovechar los datos masivos, las matemáticas avanzadas y las tecnologías de última generación. Este enfoque permite tomar decisiones de inversión más rápidas y precisas, imposibles de alcanzar mediante métodos tradicionales.

Desde la creación de modelos predictivos hasta sistemas de ejecución de alta frecuencia, el éxito en el trading algorítmico radica en la capacidad de analizar patrones

invisibles al ojo humano y ejecutar estrategias con una velocidad que supera cualquier limitación humana.

JIM SIMONS Y RENAISSANCE TECHNOLOGIES



Fuente: [Jim Simons](#)

Jim Simons es uno de los traders más exitosos de la historia, conocido como el "Rey del Trading Algorítmico". Su éxito está íntimamente ligado a **Renaissance Technologies**, el fondo de inversión que fundó en 1982 y que revolucionó el mundo de las finanzas mediante el uso de modelos matemáticos y algoritmos. [27]

El Éxito de Jim Simons en el Trading Algorítmico

Jim Simons, a través de **Renaissance Technologies**, revolucionó el trading algorítmico. Su fondo más famoso, el **Medallion Fund**, ha logrado retornos excepcionales, superando el **66%** anual bruto. Renaissance utiliza modelos matemáticos complejos y Big Data para predecir precios y gestionar riesgos, basándose completamente en algoritmos y datos, sin espacio para la intuición humana.

Simons reclutó matemáticos, físicos e informáticos, en lugar de financieros tradicionales, para crear sus modelos predictivos. Además, fue pionero en el uso de **Big Data**, comenzando a analizar grandes volúmenes de información antes de que fuera popular, y desarrolló una infraestructura tecnológica avanzada para ejecutar operaciones de alta frecuencia.

El **éxito** de Renaissance se debe a su capacidad para detectar ineficiencias en los mercados y ejecutar millones de operaciones rápidamente con márgenes pequeños pero constantes. Su enfoque científico y secreto ha inspirado la creación de numerosos fondos cuantitativos en todo el mundo, popularizando el **trading algorítmico** y demostrando su efectividad frente al enfoque tradicional.

IRENE ALDRIDGE: MEJOR TRADER ALGORÍTMICA



Fuente: [Irene Aldridge](#)

Irene Aldridge es una figura destacada en el mundo del trading algorítmico, conocida por su contribución al desarrollo y entendimiento de estrategias cuantitativas y de alta frecuencia (HFT). Ha sido reconocida como una líder intelectual y práctica en el campo, gracias a sus aportes tanto como autora, académica y profesional en el desarrollo de algoritmos financieros. [28]

El Éxito de Irene Aldridge en el Trading Algorítmico

Es autora del libro ***High-Frequency Trading: A Practical Guide to Algorithmic Strategies and Trading Systems***, considerado esencial para entender las estrategias algorítmicas y de alta frecuencia. Su trabajo ha sido ampliamente citado en la industria financiera. Además, es presidenta y directora de **AbleMarkets**, una firma especializada en análisis cuantitativo y datos avanzados para mercados financieros. A lo largo de su carrera, ha impartido talleres y conferencias internacionales sobre algoritmos de *trading* y su impacto en los mercados.

Ha desarrollado estrategias algorítmicas para aprovechar patrones de mercado a alta velocidad, integrando datos masivos y tecnología avanzada. Su trabajo abarca desde el análisis de la microestructura del mercado hasta la gestión del riesgo en *trading* de alta frecuencia (**HFT**) y la predicción de volatilidad. También ha sido pionera en el uso de datos de alta frecuencia para identificar dinámicas de mercado no evidentes, mejorando la liquidez y eficiencia en los mercados financieros.

Su enfoque ha ayudado a estandarizar el conocimiento sobre HFT, haciendo que un campo complejo sea más accesible. Además, ha estudiado el impacto del trading algorítmico en la liquidez y volatilidad, contribuyendo al debate sobre sus ventajas y desventajas. Como mujer en una industria predominantemente masculina, ha inspirado a nuevas generaciones de profesionales, promoviendo la diversidad en el ámbito del *trading* algorítmico.

2.4.7. ¿Funcionan los Bots de Trading?

La efectividad de los bots de trading es un tema de debate entre inversores, *traders* y académicos. A continuación, se analizan los factores que determinan si un bot de trading puede ser exitoso:

FACTORES QUE INFLUYEN EN EL ÉXITO DE LOS BOTS DE TRADING

1. Calidad del Algoritmo:

- **Diseño del Algoritmo:** Un algoritmo bien diseñado que incorpora análisis robustos y estrategias efectivas es fundamental para el éxito del bot.
- **Adaptabilidad:** La capacidad del bot para adaptarse a cambios en las condiciones del mercado es crucial para mantener su rendimiento a largo plazo.

2. Calidad de los Datos:

- **Precisión y Actualización:** Los bots dependen de datos precisos y actualizados para tomar decisiones informadas. Datos de mala calidad pueden llevar a decisiones erróneas.
- **Fuentes de Datos:** Utilizar múltiples fuentes de datos puede mejorar la capacidad del bot para analizar y predecir movimientos de mercado.

3. Gestión del Riesgo:

- **Implementación de Estrategias de Gestión de Riesgo:** Establecer límites de pérdida, diversificación y otras estrategias de gestión del riesgo es esencial para proteger el capital.
- **Backtesting y Optimización:** Realizar pruebas retrospectivas y optimizar las estrategias basándose en los resultados históricos puede mejorar la efectividad del bot.

4. Infraestructura Tecnológica:

- **Velocidad y Fiabilidad:** Una infraestructura tecnológica robusta que garantice ejecuciones rápidas y sin fallos es fundamental para el rendimiento del bot.
- **Seguridad:** Proteger el bot contra ciberataques y garantizar la integridad de las operaciones es esencial para evitar pérdidas y fraudes.

VENTAJAS DE LOS BOTS DE TRADING EXITOSOS

- **Consistencia:** Los bots pueden seguir estrictamente las reglas establecidas sin dejarse llevar por emociones, manteniendo una consistencia en las operaciones.
- **Eficiencia en la Ejecución:** Pueden ejecutar órdenes más rápidamente y con mayor precisión que los *traders* humanos, aprovechando oportunidades que de otro modo se perderían.
- **Capacidad de Monitoreo:** Pueden operar 24/7, monitoreando múltiples mercados y activos simultáneamente.

DESAFÍOS Y RIESGOS

- **Sobreoptimización:** Existe el riesgo de que los bots estén demasiado ajustados a datos históricos, lo que puede disminuir su rendimiento en condiciones de mercado futuras no previstas.
- **Cambios en el Mercado:** Los mercados financieros son dinámicos y los bots deben ser actualizados regularmente para adaptarse a nuevas condiciones y regulaciones.
- **Fallas Técnicas:** Problemas como desconexiones, errores de programación o ataques cibernéticos pueden afectar negativamente el rendimiento del bot.
- **Competencia:** En mercados altamente competitivos, los bots deben ser continuamente mejorados para mantener una ventaja competitiva.

CONCLUSIÓN SOBRE LA EFECTIVIDAD DE LOS BOTS DE TRADING

Los bots de *trading* pueden ser una herramienta poderosa para mejorar la eficiencia y consistencia en las operaciones financieras. Sin embargo, su efectividad depende en gran medida de la calidad de los algoritmos, la gestión del riesgo, la infraestructura tecnológica y la capacidad de adaptarse a las condiciones cambiantes del mercado. Los inversores deben realizar una diligencia debida exhaustiva y mantener una supervisión continua para maximizar los beneficios y minimizar los riesgos asociados con el uso de bots de *trading*.

3. DESARROLLO

3.1. Creando un bot de trading

3.1.1. ¿Qué mercados escogemos?



Fuente: [Mercado de Criptomonedas](#)

Debido a la facilidad de conseguir datos precisos de los mercados financieros, se decidió utilizar el mercado de criptomonedas el cuál es el mercado más emergente en la actualidad.

3.1.2. ¿Qué Exchange operamos?

Para la implementación y operación de los bots de *trading*, se utilizaron dos tipos de *exchanges*:

PHEMEX



Fuente: [Phemex](#)

Phemex es un *exchange* centralizado (**CEX**) que ofrece *trading* con criptomonedas. Al ser un CEX, todas las transacciones se gestionan a través de una plataforma central, lo que implica que los fondos de los usuarios se almacenan en las billeteras del *exchange*.

Phemex es conocido por su baja latencia y la amplia variedad de pares de *trading* que ofrece. La facilidad de uso y la disponibilidad de API robustas para la integración de bots hacen que sea una opción popular para el *trading* automatizado.

HYPERLIQUID



Fuente: [Hyperliquid](#)

Hyperliquid es un *exchange* descentralizado (**DEX**) que permite a los usuarios operar directamente desde sus billeteras sin necesidad de intermediarios. Los DEX son conocidos por ofrecer mayor privacidad y seguridad, ya que los usuarios mantienen el control total de sus fondos.

Hyperliquid utiliza contratos inteligentes para facilitar el *trading*, y su arquitectura descentralizada permite operaciones sin necesidad de un registro o verificación de

identidad. Aunque los DEX suelen tener una menor liquidez en comparación con los CEX, ofrecen una alternativa más segura y transparente.

3.1.3. ¿Qué activo operamos?



Fuente 1: [Bitcoin](#)

Fuente 2: [WIFUSDT](#)

Para realizar las pruebas históricas (*backtesting*), hemos utilizado el activo **Bitcoin**, debido a su menor volatilidad en comparación con otros activos, lo que nos permite evaluar con mayor precisión los resultados de la estrategia.

Sin embargo, para la implementación del bot en el mercado real, hemos optado por operar con el activo **WIFUSDT**. Este activo es confiable y se ajusta mejor a nuestra estrategia, permitiendo una ejecución más eficiente.

No hemos empleado Bitcoin en el mercado real debido a problemas relacionados con la ejecución de órdenes a mercado con poco capital. El alto valor de Bitcoin, que supera los 100,000 USD, impide realizar compras con montos pequeños en el *exchange*, ya que no se permite operar con cantidades mínimas tan reducidas.

3.1.3. ¿Qué temporalidad escogemos?

Hemos decidido utilizar una temporalidad de **6 horas** en los tres primeros bots desarrollados para el *exchange* **Phemex**. Esta temporalidad nos permite reducir el número de operaciones realizadas por el bot, enfocándonos únicamente en aquellas de mayor calidad. Esto se debe a que los patrones detectados en temporalidades más amplias, como 1 día, tienden a ser más robustos que los de intervalos más cortos, como 1 minuto.

Por otro lado, en los tres primeros bots diseñados para **Hyperliquid**, hemos optado por una temporalidad de **1 hora**. Esta decisión busca diversificar las estrategias y garantizar que los bots de ambos *exchanges* no realicen las mismas operaciones.

Finalmente, en el bot que combina dos temporalidades, hemos utilizado una temporalidad alta de **1 hora** y una temporalidad baja de **15 minutos** en **ambos exchanges**. Esto nos permite aprovechar lo mejor de ambas escalas temporales: la

robustez de una temporalidad alta y la precisión en la entrada de operaciones que ofrece una temporalidad baja.

3.2. Descripción de los Bots de Trading

Todos los bots de trading están configurados para utilizar niveles de *take profit* y *stop loss*, asegurando una gestión del riesgo adecuada. Cada operación realizada por los bots arriesga únicamente un 1% del capital total, lo que permite minimizar posibles pérdidas mientras se maximiza la relación riesgo-beneficio.

3.2.1. Bot de EMA y RSI



Figura 1: Ejecución de la estrategia del Bot EMA y RSI

Este bot realiza la estrategia en la temporalidad de 6h utilizando dos indicadores(la EMA y el RSI). La combinación de estos indicadores permitirá realizar un mejor análisis y por ende una mejor predicción del precio.

Se utilizará la **EMA de 100 periodos** para determinar la tendencia del mercado y el **RSI** para determinar si el precio está sobrecomprado o sobrevendido. La lógica para relizar órdenes de compra y venta es la siguiente:

- **Entrada en compra:** Cuando el precio esté por encima de la EMA de 100 periodos indicando que está en una tendencia alcista y el **RSI** esté por debajo de los **30** puntos indicando que está sobrevendido. Cuando se cumplan ambas

condiciones realizará una orden de compra a mercado con un *stop los* debajo de la EMA con un pequeño *buffer* y el *take profit* de un riesgo beneficio de 1 a 3.

- **Entrada en venta:** Cuando el precio esté por debajo de la EMA de 100 periodos indicando que está en una tendencia bajista y el **RSI** esté por encima de los **70** puntos indicando que está sobrecomprado. Cuando se cumplan ambas condiciones realizará una orden de venta a mercado con un *stop loss* encima de la EMA con un pequeño *buffer* y el *take profit* de un riesgo beneficio de 1 a 3.

3.2.2. Bot de Order Block



Figura 2: Ejecución de la estrategia del Bot EMA y Order Block

Este bot realiza la estrategia en la temporalidad de 6h utilizando dos indicadores(la EMA y el order block). La combinación de ambos indicadores permite realizar un mejor análisis realizando operaciones precisas.

Se utilizará la **EMA de 100 periodos** para determinar la tendencia del mercado y el **order block** para determinar donde entraron grandes instituciones y poder hacer entradas precisas. El order block tiene un mínimo de tamaño para considerarlo relevante que es previamente definido. La lógica para relizar órdenes de compra y venta es la siguiente:

- **Entrada en compra:** Cuando el precio se encuentra por encima de la EMA de 100 periodos, lo que confirma una tendencia alcista, el sistema escanea los order blocks previamente identificados (que no han sido mitigados) y, si detecta que la vela actual está interactuando con un order block alcista, coloca una orden límite de compra en el máximo de ese order block.
- El *stop loss* se sitúa justo debajo del mínimo del orderblock con un pequeño *buffer*, mientras que el *take profit* se fija para alcanzar una relación riesgo-beneficio de 1:3. **Entrada en venta:** Si el precio está por debajo de la EMA de 100 periodos indicando que la tendencia es bajista, buscará entre los order blocks detectados hasta esa vela y que no hayan sido mitigados con anterioridad. Si es un **orderblock bajista** y la vela actual mitiga el orderblock pondrá una orden de venta en el mínimo del order block, con el stop loss encima del máximo del order block dándole un pequeño *buffer* y con un *take profit* de riesgo beneficio 1 a 3.

3.2.3. Bot de Fair Value Gap (FVG)



Figura 3: Ejecución de la estrategia del Bot EMA y Fair Value Gap

Este bot realiza la estrategia en la temporalidad de 6h utilizando dos indicadores(la EMA y el fair value gap). La combinación de ambos indicadores permite realizar un mejor análisis realizando operaciones precisas.

Se utilizará la **EMA de 100 periodos** para determinar la tendencia del mercado y el **fair value gap** para determinar hay un desbalance entre oferta y demanda y poder hacer entradas precisas continuando la tendencia. El fair value gap tiene un mínimo de tamaño para considerarlo relevante que es previamente definido. La lógica para relizar órdenes de compra y venta es la siguiente:

- **Entrada en compra:** Si el precio está por encima de la EMA de 100 periodos indicando que la tendencia es alcista, buscará entre los fair value gaps detectados hasta esa vela y que no hayan sido mitigados con anterioridad Si es un **fair value gap alcista** y la vela actual mitiga el fair value gap pondrá una orden límite de compra en el máximo del fair value gap, con el stop los debajo del mínimo del fair value gap dándole un pequeño *buffer* y con un *take profit* de riesgo beneficio 1 a 3.
- **Entrada en venta:** Si el precio está por debajo de la EMA de 100 periodos indicando que la tendencia es bajista, buscará entre los fair value gaps detectados hasta esa vela y que no hayan sido mitigados con anterioridad y si es un **fair value gap bajista** y la vela actual mitiga el fair value gap pondrá una orden límite de venta en el mínimo del fair value gap, con el *stop loss* encima del máximo del fair value gap dándole un pequeño *buffer* y con un *take profit* de riesgo beneficio 1 a 3.

3.2.4. Bot usando Multi TimeFrame Analysis

Multi Timeframe Analysis es la combinación de una temporalidad alta y una temporalidad baja que ayuda a ser o más precisos o tener una visión más genérica. Esto es útil ya que el mercado es **fractal** lo que quiere decir que lo que ocurre en una temporalidad, de la misma forma ocurre en una temporalidad inferior. Hay dos formas de usarla:

La primera es haciendo “**Upsampling**” que es usar la temporalidad pequeña como principal en la que tienes definida la lógica de compra y venta basada tanto en indicadores como el precio y usa la temporalidad alta para confirmar que está operando alineado con la temporalidad grande ya que aumenta las probabilidades de la operación.

La segunda forma es haciendo “**Downsampling**” en el que la lógica principal de la estrategia se basa en la temporalidad alta en la que ves la tendencia principal y en la temporalidad baja entras en el detalle viendo donde se alinean ambas temporalidades para dar más probabilidad a la operación.



Figura 4: Ejecución de la estrategia del Bot Multi Timeframe Analysis

En este bot he utilizado “**Downsampling**”, usando como temporalidad alta el **6h** y como temporalidad baja el **1h**.

Este bot no utiliza las **EMAS** para determinar la tendencia y solo se basa en buscar pequeños movimientos del precio utilizando **order blocks** que tienen un **fair value gap** en la siguiente vela para conseguir una entrada precisa. La lógica del compra/venta es la siguiente:

- **Entrada en compra:**

La estrategia de entrada se basa en la interacción entre dos temporalidades:

- **Temporalidad alta:** Se busca que el precio mitigue un Order Block alcista.
- **Temporalidad baja:** Una vez identificado el Order Block de la temporalidad alta, se comprueban los Order Blocks en la temporalidad baja que estén alineados dentro del de la temporalidad alta.

El bot continuará buscando órdenes de compra hasta que el precio rebase el mínimo del Order Block de la temporalidad alta o se forme una nueva vela en la temporalidad alta.

Se colocará una orden límite de compra en el máximo del Order Block de la temporalidad baja.

El *stop loss* se ubicará por debajo del mínimo del Order Block de la temporalidad baja, añadiendo un pequeño.

El *take profit* se establecerá con una relación riesgo-beneficio de 1 a 3, maximizando el potencial de ganancias frente al riesgo asumido.

- **Entrada en venta:**

La estrategia de entrada se basa en la interacción entre dos temporalidades:

- **Temporalidad alta:** Se busca que el precio mitigue un Order Block bajista.
- **Temporalidad baja:** Una vez identificado el Order Block de la temporalidad alta, se comprueban los Order Blocks en la temporalidad baja que estén alineados dentro del de la temporalidad alta.

El bot continuará buscando órdenes de venta hasta que el precio rebase el máximo del Order Block de la temporalidad alta o se forme una nueva vela en la temporalidad alta.

Se colocará una orden límite de venta en el mínimo del Order Block de la temporalidad baja.

El *stop loss* se ubicará por encima del máximo del Order Block de la temporalidad baja, añadiendo un pequeño *buffer*.

El *take profit* se establecerá con una relación riesgo-beneficio de 1:3, maximizando el potencial de ganancias frente al riesgo asumido.

Se puede plantear el uso de emplear la EMA para tener una mayor confirmación de que el Order Block va a ser respetado por ir a favor de la tendencia.

3.3. Implementación de las estrategias

3.3.1 Herramientas y Lenguajes de Programación Utilizados

Para la implementación de los bots, se ha utilizado Python como lenguaje de programación principal debido a su simplicidad y la amplia variedad de bibliotecas especializadas disponibles para el análisis de datos financieros, como Pandas, NumPy y TA-Lib.

La conexión con las plataformas de trading se ha realizado mediante las APIs proporcionadas por *brokers* como Phemex, utilizando la biblioteca *ccxt* para interactuar con los mercados de criptomonedas. Además, se han importado librerías específicas para trabajar con el *exchange* Hyperliquid, como **eth_account**, que permite firmar las operaciones de forma segura.

3.3.2 Indicadores Realizados

Se desarrollaron los indicadores de detección de **Order Blocks** y **Fair Value Gaps** con el objetivo de proporcionar una representación visual más clara. Esto permitió verificar durante el *backtesting* que el bot ejecutaba las operaciones esperadas y, al mismo tiempo, visualizar de forma dinámica tanto los Order Blocks como los Fair Value Gaps directamente en el gráfico.

INDICADOR DE DETECCIÓN DE ORDER BLOCKS



Figura 5: Indicador de detección de Order Blocks

El indicador de detección de Order Blocks identifica zonas clave de oferta y demanda basándose en datos históricos (**BTC/USD**). Su funcionamiento se resume así:

Identificación de Pivotes: Detecta pivotes altos (alcistas) y bajos (bajistas) evaluando si el máximo o mínimo de una vela cumple criterios en relación con las 3 velas anteriores y posteriores.

Aquellos que ya fueron mitigados, es decir, que el precio ya llegó a esos niveles no se consideran válidos para futuras operaciones.

Cálculo de EMA: Calcula una media móvil exponencial (por defecto, periodo 100) para visualizar la tendencia general.

Visualización: Grafica los Order Blocks (verdes: alcistas, rojos: bajistas) y la EMA con **Plotly** para análisis interactivo.



Pseudocódigo:

Función identificar_bloques_de_órdenes(datos):

Inicializar señales_ob como una lista de ceros con la misma longitud que datos

Inicializar tops y bottoms como listas vacías

Inicializar índice_mitigación como una lista de ceros con la misma longitud que datos

Para i desde 1 hasta longitud de datos - 2:

pivote = identificar_pivote(datos, i)

Si pivote es alcista:

Si no hay un bloque de órdenes reciente o el bloque anterior no contiene al actual:

señales_ob[i] = 1

tops.append(datos[i].alto)

bottoms.append(datos[i].bajo)

Sino si pivote es bajista:

Si no hay un bloque de órdenes reciente o el bloque anterior no contiene al actual:

señales_ob[i] = -1

tops.append(datos[i].alto)

bottoms.append(datos[i].bajo)

Para cada índice i en señales_ob donde señales_ob[i] no es cero:

Si señales_ob[i] es alcista:

Para j desde $i + 1$ hasta longitud de datos:

Si datos[j].bajo \leq tops[i]:

Si hay un retroceso que confirma la dirección alcista:

índice_mitigación[i] = j

Romper el bucle

Sino si señales_ob[i] es bajista:

Para j desde $i + 1$ hasta longitud de datos:

Si datos[j].alto \geq bottoms[i]:

Si hay un retroceso que confirma la dirección bajista:

índice_mitigación[i] = j

Romper el bucle

Retornar una estructura con señales_ob, tops, bottoms e índice_mitigación

INDICADOR DE DETECCIÓN DE FAIR VALUE GAPS



Figura 6: Indicador de detección de Fair Value Gaps

El indicador de detección de Fair Value Gaps identifica zonas clave de oferta y demanda debido a grandes movimientos que causan ineficiencias en el mercado basándose en datos históricos (**BTC/USD**). Su funcionamiento se resume así:

Detección de FVGs:

Identifica FVGs alcistas (Máximo de la vela anterior < Mínimo de la siguiente) y bajistas (Mínimo de la vela anterior > Máximo de la siguiente).

Verifica si el FVG es mitigado, es decir, si el precio llena el gap.

Registra señales (1 para alcista, -1 para bajista), límites del FVG y el índice de mitigación.

Visualización:

Genera un gráfico interactivo con velas donde se destacan los FVGs: verdes (alcistas) y rojos (bajistas), mostrando su evolución desde su creación hasta su mitigación.



Pseudocódigo:

Función detectar_FVG(datos):

Inicializar señales_FVG como una lista de valores nulos con la misma longitud que datos

Inicializar tops y bottoms como listas vacías

Inicializar índice_mitigación como una lista de valores nulos con la misma longitud que datos

Para cada índice i desde 1 hasta longitud de datos - 2:

Si $\text{datos}[i-1].\text{alto} < \text{datos}[i+1].\text{bajo}$ Y $\text{datos}[i].\text{cierre} > \text{datos}[i].\text{apertura}$:

señales_FVG[i] = 1 // FVG alcista

tops[i] = datos[i+1].bajo

bottoms[i] = datos[i].alto

Sino si $\text{datos}[i-1].\text{bajo} > \text{datos}[i+1].\text{alto}$ Y $\text{datos}[i].\text{cierre} < \text{datos}[i].\text{apertura}$:

señales_FVG[i] = -1 // FVG bajista

tops[i] = datos[i].bajo

bottoms[i] = datos[i-1].alto

// Calcular índices de mitigación

Para cada índice i donde señales_FVG[i] no es nulo:

Si señales_FVG[i] es 1: // FVG alcista

Para cada índice j desde i+2 hasta longitud de datos:

Si $\text{datos}[j].\text{bajo} \leq \text{tops}[i]$:

Si existe un retroceso que confirma la dirección alcista:

índice_mitigación[i] = j

Salir del bucle

Sino si señales_FVG[i] es -1: // FVG bajista

Para cada índice j desde i+2 hasta longitud de datos:

Si $\text{datos}[j].\text{alto} \geq \text{bottoms}[i]$:

Si existe un retroceso que confirma la dirección bajista:

índice_mitigación[i] = j

Salir del bucle

Retornar una estructura con señales_FVG, tops, bottoms e índice_mitigación

3.3.3. Backtesting de las estrategias

En esta sección se describe la implementación del *backtest* para evaluar estrategias de *trading*. Sin embargo, las estadísticas de la librería **backtesting.py** no son completamente precisas, lo que obliga a calcular manualmente el retorno real de la estrategia en condiciones de mercado en vivo.

Esto ocurre porque tanto el **take profit** como el **stop loss** solo se ejecutan al cierre de la vela que activa la orden, lo que genera discrepancias en los precios de ejecución. Por ejemplo:

- Si entras a \$100 con un *stop loss* en \$90 y un *take profit* en \$130, pero la vela alcanza \$130 y luego cierra en \$120, la librería considera el cierre en \$120, reduciendo la ganancia al 2% en lugar del 3%.
- Lo mismo aplica para el *stop loss*: dependiendo del cierre de la vela, la pérdida puede ser mayor o menor al 1%.

Esta imprecisión se debe a que **backtesting.py** solo utiliza datos OHLCV (apertura, máximo, mínimo, cierre y volumen) y no sabe cómo evoluciona el precio dentro de cada vela. Por lo tanto, no puede determinar el orden exacto en que se alcanzan los precios, lo que impacta en la precisión del cálculo de retornos.

BOT DE EMA Y RSI

1. Preparación de datos:

Se utilizó un *dataset* histórico de BTC/USD con velas de 6 horas, incluyendo columnas estándar: Open, High, Low, Close y Volume. El *backtest* partió con un capital inicial de \$2,000,000, considerando comisiones del 0.2% por operación.

Los indicadores clave para la estrategia son:

- **EMA** (Media Móvil Exponencial) para identificar la tendencia.
- **RSI** (Índice de Fuerza Relativa) para detectar niveles de sobrecompra (<30) y sobreventa (>70).

2. Lógica de la estrategia:

Órdenes de compra o venta se ejecutan con las siguientes condiciones:

- **Compra:**
 - Precio > EMA (tendencia alcista) y RSI < 30.
 - Orden a mercado con *stop loss* 1% por debajo de la EMA y *take profit* con un ratio 1:3.
- **Venta:**
 - Precio < EMA (tendencia bajista) y RSI > 70.
 - Orden a mercado con *stop loss* 1% por encima de la EMA y *take profit* con un ratio 1:3.

4. Gestión del riesgo

La exposición por operación se limita al 1% del capital total, calculada en base al tamaño del *stop loss* y la relación riesgo-recompensa definida.

Resultado:



Figura 7: Resultados de la estrategia EMA y RSI

Start	2019-01-02 18:00:00
End	2024-01-01 18:00:00
Nº Trades	99
Return [%]	14.15%
Max Drawdown [%]	-5.52%
Win Rate	73%
Worst Trade	-11%

Tabla 1: Resultados del backtest de la estrategia EMA y RSI

Evaluación de los resultados del backtest

Según las estadísticas generadas por la librería `backtesting.py`, la estrategia reportó un 14% de retorno y una pérdida máxima del 5% en un período de 5 años. Sin embargo, estos resultados subestiman el rendimiento real, ya que la librería es imprecisa al cerrar las posiciones: las órdenes de *stop loss* y *take profit* se ejecutan al cierre de la vela en lugar del momento exacto en que se activan.

Esto se refleja en las métricas, como el *Worst Trade*, que muestra una pérdida de -11% en lugar del límite teórico de -1%.

Cálculo del retorno real

Para estimar el rendimiento real de la estrategia sin contar comisiones:

- Número total de operaciones: **99**

- Tasa de acierto (*Win Rate*): **73%**
- Relación riesgo-beneficio: **1:3** (3% en operaciones ganadoras, -1% en perdedoras).

1. **Cálculo de operaciones ganadoras y perdedoras:**

- Ganadoras: $99 \times 0.73 = 73$
- Perdedoras: $99 - 73 = 26$

2. **Cálculo del retorno total:**

- Ganancias: $73 \times 3\% = 219$
- Pérdidas: $26 \times -1\% = -26$
- Retorno neto: $219\% - 26\% = 193$

3. **Retorno anual promedio:**

$193\% \div 5 \text{ años} = 38.6\%$

4. **Conclusión**

Si la estrategia se hubiera ejecutado en vivo, con órdenes al momento de activarse y no al cierre de la vela, el retorno estimado habría sido significativamente mayor, alcanzando un promedio anual del **38.6%** en lugar del 14% reportado por la librería.

5. **Pseudocódigo:**

Función calcular_RSI(precios, periodo):

diferencias = calcular_diferencias(precios)

ganancias = calcular_ganancias(diferencias)

pérdidas = calcular_pérdidas(diferencias)

promedio_ganancias = calcular_media_móvil(ganancias, periodo)

promedio_pérdidas = calcular_media_móvil(pérdidas, periodo)

rs = promedio_ganancias / promedio_pérdidas

rsi = $100 - (100 / (1 + rs))$

Retornar rsi

Clase Estrategia_EMA_RSI:

periodo_ema = 150

periodo_rsi = 10

sobrecompra_rsi = 70

sobreventa_rsi = 30

Función inicializar():

Calcular EMA con periodo periodo_ema

Calcular RSI con periodo periodo_rsi

Función ejecutar():

precio_actual = obtener_precio_actual()

ema_actual = obtener_valor_ema()

rsi_actual = obtener_valor_rsi()

saldo_cuenta = obtener_saldo_cuenta()

monto_riesgo = saldo_cuenta * 0.01

tendencia_alcista = precio_actual > ema_actual

Si tendencia_alcista Y rsi_actual < sobreventa_rsi:

Si no hay posición abierta:

precio_entrada = precio_actual

take_profit = precio_entrada * 1.03

stop_loss = ema_actual * 0.99

riesgo_por_accion = precio_entrada - stop_loss

tamaño_posición=calcular_tamaño_posición(monto_riesgo,riesgo_por_accion,
precio_entrada)

Abrir posición de compra con take_profit, stop_loss y tamaño_posición

Sino si no tendencia_alcista Y rsi_actual > sobrecompra_rsi:

Si hay posición abierta:

precio_entrada = precio_actual

take_profit = precio_entrada * 0.97

stop_loss = ema_actual * 1.01

riesgo_por_accion = stop_loss - precio_entrada

tamaño_posición=calcular_tamaño_posición(monto_riesgo,riesgo_por_accion,
precio_entrada)

Abrir posición de venta con take_profit, stop_loss y tamaño_posición

BOT DE ORDER BLOCK

1. Preparación de los datos

Se utilizó un *dataset* histórico de BTC/USD con velas de 6 horas que incluye las columnas estándar (*Open, High, Low, Close, Volume*).

El capital inicial para el backtest fue de **\$2,000,000**, considerando comisiones del **0.2% por operación**.

2. Lógica de la estrategia

- **Condiciones generales:**
 - Se calcula la EMA de 100 periodos.
 - Se identifican *Order Blocks* alcistas o bajistas que no hayan sido mitigados.
- **Entradas en compra:**
 - Precio por encima de la EMA.
 - Si la vela actual mitiga un *Order Block* alcista, se coloca una orden límite en el límite superior del bloque.
 - Stop loss: ligeramente por debajo del límite inferior del bloque.
 - Take profit: ratio riesgo-beneficio de **1:3**.
- **Entradas en venta:**
 - Precio por debajo de la EMA.
 - Si la vela actual mitiga un *Order Block* bajista, se coloca una orden límite en el límite inferior del bloque.
 - Stop loss: ligeramente por encima del límite superior del bloque.
 - Take profit: ratio riesgo-beneficio de **1:3**.

4. Gestión del riesgo

Se limita la exposición al **1% del capital total por operación**, calculando el tamaño de la posición con base en el *stop loss* y el ratio riesgo-beneficio definido.

Resultado:



Figura 8: Resultados de la estrategia EMA y Order Block

Start	2019-01-02 18:00:00
End	2024-01-01 18:00:00
Nº Trades	350
Return [%]	-3.54%
Max Drawdown [%]	-9.2%
Win Rate	34%
Worst Trade	-10%

Tabla 2: Resultados del backtest de la estrategia EMA y Order Block

Evaluación de los resultados del backtest

Según las estadísticas generadas por la librería `backtesting.py`, la estrategia reportó un **-3.5% de retorno** y una **pérdida máxima del -9.2%** en un período de 5 años. Sin embargo, estos resultados subestiman el rendimiento real debido a que la librería es imprecisa al cerrar posiciones: las órdenes de *stop loss* y *take profit* se ejecutan al cierre de la vela, en lugar del momento exacto en que se activan.

Esta imprecisión queda reflejada en métricas como el **Worst Trade**, que reporta una pérdida del -10% en lugar del límite teórico del -1%.

Cálculo del retorno real

Para estimar el rendimiento real de la estrategia, excluyendo las comisiones, consideramos los siguientes parámetros:

- **Número total de operaciones:** 350.
- **Tasa de acierto (*Win Rate*):** 34%.
- **Relación riesgo-beneficio:** 1:3 (3% en operaciones ganadoras, -1% en perdedoras).

1. Cálculo de operaciones ganadoras y perdedoras:

- Operaciones ganadoras: $350 \times 0.34 = 119$
- Operaciones perdedoras: $350 - 119 = 231$

2. Cálculo del retorno total:

- Ganancias: $119 \times 3\% = 357\%$
- Pérdidas: $231 \times -1\% = -231\%$
- Retorno neto: $357\% - 231\% = 126\%$

3. Retorno promedio anual:

$126\% \div 5 = 25.2\%$ anual

4. Conclusión

Si la estrategia se hubiera ejecutado en vivo, con órdenes colocadas al momento en que se activan y no al cierre de la vela, el retorno estimado habría sido considerablemente

mayor, alcanzando un promedio anual del **25.2%**, en lugar del **-3.5%** reportado por la librería.

5. Pseudocódigo

Clase OBStrategy:

Parámetros:

- min_ob_height: Altura mínima del OB
- ema_period: Periodo de la EMA
- risk_reward_ratio: Relación riesgo/recompensa
- sl_offset: Desviación para el Stop Loss

Método init():

- Calcular self.obs usando identify_ob con los datos de mercado
- Inicializar self.processed_obs como conjunto vacío
- Calcular self.ema como la EMA de los precios de cierre con el periodo definido

Método next():

- Obtener current_index como el índice de la última vela
- Filtrar valid_obs de self.obs donde OBIndex <= current_index
- Calcular account_balance y risk_amount (1% del saldo)
- Determinar macro_trend_bullish basado en la relación entre el precio mínimo y la EMA

Para cada ob en valid_obs:

- Desempaquetar ob en ob_type, ob_top, ob_bottom, ob_mit_index, ob_index
- Calcular ob_height como la diferencia entre ob_top y ob_bottom
- Si ob_height < min_ob_height, continuar con el siguiente OB
- Verificar si el OB ha sido llenado y si ob_mit_index es válido
- Si macro_trend_bullish y ob_type es 1 (OB alcista):
 - Si current_index == ob_mit_index:
 - Calcular entry_price, stop_loss y take_profit
 - Determinar position_size basado en risk_amount y account_balance
 - Ejecutar orden de compra si las condiciones son favorables
- Si no es macro_trend_bullish y ob_type es -1 (OB bajista):

- Si `current_index == ob_mit_index`:
 - Calcular `entry_price`, `stop_loss` y `take_profit`
 - Determinar `position_size` basado en `risk_amount` y `account_balance`
 - Ejecutar orden de venta si las condiciones son favorables

BOT DE FAIR VALUE GAP

1. Preparación de los datos

Se utilizó un *dataset* histórico de **BTC/USD** con velas de 6 horas que incluye las columnas estándar (*Open, High, Low, Close, Volume*).

- **Capital inicial:** \$2,000,000.
- **Comisiones:** 0.2% por operación.

2. Lógica de la estrategia

• **Condiciones generales:**

- Se calcula la **EMA de 100 periodos**.
- Se identifican los **Fair Value Gaps (FVG)** alcistas o bajistas que no hayan sido mitigados.

• **Entradas en compra** (*tendencia alcista*):

- Precio por encima de la EMA.
- Si la vela actual mitiga un *FVG* alcista, se coloca una orden límite en el límite superior del *FVG*.
- **Stop loss:** ligeramente por debajo del límite inferior del *FVG*.
- **Take profit:** ratio riesgo-beneficio de 1 a 3.

• **Entradas en venta** (*tendencia bajista*):

- Precio por debajo de la EMA.
- Si la vela actual mitiga un *FVG* bajista, se coloca una orden límite en el límite inferior del *FVG*.
- **Stop loss:** ligeramente por encima del límite superior del *FVG*.
- **Take profit:** ratio riesgo-beneficio de 1:3.

3. Gestión del riesgo

Se limita la exposición al **1% del capital total por operación**, calculando el tamaño de la posición con base en el *stop loss* y el ratio riesgo-beneficio definido.

Resultado:



Figura 9: Resultados de la estrategia EMA y Fair Value Gap

Start	2019-01-02 18:00:00
End	2024-01-01 18:00:00
Nº Trades	995
Return [%]	1.56%
Max Drawdown [%]	-8.62%
Win Rate	37%
Worst Trade	-12%

Tabla 3: Resultados del backtest de la estrategia EMA y Fair Value Gap

4. Evaluación de los resultados del backtest

Según las estadísticas generadas por la librería `backtesting.py`, la estrategia reportó un **8%** de retorno y una pérdida máxima de **-8.6%** en un período de **5 años**. Sin embargo, estos resultados subestiman el rendimiento real debido a que la librería es imprecisa al cerrar las posiciones: las órdenes de *stop loss* y *take profit* se ejecutan al cierre de la vela, en lugar del momento exacto en que se activan. Esta imprecisión se refleja en métricas como:

- **Worst Trade**, que reporta una pérdida de -12% en lugar del límite teórico de -1%.
- **Best Trade**, que muestra un beneficio de +32% en lugar del +3% esperado.

Cálculo del retorno real

Para estimar el rendimiento real de la estrategia, excluyendo las comisiones, consideramos los siguientes parámetros:

- Número total de operaciones: 995

- Tasa de acierto (Win Rate): 37%
- Relación riesgo-beneficio: 1 a 3 (3% en operaciones ganadoras, -1% en perdedoras).

Cálculo de operaciones ganadoras y perdedoras:

- Operaciones ganadoras: $995 \times 0.37 = 368$
- Operaciones perdedoras: $995 - 368 = 627$

Cálculo del retorno total:

- Ganancias: $368 \times 3\% = 1104\%$
- Pérdidas: $627 \times -1\% = -627\%$
- Retorno neto: $1104\% - 627\% = 477\%$

Retorno **promedio** **anual:**
 $477\% \div 5 = 95.4\%$ anual

5. Conclusión

Si la estrategia se hubiera ejecutado en vivo, con órdenes colocadas al momento en que se activan y no al cierre de la vela, el retorno estimado habría sido considerablemente mayor, alcanzando un promedio anual de 95.4%, en lugar del 8% reportado por la librería.

6. Pseudocódigo

Clase EstrategiaFVG:

Parámetros:

```
min_fvg_height = 10 // Altura mínima del FVG para considerar una operación
ema_period = 100 // Periodo de la EMA
risk_reward_ratio = 3 // Relación riesgo/recompensa
sl_offset = 0.001 // Desviación para el Stop Loss
```

Método inicializar():

```
Calcular FVGS utilizando la función detect_fvg con los datos de precios
Crear un conjunto vacío processed_fvgs para rastrear los FVG procesados
Calcular la EMA con el periodo especificado utilizando los precios de cierre
```

Método siguiente():

```
current_index = Índice de la vela actual
Filtrar FVGS hasta el índice actual
Determinar la tendencia macro: alcista si el mínimo actual es mayor que la EMA
```

```

Para cada FVG en FVGS válidos:
    Obtener tipo, top, bottom, mitigated_index, fvg_index de la FVG

    // Verificar si la altura del FVG cumple con el requisito mínimo
    fvg_height = Valor absoluto de (top - bottom)
    Si fvg_height < min_fvg_height:
        Continuar con la siguiente FVG

    // Asegurar que los índices sean válidos
    Si fvg_index es nulo o ya está en processed_fvgs:
        Continuar con la siguiente FVG

    // Verificar si el FVG ya ha sido llenado
    Si is_fvg_filled(fvg, datos, fvg_index + 2):
        Continuar con la siguiente FVG

    Si la tendencia es alcista:
        Si el tipo de FVG es alcista:
            // Verificar si el precio actual ha tocado o cruzado el top del FVG
            Si el máximo actual es mayor o igual al top del FVG y el bottom del FVG es
mayor que la EMA:
                Establecer entry_price en el top del FVG
                Calcular stop_loss como (bottom del FVG) * (1 - sl_offset)
                Calcular take_profit como entry_price + ((entry_price - stop_loss) *
risk_reward_ratio)

            // Asegurar que el stop_loss y take_profit sean lógicos
            Si stop_loss < entry_price < take_profit:
                Ejecutar una orden de compra con los parámetros definidos

            Añadir fvg_index a processed_fvgs

```

Sino (tendencia bajista):

Si el tipo de FVG es bajista:

```
// Verificar si el precio actual ha tocado o cruzado el bottom del FVG
```

Si el mínimo actual es menor o igual al bottom del FVG y el bottom del FVG es menor que la EMA:

```
Establecer entry_price en el bottom del FVG
Calcular stop_loss como (top del FVG) * (1 + sl_offset)
Calcular take_profit como entry_price - ((stop_loss - entry_price) *
risk_reward_ratio)
```

```
// Asegurar que el stop_loss y take_profit sean lógicos
```

Si $\text{stop_loss} > \text{entry_price} > \text{take_profit}$:

```
Ejecutar una orden de venta con los parámetros definidos
Añadir fvg_index a processed_fvgs
```

BOT MULTI TIMEFRAME ANALYSIS

1. Preparación de los datos

Se utilizó un *dataset* histórico de BTC/USD con velas de 6 horas que incluye las columnas estándar (Open, High, Low, Close, Volume).

- **Capital inicial:** \$2,000,000.
- **Comisiones:** 0.2% por operación.

2. Lógica de la estrategia

Condiciones generales:

- Se calcula la **EMA de 100 periodos**.
- Se identifican los **Order Blocks** (OBs) alcistas y bajistas que no hayan sido mitigados en ambas temporalidades.

Entradas en compra (tendencia alcista):

- El precio mitiga un OB alcista en la temporalidad alta y está por encima de la EMA de 100 periodos.
- En la temporalidad baja, se buscan OBs alineados y se coloca una orden límite en el máximo del OB.
- **Stop loss:** ligeramente por debajo del máximo del OB de la temporalidad baja.
- **Take profit:** ratio riesgo-beneficio de 1 a 3.

Entradas en venta (tendencia bajista):

- El precio mitiga un OB bajista en la temporalidad alta y está por debajo de la EMA de 100 periodos.
- En la temporalidad baja, se buscan OBs alineados y se coloca una orden límite en el mínimo del OB.
- **Stop loss:** ligeramente por encima del máximo del OB de la temporalidad baja.
- **Take profit:** ratio riesgo-beneficio de 1 a 3.

Resultado:



Figura 10: Resultados de la estrategia Multi Timeframe Analysis

Start	2019-01-02 18:00:00
End	2024-01-01 18:00:00
Nº Trades	19
Return [%]	1.31%
Max Drawdown [%]	-0.5%
Win Rate	42%
Worst Trade	-3.4%

Tabla 4: Resultados del backtest de la estrategia Multi Timeframe Analysis

Evaluación de los resultados del backtest

Según las estadísticas generadas por la librería backtesting.py, la estrategia reportó un 1.3% de retorno y una pérdida máxima de -0.5% en un período de 5 años. Sin embargo, estos resultados subestiman el rendimiento real debido a que la librería es imprecisa al cerrar las posiciones: las órdenes de *stop loss* y *take profit* se ejecutan al cierre de la vela, en lugar del momento exacto en que se activan.

Esta imprecisión se refleja en métricas como:

- **Worst Trade**, que reporta una pérdida de -3% en lugar del límite teórico de -1%.
- **Best Trade**, que muestra un beneficio de +7% en lugar del +3% esperado.

Cálculo del retorno real

Para estimar el rendimiento real de la estrategia, excluyendo las comisiones, consideramos los siguientes parámetros:

- **Número total de operaciones:** 19
- **Tasa de acierto (Win Rate):** 42%
- **Relación riesgo-beneficio:** 1:3 (3% en operaciones ganadoras, -1% en perdedoras).

1. Cálculo de operaciones ganadoras y perdedoras:

- Operaciones ganadoras: $19 \times 0.42 = 8$
- Operaciones perdedoras: $19 - 8 = 11$

2. Cálculo del retorno total:

- Ganancias: $8 \times 3\% = 24\%$
- Pérdidas: $11 \times -1\% = -11\%$
- Retorno neto: $24\% - 11\% = 13\%$

3. Retorno **promedio** **anual:** $13\% \div 5 = 2.6\%$ anual

4. Conclusión

Si la estrategia se hubiera ejecutado en vivo, con órdenes colocadas al momento en que se activan y no al cierre de la vela, el retorno estimado habría sido algo mayor, alcanzando un promedio anual de 2.6%, en lugar del 1.3% reportado por la librería.

5. Pseudocódigo

Clase EstrategiaMultiTimeframeOB:

Definir variables:

- datos_HTF: Datos del marco temporal alto (diario)
- datos_LTF: Datos del marco temporal bajo (15 minutos)
- ema_periodo: Periodo para el cálculo de la EMA
- min_altura_ob: Altura mínima de los bloques de órdenes
- ratio_riesgo_recompensa: Relación entre riesgo y recompensa

- sl_offset: Desviación para el stop loss
- ema: Media exponencial calculada sobre datos_HTF.Cierre
- datos_HTF, datos_LTF: Datos alineados por fecha
- observaciones_HTF, observaciones_LTF: Identificación de bloques de órdenes en ambos marcos temporales
- ob_actual_HTF: Bloque de órdenes actual en HTF
- ob_invalidar_HTF: Conjunto de índices de bloques de órdenes invalidados
- mapa_indices_hft_ltf: Mapeo entre índices de HTF y LTF

Método inicializar():

- Cargar datos_LTF desde archivo CSV
- Alinear datos_HTF y datos_LTF por fecha
- Calcular la EMA sobre datos_HTF.Cierre
- Identificar bloques de órdenes en datos_HTF y datos_LTF

Método siguiente():

- Obtener índice actual en HTF
- Determinar la tendencia macro (alcista si Cierre > EMA)
- Filtrar observaciones_HTF válidas hasta el índice actual
- Para cada bloque de orden en observaciones_HTF:
 - Si el bloque no ha sido invalidado:
 - Obtener tipo de bloque, top, bottom, índice de mitigación y índice de retroceso
 - Si el índice actual es el de mitigación:
 - Si la tendencia es alcista y el bloque es alcista:
 - Identificar rango correspondiente en LTF
 - Verificar condiciones de entrada en LTF
 - Ejecutar orden de compra si se cumplen condiciones
 - Si la tendencia es bajista y el bloque es bajista:
 - Identificar rango correspondiente en LTF
 - Verificar condiciones de entrada en LTF
 - Ejecutar orden de venta si se cumplen condiciones

3.3.4. Implementación de las estrategias en Phemex

Al ejecutar cualquiera de los bots, pueden ocurrir tres escenarios diferentes:

1. **Conexión fallida:** El bot no logra conectarse al *exchange* o se presenta un error inesperado, lo que provoca que se active una alerta de error.
2. **Conexión establecida pero sin señales de entrada:** El bot se conecta correctamente al *exchange*, pero no se cumplen las condiciones necesarias para ejecutar la operación, por lo que se queda esperando.

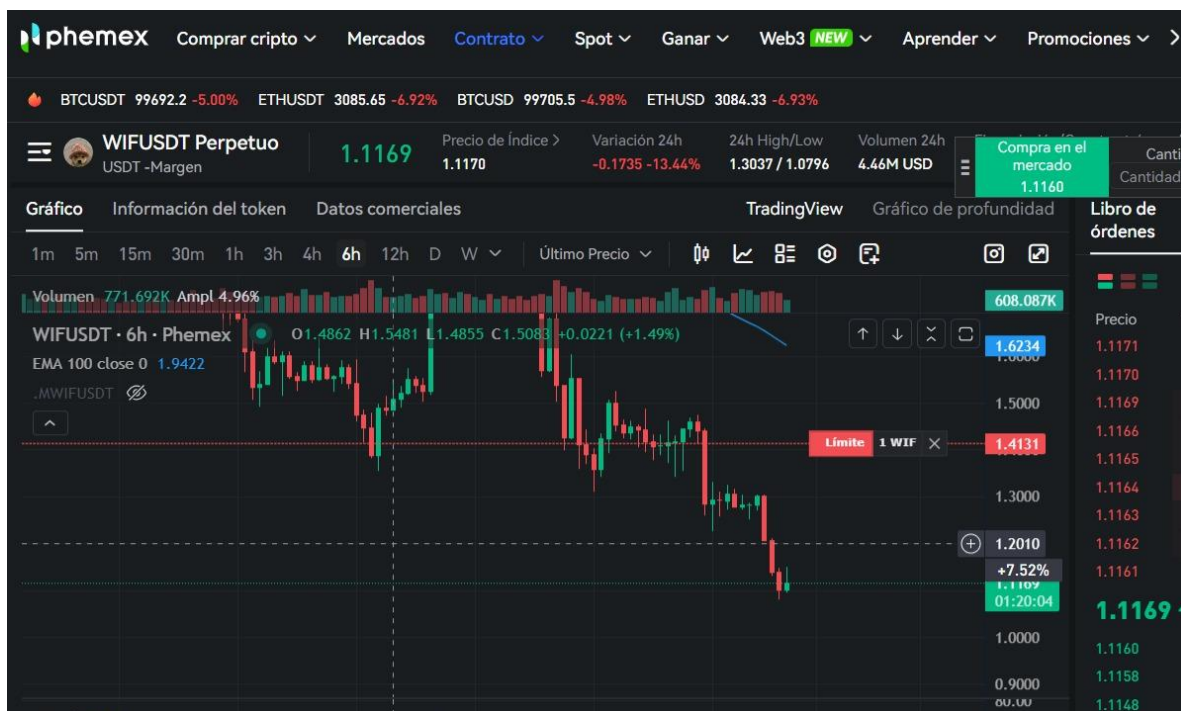
```
Connected to quant (Python 3.10.9)

'''This bot will look at the ema of 150 periods and see if we are bullish or bearish ...
... No valid conditions to place an order, current price: 1.1018, ema: 1.79771401537529, rsi: 17.182997118155626
No valid conditions to place an order, current price: 1.1018, ema: 1.79771401537529, rsi: 17.182997118155626
```

3. **Condiciones de entrada cumplidas:** El bot detecta que se cumplen las condiciones de entrada y procede a realizar una orden de compra o venta conforme a la estrategia.

```
import ccxt ...

No valid conditions to place an order, current ob top:nan, ob type: nan
No valid conditions to place an order, current ob top:nan, ob type: nan
No valid conditions to place an order, current ob top:nan, ob type: nan
No valid conditions to place an order, current ob top:nan, ob type: nan
No valid conditions to place an order, current ob top:nan, ob type: nan
No valid conditions to place an order, current ob top:nan, ob type: nan
No valid conditions to place an order, current ob top:nan, ob type: nan
No valid conditions to place an order, current ob top:nan, ob type: nan
No valid conditions to place an order, current ob top:nan, ob type: nan
OB already mitigated, ob index: 990.0, with top: 1.3308, the mit index top: 1.3251
No valid conditions to place an order, current ob top:nan, ob type: nan
Placing SELL order for bearish OB at 1.4131, current price: 1.1018
Open order exists for WIFUSDT. Skipping new order.
```



BOT DE RSI Y EMA

1. Inicialización del bot

Creamos un archivo llamado **key_file**, que contiene la **key** y **secret** del exchange Phemex, lo que nos permitirá conectarnos a él mediante la librería **ccxt**. A continuación, cargamos los datos en tiempo real en un gráfico de 6 horas del símbolo seleccionado, en este caso **WIFUSDT**, junto con el libro de órdenes que incluye los precios de **ask** y **bid**.

Antes de realizar cualquier operación, el bot verifica que no haya una operación abierta para evitar múltiples operaciones en el mismo punto, ejecutando solo una operación por idea.

2. Lógica de la estrategia

En cada operación, el bot primero comprueba si se cumplen los requisitos necesarios para entrar en la operación:

- Para una **operación de compra**, el precio debe estar por encima de la **EMA de 100 periodos** y el **RSI** debe estar por debajo de los 30 puntos.
- Para una **operación de venta**, el precio debe estar por debajo de la **EMA de 100 periodos** y el **RSI** debe estar por encima de los 70 puntos.

El bot siempre busca un **riesgo-beneficio** de 1:3 y mantiene un **riesgo del 1%** por operación para evitar periodos prolongados y pronunciados de pérdidas. La ejecución es periódica, con intervalos de 15 segundos, durante los cuales se

comprueba si se ha formado una nueva oportunidad para entrar en una operación válida. Si no se detecta ninguna oportunidad, el bot no realizará ninguna acción y esperará .

15 segundos. En caso de que ocurra un error, el bot se detendrá de inmediato.

3. Pseudocódigo:

Función ejecutar_bot():

```
// Obtener datos históricos de precios
```

```
df = obtener_datos(symbol, intervalo='6h', cantidad=1000)
```

```
Si df es None:
```

```
Retornar // Salir si no se pudieron obtener los datos
```

```
// Obtener precios de compra y venta actuales
```

```
ask, bid = obtener_precios()
```

```
Si ask es None o bid es None:
```

```
Retornar // Salir si no se pudieron obtener los precios
```

```
// Calcular EMA y RSI
```

```
ema = calcular_ema(df['Cierre'], periodo_ema)
```

```
df['rsi'] = calcular_rsi(df['Cierre'], periodo_rsi)
```

```
precio_actual = df['Cierre'].último()
```

```
valor_ema = ema.último()
```

```
valor_rsi = df['rsi'].último()
```

```
tendencia_macro_alcista = precio_actual > valor_ema
```

```
// Verificar si hay una posición abierta
```

```
pos_abierta, tamaño, lado, precio_entrada = obtener_posiciones_abiertas()
```

```
en_posicion = tamaño > 0
```

```
// Si hay una posición abierta, verificar condiciones de salida basadas en PnL
```

```
Si en_posicion y precio_entrada no es None:
```

```
pnl_pct = calcular_pnl(precio_entrada, precio_actual, lado)
```

```

// Condiciones de salida (ejemplo: ganancia objetivo 3%, pérdida máxima 1%)
Si pnl_pct >= 3 o pnl_pct <= -1:
    Imprimir("Saliendo de la posición en", precio_actual, "con PnL:", pnl_pct, "%")
    Si lado es 'Compra':
        crear_orden_venta(symbol, tamaño, ask)
    Sino si lado es 'Venta':
        crear_orden_compra(symbol, tamaño, bid)
Sino:
    Imprimir("Manteniendo posición con PnL:", pnl_pct, "%")
Retornar

// Verificar si existen órdenes abiertas para este símbolo antes de colocar nuevas
Si verificar_ordenes_abiertas(symbol):
    Imprimir("Ya existe una orden abierta para", symbol, ". Omitiendo nueva orden.")
Retornar

// Lógica de trading basada en EMA y RSI
// Comprar si tendencia alcista y RSI indica sobreventa
Si tendencia_macro_alcista y valor_rsi < nivel_sobreventa:
    Imprimir("Colocando orden de COMPRA en", precio_actual, 'EMA:', valor_ema,
'RSI:', valor_rsi)
    Intentar:
        crear_orden_compra(symbol, tamaño_posicion, ask)
    Excepto ErrorBase como e:
        Imprimir("Error al colocar orden de COMPRA:", e)

// Vender si tendencia bajista y RSI indica sobrecompra
Sino si no tendencia_macro_alcista y valor_rsi > nivel_sobrecompra:
    Imprimir("Colocando orden de VENTA en", precio_actual, 'EMA:', valor_ema, 'RSI:',
valor_rsi)
    Intentar:
        crear_orden_venta(symbol, tamaño_posicion, bid)
    Excepto ErrorBase como e:

```

```
Imprimir("Error al colocar orden de VENTA:", e)
```

Sino:

```
Imprimir('No se cumplen las condiciones para colocar una orden, precio actual:',  
precio_actual, 'EMA:', valor_ema, 'RSI:', valor_rsi)
```

BOT DE ORDER BLOCK

1. Inicialización del bot

Creamos un archivo llamado **key_file**, que contendrá la **key** y **secret** del *exchange* Phemex. Estos datos nos permitirán conectarnos al *exchange* mediante la librería **ccxt**. A continuación, cargamos los datos en tiempo real de un gráfico de 6 horas del símbolo elegido, en este caso **WIFUSDT**, junto con el libro de órdenes que muestra los precios **ask** y **bid**. Antes de realizar cualquier operación, el bot verifica que no haya ninguna operación abierta para evitar múltiples operaciones en el mismo nivel de precios. De esta forma, solo ejecutará una operación por idea de *trading*.

2. Lógica de la estrategia

En cada operación, el bot realiza una verificación para asegurarse de que se cumplen los requisitos necesarios para ejecutar una operación:

- **Operaciones de compra:** El bot espera que el precio esté por encima de la **EMA de 100 periodos** y que haya un **Order Block alcista válido** (es decir, que no esté mitigado). Si se cumplen estas condiciones, se coloca una orden límite en el máximo del **Order Block**.
- **Operaciones de venta:** El bot espera que el precio esté por debajo de la **EMA de 100 periodos** y que haya un **Order Block bajista válido** (es decir, que no esté mitigado). Si se cumplen estas condiciones, se coloca una orden límite en el mínimo del **Order Block**.

El bot siempre busca mantener una relación **riesgo-beneficio de 1:3** y establece un **riesgo del 1% por operación** para evitar periodos largos de pérdidas pronunciadas. La ejecución del bot es periódica, con un intervalo de 15 segundos, durante los cuales verifica si se ha formado una nueva oportunidad de operación válida. Si no se detecta ninguna oportunidad, el bot no realiza ninguna acción y espera otros 15 segundos. Si ocurre un error en cualquier momento, el bot se detendrá de inmediato.

3. Pseudocódigo:

Función `bot_de_trading()`:

```
// Obtener datos históricos y precios actuales  
df = obtener_datos(symbol, intervalo='6h', cantidad=1000)  
ask, bid = obtener_precios()  
obs = identificar_order_blocks(df)  
precio_actual = df['Cierre'].último()
```

```

ema = calcular_ema(df['Cierre'], periodo_ema)
valor_ema = ema.último()

// Determinar la tendencia principal
tendencia_principal_alcista = precio_actual > valor_ema

// Verificar si hay una posición abierta
posicion_abierta, tamaño, lado, precio_entrada = verificar_posiciones()
en_posicion = tamaño > 0

Si en_posicion:
    // Calcular ganancias o pérdidas
    pnl_pct = calcular_ganancia_perdida(precio_entrada, precio_actual, lado)
    Imprimir("Ganancia/Pérdida actual: ", pnl_pct, "%")

    Si pnl_pct >= ganancia_objetivo_pct o pnl_pct <= perdida_maxima_pct:
        Imprimir("Cerrando posición en ", precio_actual, " con ganancia/pérdida: ",
pnl_pct, "%")
        Si la dirección es 'Compra':
            crear_orden_venta(symbol, tamaño, ask)
        Sino si la dirección es 'Venta':
            crear_orden_compra(symbol, tamaño, bid)
        Sino:
            Imprimir("Manteniendo posición con ganancia/pérdida: ", pnl_pct, "%")
        Regresar

// Verificar si hay órdenes abiertas para el símbolo
Si verificar_ordenes_abiertas(symbol):
    Imprimir("Ya existe una orden abierta para ", symbol, ". Omitiendo nueva orden.")
    Regresar

// Procesar los Order Blocks identificados
Para cada ob en orden_inverso(obs):

```

```

ob_tipo, ob_top, ob_bottom, ob_indice_mitigacion, ob_indice = ob
Si ob_indice_mitigacion no es nulo:
    mit_index_top = df['Alto'][ob_indice_mitigacion]
    Imprimir('OB ya mitigado, índice OB: ', ob_indice, ', top: ', ob_top, ', top índice
mitigación: ', mit_index_top)
    Continuar

// Colocar una orden de COMPRA para OB alcista
Si ob_tipo es 1 y precio_actual > ob_top y tendencia_principal_alcista y ob_bottom
> valor_ema:
    Imprimir('Colocando orden de COMPRA para OB alcista en ', ob_top, ', precio
actual: ', precio_actual, ', EMA: ', valor_ema)
    crear_orden_compra(symbol, tamaño_posicion, ob_top)
    Romper

// Colocar una orden de VENTA para OB bajista
Sino si ob_tipo es -1 y precio_actual < ob_bottom y no tendencia_principal_alcista
y ob_top < valor_ema:
    Imprimir('Colocando orden de VENTA para OB bajista en ', ob_bottom, ', precio
actual: ', precio_actual)
    crear_orden_venta(symbol, tamaño_posicion, ob_bottom)
    Romper
Sino:
    Imprimir('No se cumplen las condiciones para colocar una orden, top OB: ',
ob_top, ', tipo OB: ', ob_tipo)

```

BOT DE FAIR VALUE GAP

1. Inicialización del bot

Creamos un archivo llamado **key_file**, que contiene la **key** y **secret** del Exchange Phemex, lo cual nos permitirá conectarnos a él mediante la librería **ccxt**. A continuación, cargamos los datos en tiempo real en un gráfico de 6 horas para el símbolo elegido (en este caso, **WIFUSDT**), junto con el libro de órdenes que muestra los precios **ask** y **bid**.

Antes de realizar cualquier operación, el bot verifica que no haya operaciones abiertas, evitando así colocar múltiples órdenes en el mismo nivel de precios. Esto asegura que el bot ejecute solo una operación por idea de trading.

2. Lógica de la estrategia

Para cada operación, el bot verifica que se cumplan los siguientes requisitos antes de entrar en el mercado:

- **Operaciones de compra:** El bot espera que el precio esté por encima de la **EMA de 100 periodos** y que haya un **Fair Value Gap alcista válido** (es decir, que no esté mitigado). Si ambas condiciones se cumplen, el bot coloca una orden límite en el máximo del **Fair Value Gap**.
- **Operaciones de venta:** El bot espera que el precio esté por debajo de la **EMA de 100 periodos** y que haya un **Fair Value Gap bajista válido** (es decir, que no esté mitigado). Si ambas condiciones se cumplen, el bot coloca una orden límite en el mínimo del **Fair Value Gap**.

El bot siempre busca mantener una relación **riesgo-beneficio de 1:3** y establece un **riesgo del 1% por operación** para evitar sufrir pérdidas prolongadas y grandes. La ejecución del bot ocurre cada 15 segundos, durante los cuales verifica si se ha formado una nueva oportunidad de operación válida. Si no se detecta ninguna oportunidad, el bot no realiza ninguna acción y espera otros 15 segundos. Si ocurre un error en cualquier momento, el bot se detiene inmediatamente.

3. Pseudocódigo:

Función ejecutar_bot():

```
// Obtener datos históricos y precios actuales
datos = obtener_datos(symbol, intervalo, cantidad)
precios = datos['Cierre']
precio_actual = precios[-1]

// Calcular indicadores técnicos
ema = calcular_ema(precios, periodo_ema)
rsi = calcular_rsi(precios, periodo_rsi)
tendencia_alcista = precio_actual > ema[-1]

// Verificar si hay posiciones abiertas
posiciones = obtener_posiciones_abiertas()
en_posicion = posiciones['tamaño'] > 0

Si en_posicion:
    pnl=calcular_pnl(posiciones['precio_entrada'],precio_actual,posiciones['direccion'])
    Imprimir("PnL actual: ", pnl, "%")
```

```

Si pnl >= objetivo_ganancia_pct o pnl <= max_perdida_pct:
    cerrar_posicion(posiciones)
Sino:
    Imprimir("Manteniendo posición con PnL: ", pnl, "%")
Retornar

// Detectar Fair Value Gaps
fvgs = detectar_fvg(datos)

// Analizar cada FVG detectado
Para cada fvg en fvgs:
    Si fvg['tipo'] == 'alcista' y precio_actual > fvg['top'] y tendencia_alcista y rsi[-1] < 30:
        abrir_posicion('Compra', fvg['top'])
        Romper
    Sino Si fvg['tipo'] == 'bajista' y precio_actual < fvg['bottom'] y no tendencia_alcista
    y rsi[-1] > 70:
        abrir_posicion('Venta', fvg['bottom'])
        Romper

// Funciones auxiliares
Función obtener_datos(symbol, intervalo, cantidad):
    // Implementar la obtención de datos históricos
    Retornar datos

Función calcular_ema(precios, periodo):
    // Implementar el cálculo de la EMA
    Retornar ema

Función calcular_rsi(precios, periodo):
    // Implementar el cálculo del RSI
    Retornar rsi

Función obtener_posiciones_abiertas():

```

```
// Implementar la obtención de posiciones abiertas  
Retornar posiciones
```

Función `calcular_pnl(precio_entrada, precio_actual, direccion)`:

```
// Implementar el cálculo del PnL  
Retornar pnl
```

Función `cerrar_posicion(posiciones)`:

```
// Implementar el cierre de la posición  
Imprimir("Cerrando posición en ", precio_actual)
```

Función `detectar_fvg(datos)`:

```
// Implementar la detección de FVGs  
Retornar fvg
```

Función `abrir_posicion(direccion, precio)`:

```
// Implementar la apertura de una posición  
Imprimir("Abriendo posición de ", direccion, " en ", precio)
```

BOT MULTI TIMEFRAME ANALYSIS

1. Inicialización del bot

Se crea un archivo denominado **key_file**, que contiene la **key** y **secret** del Exchange Phemex. Esto permitirá la conexión al Exchange mediante la librería **ccxt**. El bot carga los datos en tiempo real en dos temporalidades: **1 hora** y **15 minutos**, ya que esta estrategia utiliza ambas temporalidades. En este caso, se emplea el símbolo **WIFUSDT** junto con el libro de órdenes que muestra los precios **ask** y **bid**. Antes de realizar una nueva operación, el bot verifica que no haya ninguna operación abierta para evitar ejecutar múltiples operaciones en el mismo nivel de precios. De esta forma, el bot ejecutará solo una operación por idea de trading.

2. Lógica de la estrategia

En cada operación, el bot verifica que se cumplan los siguientes requisitos antes de proceder a la entrada:

- **Operaciones de compra:**

El bot espera a que el precio mitigue el **Order Block** más reciente y válido de la

temporalidad alta. Después, verifica si existe un Order Block válido en la temporalidad inferior dentro de dicha zona. Si se cumple esta condición, el bot entra en la zona más cercana, colocando una **orden límite** en el máximo y el **stop loss** en el mínimo del Order Block.

- **Operaciones de venta:**

El bot espera a que el precio mitigue el **Order Block** más reciente y válido de la temporalidad alta. Después, verifica si existe un Order Block válido en la temporalidad inferior dentro de la zona marcada de la temporalidad alta. Si se cumple esta condición, el bot entra en la zona más cercana, colocando una **orden límite** en el mínimo y el **stop loss** en el máximo del Order Block.

El bot siempre busca una relación **riesgo-beneficio de 1 a 3** y mantiene un **riesgo del 1% por operación** para evitar entrar en periodos largos de pérdidas. La ejecución del bot se realiza cada **15 segundos**, durante los cuales revisa si se ha formado una nueva oportunidad de entrada válida. Si no se detecta ninguna oportunidad, el bot no realiza ninguna acción y espera otros 15 segundos. Si ocurre un error en cualquier momento, el bot se detendrá de inmediato.

3. Pseudocódigo:

Función bot():

```
// Obtener datos históricos

df_HTF = obtener_datos(symbol, '1h', 1000) // Datos del marco de tiempo alto (HTF)
de 1 hora

df_LTF = obtener_datos(symbol, '15m', 1000) // Datos del marco de tiempo bajo (LTF)
de 15 minutos

// Calcular la EMA en HTF para detección de tendencia
ema_HTF = calcular_EMA(df_HTF['Cierre'], periodo_ema)
valor_ema_HTF = obtener_ultimo_valor(ema_HTF)

// Identificar Order Blocks en HTF y LTF
obs_HTF = identificar_OB(df_HTF)
obs_LTF = identificar_OB(df_LTF)

// Obtener el precio actual y determinar la tendencia macro
precio_actual = obtener_precio_actual(df_HTF)
tendencia_macro_alcista = precio_actual > valor_ema_HTF

// Verificar si hay una posición abierta
```

```
pos_abierta, tamaño, lado, precio_entrada = obtener_posiciones_abiertas()
en_posicion = tamaño > 0
```

```
Si en_posicion:
```

```
    // Calcular y mostrar PnL
```

```
    pnl_pct = calcular_pnl(precio_entrada, precio_actual, lado)
```

```
    Mostrar "PnL actual: ", pnl_pct, "%"
```

```
Si pnl_pct >= objetivo_ganancia_pct o pnl_pct <= max_perdida_pct:
```

```
    Mostrar "Cerrando posición en ", precio_actual, " con PnL: ", pnl_pct, "%"
```

```
    Si lado == 'Compra':
```

```
        crear_orden_venta(symbol, tamaño, precio_actual)
```

```
    Sino si lado == 'Venta':
```

```
        crear_orden_compra(symbol, tamaño, precio_actual)
```

```
Sino:
```

```
    Mostrar "Manteniendo posición con PnL: ", pnl_pct, "%"
```

```
Retornar
```

```
// Verificar si existen órdenes abiertas para el símbolo
```

```
Si verificar_ordenes_abiertas(symbol):
```

```
    Mostrar "Ya existe una orden abierta para ", symbol, ". Omitiendo nueva orden."
```

```
Retornar
```

```
// Iterar sobre los OBs en HTF y verificar mitigación
```

```
Para cada ob_HTF en reversa(obs_HTF):
```

```
    Si es_nulo(ob_HTF.OB):
```

```
        Mostrar "No se encontró un OB válido: ", ob_HTF
```

```
        Continuar
```

```
        ob_tipo_HTF, ob_top_HTF, ob_bottom_HTF, ob_indice_mit_HTF = ob_HTF.OB,  
ob_HTF.Top, ob_HTF.Bottom, ob_HTF.MitigatedIndex
```

```
        indice_actual = obtener_indice_actual(df_HTF)
```

```
// Si el OB de HTF está mitigado, verificar confirmación en LTF
```

```

Si indice_actual == ob_indice_mit_HTF:
    Si tendencia_macro_alcista:
        Si ob_tipo_HTF == 1 y precio_actual < ob_top_HTF: // OB alcista
            Mostrar "OB alcista de HTF mitigado en: ", ob_HTF, ", verificando OB en
LTF"

            Para cada ob_LTF en reversa(obs_LTF):
                ob_tipo_LTF, ob_top_LTF, ob_bottom_LTF = ob_LTF.OB, ob_LTF.Top,
ob_LTF.Bottom

                Si ob_tipo_LTF == 1 y precio_actual > ob_top_LTF y ob_bottom_LTF >=
ob_bottom_HTF y ob_top_LTF <= ob_top_HTF:
                    Mostrar "Colocando orden de COMPRA para OB alcista en ",
ob_top_LTF, ", precio actual: ", precio_actual

                    crear_orden_compra(symbol, tamaño_pos, ob_top_LTF)

                    Retornar

            Sino:
                Si ob_tipo_HTF == -1 y precio_actual > ob_bottom_HTF: // OB bajista
                    Mostrar "OB bajista de HTF mitigado en: ", ob_HTF, ", verificando OB en
LTF"

                    Para cada ob_LTF en reversa(obs_LTF):
                        ob_tipo_LTF, ob_top_LTF, ob_bottom_LTF = ob_LTF.OB, ob_LTF.Top,
ob_LTF.Bottom

                        Si ob_tipo_LTF == -1 y precio_actual < ob_bottom_LTF y ob_bottom_LTF
>= ob_bottom_HTF y ob_top_LTF <= ob_top_HTF:
                            Mostrar "Colocando orden de VENTA para OB bajista en ",
ob_bottom_LTF, ", precio actual: ", precio_actual

                            crear_orden_venta(symbol, tamaño_pos, ob_bottom_LTF)

                            Retornar

                        Sino:
                            Mostrar "OB de HTF no ha sido mitigado: ", ob_HTF

                            Retornar

```

3.3.5. Implementación de las estrategias en Hyperliquid

Al ejecutar cualquiera de los bots, pueden ocurrir tres escenarios diferentes:

1. **Conexión fallida:** El bot no logra conectarse al *exchange* o se presenta un error inesperado, lo que provoca que se active una alerta de error.
2. **Conexión establecida pero sin señales de entrada:** El bot se conecta correctamente al *exchange*, pero no se cumplen las condiciones necesarias para ejecutar la operación, por lo que se queda esperando.

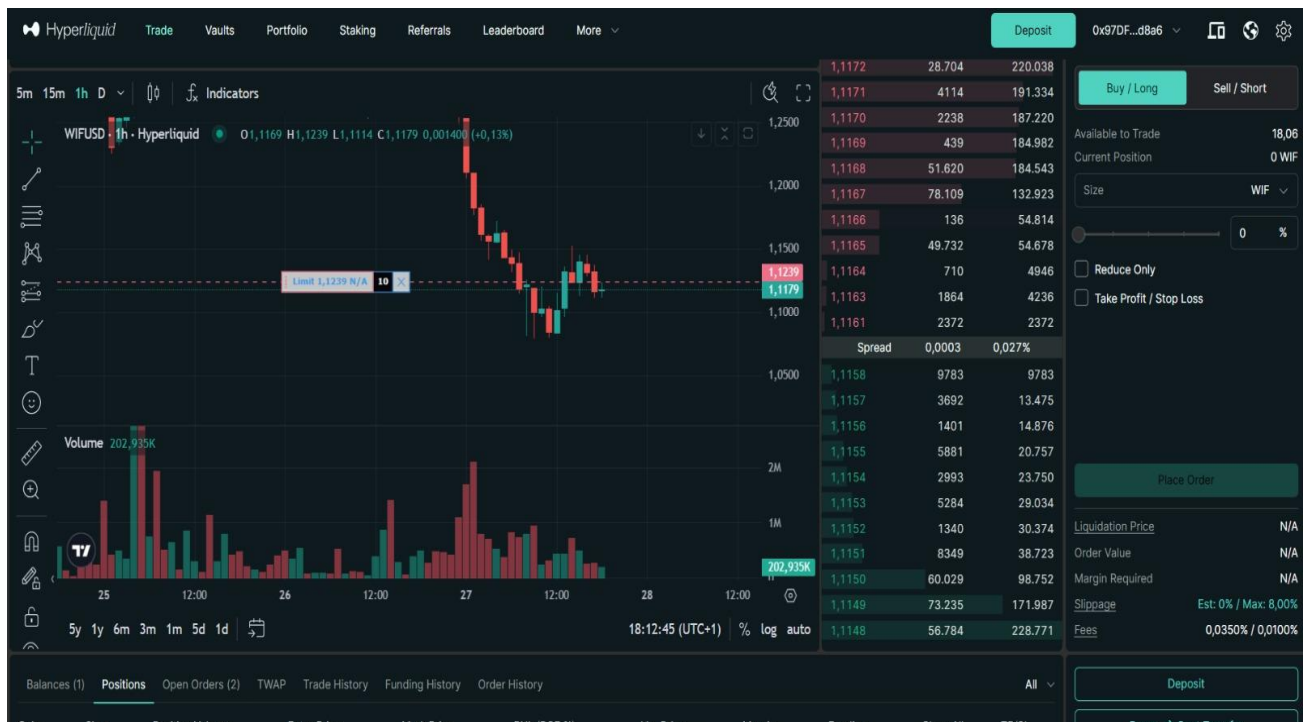
```
Connected to quant (Python 3.10.9)

Q '''This bot will look at the ema of 150 periods and see if we are bullish or bearish ...
... No valid conditions to place an order, current price: 1.1018, ema: 1.79771401537529, rsi: 17.182997118155626
  No valid conditions to place an order, current price: 1.1018, ema: 1.79771401537529, rsi: 17.182997118155626
```

3. **Condiciones de entrada cumplidas:** El bot detecta que se cumplen las condiciones de entrada y procede a realizar una orden de compra o venta conforme a la estrategia.

```
Connected to quant (Python 3.10.9)

Q '''This bot will buy when price touches the top of a bullish FVG and sell if it touches the bottom of
. Placing SELL order for bearish FVG at 1.1239, current price: 1.1179
WIF this is the price 0 decimals(s)
coin: WIF, type: <class 'str'>
is_buy: False, type: <class 'bool'>
sz: 10, type: <class 'int'>
limit_px: 1.1239, type: <class 'float'>
reduce_only: False, type: <class 'bool'>
placing limit order for WIF 10 @ 1.1239
limit SELL order placed thanks moon, resting: {'resting': {'oid': 66999213366}}
Placing SELL order for bearish FVG at 1.1239, current price: 1.1179
WIF this is the price 0 decimals(s)
coin: WIF, type: <class 'str'>
is_buy: False, type: <class 'bool'>
sz: 10, type: <class 'int'>
limit_px: 1.1239, type: <class 'float'>
reduce_only: False, type: <class 'bool'>
placing limit order for WIF 10 @ 1.1239
limit SELL order placed thanks moon, resting: {'resting': {'oid': 66999278215}}
```



BOT DE EMA Y RSI

1. Inicialización del bot

Creamos un archivo llamado **dont_share**, que contendrá la **key** del Exchange **Hyperliquid**. Esto nos permitirá conectarnos al *exchange* utilizando **eth_account**, que es una clase de Ethereum que permite crear, firmar y gestionar claves privadas, públicas y direcciones asociadas. Esto será útil para obtener las posiciones abiertas en nuestra cuenta del *exchange*.

El bot carga los datos en tiempo real en un gráfico de **1 hora** del símbolo elegido (en este caso, **WIF**), junto con el libro de órdenes, que incluye los precios de **ask** y **bid**. Antes de realizar una nueva operación, el bot verifica que no haya ninguna operación abierta para evitar ejecutar múltiples operaciones en el mismo lugar. Solo se ejecutará una operación por idea de *trading*.

2. Lógica de la estrategia

En cada operación, el bot verifica si se cumplen los requisitos necesarios para entrar:

- **Operaciones de compra:**

El bot espera a que el precio esté por encima de la **EMA de 100 periodos** y que el **RSI** esté por debajo de los **30 puntos**.

- **Operaciones de venta:**

El bot espera a que el precio esté por debajo de la **EMA de 100 periodos** y que el **RSI** esté por encima de los **70 puntos**.

El bot siempre busca un **riesgo-beneficio de 1:3** y mantiene un **riesgo del 1% por operación** para evitar caer en periodos largos y pronunciados de pérdidas. La ejecución del bot se realiza cada **15 segundos**, durante los cuales revisa si se ha formado una nueva oportunidad para entrar en una operación válida. Si no hay ninguna oportunidad, el bot no realiza ninguna acción y espera otros 15 segundos. Si ocurre un error en cualquier momento, el bot se detendrá de inmediato.

3. Pseudocódigo:

Función ejecutar_bot():

```
// Obtener datos de mercado y precios actuales
datos = obtener_datos(symbol, intervalo='1h', cantidad=1)
ask, bid = obtener_precios(symbol)

// Calcular EMA y RSI
ema = calcular_ema(datos['cierre'], periodo_ema)
rsi = calcular_rsi(datos['cierre'], periodo_rsi)

// Obtener valores actuales
precio_actual = obtener_precio_actual(datos)
valor_ema = ema[-1]
valor_rsi = rsi[-1]

// Determinar tendencia general
tendencia_bullish = precio_actual > valor_ema

// Verificar si hay una posición abierta
posicion_abierta, tamaño, direccion, precio_entrada = obtener_posicion_abierta()
en_posicion = tamaño > 0

Si en_posicion:
    // Calcular y mostrar PnL
    pnl_pct = calcular_pnl(precio_entrada, precio_actual, direccion)
```

```

mostrar("PnL actual: " + pnl_pct + "%")

Si pnl_pct >= objetivo_ganancia_o pnl_pct <= max_perdida:
    // Cerrar posición
    mostrar("Cerrando posición en " + precio_actual + " con PnL: " + pnl_pct + "%")
    ejecutar_orden(symbol, direccion == 'Venta', tamaño.abs(), ask_si_compra =
(direccion == 'Compra'), es_limite = Verdadero)
Sino:
    // Mantener posición
    mostrar("Manteniendo posición con PnL: " + pnl_pct + "%")
Retornar

// Verificar si ya existen órdenes abiertas para este símbolo
Si verificar_ordenes_abiertas(symbol):
    mostrar("Ya existe una orden abierta para " + symbol + ". Omitiendo nueva orden.")
Retornar

// Lógica de trading basada en EMA y RSI
Si tendencia_bullish Y valor_rsi < nivel_sobrevendido:
    // Señal de compra
    mostrar("Colocando orden de COMPRA en " + precio_actual + ", EMA: " +
valor_ema + ", RSI: " + valor_rsi)
    ejecutar_orden(symbol, es_compra = Verdadero, tamaño_posicion =
tamaño_posicion, precio = ask, es_limite = Falso)
Sino Si NO tendencia_bullish Y valor_rsi > nivel_sobrecomprado:
    // Señal de venta
    mostrar("Colocando orden de VENTA en " + precio_actual + ", EMA: " + valor_ema
+ ", RSI: " + valor_rsi)
    ejecutar_orden(symbol, es_compra = Falso, tamaño_posicion = tamaño_posicion,
precio = bid, es_limite = Falso)
Sino:
    // No hay condiciones válidas para operar
    mostrar("No hay condiciones válidas para colocar una orden, precio actual: " +
precio_actual + ", EMA: " + valor_ema + ", RSI: " + valor_rsi)

```

BOT DE ORDER BLOCK

1. Inicialización del bot

Creamos un archivo llamado **dont_share** que contendrá la **key** del exchange **Hyperliquid**, lo que nos permitirá conectarnos a él mediante el uso de **eth_account**, una clase de Ethereum que permite crear, firmar y gestionar claves privadas, públicas y direcciones asociadas. Esto será útil para obtener las posiciones abiertas en nuestra cuenta del Exchange.

Cargamos los datos en tiempo real en un gráfico de **1 hora** del símbolo elegido (en este caso, **WIF**), junto con el libro de órdenes, que contiene los precios de **ask** y **bid**. Antes de realizar una nueva operación, el bot comprueba que no esté en ninguna operación abierta, evitando ejecutar múltiples operaciones en el mismo lugar. Solo se realizará una operación por idea.

2. Lógica de la estrategia

En cada operación, el bot comprueba que se cumplan los requisitos necesarios para entrar:

- **Operación de compra:**

El bot espera a que el precio esté por encima de la **EMA de 100 periodos** y que haya un **Order Block alcista válido** (que no esté mitigado). Si se cumple, se coloca una orden límite en el máximo del **Order Block**.

- **Operación de venta:**

El bot espera a que el precio esté por debajo de la **EMA de 100 periodos** y que haya un **Order Block bajista válido** (que no esté mitigado). Si se cumple, se coloca una orden límite en el mínimo del **Order Block**.

El bot siempre busca un **riesgo-beneficio de 1 a 3** y mantiene un **riesgo del 1% por operación** para evitar periodos largos y pronunciados de pérdidas. La ejecución del bot se realiza cada **15 segundos**, durante los cuales comprueba si se ha formado una nueva oportunidad de entrada. Si no se encuentra ninguna oportunidad, no realiza ninguna acción y vuelve a esperar otros 15 segundos. Si ocurre algún error, el bot se detendrá de inmediato.

3. Pseudocódigo:

Función bot():

```
// Recuperar datos históricos de precios
df = obtener_datos(symbol, '1h', 1000)

// Obtener precios actuales de compra y venta
ask, bid, _ = obtener_ask_bid(symbol)
```

```

// Identificar Order Blocks en los datos
obs = identificar_OB(df)

// Obtener el precio de cierre más reciente
precio_actual = convertir_a_float(df['close'].último())

// Calcular el valor de la EMA
ema = calcular_EMA(df['close'], periodo_ema)
valor_ema = convertir_a_float(ema.último())

// Determinar la tendencia macroeconómica
tendencia_macro_bullish = precio_actual > valor_ema

// Verificar si hay una posición abierta
pos_abierta, tamaño, lado, precio_entrada = obtener_posiciones_abiertas()
en_posicion = tamaño > 0

// Si hay una posición abierta, verificar PnL y condiciones de salida
Si en_posicion Y precio_entrada NO ES NULO:
    pnl_pct = calcular_PnL(precio_entrada, precio_actual, lado)
    Imprimir("PnL actual: " + pnl_pct + "%")

    Si pnl_pct >= objetivo_ganancia_pct O pnl_pct <= maximo_perdida_pct:
        Imprimir("Saliendo de la posición en " + precio_actual + " con PnL: " + pnl_pct +
"%")
        colocar_orden(symbol, lado == 'Sell', valor_absoluto(tamaño), ask SI lado ==
'Buy' SINO bid, TRUE, cuenta)
    SINO:
        Imprimir("Manteniendo posición con PnL: " + pnl_pct + "%")
    Regresar

// Verificar si ya existen órdenes abiertas para este símbolo antes de colocar nuevas

```

```

Si verificar_ordenes_abiertas(symbol):
    Imprimir("Ya existe una orden abierta para " + symbol + ". Omitiendo nueva orden.")
    Regresar

// Iterar sobre los Order Blocks identificados en orden inverso
Para cada ob en reversa(lista(obs)):
    ob_tipo, ob_top, ob_bottom, ob_indice_mitigacion, ob_indice = ob.OB, ob.Top,
ob.Bottom, ob.MitigatedIndex, ob.OBIndex

    Imprimir("Tipo OB: " + ob.OB + ", OB Top: " + ob.Top + ", OB Bottom: " + ob.Bottom
+ ", Precio Actual: " + precio_actual + ", valor_ema: " + valor_ema)

    Si ob_indice_mitigacion NO ES NULO Y NO es_NaN(ob_indice_mitigacion):
        mit_index_top = df['high'].en_posicion(int(ob_indice_mitigacion))

        Imprimir("OB ya mitigado, índice OB: ' + ob_indice + ', con top: ' + ob_top + ',
índice mitigación top: ' + mit_index_top)

        Continuar

    // Colocar una orden de COMPRA para el OB alcista más reciente

    Si ob_tipo == 1 Y precio_actual > ob_top Y tendencia_macro_bullish Y ob_bottom
> valor_ema:

        Imprimir("Colocando orden de COMPRA para OB alcista en ' + ob_top + ', precio
actual: ' + precio_actual + ', EMA: ' + valor_ema)

        colocar_orden(symbol, TRUE, tamaño_posicion, ob_top, FALSE, cuenta)

        Romper

    // Colocar una orden de VENTA para el OB bajista más reciente

    Sino si ob_tipo == -1 Y precio_actual < ob_bottom Y NO tendencia_macro_bullish
Y ob_top < valor_ema:

        Imprimir("Colocando orden de VENTA para OB bajista en ' + ob_bottom + ', precio
actual: ' + precio_actual)

        colocar_orden(symbol, FALSE, tamaño_posicion, ob_bottom, FALSE, cuenta)

        Romper

SINO:

    Imprimir('No se cumplieron las condiciones para colocar una orden')

```

BOT DE FAIR VALUE GAP

1. Inicialización del bot

Creamos un archivo llamado **dont_share** que contendrá la **key** del exchange **Hyperliquid**, lo que nos permitirá conectarnos a él mediante el uso de **eth_account**, una clase de Ethereum que permite crear, firmar y gestionar claves privadas, públicas y direcciones asociadas. Esto será útil para obtener las posiciones abiertas en nuestra cuenta del *exchange*.

Cargamos los datos en tiempo real en un gráfico de **1 hora** del símbolo elegido (en este caso **WIF**), junto con el libro de órdenes que muestra los precios de **ask** y **bid**. Antes de realizar una nueva operación, el bot comprueba que no esté en ninguna operación abierta, evitando así realizar múltiples operaciones en el mismo lugar. Solo se ejecutará una operación por idea.

2. Lógica de la estrategia

En cada operación, el bot verifica que se cumplan los requisitos necesarios para entrar:

- **Operación de compra:**

El bot espera a que el precio esté por encima de la **EMA de 100 periodos** y que haya un **Fair Value Gap alcista válido** (que no esté mitigado). Si se cumplen estas condiciones, se colocará una orden límite en el máximo del **Fair Value Gap**.

- **Operación de venta:**

El bot espera a que el precio esté por debajo de la **EMA de 100 periodos** y que haya un **Fair Value Gap bajista válido** (que no esté mitigado). Si se cumplen estas condiciones, se colocará una orden límite en el mínimo del **Fair Value Gap**.

El bot siempre busca un **riesgo-beneficio de 1:3** y mantiene un **riesgo del 1% por operación** para evitar largos periodos de pérdidas pronunciadas. La ejecución se realiza cada **15 segundos**, durante los cuales el bot verifica si se ha formado una nueva oportunidad para entrar en una operación válida. Si no se encuentra ninguna oportunidad, no se ejecutará ninguna acción y se volverá a esperar otros 15 segundos.

Si ocurre algún error, el bot se detendrá inmediatamente.

3.Pseudocódigo

Función bot():

```
// Obtener datos históricos de precios  
df = obtener_datos(symbol, '1h', 1000)
```

```
// Obtener precios actuales de compra y venta
```

```

ask, bid, _ = obtener_precios(symbol)

// Detectar Fair Value Gaps en los datos
fvgs = detectar_fvg(df)

// Obtener el precio de cierre más reciente
precio_actual = convertir_a_float(df["close"].último())

// Calcular la media móvil exponencial (EMA)
ema = calcular_ema(df["close"], periodo_ema)
valor_ema = convertir_a_float(ema.último())

// Determinar la tendencia macroeconómica
tendencia_macroeconomica_alcista = precio_actual > valor_ema

// Verificar si hay una posición abierta
posición_abierta, tamaño, lado, precio_entrada = obtener_posiciones_abiertas()
en_posición = tamaño > 0

// Si hay una posición abierta, verificar ganancias y condiciones de salida
Si en_posición Y precio_entrada NO es NULO:
    pnl_pct = calcular_pnl(precio_entrada, precio_actual, lado)
    Imprimir("Ganancia/Pérdida actual: " + pnl_pct + "%")

    Si pnl_pct >= porcentaje_ganancia_objetivo O pnl_pct <=
    porcentaje_pérdida_maxima:
        Imprimir("Cerrando posición en " + precio_actual + " con ganancia/pérdida: " +
        pnl_pct + "%")
        colocar_orden(symbol, lado == 'Venta', valor_absoluto(tamaño), ask SI lado ==
        'Compra' SINO bid, True, cuenta)
    SINO:
        Imprimir("Manteniendo posición con ganancia/pérdida: " + pnl_pct + "%")
    Regresar

```

```

// Verificar si existen órdenes abiertas para este símbolo antes de colocar nuevas
Si verificar_órdenes_abiertas(symbol):
    Imprimir("Ya existe una orden abierta para " + symbol + ". Omitiendo nueva orden.")
    Regresar

// Procesar las FVGs en orden inverso para priorizar la más reciente
Para cada fvg en orden_invertido(fvgs):
    fvg_indice, fvg_tipo, fvg_inferior, fvg_superior, fvg_indice_mitigacion = fvg

    // Convertir los niveles de FVG a números flotantes
    fvg_inferior = convertir_a_float(fvg_inferior)
    fvg_superior = convertir_a_float(fvg_superior)

    // Omitir si la FVG ya ha sido cubierta o mitigada
    Si es_fvg_cubierta(fvg, df):
        Continuar

    // Colocar una orden de COMPRA para la FVG alcista más reciente
    Si fvg_tipo ES 'alcista' Y precio_actual > fvg_superior Y
    tendencia_macroeconomica_alcista Y precio_actual > valor_ema:
        Imprimir('Colocando orden de COMPRA para FVG alcista en ' + fvg_superior +
        ', precio actual:' + precio_actual + ',EMA:' + valor_ema + ',
        tendencia_macroeconomica_alcista: ' + tendencia_macroeconomica_alcista)
        colocar_orden(symbol, True, tamaño_posición, fvg_superior, False, cuenta)
        Romper // Salir del bucle después de colocar la orden

    // Colocar una orden de VENTA para la FVG bajista más reciente
    SINO SI fvg_tipo ES 'bajista' Y precio_actual < fvg_inferior:
        Si NO tendencia_macroeconomica_alcista:
            Si precio_actual < valor_ema:
                Imprimir('Colocando orden de VENTA para FVG bajista en ' + fvg_inferior +
                ', precio actual: ' + precio_actual)
                colocar_orden(symbol, False, tamaño_posición, fvg_inferior, False, cuenta)

```

Romper // Salir del bucle después de colocar la orden

SINO:

Imprimir('No se cumplen las condiciones para colocar una orden para el tipo de FVG: ' + fvg_tipo + ', precio actual: ' + precio_actual + ', FVG superior: ' + fvg_superior + ', FVG inferior: ' + fvg_inferior)

BOT MULTI TIMEFRAME ANALYSIS

Creemos un archivo llamado **dont_share** que contendrá la **key** del *exchange Hyperliquid*, lo que nos permitirá conectarnos a él mediante el uso de **eth_account**, una clase de Ethereum que permite crear, firmar y gestionar claves privadas, públicas y direcciones asociadas. Esto nos será útil para obtener las posiciones abiertas en nuestra cuenta del Exchange.

Cargamos los datos en tiempo real en el gráfico de **1 hora y 15 minutos** del símbolo elegido, ya que en esta estrategia se emplean dos temporalidades. En este caso, el símbolo es **WIF**, junto con el libro de órdenes que muestra los precios de **ask** y **bid**. Antes de realizar una nueva operación, el bot comprueba que no esté en ninguna operación abierta, para evitar realizar múltiples operaciones en el mismo lugar. Solo se ejecutará una operación por idea.

Lógica de la estrategia

En cada operación, el bot primero verifica que se cumplan los requisitos necesarios para entrar:

- **Operación de compra**

El bot espera a que el precio mitigue el **Order Block** válido más reciente de la temporalidad alta. Finalmente, verifica si existe algún **Order Block** válido en la temporalidad inferior que esté dentro del **Order Block** de la temporalidad alta y entra en el más cercano, colocando una orden límite en el máximo y el **stop loss** en el mínimo del **Order Block**.

- **Operación de venta:**

El bot espera a que el precio mitigue el **Order Block** válido más reciente de la temporalidad alta. Finalmente, verifica si existe algún **Order Block** válido en la temporalidad inferior que esté dentro del **Order Block** de la temporalidad alta y entra en el más cercano, colocando una orden límite en el mínimo y el **stop loss** en el máximo del **Order Block**.

El bot siempre busca un **riesgo-beneficio de 1 a 3** y mantiene un **riesgo del 1% por operación** para evitar caer en periodos largos y pronunciados de pérdidas. Realiza una ejecución periódica cada **15 segundos**, en la que comprueba si se ha formado una nueva oportunidad para entrar en una operación válida. En caso de que no haya ninguna oportunidad, no realiza ninguna acción y vuelve a esperar otros 15 segundos.

Si ocurre un error, el bot se detendría de inmediato.

3.Pseudocódigo

Función bot():

```
// Obtener datos históricos de precios
df = obtener_datos(symbol, '1h', 1000)

// Obtener precios actuales de compra y venta
ask, bid, _ = obtener_precios(symbol)

// Detectar Fair Value Gaps en los datos
fvgs = detectar_fvg(df)

// Obtener el precio de cierre más reciente
precio_actual = convertir_a_float(df['close'].último())

// Calcular la media móvil exponencial (EMA)
ema = calcular_ema(df['close'], periodo_ema)
valor_ema = convertir_a_float(ema.último())

// Determinar la tendencia macroeconómica
tendencia_macroeconomica_alcista = precio_actual > valor_ema

// Verificar si hay una posición abierta
posición_abierta, tamaño, lado, precio_entrada = obtener_posiciones_abiertas()
en_posición = tamaño > 0

// Si hay una posición abierta, verificar ganancias y condiciones de salida
Si en_posición Y precio_entrada NO es NULO:
    pnl_pct = calcular_pnl(precio_entrada, precio_actual, lado)
    Imprimir("Ganancia/Pérdida actual: " + pnl_pct + "%")

    Si pnl_pct >= porcentaje_ganancia_objetivo O pnl_pct <=
    porcentaje_pérdida_maxima:
```

```
    Imprimir("Cerrando posición en " + precio_actual + " con ganancia/pérdida: " +
pnl_pct + "%")
```

```
    colocar_orden(symbol, lado == 'Venta', valor_absoluto(tamaño), ask SI lado ==
'Compra' SINO bid, True, cuenta)
```

```
    SINO:
```

```
        Imprimir("Manteniendo posición con ganancia/pérdida: " + pnl_pct + "%")
```

```
    Regresar
```

```
// Verificar si existen órdenes abiertas para este símbolo antes de colocar nuevas
```

```
Si verificar_órdenes_abiertas(symbol):
```

```
    Imprimir("Ya existe una orden abierta para " + symbol + ". Omitiendo nueva orden.")
```

```
    Regresar
```

```
// Procesar las FVGs en orden inverso para priorizar la más reciente
```

```
Para cada fvg en orden_invertido(fvgs):
```

```
    fvg_indice, fvg_tipo, fvg_inferior, fvg_superior, fvg_indice_mitigacion = fvg
```

```
// Convertir los niveles de FVG a números flotantes
```

```
    fvg_inferior = convertir_a_float(fvg_inferior)
```

```
    fvg_superior = convertir_a_float(fvg_superior)
```

```
// Omitir si la FVG ya ha sido cubierta o mitigada
```

```
Si es_fvg_cubierta(fvg, df):
```

```
    Continuar
```

```
// Colocar una orden de COMPRA para la FVG alcista más reciente
```

```
Si fvg_tipo ES 'alcista' Y precio_actual > fvg_superior Y
tendencia_macroeconomica_alcista Y precio_actual > valor_ema:
```

```
    Imprimir('Colocando orden de COMPRA para FVG alcista en ' + fvg_superior +
'; precio actual: ' + precio_actual + ', EMA: ' + valor_ema + ',
tendencia_macroeconomica_alcista: ' + tendencia_macroeconomica_alcista)
```

```
    colocar_orden(symbol, True, tamaño_posición, fvg_superior, False, cuenta)
```

```
    Romper // Salir del bucle después de colocar la orden
```

```

// Colocar una orden de VENTA para la FVG bajista más reciente
SINO SI fvg_tipo ES 'bajista' Y precio_actual < fvg_inferior:
    Si NO tendencia_macroeconomica_alcista:
        Si precio_actual < valor_ema:
            Imprimir('Colocando orden de VENTA para FVG bajista en ' + fvg_inferior +
', precio actual: ' + precio_actual)
            colocar_orden(symbol, False, tamaño_posición, fvg_inferior, False, cuenta)
            Romper // Salir del bucle después de colocar la orden

SINO:
    Imprimir('No se cumplen las condiciones para colocar una orden para el tipo de
FVG: ' + fvg_tipo + ', precio actual: ' + precio_actual + ', FVG superior: ' + fvg_superior +
', FVG inferior: ' + fvg_inferior)

```

3.4. Análisis de Resultados

3.4.1. Evaluación de la Efectividad de Cada Bot

BOT DE EMA Y RSI

Este bot alcanzó un rendimiento del **193%** durante los 5 años de pruebas, posicionándose como el **segundo bot con mejor rentabilidad**. Este resultado es notable, ya que supera ampliamente el retorno anual promedio del **SP500**, que históricamente ha sido de alrededor del **10%**. En comparación, el bot logró un retorno anual promedio del **38%**, sin contar las comisiones.

Aunque las comisiones podrían reducir este beneficio, el impacto no sería significativo, ya que el número de operaciones realizadas (**99**) es relativamente bajo. Esto sugiere que los rendimientos seguirían siendo atractivos incluso después de considerar las comisiones.

Esta estrategia es una opción valiosa a tener en cuenta al construir un portafolio de estrategias, debido a su simplicidad y efectividad. Solo utiliza indicadores, lo que la hace fácil de calcular y aplicar.

BOT DE ORDER BLOCK

Este bot logró un rendimiento del **126%** durante los **5 años** de pruebas, lo cual es un resultado destacado, superando ampliamente el rendimiento promedio anual del **SP500**, que históricamente ha sido de alrededor del **10%**. El bot, por su parte, obtuvo un retorno anual promedio del **25%**, sin contar las comisiones. Aunque las comisiones podrían reducir el beneficio, el impacto no sería significativo, dado que realizó **350 operaciones en 5 años**. Este número de operaciones no es excesivo, por lo que el retorno seguiría siendo atractivo.

La estrategia es algo más compleja, ya que utiliza **acción del precio**, como los **Order Blocks**, junto con indicadores como la **EMA**. Sin embargo, sigue siendo relativamente sencilla y objetiva, lo cual es un factor clave al diseñar bots de *trading*.

BOT DE FAIR VALUE GAP

Esta estrategia logró un rendimiento excepcional del **477%** durante los **5 años** de pruebas, lo que la posiciona como la de mejor desempeño, superando ampliamente el retorno promedio del **SP500** el cuál es de **10%** anual y el bot obtuvo **95.4%**.

Es importante señalar que esta estrategia realizó un total de **995 operaciones**, lo que representa una cantidad considerable y resulta en una pérdida significativa debido a las comisiones. Sin embargo, incluso después de tener en cuenta las comisiones, sigue siendo la estrategia más rentable.

BOT MULTI TIMEFRAME ANALYSIS

Esta estrategia obtuvo un rendimiento del **13%** durante los **5 años** de pruebas, lo que implica un promedio anual de **2.6%**. Además, al considerar las comisiones, habría generado una pérdida neta, lo que la convierte en una estrategia poco atractiva para implementar.

Uno de los motivos principales por el cual obtuvo una rentabilidad tan baja fue la cantidad de operaciones que realizó (19), un número bastante inferior al del resto de las estrategias.

La estrategia se basa únicamente en la acción del precio, lo que podría limitar su efectividad al carecer de indicadores adicionales o confluencias que mejoren su rendimiento.

3.4.2. Comparación del Rendimiento de los Bots

Los resultados de las estrategias en general son bastante positivos, ya que **3 de las 4** estrategias (1, 2 y 3) **superan** el rendimiento del **SP500**, lo que las hace rentables e interesantes para implementar en vivo. En cambio, la **Estrategia 4** obtiene un **retorno** anual del 2.5%, muy por **debajo del 10%** del SP500, por lo que, considerando comisiones y costos de operación, es probable que termine generando pérdidas netas.

Sin embargo, no solo el retorno importa, sino también la eficiencia operativa. La **Estrategia 3**, a pesar de ser la más rentable (95.4% anual), **ejecuta 995 operaciones**, lo que puede implicar **costos elevados en comisiones, spreads y deslizamientos**. Este volumen de operaciones podría reducir significativamente la rentabilidad neta.

Por otro lado, la **Estrategia 1** destaca por tener un buen **equilibrio** entre **retorno** (38%) y **número de operaciones** (99), lo que la hace más eficiente en términos de rentabilidad por operación. La Estrategia 2 también es rentable, pero su relación entre retorno y cantidad de operaciones es menos atractiva.

Sería recomendable **optimizar la Estrategia 3** para reducir la cantidad de operaciones sin afectar demasiado el retorno, posiblemente añadiendo filtros o confluencias adicionales. Esto ayudaría a mejorar la tasa de acierto, hacer el portafolio más consistente y reducir periodos de pérdidas significativas.

ESTRATEGIA	RETORNO ANUAL(%)	Nº OPERACIONES	RETORNO SP500(%)	DIFERENCIA VS SP500(%)	RENTABILIDAD/ OPERACIONES
EMA Y RSI	38%	99	10	28%	0.384
EMA Y ORDER BLOCK	25%	350	10	15%	0.071
EMA Y FVG	95.4%	995	10	85,4%	0.096
MULTI TIMEFRAME	2.6%	19	10	-7,5%	0.132

Tabla 5: Resultados del backtest de las estrategias

3.5. Optimización

3.5.1. ¿Por qué usar algoritmos de optimización?

El uso de algoritmos de optimización en el trading algorítmico es esencial, ya que permiten encontrar de manera eficiente los parámetros óptimos de una estrategia para **maximizar su rendimiento y minimizar los riesgos**. Estos algoritmos analizan grandes volúmenes de datos y prueban múltiples combinaciones de variables (como periodos de indicadores, umbrales de compra/venta, niveles de *stop loss*, etc.) para identificar configuraciones que ofrezcan el mejor equilibrio entre retorno y pérdida máxima.

Es importante tener en cuenta el riesgo de **sobreoptimización**, que puede llevar a que una estrategia parezca funcionar bien durante el **backtest**, pero que, al ejecutarla en vivo, no tenga el mismo desempeño o incluso genere pérdidas. En resumen, la optimización puede mejorar tanto la efectividad como la confiabilidad de las estrategias algorítmicas, siempre que se realice correctamente.

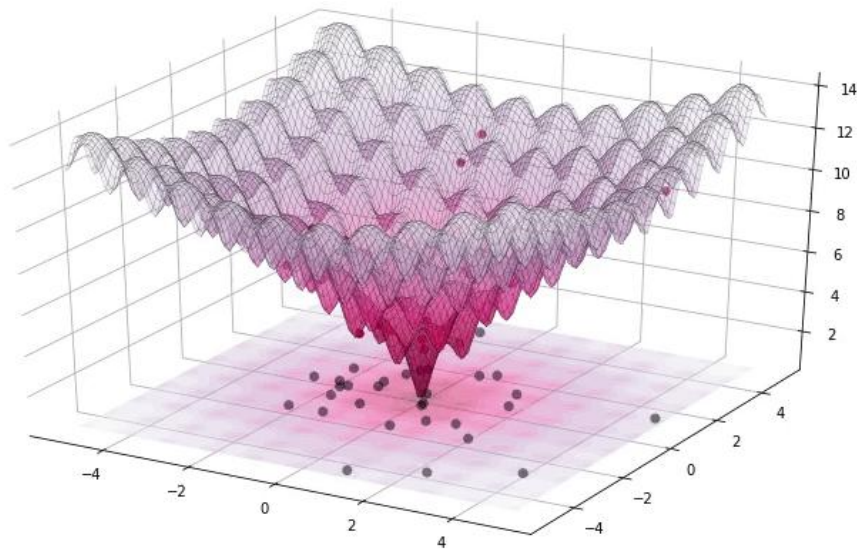
3.5.2. Librerías de Optimización

En esta sección se describen diferentes métodos de optimización de la estrategia EMA y RSI con el objetivo de maximizar el retorno y minimizar el *drawdown*. Para ello, se han utilizado librerías que implementan algoritmos de optimización ya desarrollados. En particular:

- **Differential Evolution:** se ha implementado utilizando la función `differential_evolution` de la librería `scipy.optimize`.[\[29\]](#)
- **Genetic Algorithm:** ha sido desarrollado con la librería DEAP, utilizando los módulos `base`, `creator`, `tools` y `algorithms`.[\[30\]](#)

- **Particle Swarm Optimization:** se ha aplicado mediante la función pso de la librería pyswarm.[\[31\]](#)

DIFFERENTIAL EVOLUTION



Fuente: [DIFFERENTIAL EVOLUTION](#)

El algoritmo Differential Evolution (DE) es una técnica metaheurística de optimización diseñada para resolver problemas no lineales en espacios continuos, con múltiples dimensiones y óptimos locales. Es útil para maximizar o minimizar una función objetivo.

Además, es importante destacar que DE ha sido aplicado en la optimización de bots y en operaciones de trading en general. Diversos trabajos en la literatura han demostrado su utilidad para sintonizar parámetros de estrategias de trading, optimizar carteras y mejorar la rentabilidad de sistemas automatizados. Por ejemplo, Storn y Price (1997) establecieron las bases teóricas de DE, y trabajos posteriores han empleado este algoritmo para ajustar sistemas de trading en mercados como el Forex, optimizando tanto la selección de señales como la gestión del riesgo (véase, por ejemplo, Müller, 2014 y Torres et al., 2012). En mi trabajo se aplicó DE para afinar la estrategia de trading, lo que permitió obtener mejoras significativas en la eficiencia del bot sin caer en óptimos locales.[\[29\]](#) [\[32\]](#) [\[33\]](#)

1. Idea General

DE pertenece a los algoritmos evolutivos, similar a los genéticos, pero con un enfoque diferente. Genera, muta, cruza y selecciona vectores en el espacio de búsqueda continuo, usando diferencias ponderadas entre vectores para guiar la exploración de soluciones.

2. Componentes Clave

- **Población inicial:** Conjunto de vectores generados aleatoriamente en el espacio de búsqueda.
- **Función objetivo:** Función a maximizar o minimizar. Cada vector se evalúa en ella.
- **Parámetros principales:**
 - **Tamaño de la población (NP):** Número de vectores en la población.
 - **Factor de mutación (F):** Controla la escala de la mutación (0.4–1.0).
 - **Probabilidad de recombinación (CR):** Mezcla entre padres e hijos (0–1).
 - **Número de iteraciones (max_iter):** Número máximo de generaciones.

3. Pasos del Algoritmo

1. **Inicialización:** Se genera una población de NP vectores.
2. **Evaluación inicial:** Se calcula el valor de cada vector en la función objetivo y se guarda el mejor.
3. **Generación de nuevos vectores:** Cada vector pasa por mutación, recombinación y selección:

- **Mutación:** Se combina linealmente tres vectores aleatorios para generar un nuevo vector.

$$\text{Vector Mutado} = \text{Vector Base} + \text{Factor de Mutación} \times (\text{Vector Aleatorio 1} - \text{Vector Aleatorio 2})$$

Donde:

- **Vector Mutado (v_p^g):** Es el nuevo candidato que se crea para probar en el proceso de optimización.
 - **Vector Base ($x_{r_0}^g$):** Se selecciona de manera aleatoria de la población actual y sirve como punto de partida.
 - **Vector Aleatorio 1 ($x_{r_1}^g$) y Vector Aleatorio 2 ($x_{r_2}^g$):** Son otros dos vectores elegidos al azar de la población. Se calcula la diferencia entre estos dos.
 - **Factor de Mutación (F):** Es un número que controla cuánto se amplifica la diferencia entre los dos vectores aleatorios. Si F es mayor, la diferencia se multiplica más, lo que genera mutaciones más grandes; si es menor, la mutación será más sutil.
- **Recombinación:** Se crea un vector candidato combinando el vector original y el mutado.
 - **Selección:** El vector candidato reemplaza al original si mejora el valor de la función objetivo.
4. **Repetición:** Se repiten los pasos hasta alcanzar el criterio de parada (máximo número de iteraciones o tolerancia en la función objetivo).

Implementación

En la implementación del DE en el bot EMA y RSI, se buscó maximizar el retorno y minimizar el drawdown.

Los parámetros utilizados para encontrar la mejor combinación fueron:

- **ema_period**: Entre 50 y 200.
- **rsi_period**: Entre 5 y 20.
- **overbought**: Entre 70 y 90.
- **oversold**: Entre 10 y 30.

Se definió el máximo número de generaciones en **100** y se estableció un **tamaño de población** de 50 para realizar las mutaciones.

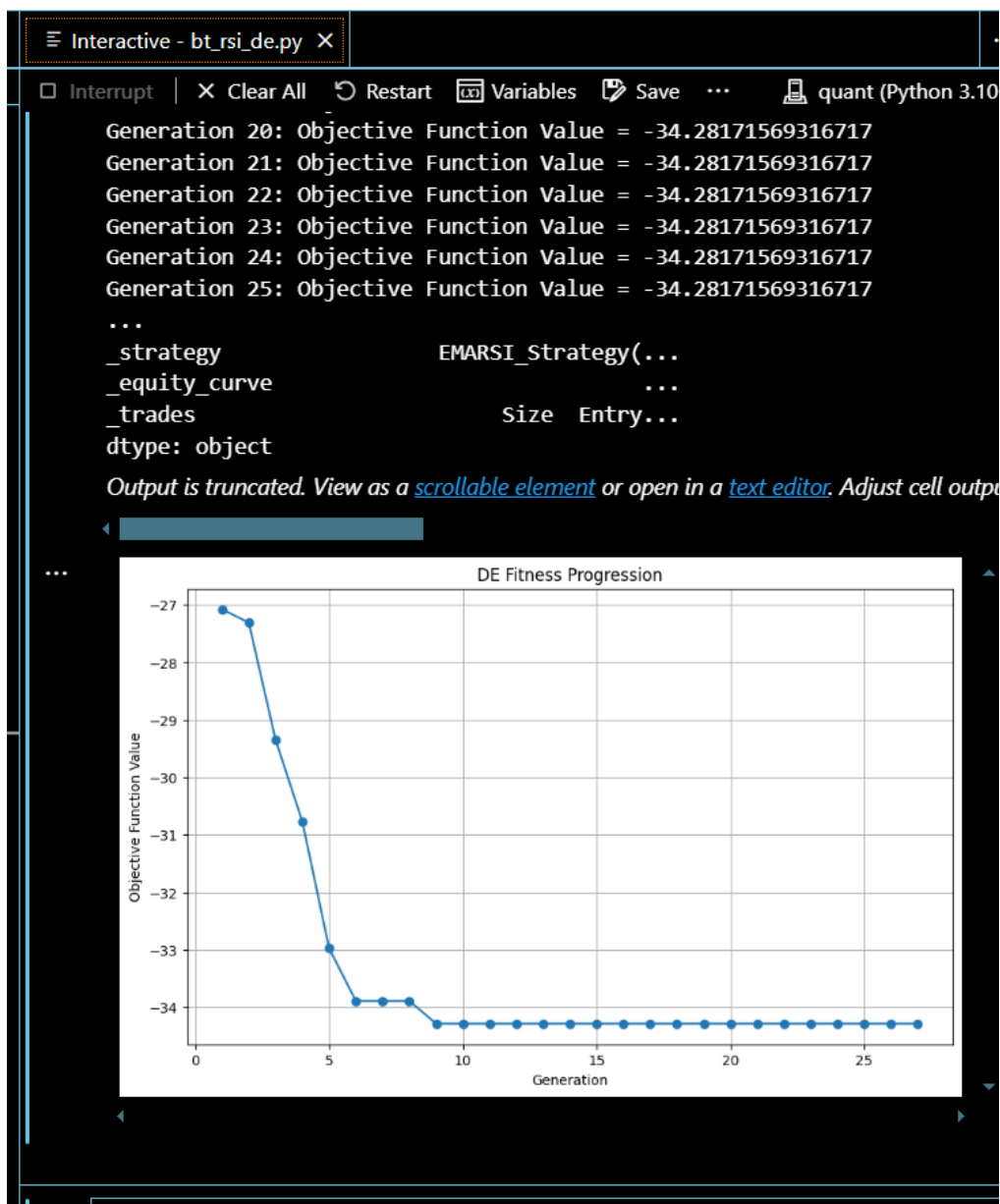


Figura 11: Resultado de la optimización de la estrategia usando Differential Evolution

Start	2019-01-02 18:00:00
End	2024-01-01 18:00:00
Nº Trades	116
Return [%]	30.53%
Max Drawdown [%]	-3.75%
Win Rate	81%
Worst Trade	-10,68%

Tabla 6: Resultados de la optimización de la estrategia usando Differential Evolution

Evaluación de los resultados del backtest

Según las estadísticas generadas por la librería `backtesting.py`, la estrategia reportó un **31%** de retorno y una pérdida máxima del **-4%** en un **período de 5 años**. Sin embargo, estos resultados subestiman el rendimiento real, ya que la librería es imprecisa al cerrar las posiciones: las órdenes de *stop loss* y *take profit* se ejecutan al cierre de la vela en lugar del momento exacto en que se activan.

Esto se refleja en las métricas, como el *Worst Trade*, que muestra una pérdida de -10% en lugar del límite teórico de -1%.

Cálculo del retorno real

Para estimar el rendimiento real de la estrategia sin contar comisiones:

- Número total de operaciones: **116**
- Tasa de acierto (*Win Rate*): **81%**
- Relación riesgo-beneficio: **1:3** (3% en operaciones ganadoras, -1% en perdedoras).

6. Cálculo de operaciones ganadoras y perdedoras:

- Ganadoras: $116 \times 0.81 = 94$
- Perdedoras: $116 - 94 = 22$

7. Cálculo del retorno total:

- Ganancias: $94 \times 3\% = 282$
- Pérdidas: $22 \times -1\% = -22$
- Retorno neto: $282\% - 22\% = \mathbf{260\%}$

8. Retorno anual promedio:

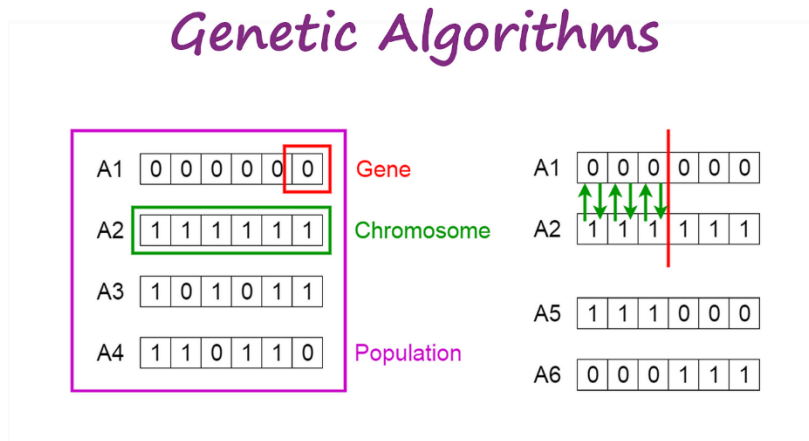
$$260\% \div 5 \text{ años} = 52\%$$

9. Conclusión

Si la estrategia se hubiera ejecutado en vivo, con órdenes al momento de activarse y no al cierre de la vela, el retorno estimado habría sido

significativamente mayor, alcanzando un promedio anual del **52%** en lugar del 31% reportado por la librería.

ALGORITMO GENÉTICO



Fuente: [ALGORITMO GENÉTICO](#)

El **algoritmo genético (GA)** busca encontrar una solución óptima mediante una población de posibles soluciones que evolucionan utilizando operadores genéticos. Este enfoque ha sido ampliamente utilizado en la optimización de estrategias de trading y bots de inversión.

Por ejemplo, en la plataforma MetaTrader, se emplean algoritmos genéticos para optimizar estrategias de trading, permitiendo realizar optimizaciones con un alto número de combinaciones de parámetros de manera más eficiente que los métodos tradicionales. En mi trabajo se aplicó GA para afinar la estrategia de trading, lo que permitió obtener mejoras significativas en la eficiencia del bot sin caer en óptimos locales. [34]

1. **Representación de soluciones:** Las soluciones se representan como individuos en la población, usando cadenas binarias, vectores reales o listas de enteros, dependiendo del tipo de problema.
2. **Población inicial:** Se crea aleatoriamente una población de individuos, por ejemplo, 50 vectores con valores dentro de los límites del problema.
3. **Función de fitness:** Cada individuo se evalúa según una función de fitness que mide qué tan buena es la solución, como maximizar el retorno y minimizar el *drawdown* en un problema de *trading*.
4. **Operadores genéticos:**
 - **Selección:** Los mejores individuos se eligen para pasar sus genes a la siguiente generación, utilizando métodos como ruleta, torneo o rango.

- **Cruce (Crossover):** Los genes de dos padres se combinan para generar nuevos hijos. Ejemplo: cruce de un punto o cruce uniforme.
 - **Mutación:** Algunos genes se alteran aleatoriamente para introducir diversidad y evitar óptimos locales.
5. **Generaciones:** El proceso de selección, cruce y mutación se repite para crear nuevas generaciones, con la posibilidad de conservar los mejores individuos (elitismo).
 6. **Criterio de parada:** El algoritmo termina al alcanzar un número máximo de generaciones, una mejora mínima en el fitness o un objetivo de fitness predefinido.

Implementación

En la implementación del GA en el bot EMA y RSI, el objetivo fue maximizar el retorno y minimizar el *drawdown*.

Los parámetros utilizados para encontrar la mejor combinación fueron:

- **ema_period:** Entre 50 y 200.
- **rsi_period:** Entre 5 y 20.
- **overbought:** Entre 70 y 90.
- **oversold:** Entre 10 y 30.

Se asignaron pesos de **0.5** tanto al retorno como al *drawdown*, sin priorizar ninguno, buscando un equilibrio entre ambas variables.

El método de selección utilizado fue el torneo, con una población de **50** individuos y **100** generaciones.

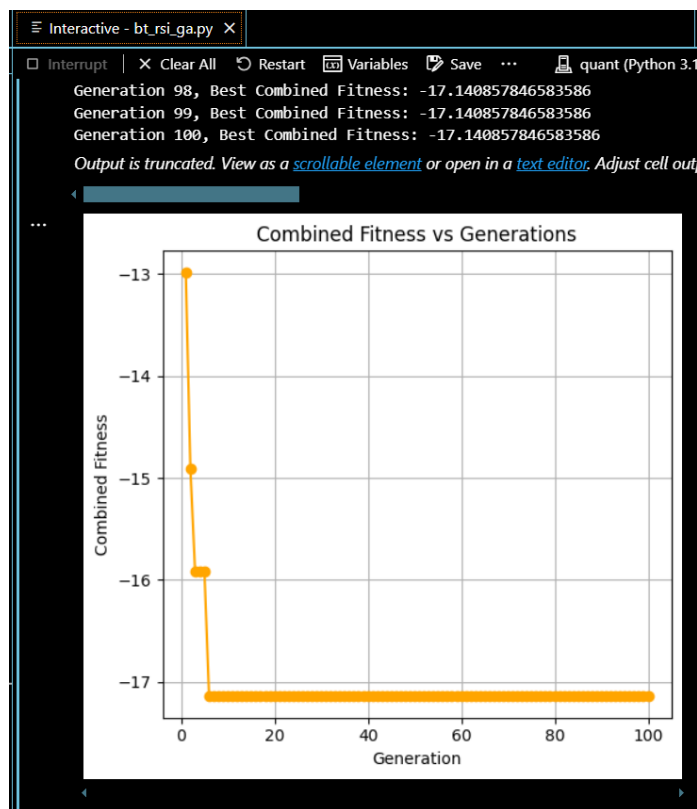


Figura 12: Resultado de la optimización de la estrategia usando Genetic Algorithm

Start	2019-01-02 18:00:00
End	2024-01-01 18:00:00
Nº Trades	116
Return [%]	30.53%
Max Drawdown [%]	-3.75%
Win Rate	81%
Worst Trade	-10,68%

Tabla 7: Resultados de la optimización de la estrategia usando Genetic Algorithm

El algoritmo genético obtuvo los mismos resultados que el Differential Evolution lo que quiere decir que están proponiendo una **solución robusta**. Ambos algoritmos están convergiendo a la misma solución óptima porque esta es consistente y representativa del problema planteado. Esto sugiere que no hay múltiples óptimos locales significativos que los lleven a resultados distintos.

Además, esto indica una baja probabilidad de sobreajuste (overfitting): Si el resultado es consistente en diferentes enfoques de optimización, es menos probable que los algoritmos estén ajustándose únicamente a un ruido o características específicas de los datos (es decir, sobreajustando).

Evaluación de los resultados del backtest

Según las estadísticas generadas por la librería backtesting.py, la estrategia reportó un

31% de retorno y una pérdida máxima del **-4%** en un **período de 5 años**. Sin embargo, estos resultados subestiman el rendimiento real, ya que la librería es imprecisa al cerrar las posiciones: las órdenes de *stop loss* y *take profit* se ejecutan al cierre de la vela en lugar del momento exacto en que se activan.

Esto se refleja en las métricas, como el *Worst Trade*, que muestra una pérdida de -10% en lugar del límite teórico de -1%.

Cálculo del retorno real

Para estimar el rendimiento real de la estrategia sin contar comisiones:

- Número total de operaciones: **116**
- Tasa de acierto (*Win Rate*): **81%**
- Relación riesgo-beneficio: **1:3** (3% en operaciones ganadoras, -1% en perdedoras).

10. Cálculo de operaciones ganadoras y perdedoras:

- Ganadoras: $116 \times 0.81 = 94$
- Perdedoras: $116 - 94 = 22$

11. Cálculo del retorno total:

- Ganancias: $94 \times 3\% = 282$
- Pérdidas: $22 \times -1\% = -22$
- Retorno neto: $282\% - 22\% = \mathbf{260\%}$

12. Retorno anual promedio:

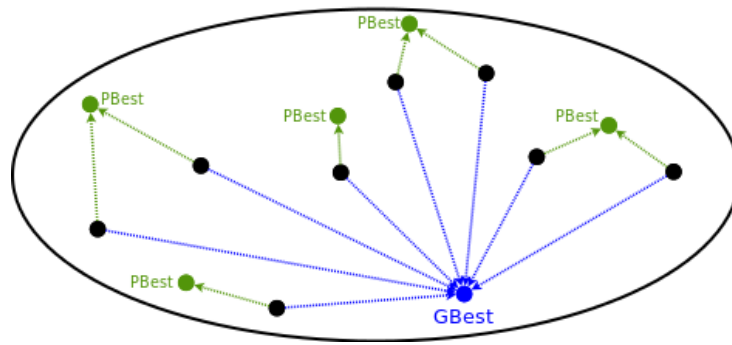
$$260\% \div 5 \text{ años} = 52\%$$

Conclusión

Si la estrategia se hubiera ejecutado en vivo, con órdenes al momento de activarse y no al cierre de la vela, el retorno estimado habría sido significativamente mayor, alcanzando un promedio anual del **52%** en lugar del 31% reportado por la librería

ALGORITMO OPTIMIZACIÓN POR ENJAMBRE DE PARTÍCULAS (PSO)

Standard Particle Swarm Optimization (PSO)



El algoritmo **PSO** (Optimización por Enjambre de Partículas) es una técnica de optimización inspirada en el comportamiento de grupos de animales. En este algoritmo, un conjunto de partículas (soluciones) se mueve en un espacio de búsqueda, ajustando su posición y velocidad según su propia experiencia y la del grupo, para encontrar la mejor solución posible a un problema. Combina la exploración de nuevas áreas con la explotación de soluciones ya encontradas.

Ha sido ampliamente utilizado en la optimización de estrategias de trading y en el desarrollo de bots de trading. Por ejemplo, en el artículo "Fine-Tuning the Strategy Using a Particle Swarm Optimization" se describe cómo el PSO puede ajustar parámetros de estrategias de trading para maximizar funciones objetivo específicas, mejorando así el rendimiento de los algoritmos de trading. En mi trabajo se aplicó PSO para afinar la estrategia de trading, lo que permitió obtener mejoras significativas en la eficiencia del bot sin caer en óptimos locales.[\[35\]](#)

1. Conceptos básicos

En PSO, las partículas son soluciones posibles a un problema que se mueven en un espacio de búsqueda. El objetivo es encontrar la mejor solución optimizando una función objetivo.

- **Partícula:** Representa una solución al problema.
- **Espacio de búsqueda:** El área donde las partículas buscan la solución óptima.
- **Velocidad:** Determina cómo se mueve cada partícula.
- **Objetivo:** Mejorar la función objetivo (maximizar o minimizar).

2. Colaboración en el enjambre

Las partículas trabajan juntas, compartiendo información sobre sus mejores soluciones:

- **Mejor personal:** La mejor solución que cada partícula ha encontrado.
- **Mejor global:** La mejor solución encontrada por el grupo entero.

3. Movimiento de las partículas

Las partículas se mueven combinando dos aspectos:

- **Exploración:** Buscar nuevas áreas del espacio de búsqueda.
- **Explotación:** Refinar soluciones ya conocidas. Cada partícula ajusta su posición según estas dos fuerzas.

4. Proceso iterativo

En cada iteración del PSO:

1. **Evaluación:** Se calcula qué tan buena es la solución actual.
2. **Actualización:** Se actualizan las mejores soluciones personales y globales.
3. **Movimiento:** Las partículas ajustan su velocidad y posición para explorar mejor el espacio.

5. Equilibrio exploración-explotación

El algoritmo busca un balance entre explorar nuevas áreas y refinar las soluciones cercanas a la mejor encontrada, evitando quedar atrapado en óptimos locales.

Implementación

En el bot EMA y RSI, PSO se usa para maximizar el retorno y minimizar el drawdown.

Los parámetros utilizados para encontrar la mejor combinación fueron:

- **ema_period:** Entre 50 y 200.
- **rsi_period:** Entre 5 y 20.
- **overbought:** Entre 70 y 90.
- **oversold:** Entre 10 y 30.

Se asignaron pesos de 1 a ambos, retorno y *drawdown*, sin priorizar ninguna de las dos variables, buscando un equilibrio entre ambas.

Se utilizó una población de 50 individuos y 100 generaciones. Debido a que el objetivo del algoritmo es minimizar el resultado, se invirtieron los signos al establecer los criterios de maximizar y minimizar, para poder minimizar el *drawdown* y maximizar el retorno.

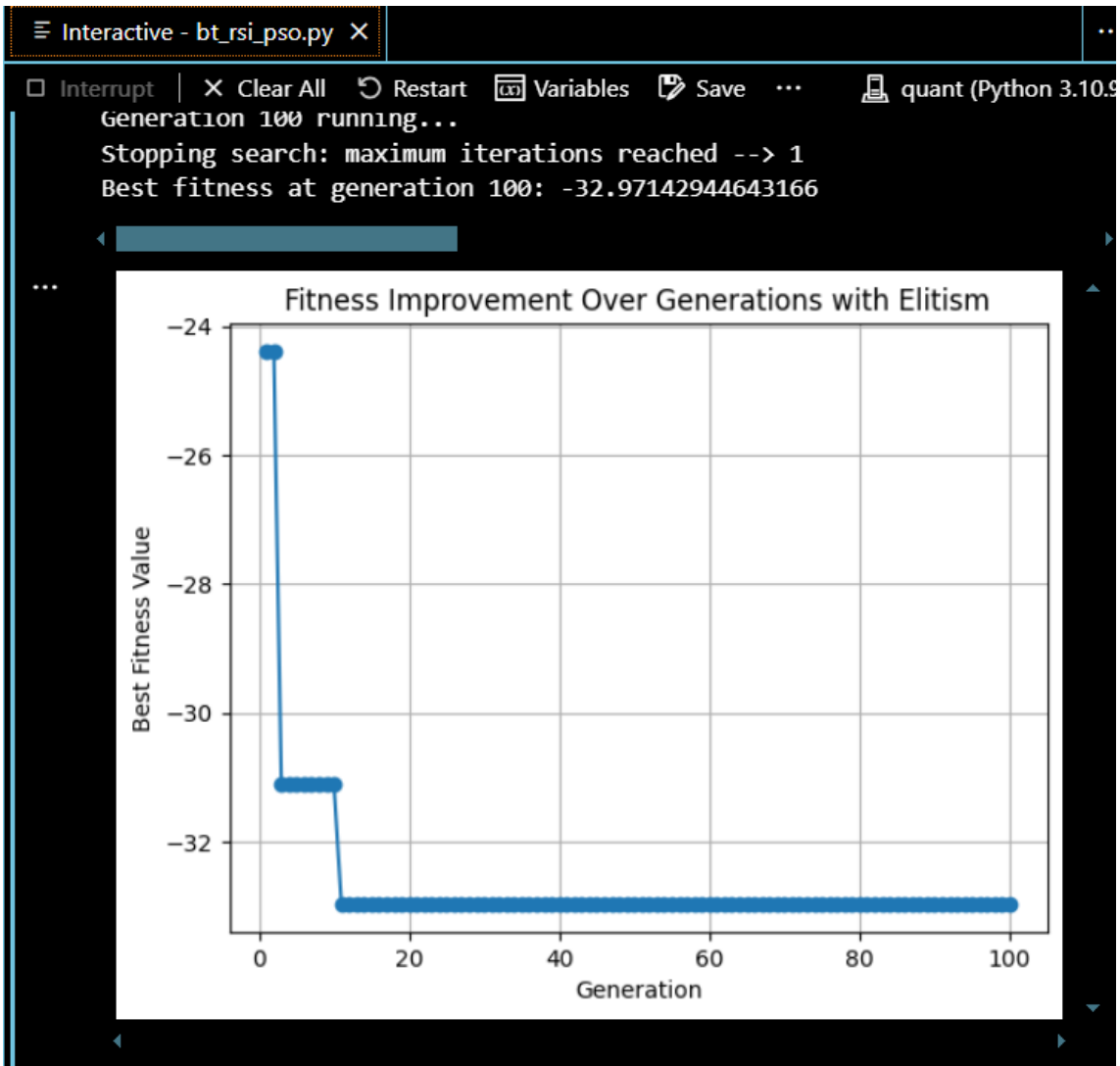


Figura 13: Resultado de la optimización de la estrategia usando PSO

Start	2019-01-02 18:00:00
End	2024-01-01 18:00:00
Nº Trades	96
Return [%]	27.63%
Max Drawdown [%]	-4.23%
Win Rate	81%
Worst Trade	-8,19%

Tabla 8: Resultados de la optimización de la estrategia usando PSO

Evaluación de los resultados del backtest

Según las estadísticas generadas por la librería backtesting.py, la estrategia reportó un **27%** de retorno y una pérdida máxima del **-4%** en un **período de 5 años**. Sin embargo, estos resultados subestiman el rendimiento real, ya que la librería es imprecisa al cerrar

las posiciones: las órdenes de *stop loss* y *take profit* se ejecutan al cierre de la vela en lugar del momento exacto en que se activan.

Esto se refleja en las métricas, como el *Worst Trade*, que muestra una pérdida de -8% en lugar del límite teórico de -1%.

Cálculo del retorno real

Para estimar el rendimiento real de la estrategia sin contar comisiones:

- Número total de operaciones: **96**
- Tasa de acierto (*Win Rate*): **81%**
- Relación riesgo-beneficio: **1:3** (3% en operaciones ganadoras, -1% en perdedoras).

13. Cálculo de operaciones ganadoras y perdedoras:

- Ganadoras: $96 \times 0.81 = 78$
- Perdedoras: $96 - 78 = 18$

14. Cálculo del retorno total:

- Ganancias: $78 \times 3\% = 234$
- Pérdidas: $18 \times -1\% = -18$
- Retorno neto: $234\% - 18\% = \mathbf{216\%}$

15. Retorno anual promedio:

$$216\% \div 5 \text{ años} = 43\%$$

Conclusión

Si la estrategia se hubiera ejecutado en vivo, con órdenes al momento de activarse y no al cierre de la vela, el retorno estimado habría sido significativamente mayor, alcanzando un promedio anual del **43%** en lugar del 27% reportado por la librería

3.5.2.4. Comparativa de Resultados

Después de implementar los tres algoritmos de optimización en la estrategia base, hemos obtenido resultados interesantes que permiten una comparación clara tanto entre los algoritmos como con la estrategia sin optimizar.

En primer lugar, la estrategia sin optimización presentó un retorno anual de 38% con 99 operaciones, lo cual fue el punto de partida. Al aplicar los algoritmos de optimización, los resultados mejoraron notablemente. Dos de los algoritmos, Algoritmo DE y Algoritmo GA, lograron un retorno del 52% con 116 operaciones, un aumento considerable en comparación con la estrategia base. Sin embargo, a pesar de este aumento en el retorno, el número de operaciones también se incrementó, lo que podría generar mayores costos en comisiones y deslizamientos.

El Algoritmo PSO, por otro lado, obtuvo un 43% de retorno anual con 96 operaciones, siendo el que menos operaciones ejecutó, pero con un retorno algo inferior al de los

otros dos algoritmos. A pesar de tener menos operaciones, la rentabilidad por operación fue similar a la de los algoritmos 1 y 2, con un 0.448% de rentabilidad por operación.

Si comparamos entre los algoritmos de optimización, el Algoritmo DE y el Algoritmo GA fueron los más efectivos, logrando el mejor retorno (52%) con el mismo número de operaciones (116). Sin embargo, a pesar de este buen rendimiento, el aumento en las operaciones podría haber reducido su eficiencia en términos de costos. El Algoritmo PSO, aunque con un retorno más bajo, presentó un balance favorable con menos operaciones, lo que podría hacerlo más eficiente en términos de costos operativos.

En resumen, los algoritmos de optimización demostraron ser efectivos, duplicando prácticamente los resultados de la estrategia base sin optimización. No obstante, ninguno de los algoritmos logró superar la rentabilidad del bot de mejor rendimiento (el bot de FVG) en cuanto a retorno que ese era uno de los objetivos, que seguía siendo más eficiente en cuanto a retorno por operación.

En conclusión, los algoritmos mejoraron notablemente el rendimiento global de la estrategia.

ESTRATEGIA	RETORNO ANUAL(%)	Nº OPERACIONES	RENTABILIDAD/ OPERACIONES
SIN OPTIMIZAR	38%	99	0.384
ALGORITMO DE	52%	116	0.448
ALGORITMO GA	52%	116	0.448
ALGORITMO PSO	43%	96	0.447

Tabla 9: Comparativa de resultados de los Algoritmos de Optimización

3.6. DESAFÍOS ENCONTRADOS DURANTE LA IMPLEMENTACIÓN

Durante la implementación de los bots, se encontraron varios desafíos, como la gestión del riesgo y la configuración de los parámetros óptimos para cada estrategia:

- El principal problema, que fue bastante tedioso de resolver, fue la falta de rentabilidad esperada en las operaciones. Es decir, si se realizaban 10 operaciones y se ganaban 5 con un riesgo-beneficio de 1 a 3, en lugar de obtener un 10% de ganancia ($5 \cdot 3 - 5$), solo se lograba un 1% o 2%. Tras investigar, se descubrió que la causa no era un problema en los algoritmos, sino en el diseño de la librería utilizada.
- Además, se presentaron dificultades al implementar los algoritmos de optimización, ya que inicialmente el algoritmo genético mostraba una curva ascendente, mientras que el DE y el PSO presentaban curvas descendentes. Esto requirió ajustar los signos para que todas las curvas fueran coherentes y poder compararlas adecuadamente.

4. CONCLUSIONES

El *trading* algorítmico se ha consolidado como una herramienta eficaz para rentabilizar ahorros y preservar el poder adquisitivo frente a la inflación. A lo largo de este trabajo, se han explorado y desarrollado estrategias automatizadas con el objetivo de optimizar su rendimiento y adaptabilidad a diversos escenarios de mercado.

Inicialmente, se abordaron dos enfoques principales en las estrategias de trading: seguir la tendencia y operar en contra de ella. Ambas metodologías demostraron ser efectivas dependiendo del contexto del mercado y su correcta implementación. Para garantizar la viabilidad de estas estrategias, se llevaron a cabo pruebas exhaustivas que confirmaron una esperanza matemática positiva, evidenciando su potencial rentabilidad.

La diversificación, tanto en estrategias como en mercados, emergió como un factor crucial para mitigar riesgos y adaptarse a distintos entornos económicos. La automatización mediante bots de trading permitió eliminar el componente emocional en la toma de decisiones, incrementando la eficiencia y consistencia operativa. Sin embargo, se destacó la importancia de seleccionar plataformas adecuadas, considerando costos como comisiones y posibles inconvenientes técnicos que podrían afectar los resultados.

Aunque las simulaciones proporcionaron una visión preliminar del desempeño de las estrategias, se reconoció que no replican completamente las condiciones del mundo real, donde factores como errores de conexión o deslizamientos pueden influir significativamente. Por ello, se enfatizó la necesidad de validar las estrategias en entornos reales antes de su implementación definitiva.

El desarrollo de estrategias rentables no requirió innovaciones radicales; en cambio, el estudio de trabajos y publicaciones de autores reconocidos brindó una base sólida para alcanzar los resultados deseados. El control del *drawdown* se identificó como esencial para mantener el riesgo en niveles aceptables y proteger el capital invertido. Además, se resaltó que una gestión adecuada de las salidas en las operaciones es más determinante que las entradas, ya que permite controlar el riesgo y maximizar beneficios.

La gestión del riesgo, incluyendo la diversificación, aseguró la sostenibilidad de las estrategias incluso en condiciones adversas del mercado. Se concluyó que las estrategias deben ser dinámicas y adaptarse a las condiciones cambiantes del mercado para evitar su obsolescencia. Asimismo, se subrayó la importancia de la ética en el uso de bots, garantizando que no se utilicen para manipular el mercado o generar ventajas injustas.

En conclusión, el trading algorítmico ha demostrado ser una disciplina de creciente relevancia en los mercados financieros. Se estima que un gran porcentaje de las operaciones en las bolsas de valores son realizadas mediante trading algorítmico, lo que refleja su impacto en la economía global. Este trabajo ha buscado contribuir al conocimiento de esta área, esperando despertar el interés del lector y motivarlo a profundizar en esta fascinante intersección entre finanzas, matemáticas y tecnología.

Este proyecto ha sido una experiencia profundamente enriquecedora que ha ampliado significativamente mi comprensión del trading algorítmico. A través del desarrollo y optimización de algoritmos de trading, he aprendido que la disciplina, la adaptabilidad y la colaboración interdisciplinaria son fundamentales para crear estrategias exitosas. Esta experiencia me ha permitido integrar conocimientos de finanzas, matemáticas y

programación, resaltando la importancia de una formación integral en este campo. Además, he comprendido que, aunque la automatización es una herramienta poderosa, la supervisión humana sigue siendo esencial para adaptarse a las condiciones dinámicas del mercado y garantizar la eficacia de las estrategias implementadas.

5. LÍNEAS FUTURAS

A continuación, se presentan algunas propuestas para ampliar y mejorar el proyecto, así como nuevas direcciones que podrían explorarse en el futuro:

1. Creación de estrategias adaptadas a empresas de fondeo

Una posible expansión del proyecto sería diseñar estrategias específicas para traders que operan con capital proporcionado por empresas como **Darwinex** o **FTMO**. Estas empresas evalúan el desempeño de los traders y les ofrecen fondos para operar si demuestran rentabilidad y control de riesgos.

- Se podrían crear bots que optimicen la relación riesgo/beneficio para cumplir con los criterios estrictos de estas empresas.
- Esto requeriría adaptar las estrategias a sus reglas, como límites de *drawdown*, objetivos de beneficio y plazos de evaluación.

2. Creación de una página web o blog de comunidad

Otra línea de desarrollo sería crear una plataforma en línea, como una página web o blog, que permita a los usuarios debatir y compartir estrategias de trading.

- **Objetivo:** Crear un espacio colaborativo donde *traders* e investigadores puedan compartir ideas, analizar estrategias y aprender mutuamente.
- Funcionalidades posibles:
 - Foro de discusión sobre estrategias.
 - Recursos educativos y tutoriales.
 - Simuladores o herramientas interactivas para probar estrategias compartidas.

Esto fomentaría la creación de una comunidad activa en torno al trading algorítmico, promoviendo el aprendizaje colectivo.

3. Pruebas más exhaustivas con técnicas avanzadas

Para garantizar que las estrategias son robustas y sostenibles, se pueden realizar análisis más avanzados como:

- **Simulación de Montecarlo:** Este método evalúa la robustez de una estrategia al simular miles de escenarios aleatorios basados en los datos históricos. Permite medir cómo podría comportarse la estrategia bajo diferentes condiciones de mercado y niveles de ruido, proporcionando una visión más completa de su viabilidad.

- **Análisis Walkforward:** Es una técnica que divide los datos históricos en períodos de entrenamiento y prueba, optimizando la estrategia en un conjunto de datos y validándola en un período posterior. Esto imita cómo se comportará la estrategia en datos futuros y evita el sobreajuste (**overfitting**).

Estas pruebas garantizarían que las estrategias no solo son rentables en el pasado, sino también en condiciones futuras impredecibles.

4. Participación en competiciones de trading

Otra oportunidad es diseñar estrategias orientadas a competiciones de trading, como las organizadas por **RoboForex** o los campeonatos mundiales de trading.

- Estas competiciones suelen premiar tanto la rentabilidad como la innovación en las estrategias.
- Participar en estos eventos no solo serviría como validación pública de las estrategias, sino que también permitiría obtener reconocimiento en la comunidad financiera.

5. Implementación de técnicas de Machine Learning e IA

Un área prometedora es la incorporación de técnicas de **inteligencia artificial (IA)** y **machine learning (ML)**, como:

- **Redes neuronales:** Pueden usarse para identificar patrones complejos en los datos de mercado y predecir movimientos futuros.
- **LSTM (Long Short-Term Memory):** Estas redes neuronales son ideales para analizar series temporales, como datos de precios, ya que pueden captar dependencias a largo plazo en los movimientos del mercado.
- **Aprendizaje por refuerzo:** Consiste en entrenar un agente para que aprenda las mejores decisiones (compra, venta, o mantenerse fuera del mercado) maximizando recompensas acumuladas a lo largo del tiempo.

La incorporación de estas técnicas podría aumentar la capacidad predictiva de los algoritmos y hacerlos más adaptables a condiciones cambiantes.

6. PLANIFICACIÓN TEMPORAL Y PRESUPUESTO

6.1. Planificación Temporal

FASE I.

1. Elección del tema general del trabajo. (1 semana)
2. Planteamiento del trabajo y búsqueda de información en libros, podcast y papers. (3 semanas)
3. Búsqueda de un tutor y propuesta del trabajo. Carlos Camacho, tutor de este proyecto, se mostró interesado desde el principio y aceptó llevar el tutelaje del mismo. (1 semana)

FASE II.

1. Enfoque más específico del tema y elaboración y estructuración de las bases del trabajo. (1 semana)
2. Búsqueda de información en papers de estrategias e indicadores aplicados a los bots de trading. (2 semanas)

FASE III.

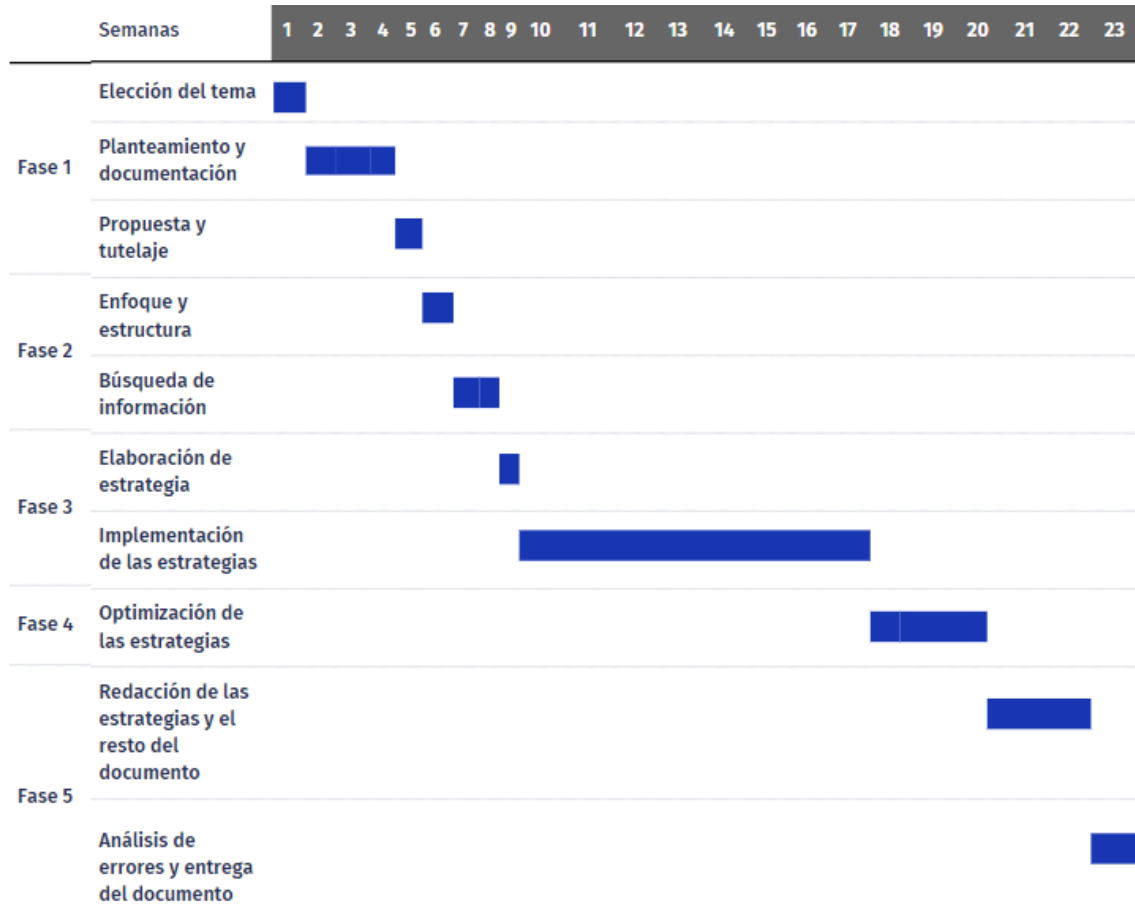
1. Elaboración de las 4 estrategias para los diferentes bots de trading (1 semana)
2. Implementación del backtesting de las 4 estrategias. (3 semanas)
3. Implementación de los bots de las 4 estrategias para el Exchange de Phemex (3 semanas)
4. Implementación de los bots de las 4 estrategias para el Exchange de Hyperliquid (2 semanas)

FASE IV.

1. Búsqueda de información de algoritmos de optimización (1 semana)
2. Implementación de los algoritmos de optimización (2 semanas)

FASE V.

1. Redacción de las estrategias y del resto del documento (2 semanas)
2. Análisis de errores y entrega del documento (1 semana)



6.2. Presupuesto

El presupuesto incluye los gastos de recursos humanos:

- Horas de trabajo del autor en el proyecto
- Horas dedicadas del tutor a la elaboración y valoración del proyecto.

COSTE CAPITAL HUMANO			
COLABORADOR	Nº HORAS	COSTE POR HORA	COSTE TOTAL
AUTOR	400	15€	6.000€
TUTOR	15	30€	450€
TOTAL			6.450€

Tabla 10: Tabla de Presupuesto

6.3 Impactos Sociales y Ambientales del Proyecto

El desarrollo y uso de sistemas automatizados de trading tienen diversas implicaciones en la sociedad y en el medio ambiente. En el contexto social, estos sistemas han transformado la manera en que los inversores participan en los mercados financieros, facilitando el acceso a herramientas que antes eran exclusivas de grandes instituciones. Sin embargo, este avance también conlleva riesgos, especialmente para los usuarios sin la formación adecuada. La proliferación de bots de *trading* que prometen altos rendimientos sin considerar la gestión de riesgos ha llevado a que muchas personas sufran pérdidas económicas significativas, lo que puede generar desconfianza en el mercado y afectar la estabilidad financiera de individuos y familias.

Otro impacto social relevante es la posible reducción de empleo en el sector financiero tradicional. Con la automatización de operaciones, la necesidad de traders humanos disminuye, lo que puede provocar la reestructuración de puestos de trabajo en instituciones financieras. No obstante, también se generan nuevas oportunidades laborales en el desarrollo, mantenimiento y regulación de estos sistemas, lo que requiere una readaptación en la formación y capacitación profesional.

Desde el punto de vista ambiental, si bien los bots de *trading* no tienen un impacto directo en el medio ambiente, su funcionamiento depende del uso intensivo de infraestructura tecnológica, incluyendo servidores y centros de datos que consumen grandes cantidades de energía. La creciente digitalización de los mercados financieros y la necesidad de procesamiento de datos en tiempo real aumentan la demanda de energía, contribuyendo indirectamente a la huella de carbono. La implementación de soluciones energéticamente eficientes y el uso de fuentes renovables pueden mitigar este impacto.

6.4 Responsabilidad Ética y Profesional

El desarrollo y uso de bots de *trading* implican diversas responsabilidades éticas y profesionales. Uno de los principales desafíos éticos es la transparencia en la información proporcionada a los usuarios. Muchas plataformas promocionan sus bots

con datos sesgados o manipulados para aparentar un rendimiento superior al real. Esta práctica puede inducir a error a los inversores, especialmente a aquellos sin experiencia, llevándolos a tomar decisiones basadas en expectativas irreales.

La responsabilidad profesional en este ámbito requiere el diseño de sistemas que prioricen la gestión del riesgo y la sostenibilidad a largo plazo sobre la mera búsqueda de rentabilidad. Un bot de *trading* bien diseñado debe ofrecer a los usuarios herramientas para comprender mejor el mercado y gestionar sus inversiones de manera responsable. Además, los desarrolladores deben garantizar que sus algoritmos sean probados en diversas condiciones de mercado para evitar que estrategias optimizadas solo para datos históricos generen pérdidas inesperadas en situaciones reales.

Otro aspecto ético fundamental es la regulación y supervisión de estos sistemas. La ausencia de normativas claras en algunos mercados permite la proliferación de bots que operan con prácticas cuestionables, como el uso de estrategias de manipulación de precios o la explotación de ineficiencias del mercado en perjuicio de otros inversores. La colaboración entre desarrolladores, reguladores y usuarios es clave para fomentar un ecosistema financiero más justo y transparente.

Finalmente, la educación financiera juega un papel crucial en la mitigación de los riesgos asociados al uso de bots de *trading*. Es responsabilidad de los desarrolladores y promotores de estas herramientas proporcionar materiales educativos que ayuden a los usuarios a comprender su funcionamiento, limitaciones y riesgos asociados. Solo a través de un enfoque ético y profesional se puede garantizar que la automatización del trading contribuya de manera positiva al desarrollo de los mercados financieros y al bienestar de los inversores.

7. BIBLIOGRAFÍA

- [1] [¿Qué son las inversiones financieras y qué tipos existen? BBVA](#)
- [2] [Crypto, art, NFTs, real estate: what are the best alternative investments? MYND](#)
- [3] [Perfil inversor o de riesgo: ¿qué es y qué tipos existen? Santander](#)
- [4] [¿Qué son los mercados financieros? Santander](#)
- [5] [¿Qué son los mercados financieros? Raisin](#)
- [6] [Factores que influyen en el precio de las acciones BBVA](#)
- [7] [Factores Internos y Externos que Afectan el Precio de las Acciones](#)
- [8] [Wikipedia Mercado de valores](#)
- [9] [¿Qué es fintech?](#)
- [10] [COVID-19](#)
- [11] [ANÁLISIS TÉCNICO,FUNDAMENTAL Y CUANTITATIVO eToro](#)
- [12] [Cómo superar los 5 sesgos más comunes en el trading Earn2Trade Blog](#)
- [13] [Análisis del comportamiento: el estudio de la psicología de los traders NewTrading](#)

- [\[14\] ¿Cómo afecta la psicología financiera a la inversión bursátil? Inesem business school](#)
- [\[15\] Flash Crash de 2010](#)
- [\[16\] Trading Bots Robots al mando El impacto de los Trading Bots en la dinamica del trading social FasterCapital](#)
- [\[17\] 7 Unexpected Ways AI Can Transform Your Investment Strategy Investopedia](#)
- [\[18\] Historia del Trading Automático tradEasy](#)
- [\[19\] Stock Market Prediction Using LSTM Recurrent Neural Network](#)
- [\[20\] How Using Natural Language Processing Improves Market Predictions](#)
- [\[21\] La revolución de los bots: Cómo el trading algorítmico está transformando los mercados HistoriaCompu](#)
- [\[22\] Trading manual vs. trading automático JustMarkets](#)
- [\[23\] Los mejores indicadores técnicos de trading Inverbots](#)
- [\[24\] La Evolución del Trading: Aprovechando la IA y los Bots para el Éxito Medium](#)
- [\[25\] ¿Qué es IA Trading Bot y cómo funciona? CriptoNoticias](#)
- [\[26\] Are Trading Bots Legal & Legit for Trading? DailyForex](#)
- [\[27\] Jim Simmons Insights Academy.](#)
- [\[28\] Irene Alridge CornellEngineering](#)
- [\[29\] Differential evolution](#)
- [\[30\] DEAP documentation](#)
- [\[31\] PSO: Optimización por enjambres de partículas](#)
- [\[32\] DE by Rainer Storn¹ and Kenneth Price²](#)
- [\[33\] DE by Müller and Roger](#)
- [\[34\] Algoritmo Genético MetaTrader](#)
- [\[35\] Fine-Tuning the Algorithmic Strategy Using a Particle Swarm](#)